Final Report

Title: A Density-Ratio Approach to Machine Learning

AFOSR/AOARD Reference Number: AOARD-09-4071

AFOSR/AOARD Program Manager: Hiroshi Motoda, Ph.D.

Period of Performance: 24 months from 9 April 2009

Submission Date: 12 March 2011

PI: Masashi Sugiyama, Tokyo Institute of Technology

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
20 APR 2011	**Final**	**09-04-2009 to 09-04-2011**

4. TITLE AND SUBTITLE
A density ratio approach to machine learning

5a. CONTRACT NUMBER
FA23860914071

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Masashi Sugiyama

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Tokyo Institute of Technology, Department of Computer Science,2-12-1-W8-74, O-okayama, Meguro-ku,Tokyo 152-8552 Japan,NA,NA,NA

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD, UNIT 45002, APO, AP, 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AOARD

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD-094071

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A new method of density-ratio estimation that can avoid density estimation was developed. This method gives the solution analytically just by solving a system of linear equations, so it can be applied to large-scale problems. Statistical properties of density ratio estimators have also theoretically investigated. Based on these results, practical machine learning algorithms were developed which include outlier detection, supervised dimensionality reduction, causal direction inference, independent component analysis, conditional density estimation, probabilistic classification, and their performance are shown to be comparable to the state-of-the-art. These algorithms have been applied to solve several real-world problems, which includes speaker identification, audio tagging, nonstationarity adaptation in brain-computer interface, efficient sample reuse in robot control, active exploration in robot control, feature selection in robot control, adaptation of lighting-condition change in face-based age recognition, detection of regions of interest in images , and multi-user adaptation in accelerometer-based human activity recognition.

15. SUBJECT TERMS
Machine Learning, Dimensinality Reduction, Non Stationary Adaptation, Outlier Detection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**211**	

1 Objectives

Given data samples, inferring the rules underlying behind the data is a major challenge in the area of machine learning. Such a learning method can be applied to solving a wide range of real-world problems such as robot control, bioinformatics, brain signal analysis, computer vision, speech recognition, and natural language processing. For this reason, a great deal of effort has been made recently to develop various machine learning algorithms. Our goal is to propose a general machine learning framework that can be employed for improving the state-of-the-art performance in these application domains.

More specifically, we establish a novel approach that accommodates various challenging machine learning tasks such as non-stationarity adaptation, outlier detection, dimensionality reduction, and conditional probability estimation. Our key idea is to directly estimate the *ratio* of two probability densities without going through density estimation, which is a novel paradigm in the machine learning community. Under the common concept of direct density-ratio estimation, we develop tailored machine learning algorithms for each task and show their usefulness in various application domains.

2 Status of effort

My project consists of four layers: (A) theory of density ratio estimation, (B) algorithms of density ratio estimation, (C) machine learning algorithms based on density ratio estimation, and (D) real-world application of density ratio estimation.

For the layer (B), we developed a new method of direct density-ratio estimation that is significantly more accurate than naively estimating the ratio via density estimation. We also developed density-ratio estimation algorithms that can handle correlated and rank-deficient data. We further proposed to combine density ratio estimation with dimensionality reduction, which improves the estimation accuracy in high-dimensional problems.

For the layer (A), we elucidated statistical properties of density ratio estimators for parametric and non-parametric cases. Such results are expected to contribute to further improving the estimation accuracy and computational efficiency of density ratio estimators.

For the layer (C), we employed our density-ratio estimation methods for designing practical machine learning algorithms including non-stationarity adaptation, outlier detection, supervised dimensionality reduction, causal direction inference, independent component analysis, conditional density estimation, probabilistic classification, and multi-task classification.

Finally, for the layer (D), we demonstrated the usefulness of the above machine learning algorithms in several real-world applications such as brain-computer interface, robot control, speech and audio recognition, image processing, and sensor data analysis.

3 Abstract

The basis of our project is a method to accurately estimate the ratio of probability densities. First, we developed a new method of density-ratio estimation that can avoid density estimation (publication 1). This method gives the solution *analytically* just by solving a system of linear equations, so it can be applied to large-scale problems. Furthermore, we developed methods of direct density-ratio estimation suitable for highly-correlated data (publication 2) and rank-deficient data (publication 3). We also proposed two methods for handling high-dimensional data: The first method is heuristic but computationally efficient (publication 4), and the other method is theoretically justifiable but computationally expensive (publication 5). We have also theoretically investigated statistical properties of density ratio estimators for parametric models (publication 6) and non-parametric models (publication 7).

Then we designed practical machine learning algorithms based on density ratio estimators. This includes outlier detection (publication 8) supervised dimensionality reduction (publication 9), causal direction inference (publication 10), independent component analysis (publication 11), conditional density estimation (publication 12), probabilistic classification (publication 13), and its multi-task version (publication 14). Through extensive experiments, these methods were shown to compare favorably with existing approaches in terms of accuracy and/or computational efficiency.

We also explored various real-world applications using the density ratio approaches. This includes speaker identification (publication 15), audio tagging (publication 16), non-stationarity adaptation in brain-computer interface (publication 17), efficient sample reuse in robot control (publications 18 and 19), active exploration in robot control (publication 20), feature selection in robot control (publication 21), adaptation of lighting-condition change in face-based age recognition (publication 22), detection of regions of interest in images (publication 23), and multi-user adaptation in accelerometer-based human activity recognition (publication 24).

Finally, the framework of density ratio estimation was published as review articles (publication 25 and 26). We will also publish a book from the MIT Press on our density-ratio-based non-stationarity adaptation approach (publication 27; approx. 300 pages, the final version was already sent to the publisher). Furthermore, the entire framework of the density ratio estimation will be published as a book from the Cambridge University Press (publication 28; beyond 400 pages, 90% of the material was already prepared).

4 Personnel Supported

The research activity of the following people was supported.

- Masashi Sugiyama (Tokyo Institute of Technology),
- Hirotaka Hachiya (Tokyo Institute of Technology),
- Makoto Yamada (Tokyo Institute of Technology),

- Gang Niu (Tokyo Institute of Technology),
- Takayuki Akiyama (Tokyo Institute of Technology),
- Pui-Ling Chui (Tokyo Institute of Technology).

5 Publications

During the 24 months, the following papers were published. The papers indicated by '*' were attached to this report, and all the publications are available from `http://sugiyama-www.cs.titech.ac.jp/~sugi/publications.html`.

1. * Kanamori, T., Hido, S., & Sugiyama, M. A least-squares approach to direct importance estimation. Journal of Machine Learning Research, vol.10 (Jul.), pp.1391–1445, 2009.

2. Yamada, M. & Sugiyama, M. Direct importance estimation with Gaussian mixture models. IEICE Transactions on Information and Systems, vol.E92-D, no.10, pp.2159–2162, 2009.

3. Yamada, M., Sugiyama, M., Wichern, G., & Simm, J. Direct importance estimation with a mixture of probabilistic principal component analyzers. IEICE Transactions on Information and Systems, vol.E93-D, no.10, pp.2846–2849, 2010.

4. Sugiyama, M., Kawanabe, M., & Chui, P. L. Dimensionality reduction for density ratio estimation in high-dimensional spaces. Neural Networks, vol.23, no.1, pp.44–59, 2010.

5. * Sugiyama, M., Yamada, M., von Buñau, P., Suzuki, T., Kanamori, T., & Kawanabe, M. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. Neural Networks, vol.24, no.2, pp.183–198, 2011.

6. * Kanamori, T., Suzuki, T., & Sugiyama, M. Theoretical analysis of density ratio estimation. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E93-A, no.4, pp.787–798, 2010.

7. Suzuki, T., Sugiyama, M., & Tanaka, T. Mutual information approximation via maximum likelihood estimation of density ratio. In Proceedings of 2009 IEEE International Symposium on Information Theory (ISIT2009), pp.463-467, Seoul, Korea, Jun. 28-Jul. 3, 2009.

8. Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., & Kanamori, T. Statistical outlier detection using direct density ratio estimation. Knowledge and Information Systems, vol.26, no.2, pp.309–336, 2011.

9. Suzuki, T. & Sugiyama, M. Sufficient dimension reduction via squared-loss mutual information estimation. In Proceedings of Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS2010), JMLR Workshop and Conference Proceedings, vol.9, pp.804–811, Sardinia, Italy, May 13-15, 2010.

10. Yamada, M. & Sugiyama, M. Dependence minimizing regression with model selection for non-linear causal inference under non-Gaussian noise. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI2010), pp.643–648, Atlanta, Georgia, USA, Jul. 11-15, 2010.

11. * Suzuki, T. & Sugiyama, M. Least-squares independent component analysis. Neural Computation, vol.23, no.1, pp.284–301, 2011.

12. Sugiyama, M., Takeuchi, I., Kanamori, T., Suzuki, T., Hachiya, H., & Okanohara, D. Least-squares conditional density estimation. IEICE Transactions on Information and Systems, vol.E93-D, no.3, pp.583–594, 2010.

13. * Sugiyama, M. Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting. IEICE Transactions on Information and Systems, vol.E93-D, no.10, pp.2690–2701, 2010.

14. Simm, J., Sugiyama, M., & Kato, T. Computationally efficient multi-task learning with least-squares probabilistic classifiers. IPSJ Transactions on Computer Vision and Applications, vol.3, pp.1–8, 2011.

15. Yamada, M., Sugiyama, M., & Matsui, T. Semi-supervised speaker identification under covariate shift. Signal Processing, vol.90, no.8, pp.2353–2361, 2010.

16. Wichern, G., Yamada, M., Thornburg, H., Sugiyama, M., & Spanias, A. Automatic audio tagging using covariate shift adaptation. To appear in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2010), Dallas, Texas, USA, Mar. 14-19, 2010.

17. * Li, Y., Kambara, H., Koike, Y., & Sugiyama, M. Application of covariate shift adaptation techniques in brain computer interfaces. IEEE Transactions on Biomedical Engineering, vol.57, no.6, pp.1318–1324, 2010.

18. Hachiya, H., Akiyama, T., Sugiyama, M., & Peters, J. Adaptive importance sampling for value function approximation in off-policy reinforcement learning. Neural Networks, vol.22, no.10, pp.1399–1410, 2009.

19. Hachiya, H., Peters, J., & Sugiyama, M. Efficient sample reuse in EM-based policy search. In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD2009), pp.469–484, Bled, Slovenia, Sep. 7-11, 2009.

20. * Akiyama, T., Hachiya, H., & Sugiyama, M. Efficient exploration through active learning for value function approximation in reinforcement learning. Neural Networks, vol.23, no.5, pp.639–648, 2010.

21. Hachiya, H. & Sugiyama, M. Feature selection for reinforcement learning: Evaluating implicit state-reward dependency via conditional mutual information. In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD2010), pp.474–489, Barcelona, Spain, Sep. 20-24, 2010.

22. * Ueki, K., Sugiyama, M., & Ihara, Y. Lighting condition adaptation for perceived age estimation. IEICE Transactions on Information and Systems, to appear.

23. Yamanaka, M., Matsugu, M. & Sugiyama, M. Automatic detection of regions of interest using multiple visual saliency measures based on density ratio estimation. In Proceedings of Vision Engineering Workshop 2010 (ViEW2010), pp.7–8, Yokohama, Japan, Dec. 9-10, 2010.

24. Hachiya, H., Sugiyama, M. & Ueda, N. Coping with new user problems: Transfer learning in accelerometer-based human activity recognition. NIPS 2010 Workshop on Transfer Learning by Learning Rich Generative Models, Whistler, British Columbia, Canada, Dec. 11, 2010.

25. Sugiyama, M., Kanamori, T., Suzuki, T., Hido, S., Sese, J., Takeuchi, I., & Wang, L. A density-ratio framework for statistical data processing. IPSJ Transactions on Computer Vision and Applications, vol.1, pp.183–208, 2009.

26. Sugiyama, M. A new approach to machine learning based on density ratios. Proceedings of the Institute of Statistical Mathematics, vol.58, no.2, pp.141–155, 2010.

27. Sugiyama, M. & Kawanabe, M. Covariate Shift Adaptation: Towards Machine Learning under Non-Stationary Environment, MIT Press, Cambridge, MA, USA, to appear.

28. Sugiyama, M., Suzuki, T., & Kanamori, T. Density Ratio Estimation in Machine Learning: A Versatile Tool for Statistical Data Processing, Cambridge University Press, Cambridge, UK, in preparation.

6 Interactions

On Jun. 25, 2009 (at the AOARD Roppongi office), Nov. 3, 2009 (at the ACML conference), and Dec. 21, 2010 (at my office at Tokyo Tech), I had technical discussions with my program manager, Dr. Hiroshi Motoda, and received detailed comments and suggestions.

Below is the list of my presentations related to the project.

1. Dec. 22, 2010. Toshiba, Kawasaki, Japan.

2. Dec. 1, 2010. Fujitsu, Kawasaki, Japan.
3. Nov. 30, 2010. GCOE Symposium at Tokyo Institute of Technology, Tokyo, Japan.
4. Nov. 25, 2010. LEAP Symposium at Tokyo Institute of Technology, Tokyo, Japan.
5. Nov. 24, 2010. Kyoto University, Kyoto, Japan.
6. Oct. 12, 2010. Sony, Tokyo, Japan.
7. Oct. 4, 2010. Google, Tokyo, Japan.
8. Sep. 20, 2010. Yahoo, Barcelona, Spain.
9. Sep. 14, 2010. Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany.
10. Sep. 13, 2010. Technical University Berlin, Berlin, Germany.
11. Aug. 3, 2010. Aichi Science and Technology Foundation, Nagoya, Japan.
12. Aug. 2, 2010. Nagoya Institute of Technology, Nagoya, Japan.
13. Jul. 20, 2010. NEC, Kawasaki, Japan.
14. Jul. 15, 2010. Georgia Institute of Technology, Atlanta, GA, USA.
15. Jun. 14, 2010. Institute of Electronics, Information and Communication Engineers, Tokyo, Japan.
16. Jun. 7, 2010. Institute of Systems, Control and Information Engineers, Osaka, Japan.
17. May 25, 2010. The Society of Instrument and Control Engineers, Tokyo, Japan.
18. Apr. 26, 2010. Carnegie Mellon University, Pittsburgh, PA, USA.
19. Apr. 21, 2010. NEC Soft, Tokyo, Japan.
20. Mar. 8, 2010. Kyoto University, Kyoto, Japan.
21. Dec. 21, 2009. Science Council of Japan, Tokyo, Japan.
22. Dec. 3, 2009. GCOE Symposium at Tokyo Institute of Technology, Tokyo, Japan.
23. Nov. 27, 2009. National Institute of Information and Communications Technology, Kyoto, Japan.
24. Nov. 17, 2009. NECsoft, Tokyo, Japan.

25. Nov. 3, 2009. 1st Asian Conference on Machine Learning (ACML2009), Nanjing, China.

26. Oct. 16, 2009. NLP workshop, Tokyo, Japan.

27. Jun. 25, 2009. Asian Office of Aerospace Research & Development, Tokyo, Japan.

28. May. 21, 2009. NTT Communication Science Laboratories, Kanagawa, Japan.

7 Inventions

None.

8 Honors/Award

I received two awards related to the current project.

1. May 13, 2010. Incentive Award, The Institute of Electronics, Information and Communication Engineers (IEICE), PRMU Technical Group.

2. Jun. 10, 2010. Incentive Award, Japanese Society for Artificial Intelligence (JSAI) SIG-DMSM

The award 1 was given for a conference version of the publication 20, while the award 2 was given for a conference version of the publication 13.

9 Archival Documentation

Selected journal articles (1, 5, 6, 11, 13, 17, 20, and 22) are attached as archival documentation. All the publications listed in Section 5 are available from `http://sugiyama-www.cs.titech.ac.jp/~sugi/publications.html`.

10 Software

Implementation of various density-ratio methods (mostly in MATLAB) is available from my web page: `http://sugiyama-www.cs.titech.ac.jp/~sugi/software/index.html`.

Journal of Machine Learning Research 10 (2009) 1391-1445 Submitted 3/08; Revised 4/09; Published 7/09

A Least-squares Approach to Direct Importance Estimation*

Takafumi Kanamori KANAMORI@IS.NAGOYA-U.AC.JP
Department of Computer Science and Mathematical Informatics
Nagoya University
Furocho, Chikusaku, Nagoya 464-8603, Japan

Shohei Hido HIDO@JP.IBM.COM
IBM Research
Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato-shi, Kanagawa 242-8502, Japan

Masashi Sugiyama SUGI@CS.TITECH.AC.JP
Department of Computer Science
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan

Editor: Bianca Zadrozny

Abstract

We address the problem of estimating the ratio of two probability density functions, which is often referred to as the *importance*. The importance values can be used for various succeeding tasks such as *covariate shift adaptation* or *outlier detection*. In this paper, we propose a new importance estimation method that has a closed-form solution; the leave-one-out cross-validation score can also be computed analytically. Therefore, the proposed method is computationally highly efficient and simple to implement. We also elucidate theoretical properties of the proposed method such as the convergence rate and approximation error bounds. Numerical experiments show that the proposed method is comparable to the best existing method in accuracy, while it is computationally more efficient than competing approaches.

Keywords: importance sampling, covariate shift adaptation, novelty detection, regularization path, leave-one-out cross validation

1. Introduction

In the context of *importance sampling* (Fishman, 1996), the ratio of two probability density functions is called the *importance*. The problem of estimating the importance is attracting a great deal of attention these days since the importance can be used for various succeeding tasks such as *covariate shift adaptation* or *outlier detection*.

> **Covariate Shift Adaptation:** Covariate shift is a situation in supervised learning where the distributions of inputs change between the training and test phases but the conditional distribution of outputs given inputs remains unchanged (Shimodaira, 2000; Quiñonero-Candela et al., 2008). Covariate shift is conceivable in many real-world

*. A MATLAB® or R implementation of the proposed importance estimation algorithm, *unconstrained Least-Squares Importance Fitting* (uLSIF), is available from `http://sugiyama-www.cs.titech.ac.jp/~sugi/software/uLSIF/`.

applications such as bioinformatics (Baldi and Brunak, 1998; Borgwardt et al., 2006), brain-computer interfaces (Wolpaw et al., 2002; Sugiyama et al., 2007), robot control (Sutton and Barto, 1998; Hachiya et al., 2008), spam filtering (Bickel and Scheffer, 2007), and econometrics (Heckman, 1979). Under covariate shift, standard learning techniques such as maximum likelihood estimation or cross-validation are biased and therefore unreliable—the bias caused by covariate shift can be compensated by weighting the loss function according to the importance (Shimodaira, 2000; Zadrozny, 2004; Sugiyama and Müller, 2005; Sugiyama et al., 2007; Huang et al., 2007; Bickel et al., 2007).

Outlier Detection: The outlier detection task addressed here is to identify irregular samples in a validation data set based on a model data set that only contains regular samples (Schölkopf et al., 2001; Tax and Duin, 2004; Hodge and Austin, 2004; Hido et al., 2008). The values of the importance for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus the values of the importance could be used as an index of the degree of outlyingness.

Below, we refer to the two sets of samples as the *training* set and the *test* set.

A naive approach to estimating the importance is to first estimate the training and test density functions from the sets of training and test samples separately, and then take the ratio of the estimated densities. However, density estimation is known to be a hard problem particularly in high-dimensional cases if we do not have simple and good parametric density models (Vapnik, 1998; Härdle et al., 2004). In practice, such an appropriate parametric model may not be available and therefore this naive approach is not so effective.

To cope with this problem, direct importance estimation methods which do not involve density estimation have been developed recently. The *kernel mean matching* (KMM) method (Huang et al., 2007) directly gives estimates of the importance at the training inputs by matching the two distributions efficiently based on a special property of *universal reproducing kernel Hilbert spaces* (Steinwart, 2001). The optimization problem involved in KMM is a convex quadratic program, so the unique global optimal solution can be obtained using a standard optimization software. However, the performance of KMM depends on the choice of tuning parameters such as the kernel parameter and the regularization parameter. For the kernel parameter, a popular heuristic of using the median distance between samples as the Gaussian width could be useful in some cases (Schölkopf and Smola, 2002; Song et al., 2007). However, there seems no strong justification for this heuristic and the choice of other tuning parameters is still open.

A probabilistic classifier that separates training samples from test samples can be used for directly estimating the importance, for example, a *logistic regression* (LogReg) classifier (Qin, 1998; Cheng and Chu, 2004; Bickel et al., 2007). Maximum likelihood estimation of LogReg models can be formulated as a convex optimization problem, so the unique global optimal solution can be obtained. Furthermore, since the LogReg-based method only involves a standard supervised classification problem, the tuning parameters such as the kernel width and the regularization parameter can be optimized based on the standard cross-validation procedure. This is a very useful property in practice.

The *Kullback-Leibler importance estimation procedure* (KLIEP) (Sugiyama et al., 2008b; Nguyen et al., 2008) also directly gives an estimate of the importance function by matching the two distributions in terms of the Kullback-Leibler divergence (Kullback and Leibler, 1951). The optimization

problem involved in KLIEP is convex, so the unique global optimal solution—which tends to be sparse—can be obtained, when linear importance models are used. In addition, the tuning parameters in KLIEP can be optimized based on a variant of cross-validation.

As reviewed above, LogReg and KLIEP are more advantageous than KMM since the tuning parameters can be objectively optimized based on cross-validation. However, optimization procedures of LogReg and KLIEP are less efficient in computation than KMM due to high non-linearity of the objective functions to be optimized—more specifically, exponential functions induced by the LogReg model or the log function induced by the Kullback-Leibler divergence. The purpose of this paper is to develop a new importance estimation method that is equipped with a build-in model selection procedure as LogReg and KLIEP and is computationally more efficient than LogReg and KLIEP.

Our basic idea is to formulate the direct importance estimation problem as a least-squares function fitting problem. This formulation allows us to cast the optimization problem as a convex quadratic program, which can be efficiently solved using a standard quadratic program solver. Cross-validation can be used for optimizing the tuning parameters such as the kernel width or the regularization parameter. We call the proposed method *least-squares importance fitting* (LSIF). We further show that the solutions of LSIF is piecewise linear with respect to the ℓ_1-regularization parameter and the entire regularization path (that is, all solutions for different regularization parameter values) can be computed efficiently based on the *parametric optimization technique* (Best, 1982; Efron et al., 2004; Hastie et al., 2004). Thanks to this regularization path tracking algorithm, LSIF is computationally efficient in model selection scenarios. Note that in the regularization path tracking algorithm, we can trace the solution path without a quadratic program solver—we just need to compute matrix inverses.

LSIF is shown to be efficient in computation, but it tends to share a common weakness of regularization path tracking algorithms, that is, *accumulation of numerical errors* (Scheinberg, 2006). The numerical problem tends to be severe if there are many change points in the regularization path. To cope with this problem, we develop an approximation algorithm in the same least-squares framework. The approximation version of LSIF, which we call *unconstrained LSIF* (uLSIF), allows us to obtain the closed-form solution that can be computed just by solving a system of linear equations. Thus uLSIF is numerically stable when regularized properly. Moreover, the leave-one-out cross-validation score for uLSIF can also be computed analytically (cf. Wahba, 1990; Cawley and Talbot, 2004), which significantly improves the computational efficiency in model selection scenarios. We experimentally show that the accuracy of uLSIF is comparable to the best existing method while its computation is faster than other methods in covariate shift adaptation and outlier detection scenarios.

Our contributions in this paper are summarized as follows. A proposed density-ratio estimation method, LSIF, is equipped with cross-validation (which is an advantage over KMM) and is computationally efficient thanks to regularization path tracking (which is an advantage over KLIEP and LogReg). Furthermore, uLSIF is computationally even more efficient since its solution and leave-one-out cross-validation score can be computed analytically in a stable manner. The proposed methods, LSIF and uLSIF, are similar in spirit to KLIEP, but the loss functions are different: KLIEP uses the log loss while LSIF and uLSIF use the squared loss. The difference of the log functions allows us to improve computational efficiency significantly.

The rest of this paper is organized as follows. In Section 2, we propose a new importance estimation procedure based on least-squares fitting (LSIF) and show its theoretical properties. In

Section 3, we develop an approximation algorithm (uLSIF) which can be computed efficiently. In Section 4, we illustrate how the proposed methods behave using a toy data set. In Section 5, we discuss the characteristics of existing approaches in comparison with the proposed methods and show that uLSIF could be a useful alternative to the existing methods. In Section 6, we experimentally compare the performance of uLSIF and existing methods. Finally in Section 7, we summarize our contributions and outline future prospects. Those who are interested in practical implementation may skip the theoretical analyses in Sections 2.3, 3.2, and 3.3.

2. Direct Importance Estimation

In this section, we propose a new method of direct importance estimation.

2.1 Formulation and Notation

Let $\mathcal{D} \subset (\mathbb{R}^d)$ be the data domain and suppose we are given independent and identically distributed (i.i.d.) training samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ from a training distribution with density $p_{\mathrm{tr}}(x)$ and i.i.d. test samples $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ from a test distribution with density $p_{\mathrm{te}}(x)$:

$$\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}} \overset{i.i.d.}{\sim} p_{\mathrm{tr}}(x),$$
$$\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \overset{i.i.d.}{\sim} p_{\mathrm{te}}(x).$$

We assume that the training density is strictly positive, that is,

$$p_{\mathrm{tr}}(x) > 0 \text{ for all } x \in \mathcal{D}.$$

The goal of this paper is to estimate the *importance* $w(x)$ from $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$:

$$w(x) = \frac{p_{\mathrm{te}}(x)}{p_{\mathrm{tr}}(x)},$$

which is non-negative by definition. Our key restriction is that we want to avoid estimating densities $p_{\mathrm{te}}(x)$ and $p_{\mathrm{tr}}(x)$ when estimating the importance $w(x)$.

2.2 Least-squares Approach to Direct Importance Estimation

Let us model the importance $w(x)$ by the following linear model:

$$\widehat{w}(x) = \sum_{\ell=1}^{b} \alpha_\ell \varphi_\ell(x), \tag{1}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_b)^\top$ are parameters to be learned from data samples, $^\top$ denotes the transpose of a matrix or a vector, and $\{\varphi_\ell(x)\}_{\ell=1}^{b}$ are basis functions such that

$$\varphi_\ell(x) \geq 0 \text{ for all } x \in \mathcal{D} \text{ and for } \ell = 1, 2, \ldots, b.$$

Note that b and $\{\varphi_\ell(x)\}_{\ell=1}^{b}$ could be dependent on the samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, for example, *kernel* models are also allowed. We explain how the basis functions $\{\varphi_\ell(x)\}_{\ell=1}^{b}$ are chosen in Section 2.5.

We determine the parameters $\{\alpha_\ell\}_{\ell=1}^b$ in the model $\widehat{w}(x)$ so that the following squared error J_0 is minimized:

$$\begin{aligned}
J_0(\alpha) &= \frac{1}{2}\int \left(\widehat{w}(x) - w(x)\right)^2 p_{\mathrm{tr}}(x)dx \\
&= \frac{1}{2}\int \widehat{w}(x)^2 p_{\mathrm{tr}}(x)dx - \int \widehat{w}(x)w(x)p_{\mathrm{tr}}(x)dx + \frac{1}{2}\int w(x)^2 p_{\mathrm{tr}}(x)dx \\
&= \frac{1}{2}\int \widehat{w}(x)^2 p_{\mathrm{tr}}(x)dx - \int \widehat{w}(x)p_{\mathrm{te}}(x)dx + \frac{1}{2}\int w(x)^2 p_{\mathrm{tr}}(x)dx,
\end{aligned}$$

where in the second term the probability density $p_{\mathrm{tr}}(x)$ is canceled with that included in $w(x)$. The squared loss $J_0(\alpha)$ is defined as the expectation under the probability of training samples. In covariate shift adaptation (see Section 6.2) and outlier detection (see Section 6.3), the importance values on the training samples are used. Thus, the definition of $J_0(\alpha)$ well agrees with our goal.

The last term of $J_0(\alpha)$ is a constant and therefore can be safely ignored. Let us denote the first two terms by J:

$$\begin{aligned}
J(\alpha) &= \frac{1}{2}\int \widehat{w}(x)^2 p_{\mathrm{tr}}(x)dx - \int \widehat{w}(x)p_{\mathrm{te}}(x)dx \\
&= \frac{1}{2}\sum_{\ell,\ell'=1}^{b} \alpha_\ell \alpha_{\ell'} \left(\int \varphi_\ell(x)\varphi_{\ell'}(x)p_{\mathrm{tr}}(x)dx\right) - \sum_{\ell=1}^{b} \alpha_\ell \left(\int \varphi_\ell(x)p_{\mathrm{te}}(x)dx\right) \\
&= \frac{1}{2}\alpha^\top H\alpha - h^\top \alpha, \qquad (2)
\end{aligned}$$

where H is the $b \times b$ matrix with the (ℓ,ℓ')-th element

$$H_{\ell,\ell'} = \int \varphi_\ell(x)\varphi_{\ell'}(x)p_{\mathrm{tr}}(x)dx, \qquad (3)$$

and h is the b-dimensional vector with the ℓ-th element

$$h_\ell = \int \varphi_\ell(x)p_{\mathrm{te}}(x)dx.$$

Approximating the expectations in J by empirical averages, we obtain

$$\begin{aligned}
\widehat{J}(\alpha) &= \frac{1}{2n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}})^2 - \frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \widehat{w}(x_j^{\mathrm{te}}) \\
&= \frac{1}{2}\sum_{\ell,\ell'=1}^{b} \alpha_\ell \alpha_{\ell'} \left(\frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})\varphi_{\ell'}(x_i^{\mathrm{tr}})\right) - \sum_{\ell=1}^{b} \alpha_\ell \left(\frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})\right) \\
&= \frac{1}{2}\alpha^\top \widehat{H}\alpha - \widehat{h}^\top \alpha,
\end{aligned}$$

where $\widehat{H}$ is the $b \times b$ matrix with the (ℓ,ℓ')-th element

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})\varphi_{\ell'}(x_i^{\mathrm{tr}}), \qquad (4)$$

and $\widehat{h}$ is the b-dimensional vector with the ℓ-th element

$$\widehat{h}_\ell = \frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}}). \tag{5}$$

Taking into account the non-negativity of the importance function $w(x)$, we can formulate our optimization problem as follows.

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^b} \quad & \left[\frac{1}{2} \alpha^\top \widehat{H} \alpha - \widehat{h}^\top \alpha + \lambda 1_b^\top \alpha \right] \\ & \text{subject to } \alpha \geq 0_b, \end{aligned} \tag{6}$$

where 0_b and 1_b are the b-dimensional vectors with all zeros and ones, respectively; the vector inequality $\alpha \geq 0_b$ is applied in the element-wise manner, that is, $\alpha_\ell \geq 0$ for $\ell = 1, 2, \ldots, b$. In Eq. (6), we included a penalty term $\lambda 1_b^\top \alpha$ for regularization purposes, where λ (≥ 0) is a regularization parameter. The above is a convex quadratic programming problem and therefore the unique global optimal solution can be computed efficiently by a standard optimization package. We call this method *Least-Squares Importance Fitting* (LSIF).

We can also use the ℓ_2-regularizer $\alpha^\top \alpha$ instead of the ℓ_1-regularizer $1_b^\top \alpha$ without changing the computational property. However, using the ℓ_1-regularizer would be more advantageous since the solution tends to be sparse (Williams, 1995; Tibshirani, 1996; Chen et al., 1998). Furthermore, as shown in Section 2.6, the use of the ℓ_1-regularizer allows us to compute the entire regularization path efficiently (Best, 1982; Efron et al., 2004; Hastie et al., 2004). The ℓ_2-regularization method will be used for theoretical analysis in Section 3.3.

2.3 Convergence Analysis of LSIF

Here, we theoretically analyze the convergence property of the solution $\widehat{\alpha}$ of the LSIF algorithm; practitioners may skip this theoretical analysis.

Let $\widehat{\alpha}(\lambda)$ be the solution of the LSIF algorithm with regularization parameter λ, and let $\alpha^*(\lambda)$ be the optimal solution of the 'ideal' problem:

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^b} \quad & \left[\frac{1}{2} \alpha^\top H \alpha - h^\top \alpha + \lambda 1_b^\top \alpha \right] \\ & \text{subject to } \alpha \geq 0_b. \end{aligned} \tag{7}$$

Below, we theoretically investigate the *learning curve* (Amari et al., 1992) of LSIF, that is, we elucidate the relation between $J(\widehat{\alpha}(\lambda))$ and $J(\alpha^*(\lambda))$ in terms of the expectation over all possible training and test samples as a function of the number of samples.

Let $\mathbb{E}$ be the expectation over all possible training samples of size n_{tr} and all possible test samples of size n_{te}. Let $\mathcal{A} \subset \{1, 2, \ldots, b\}$ be the set of *active* indices (Boyd and Vandenberghe, 2004), that is,

$$\mathcal{A} = \{\ell \mid \alpha_\ell^*(\lambda) = 0,\ \ell = 1, 2, \ldots, b\}.$$

For the active set $\mathcal{A} = \{j_1, j_2, \ldots, j_{|\mathcal{A}|}\}$ with $j_1 < j_2 < \cdots < j_{|\mathcal{A}|}$, let E be the $|\mathcal{A}| \times b$ indicator matrix with the (i, j)-th element

$$E_{i,j} = \begin{cases} 1 & j = j_i, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, let $\widehat{\mathcal{A}}$ be the active set of $\widehat{\alpha}(\lambda)$:

$$\widehat{\mathcal{A}} = \{\ell \mid \widehat{\alpha}_\ell(\lambda) = 0,\ \ell = 1,2,\ldots,b\}.$$

For the active set $\widehat{\mathcal{A}} = \{\widehat{j}_1, \widehat{j}_2, \ldots, \widehat{j}_{|\widehat{\mathcal{A}}|}\}$ with $\widehat{j}_1 < \widehat{j}_2 < \cdots < \widehat{j}_{|\widehat{\mathcal{A}}|}$, let $\widehat{E}$ be the $|\widehat{\mathcal{A}}| \times b$ indicator matrix with the (i,j)-th element similarly defined by

$$\widehat{E}_{i,j} = \begin{cases} 1 & j = \widehat{j}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

First, we show the optimality condition of (6) which will be used in the following theoretical analyses. The *Lagrangian* of the optimization problem (6) is given as

$$L(\alpha, \xi) = \frac{1}{2}\alpha^\top \widehat{H}\alpha - \widehat{h}^\top \alpha + \lambda 1_b^\top \alpha - \xi^\top \alpha,$$

where ξ is the b-dimensional *Lagrange multiplier* vector. Then the *Karush-Kuhn-Tucker (KKT) conditions* (Boyd and Vandenberghe, 2004) are expressed as follows:

$$\widehat{H}\alpha - \widehat{h} + \lambda 1_b - \xi = 0_b, \tag{9}$$
$$\alpha \geq 0_b,$$
$$\xi \geq 0_b,$$
$$\xi_\ell \alpha_\ell = 0 \text{ for } \ell = 1,2,\ldots,b. \tag{10}$$

Let $\widehat{\xi}'(\lambda)$ be the $|\widehat{\mathcal{A}}|$-dimensional vector with the i-th element being the $\widehat{j}_i$-th element of $\widehat{\xi}(\lambda)$:

$$\widehat{\xi}'_i(\lambda) = \widehat{\xi}_{\widehat{j}_i}(\lambda), \qquad i = 1,\ldots,|\widehat{\mathcal{A}}|. \tag{11}$$

We assume that $\widehat{\xi}'(\lambda)$ only contains non-zero elements of $\widehat{\xi}(\lambda)$. Let $\widehat{G}$ be

$$\widehat{G} = \begin{pmatrix} \widehat{H} & -\widehat{E}^\top \\ -\widehat{E} & O_{|\widehat{\mathcal{A}}| \times |\widehat{\mathcal{A}}|} \end{pmatrix},$$

where $O_{|\widehat{\mathcal{A}}| \times |\widehat{\mathcal{A}}|}$ is the $|\widehat{\mathcal{A}}| \times |\widehat{\mathcal{A}}|$ matrix with all zeros. Then Eqs. (9) and (10) are together expressed in a matrix form as

$$\widehat{G}\begin{pmatrix} \widehat{\alpha}(\lambda) \\ \widehat{\xi}'(\lambda) \end{pmatrix} = \begin{pmatrix} \widehat{h} - \lambda 1_b \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}. \tag{12}$$

Regarding the matrix $\widehat{G}$, we have the following lemma:

Lemma 1 *The matrix $\widehat{G}$ is invertible if $\widehat{H}$ is invertible.*

The proof of the above lemma is given in Appendix A. Below, we assume that $\widehat{H}$ is invertible. Then the inverse of $\widehat{G}$ exists and multiplying $\widehat{G}^{-1}$ from the left-hand side of Eq. (12) yields

$$\begin{pmatrix} \widehat{\alpha}(\lambda) \\ \widehat{\xi}'(\lambda) \end{pmatrix} = \widehat{G}^{-1}\begin{pmatrix} \widehat{h} - \lambda 1_b \\ 0_{|\mathcal{A}|} \end{pmatrix}. \tag{13}$$

The following inversion formula holds for block matrices (Petersen and Pedersen, 2007):

$$\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix}^{-1} = \begin{pmatrix} M_1^{-1} + M_1^{-1}M_2M_0^{-1}M_3M_1^{-1} & -M_1^{-1}M_2M_0^{-1} \\ -M_0^{-1}M_3M_1^{-1} & M_0^{-1} \end{pmatrix}, \tag{14}$$

where

$$M_0 = M_4 - M_3M_1^{-1}M_2.$$

Applying Eq. (14) to Eq. (13), we have

$$\widehat{\alpha}(\lambda) = \widehat{A}(\widehat{h} - \lambda 1_b), \tag{15}$$

where $\widehat{A}$ is defined by

$$\widehat{A} = \widehat{H}^{-1} - \widehat{H}^{-1}\widehat{E}^\top(\widehat{E}\widehat{H}^{-1}\widehat{E}^\top)^{-1}\widehat{E}\widehat{H}^{-1}. \tag{16}$$

When the Lagrange multiplier vector satisfies

$$\xi_\ell^*(\lambda) > 0 \text{ for all } \ell \in \mathcal{A}, \tag{17}$$

we say that the *strict complementarity condition* is satisfied (Bertsekas et al., 2003). An important consequence of strict complementarity is that the optimal solution and the Lagrange multipliers of convex quadratic problems are uniquely determined. Then we have the following theorem.

Theorem 2 *Let P be the probability over all possible training samples of size n_{tr} and test samples of size n_{te}. Let $\xi^*(\lambda)$ be the Lagrange multiplier vector of the problem* (7) *and suppose $\xi^*(\lambda)$ satisfies the strict complementarity condition* (17)*. Then, there exists a positive constant $c > 0$ and a natural number N such that for $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\} \geq N$,*

$$P(\widehat{\mathcal{A}} \neq \mathcal{A}) < e^{-c\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}}.$$

The proof of the above theorem is given in Appendix B. Theorem 2 shows that the probability that the active set $\widehat{\mathcal{A}}$ of the empirical problem (6) is different from the active set $\mathcal{A}$ of the ideal problem (7) is exponentially small. Thus we may regard $\widehat{\mathcal{A}} = \mathcal{A}$ in practice.

Let A be the 'ideal' counterpart of $\widehat{A}$:

$$A = H^{-1} - H^{-1}E^\top(EH^{-1}E^\top)^{-1}EH^{-1},$$

and let $C_{w,w'}$ be the $b \times b$ covariance matrix with the (ℓ, ℓ')-th element being the covariance between $w(x)\varphi_\ell(x)$ and $w'(x)\varphi_{\ell'}(x)$ under $p_{\mathrm{tr}}(x)$. Let

$$w^*(x) = \sum_{\ell=1}^{b} \alpha_\ell^*(\lambda)\varphi_\ell(x),$$

$$v(x) = \sum_{\ell=1}^{b} [A1_b]_\ell \varphi_\ell(x).$$

Let

$$f(n) = \omega(g(n))$$

denote that $f(n)$ asymptotically dominates $g(n)$; more precisely, for all $C > 0$, there exists n_0 such that

$$|Cg(n)| < |f(n)| \quad \text{for all } n > n_0.$$

Then we have the following theorem.

Theorem 3 *Assume that*

(a) *The optimal solution of the problem* (7) *satisfies the strict complementarity condition* (17)*.*

(b) n_{tr} *and* n_{te} *satisfy*

$$n_{\mathrm{te}} = \omega(n_{\mathrm{tr}}^2). \tag{18}$$

Then, for any $\lambda \geq 0$*, we have*

$$\mathbb{E}[J(\widehat{\alpha}(\lambda))] = J(\alpha^*(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(A(C_{w^*,w^*} - 2\lambda C_{w^*,v})) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{19}$$

The proof of the above theorem is given in Appendix C. This theorem elucidates the learning curve of LSIF up to the order of n_{tr}^{-1}. In Section 2.4.1, we discuss practical implications of this theorem.

2.4 Model Selection for LSIF

The practical performance of LSIF depends on the choice of the regularization parameter λ and basis functions $\{\varphi_\ell(x)\}_{\ell=1}^b$ (which we refer to as a *model*). Since our objective is to minimize the cost function J defined in Eq. (2), it is natural to determine the model such that J is minimized.

However, the value of the cost function J is inaccessible since it includes the expectation over unknown probability density functions $p_{\mathrm{tr}}(x)$ and $p_{\mathrm{te}}(x)$. The value of the empirical cost $\widehat{J}$ may be regarded as an estimate of J, but this is not useful for model selection purposes since it is heavily biased—the bias is caused by the fact that the same samples are used twice for learning the parameter α and estimating the value of J. Below, we give two practical methods of estimating the value of J in more precise ways.

2.4.1 INFORMATION CRITERION

In the same way as Theorem 3, we can obtain an asymptotic expansion of the empirical cost $\mathbb{E}\left[\widehat{J}(\widehat{\alpha}(\lambda))\right]$ as follows:

$$\mathbb{E}[\widehat{J}(\widehat{\alpha}(\lambda))] = J(\alpha^*(\lambda)) - \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(A(C_{w^*,w^*} + 2\lambda C_{w^*,v})) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{20}$$

Combining Eqs. (19) and (20), we have

$$\mathbb{E}[J(\widehat{\alpha}(\lambda))] = \mathbb{E}[\widehat{J}(\widehat{\alpha}(\lambda))] + \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(AC_{w^*,w^*}) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).$$

From this, we can immediately obtain an *information criterion* (Akaike, 1974; Konishi and Kitagawa, 1996) for LSIF:

$$\widehat{J}^{(\mathrm{IC})} = \widehat{J}(\widehat{\alpha}(\lambda)) + \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(\widehat{A}\widehat{C}_{\widehat{w},\widehat{w}}),$$

where $\widehat{A}$ is defined by Eq. (16). $\widehat{E}$ is defined by Eq. (8) and $\widehat{C}_{w,w'}$ is the $b \times b$ covariance matrix with the (ℓ,ℓ')-th element being the covariance between $w(x)\varphi_\ell(x)$ and $w'(x)\varphi_{\ell'}(x)$ over $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. Since $\widehat{A}$ and $\widehat{C}_{\widehat{w},\widehat{w}}$ are consistent estimators of A and C_{w^*,w^*}, the above information criterion is unbiased up to the order of n_{tr}^{-1}.

Note that the term $\mathrm{tr}(\widehat{A}\widehat{C}_{\widehat{w},\widehat{w}})$ may be interpreted as the *effective dimension* of the model (Moody, 1992). Indeed, when $\widehat{w}(x)=1$, we have $\widehat{H}=\widehat{C}_{\widehat{w},\widehat{w}}$ and thus

$$\mathrm{tr}(\widehat{A}\widehat{C}_{\widehat{w},\widehat{w}})=\mathrm{tr}(I_b)-\mathrm{tr}(E\widehat{C}_{\widehat{w},\widehat{w}}^{-1}E^\top(E\widehat{C}_{\widehat{w},\widehat{w}}^{-1}E^\top)^{-1})=b-|\widehat{\mathcal{A}}|,$$

which is the dimension of the *face* on which $\widehat{\alpha}(\lambda)$ lies.

2.4.2 CROSS-VALIDATION

Although the information criterion derived above is more accurate than just a naive empirical estimator, its accuracy is guaranteed only asymptotically. Here, we employ cross-validation for estimating $J(\widehat{\alpha})$, which has an accuracy guarantee for finite samples.

First, the training samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and test samples $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ are divided into R disjoint subsets $\{\mathcal{X}_r^{\mathrm{tr}}\}_{r=1}^R$ and $\{\mathcal{X}_r^{\mathrm{te}}\}_{r=1}^R$, respectively. Then an importance estimate $\widehat{w}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}(x)$ is obtained using $\{\mathcal{X}_j^{\mathrm{tr}}\}_{j\neq r}$ and $\{\mathcal{X}_j^{\mathrm{te}}\}_{j\neq r}$ (that is, without $\mathcal{X}_r^{\mathrm{tr}}$ and $\mathcal{X}_r^{\mathrm{te}}$), and the cost J is approximated using the held-out samples $\mathcal{X}_r^{\mathrm{tr}}$ and $\mathcal{X}_r^{\mathrm{te}}$ as

$$\widehat{J}^{(\mathrm{CV})}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}=\frac{1}{2|\mathcal{X}_r^{\mathrm{tr}}|}\sum_{x^{\mathrm{tr}}\in\mathcal{X}_r^{\mathrm{tr}}}\widehat{w}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}(x^{\mathrm{tr}})^2-\frac{1}{|\mathcal{X}_r^{\mathrm{te}}|}\sum_{x^{\mathrm{te}}\in\mathcal{X}_r^{\mathrm{te}}}\widehat{w}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}(x^{\mathrm{te}}).$$

This procedure is repeated for $r=1,2,\ldots,R$ and its average $\widehat{J}^{(\mathrm{CV})}$ is used as an estimate of J:

$$\widehat{J}^{(\mathrm{CV})}=\frac{1}{R}\sum_{r=1}^R\widehat{J}^{(\mathrm{CV})}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}.$$

We can show that $\widehat{J}^{(\mathrm{CV})}$ gives an almost unbiased estimate of the true cost J, where the 'almost'-ness comes from the fact that the number of samples is reduced in the cross-validation procedure due to data splitting (Luntz and Brailovsky, 1969; Wahba, 1990; Schölkopf and Smola, 2002).

Cross-validation would be more accurate than the information criterion for finite samples. However, it is computationally more expensive than the information criterion since the learning procedure should be repeated R times.

2.5 Heuristics of Basis Function Design for LSIF

A good model may be chosen by cross-validation or the information criterion, given that a family of promising model candidates is prepared. As model candidates, we propose using a Gaussian kernel model centered at the *test* points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, that is,

$$\widehat{w}(x)=\sum_{\ell=1}^{n_{\mathrm{te}}}\alpha_\ell K_\sigma(x,x_\ell^{\mathrm{te}}),$$

where $K_\sigma(x,x')$ is the Gaussian kernel with kernel width σ:

$$K_\sigma(x,x')=\exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right). \tag{21}$$

The reason why we chose the test points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ as the Gaussian centers, not the training points $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$, is as follows (Sugiyama et al., 2008b). By definition, the importance $w(x)$ tends to take

large values if the training density $p_{\mathrm{tr}}(x)$ is small and the test density $p_{\mathrm{te}}(x)$ is large; conversely, $w(x)$ tends to be small (that is, close to zero) if $p_{\mathrm{tr}}(x)$ is large and $p_{\mathrm{te}}(x)$ is small. When a function is approximated by a Gaussian kernel model, many kernels may be needed in the region where the output of the target function is large; on the other hand, only a small number of kernels would be enough in the region where the output of the target function is close to zero. Following this heuristic, we allocate many kernels at high *test* density regions, which can be achieved by setting the Gaussian centers at the test points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$.

Alternatively, we may locate $(n_{\mathrm{tr}}+n_{\mathrm{te}})$ Gaussian kernels at both $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$. However, in our preliminary experiments, this did not further improve the performance, but just slightly increased the computational cost. When n_{te} is large, just using all the test points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ as Gaussian centers is already computationally rather demanding. To ease this problem, we practically propose using a subset of $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ as Gaussian centers for computational efficiency, that is,

$$\widehat{w}(x) = \sum_{\ell=1}^{b} \alpha_\ell K_\sigma(x, c_\ell), \tag{22}$$

where $c_\ell,\ \ell = 1,2,\ldots,b$ are template points randomly chosen from $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ without replacement and $b\ (\leq n_{\mathrm{te}})$ is a prefixed number. In the rest of this paper, we usually fix the number of template points at

$$b = \min(100, n_{\mathrm{te}}),$$

and optimize the kernel width σ and the regularization parameter λ by cross-validation with grid search.

2.6 Entire Regularization Path for LSIF

We can show that the LSIF solution $\widehat{\alpha}$ is piecewise linear with respect to the regularization parameter λ (see Appendix D). Therefore, the *regularization path* (that is, solutions for all λ) can be computed efficiently based on the *parametric optimization technique* (Best, 1982; Efron et al., 2004; Hastie et al., 2004).

A basic idea of regularization path tracking is to check violation of the KKT conditions—which are necessary and sufficient for optimality of convex programs—when the regularization parameter λ is changed. The KKT conditions of LSIF are summarized in Section 2.3. The strict complementarity condition (17) assures the uniqueness of the optimal solution for a fixed λ, and thus the uniqueness of the regularization path. A pseudo code of the regularization path tracking algorithm for LSIF is described in Figure 1—its detailed derivation is summarized in Appendix D. Thanks to the regularization path algorithm, LSIF is computationally efficient in model selection scenarios.

The pseudo code implies that we no longer need a quadratic programming solver for obtaining the solution of LSIF—just computing matrix inverses is enough. Furthermore, the regularization path algorithm is computationally more efficient when the solution is sparse, that is, most of the elements are zero since the number of change points tends to be small for such sparse solutions.

Even though the regularization path tracking algorithm is computationally efficient, it tends to be numerically unreliable, as we experimentally show in Section 4. This numerical instability is caused by near singularity of the matrix $\widehat{G}$. When $\widehat{G}$ is nearly singular, it is not easy to accurately obtain the solutions u, v in Figure 1, and therefore the change point $\lambda_{\tau+1}$ cannot be accurately computed. As a result, we cannot accurately update the active set of the inequality constraints and thus

Input: $\widehat{H}$ and $\widehat{h}$ % see Eqs. (4) and (5) for the definitions
Output: entire regularization path $\widehat{\alpha}(\lambda)$ for $\lambda \geq 0$

$\tau \longleftarrow 0$;
$k \longleftarrow \mathrm{argmax}_i\{\widehat{h}_i \mid i = 1,2,\ldots,b\}$;
$\lambda_\tau \longleftarrow \widehat{h}_k$;
$\widehat{\mathcal{A}} \longleftarrow \{1,2,\ldots,b\}\backslash\{k\}$;
$\widehat{\alpha}(\lambda_\tau) \longleftarrow 0_b$; % the vector with all zeros
While $\lambda_\tau > 0$
  $\widehat{E} \longleftarrow O_{|\widehat{\mathcal{A}}|\times b}$; % the matrix with all zeros
  For $i = 1,2,\ldots,|\widehat{\mathcal{A}}|$
    $\widehat{E}_{i,\widehat{j}_i} \longleftarrow 1$; % $\widehat{\mathcal{A}} = \{\widehat{j}_1, \widehat{j}_2, \ldots, \widehat{j}_{|\widehat{\mathcal{A}}|} \mid \widehat{j}_1 < \widehat{j}_2 < \cdots < \widehat{j}_{|\widehat{\mathcal{A}}|}\}$
  end
  $\widehat{G} \longleftarrow \begin{pmatrix} \widehat{H} & -\widehat{E}^\top \\ -\widehat{E} & O_{|\widehat{\mathcal{A}}|\times|\widehat{\mathcal{A}}|} \end{pmatrix}$;
  $u \longleftarrow \widehat{G}^{-1}\begin{pmatrix} \widehat{h} \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}$;
  $v \longleftarrow \widehat{G}^{-1}\begin{pmatrix} 1_b \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}$;
  If $v \leq 0_{b+|\widehat{\mathcal{A}}|}$ % the final interval
    $\lambda_{\tau+1} \longleftarrow 0$;
    $\widehat{\alpha}(\lambda_{\tau+1}) \longleftarrow (u_1, u_2, \ldots, u_b)^\top$;
  else % an intermediate interval
    $k \longleftarrow \mathrm{argmax}_i\{u_i/v_i \mid v_i > 0,\ i = 1,2,\ldots,b+|\widehat{\mathcal{A}}|\}$;
    $\lambda_{\tau+1} \longleftarrow \max\{0, u_k/v_k\}$;
    $\widehat{\alpha}(\lambda_{\tau+1}) \longleftarrow (u_1, u_2, \ldots, u_b)^\top - \lambda_{\tau+1}(v_1, v_2, \ldots, v_b)^\top$;
    If $1 \leq k \leq b$
      $\widehat{\mathcal{A}} \longleftarrow \widehat{\mathcal{A}} \cup \{k\}$;
    else
      $\widehat{\mathcal{A}} \longleftarrow \widehat{\mathcal{A}}\backslash\{\widehat{j}_{k-b}\}$;
    end
  end
  $\tau \longleftarrow \tau + 1$;
end
$$\widehat{\alpha}(\lambda) \longleftarrow \begin{cases} 0_b & \text{if } \lambda \geq \lambda_0 \\ \frac{\lambda_{\tau+1}-\lambda}{\lambda_{\tau+1}-\lambda_\tau}\widehat{\alpha}(\lambda_\tau) + \frac{\lambda-\lambda_\tau}{\lambda_{\tau+1}-\lambda_\tau}\widehat{\alpha}(\lambda_{\tau+1}) & \text{if } \lambda_{\tau+1} \leq \lambda \leq \lambda_\tau \end{cases}$$

Figure 1: Pseudo code for computing the entire regularization path of LSIF. When the computation of $\widehat{G}^{-1}$ is numerically unstable, we may add small positive diagonals to $\widehat{H}$ for stabilization purposes.

the obtained solution $\widehat{\alpha}(\lambda)$ becomes unreliable; furthermore, such numerical error tends to be accumulated through the path-tracking process. This instability issue seems to be a common pitfall of solution path tracking algorithms in general (see Scheinberg, 2006).

When the Gaussian width σ is very small or very large, the matrix $\widehat{H}$ tends to be nearly singular and thus the matrix $\widehat{G}$ also becomes nearly singular. On the other hand, when the Gaussian width σ is not too small or too large compared with the dispersion of samples, the matrix $\widehat{G}$ is well-conditioned and therefore the path-following algorithm would be stable and reliable.

3. Approximation Algorithm

Within the quadratic programming formulation, we have proposed a new importance estimation procedure LSIF and showed its theoretical properties. We also gave a regularization path tracking algorithm that can be computed efficiently. However, as we experimentally show in Section 4, it tends to suffer from a numerical problem and therefore is not practically reliable. In this section, we give a practical alternative to LSIF which gives an approximate solution to LSIF in a computationally efficient and reliable manner.

3.1 Unconstrained Least-squares Formulation

The approximation idea we introduce here is very simple: we ignore the non-negativity constraint of the parameters in the optimization problem (6). This results in the following unconstrained optimization problem.

$$\min_{\beta\in\mathbb{R}^b}\left[\frac{1}{2}\beta^\top\widehat{H}\beta-\widehat{h}^\top\beta+\frac{\lambda}{2}\beta^\top\beta\right]. \tag{23}$$

In the above, we included a quadratic regularization term $\beta^\top\beta/2$, instead of the linear one $1_b^\top\beta$ since the linear penalty term does not work as a regularizer without the non-negativity constraint. Eq. (23) is an unconstrained convex quadratic program, so the solution can be analytically computed as

$$\widetilde{\beta}(\lambda)=(\widehat{H}+\lambda I_b)^{-1}\widehat{h},$$

where I_b is the b-dimensional identity matrix. Since we dropped the non-negativity constraint $\beta\geq 0_b$, some of the learned parameters could be negative. To compensate for this approximation error, we modify the solution by

$$\widehat{\beta}(\lambda)=\max(0_b,\widetilde{\beta}(\lambda)),$$

where the 'max' operation for a pair of vectors is applied in the element-wise manner. This is the solution of the approximation method we propose in this section.

An advantage of the above unconstrained formulation is that the solution can be computed just by solving a system of linear equations. Therefore, its computation is stable when λ is not too small. We call this method *unconstrained LSIF* (uLSIF). Due to the ℓ_2 regularizer, the solution tends to be close to 0_b to some extent. Thus, the effect of ignoring the non-negativity constraint may not be so strong—later, we analyze the approximation error both theoretically and experimentally in more detail in Sections 3.3 and 4.5.

Note that LSIF and uLSIF differ only in parameter learning. Thus, the basis design heuristic of LSIF given in Section 2.5 is also valid for uLSIF.

3.2 Convergence Analysis of uLSIF

Here, we theoretically analyze the convergence property of the solution $\widehat{\beta}(\lambda)$ of the uLSIF algorithm; practitioners may skip Sections 3.2 and 3.3.

Let $\beta^\circ(\lambda)$ be the optimal solution of the 'ideal' version of the problem (23):

$$\min_{\beta\in\mathbb{R}^b}\left[\frac{1}{2}\beta^\top H\beta - h^\top\beta + \frac{\lambda}{2}\beta^\top\beta\right].$$

Then the ideal solution $\beta^*(\lambda)$ is given by

$$\begin{aligned}\beta^*(\lambda) &= \max(0_b, \beta^\circ(\lambda)),\\ \beta^\circ(\lambda) &= B_\lambda^{-1}h,\\ B_\lambda &= H + \lambda I_b.\end{aligned} \tag{24}$$

Below, we theoretically investigate the learning curve of uLSIF.

Let $\mathcal{B}\subset\{1,2,\ldots,b\}$ be the set of negative indices of $\beta^\circ(\lambda)$, that is,

$$\mathcal{B} = \{\ell \mid \beta^\circ_\ell(\lambda) < 0,\ \ell = 1,2,\ldots,b\},$$

and $\widetilde{\mathcal{B}}\subset\{1,2,\ldots,b\}$ be the set of negative indices of $\widetilde{\beta}(\lambda)$, that is,

$$\widetilde{\mathcal{B}} = \{\ell \mid \widetilde{\beta}_\ell(\lambda) < 0,\ \ell = 1,2,\ldots,b\}.$$

Then we have the following theorem.

Theorem 4 *Assume that $\beta^\circ_\ell(\lambda)\neq 0$ for $\ell = 1,2,\ldots,b$. Then, there exists a positive constant c and a natural number N such that for $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}\geq N$,*

$$P(\mathcal{B}\neq\widetilde{\mathcal{B}}) < e^{-c\min\{n_{\mathrm{tr}},n_{\mathrm{te}}\}}.$$

The proof of the above theorem is given in Appendix E. The assumption that $\beta^\circ_\ell(\lambda)\neq 0$ for $\ell = 1,2,\ldots,b$ corresponds to the strict complementarity condition (17) in LSIF. Theorem 4 shows that the probability that $\widetilde{\mathcal{B}}$ is different from $\mathcal{B}$ is exponentially small. Thus we may regard $\widetilde{\mathcal{B}} = \mathcal{B}$ in practice.

Let D be the b-dimensional diagonal matrix with the ℓ-th diagonal element

$$D_{\ell,\ell} = \begin{cases} 0 & \ell\in\mathcal{B},\\ 1 & \text{otherwise.}\end{cases}$$

Let

$$\begin{aligned} w^\circ(x) &= \sum_{\ell=1}^b \beta^\circ_\ell(\lambda)\varphi_\ell(x),\\ u(x) &= \sum_{\ell=1}^b [B_\lambda^{-1}D(H\beta^*(\lambda)-h)]_\ell\varphi_\ell(x).\end{aligned}$$

Then we have the following theorem.

Theorem 5 *Assume that*

(a) $\beta^{\circ}_{\ell}(\lambda) \neq 0$ *for* $\ell = 1, 2, \ldots, b$.

(b) n_{tr} *and* n_{te} *satisfy Eq.* (18).

Then, for any $\lambda \geq 0$, *we have*

$$\mathbb{E}[J(\widehat{\beta}(\lambda))] = J(\beta^{*}(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(B_{\lambda}^{-1}DHDB_{\lambda}^{-1}C_{w^{\circ},w^{\circ}} + 2B_{\lambda}^{-1}C_{w^{\circ},u}) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{25}$$

The proof of the above theorem is given in Appendix F. Theorem 5 elucidates the learning curve of uLSIF up to the order of n_{tr}^{-1}. An information criterion may be obtained in the same way as Section 2.4.1. However, as shown in Section 3.4, we can have a closed-form expression of the leave-one-out cross-validation score for uLSIF, which would be practically more useful. For this reason, we do not go into the detail of information criterion.

3.3 Approximation Error Bounds for uLSIF

The uLSIF method is introduced as an approximation of LSIF. Here, we theoretically evaluate the difference between the uLSIF solution $\widehat{\beta}(\lambda)$ and the LSIF solution $\widehat{\alpha}(\lambda)$. More specifically, we use the following normalized L_2-norm on the training samples as the difference measure and derive its upper bounds:

$$\mathrm{diff}(\lambda) = \frac{\inf_{\lambda' \geq 0}\sqrt{\frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}}\left(\widehat{w}(x_i^{\mathrm{tr}};\widehat{\alpha}(\lambda')) - \widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda))\right)^2}}{\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda))}, \tag{26}$$

where the importance function $\widehat{w}(x;\alpha)$ is given by

$$\widehat{w}(x;\alpha) = \sum_{\ell=1}^{b}\alpha_{\ell}\varphi_{\ell}(x).$$

In the theoretical analysis below, we assume

$$\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda)) \neq 0.$$

For $p \in \mathbb{N} \cup \{\infty\}$, let $\|\cdot\|_p$ be the L_p-norm, and let $\|\alpha\|_{\widehat{H}}$ be

$$\|\alpha\|_{\widehat{H}} = \sqrt{\alpha^{\top}\widehat{H}\alpha}, \tag{27}$$

where $\widehat{H}$ is the $b \times b$ matrix defined by Eq. (4). Then we have the following theorem.

Theorem 6 (Norm bound) *Assume that all basis functions satisfy*

$$0 < \varphi_{\ell}(x) \leq 1.$$

Then we have

$$\mathrm{diff}(\lambda) \;\leq\; \frac{\|\widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} \tag{28}$$

$$\leq\; b^2 \left(1+\frac{b}{\lambda}\right) \frac{1}{\min_\ell \sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})} \cdot \frac{n_{\mathrm{te}}}{\min_\ell \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})}, \tag{29}$$

where b is the number of basis functions. The upper bound (29) *is reduced as the regularization parameter λ increases. For the Gaussian basis function model* (22)*, the upper bound (29) is reduced as the Gaussian width σ increases.*

The proof of the above theorem is given in Appendix G. We call Eq. (28) the *norm bound* since it is governed by the norm of $\widehat{\beta}$. Intuitively, the approximation error of uLSIF would small if λ is large since $\widetilde{\beta} \geq 0$ may not be severely violated due to the strong regularization effect. The upper bound (29) justifies this intuitive claim since the error bound tends to be small if the regularization parameter λ is large. Furthermore, the upper bound (29) shows that for the Gaussian basis function model (22), the error bound tends to be small if the Gaussian width σ is large. This is also intuitive since the Gaussian basis functions are nearly flat when the Gaussian width σ is large—a difference in parameters does not cause a significant change in the learned importance function $\widehat{w}(x)$. From the above theorem, we expect that uLSIF is a nice approximation of LSIF when λ is large and σ is large. In Section 4.5, we numerically investigate this issue.

Below, we give a more sophisticated bound on $\mathrm{diff}(\lambda)$. To this end, let us introduce an intermediate optimization problem defined by

$$\min_{\gamma \in \mathbb{R}^b} \left[\frac{1}{2}\gamma^\top \widehat{H}\gamma - \widehat{h}^\top \gamma + \frac{\lambda}{2}\gamma^\top \gamma\right]$$
$$\text{subject to } \gamma \geq 0_b, \tag{30}$$

which we refer to as *LSIF with quadratic penalty* (LSIFq). LSIFq bridges LSIF and uLSIF since the 'goodness-of-fit' part is the same as LSIF but the 'regularization' part is the same as uLSIF. Let $\widehat{\gamma}(\lambda)$ be the optimal solution of LSIFq (30). Based on the solution of LSIFq, we have the following upper bound.

Theorem 7 (Bridge bound) *For any $\lambda \geq 0$, the following inequality holds:*

$$\mathrm{diff}(\lambda) \;\leq\; \frac{\sqrt{\lambda\left(\|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty - \|\widehat{\gamma}(\lambda)\|_2^2\right)} + \|\widehat{\gamma}(\lambda) - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))}. \tag{31}$$

The proof of the above theorem is given in Appendix H. We call the above bound the *bridge bound* since the bridged estimator $\widehat{\gamma}(\lambda)$ plays a central role in the bound. Note that, in the bridge bound, the inside of the square root is assured to be non-negative due to Hölder's inequality (see Appendix H for detail). The bridge bound is generally much sharper than the norm bound (28), but not always (see Section 4.5 for numerical examples).

3.4 Efficient Computation of Leave-one-out Cross-validation Score for uLSIF

A practically important advantage of uLSIF over LSIF is that the score of leave-one-out cross-validation (LOOCV) can be computed analytically—thanks to this property, the computational complexity for performing LOOCV is the same order as just computing a single solution.

In the current setup, we are given two sets of samples, $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, which generally have different sample size. For simplicity, we assume that $n_{\mathrm{tr}} < n_{\mathrm{te}}$ and the i-th training sample x_i^{tr} and the i-th test sample x_i^{te} are held out at the same time; the test samples $\{x_j^{\mathrm{te}}\}_{j=n_{\mathrm{tr}}+1}^{n_{\mathrm{te}}}$ are always used for importance estimation. Note that this assumption is only for the sake of simplicity; we can change the order of test samples without sacrificing the computational advantages.

Let $\widehat{w}^{(i)}(x)$ be an estimate of the importance obtained without the i-th training sample x_i^{tr} and the i-th test sample x_i^{te}. Then the LOOCV score is expressed as

$$\mathrm{LOOCV} = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left[\frac{1}{2} (\widehat{w}^{(i)}(x_i^{\mathrm{tr}}))^2 - \widehat{w}^{(i)}(x_i^{\mathrm{te}}) \right]. \tag{32}$$

Our approach to efficiently computing the LOOCV score is to use the *Sherman-Woodbury-Morrison formula* (Golub and Loan, 1996) for computing matrix inverses: for an invertible square matrix A and vectors u and v such that $v^\top A^{-1} u \neq -1$,

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1} u v^\top A^{-1}}{1 + v^\top A^{-1} u}. \tag{33}$$

Efficient approximation schemes of LOOCV have often been investigated under asymptotic setups (Stone, 1974; Hansen and Larsen, 1996). On the other hand, we provide the exact LOOCV score of uLSIF, which follows the same line as that of ridge regression (Hoerl and Kennard, 1970; Wahba, 1990).

A pseudo code of uLSIF with LOOCV-based model selection is summarized in Figure 2—its detailed derivation is described in Appendix I. Note that the basis-function design heuristic given in Section 2.5 is used in the pseudo code, but the analytic form of the LOOCV score is available for any basis functions.

4. Illustrative Examples

In this section, we illustrate the behavior of LSIF and uLSIF using a toy data set.

4.1 Setup

Let the dimension of the domain be $d = 1$ and the training and test densities be

$$p_{\mathrm{tr}}(x) = \mathcal{N}(x; 1, (1/2)^2),$$
$$p_{\mathrm{te}}(x) = \mathcal{N}(x; 2, (1/4)^2),$$

where $\mathcal{N}(x; \mu, \sigma^2)$ denotes the Gaussian density with mean μ and variance σ^2. These densities are depicted in Figure 3. The task is to estimate the importance $w(x) = p_{\mathrm{te}}(x)/p_{\mathrm{tr}}(x)$.

Input: $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$
Output: $\widehat{w}(x)$

$b \longleftarrow \min(100, n_{\mathrm{te}})$; $\quad n \longleftarrow \min(n_{\mathrm{tr}}, n_{\mathrm{te}})$;
Randomly choose b centers $\{c_\ell\}_{\ell=1}^{b}$ from $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ without replacement;
For each candidate of Gaussian width σ

$$\widehat{H}_{\ell,\ell'} \longleftarrow \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \exp\left(-\frac{\|x_i^{\mathrm{tr}}-c_\ell\|^2+\|x_i^{\mathrm{tr}}-c_{\ell'}\|^2}{2\sigma^2}\right) \text{ for } \ell,\ell'=1,2,\ldots,b;$$

$$\widehat{h}_\ell \longleftarrow \frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \exp\left(-\frac{\|x_j^{\mathrm{te}}-c_\ell\|^2}{2\sigma^2}\right) \text{ for } \ell=1,2,\ldots,b;$$

$$X^{\mathrm{tr}}_{\ell,i} \longleftarrow \exp\left(-\frac{\|x_i^{\mathrm{tr}}-c_\ell\|^2}{2\sigma^2}\right) \text{ for } i=1,2,\ldots,n \text{ and } \ell=1,2,\ldots,b;$$

$$X^{\mathrm{te}}_{\ell,i} \longleftarrow \exp\left(-\frac{\|x_i^{\mathrm{te}}-c_\ell\|^2}{2\sigma^2}\right) \text{ for } i=1,2,\ldots,n \text{ and } \ell=1,2,\ldots,b;$$

For each candidate of regularization parameter λ

$$\widehat{B} \longleftarrow \widehat{H} + \frac{\lambda(n_{\mathrm{tr}}-1)}{n_{\mathrm{tr}}} I_b;$$

$$B_0 \longleftarrow \widehat{B}^{-1}\widehat{h}1_n^\top + \widehat{B}^{-1}X^{\mathrm{tr}}\,\mathrm{diag}\left(\frac{\widehat{h}^\top\widehat{B}^{-1}X^{\mathrm{tr}}}{n_{\mathrm{tr}}1_n^\top - 1_b^\top(X^{\mathrm{tr}}*\widehat{B}^{-1}X^{\mathrm{tr}})}\right);$$

$$B_1 \longleftarrow \widehat{B}^{-1}X^{\mathrm{te}} + \widehat{B}^{-1}X^{\mathrm{tr}}\,\mathrm{diag}\left(\frac{1_b^\top(X^{\mathrm{te}}*\widehat{B}^{-1}X^{\mathrm{tr}})}{n_{\mathrm{tr}}1_n^\top - 1_b^\top(X^{\mathrm{tr}}*\widehat{B}^{-1}X^{\mathrm{tr}})}\right);$$

$$B_2 \longleftarrow \max\left(O_{b\times n}, \frac{n_{\mathrm{tr}}-1}{n_{\mathrm{tr}}(n_{\mathrm{te}}-1)}(n_{\mathrm{te}}B_0 - B_1)\right);$$

$$w_{\mathrm{tr}} \longleftarrow (1_b^\top(X^{\mathrm{tr}}*B_2))^\top; \quad w_{\mathrm{te}} \longleftarrow (1_b^\top(X^{\mathrm{te}}*B_2))^\top;$$

$$\mathrm{LOOCV}(\sigma,\lambda) \longleftarrow \frac{w_{\mathrm{tr}}^\top w_{\mathrm{tr}}}{2n} - \frac{1_n^\top w_{\mathrm{te}}}{n};$$

end
end

$$(\widehat{\sigma},\widehat{\lambda}) \longleftarrow \mathrm{argmin}_{(\sigma,\lambda)}\,\mathrm{LOOCV}(\sigma,\lambda);$$

$$\widetilde{H}_{\ell,\ell'} \longleftarrow \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \exp\left(-\frac{\|x_i^{\mathrm{tr}}-c_\ell\|^2+\|x_i^{\mathrm{tr}}-c_{\ell'}\|^2}{2\widehat{\sigma}^2}\right) \text{ for } \ell,\ell'=1,2,\ldots,b;$$

$$\widetilde{h}_\ell \longleftarrow \frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \exp\left(-\frac{\|x_j^{\mathrm{te}}-c_\ell\|^2}{2\widehat{\sigma}^2}\right) \text{ for } \ell=1,2,\ldots,b;$$

$$\widehat{\alpha} \longleftarrow \max(0_b, (\widetilde{H}+\widehat{\lambda}I_b)^{-1}\widetilde{h});$$

$$\widehat{w}(x) \longleftarrow \sum_{\ell=1}^{b}\widehat{\alpha}_\ell \exp\left(-\frac{\|x-c_\ell\|^2}{2\widehat{\sigma}^2}\right);$$

Figure 2: Pseudo code of uLSIF algorithm with LOOCV. $B*B'$ denotes the element-wise multiplication of matrices B and B' of the same size, that is, the (i,j)-th element is given by $B_{i,j}B'_{i,j}$. For n-dimensional vectors b and b', $\mathrm{diag}\left(\frac{b}{b'}\right)$ denotes the $n\times n$ diagonal matrix with i-th diagonal element b_i/b'_i. A MATLAB® or R implementation of uLSIF is available from `http://sugiyama-www.cs.titech.ac.jp/~sugi/software/uLSIF/`.

4.2 Importance Estimation

First, we illustrate the behavior of LSIF and uLSIF in importance estimation. We set the number of training and test samples at $n_{\mathrm{tr}} = 200$ and $n_{\mathrm{te}} = 1000$, respectively. We use the Gaussian kernel model (22), and the number of basis functions is set at $b = 100$. The centers of the kernel function are randomly chosen from the test points $\{x_i^{\mathrm{te}}\}_{j=1}^{n^{\mathrm{te}}}$ without replacement (see Section 2.5).

We test different Gaussian widths σ and different regularization parameters λ. The following two setups are examined:

(A) λ is fixed at $\lambda = 0.2$ and σ is changed as $0.1 \leq \sigma \leq 1.0$,

(B) σ is fixed at $\sigma = 0.3$ and λ is changed as $0 \leq \lambda \leq 0.5$.

Figure 4 depicts the true importance and its estimates obtained by LSIF and uLSIF, where all importance functions are normalized so that $\int w(x)dx = 1$ for better comparison. Figures 4(a) and 4(b) show that the estimated importance $\widehat{w}(x)$ tends to be too peaky when the Gaussian width σ is small, while it tends to be overly smoothed when σ is large. If the Gaussian width is chosen appropriately, both LSIF and uLSIF seem to work reasonably well. As shown in Figures 4(c) and 4(d), the solutions of LSIF and uLSIF also significantly change when different regularization parameters λ are used. Again, given that the regularization parameter is chosen appropriately, both LSIF and uLSIF tend to perform well.

From the graphs, we also observe that model selection based on cross-validation works reasonably well for both LSIF (5-fold) and uLSIF (leave-one-out) to choose appropriate values of the Gaussian width or the regularization parameter; this will be analyzed in more detail in Section 4.4.

4.3 Regularization Path

Next, we illustrate how the regularization path tracking algorithm for LSIF behaves. We set the number of training and test samples at $n_{\mathrm{tr}} = 50$ and $n_{\mathrm{te}} = 100$, respectively. For better illustration, we set the number of basis functions at a small value as $b = 30$ in the Gaussian kernel model (22) and use the Gaussian kernels centered at equidistant points in $[0,3]$ as basis functions.

We use the algorithm described in Figure 1 for regularization path tracking. Theoretically, the inequality $\lambda_{\tau+1} < \lambda_\tau$ is assured. In numerical computation, however, the inequality is occasionally violated. In order to avoid this numerical problem, we slightly regularize $\widehat{H}$ for stabilization (see also the caption of Figure 1).

Figure 5 depicts the values of the estimated coefficients $\{\alpha_\ell\}_{\ell=1}^{b}$ as functions of $\|\alpha\|_1$ for $\sigma = 0.1, 0.3$, and 0.5. Note that small $\|\alpha\|_1$ corresponds to large λ. The figure indicates that the regularization parameter λ works as a sparseness controlling factor of the solution, that is, the larger (smaller) the value of λ ($\|\alpha\|_1$) is, the sparser the solution is.

The path following algorithm is computationally efficient and therefore practically very attractive. However, as the above experiments illustrate, the path following algorithm is numerically rather unstable. Modification of $\widehat{H}$ can ease to solve this problem, but this in turn results in accumulating numerical errors through the path tracking process. Consequently, the solutions for small λ tend to be inaccurate. This problem becomes prominent if the number of change points in the regularization path is large.

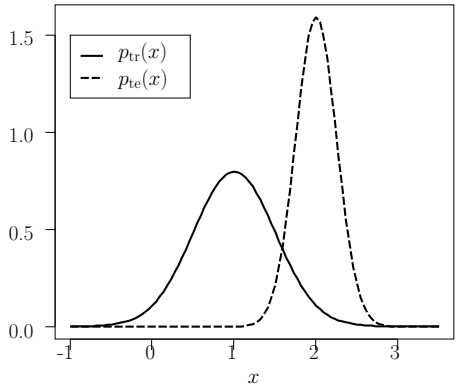
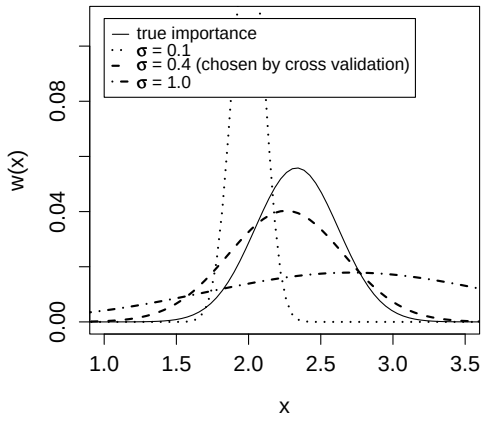
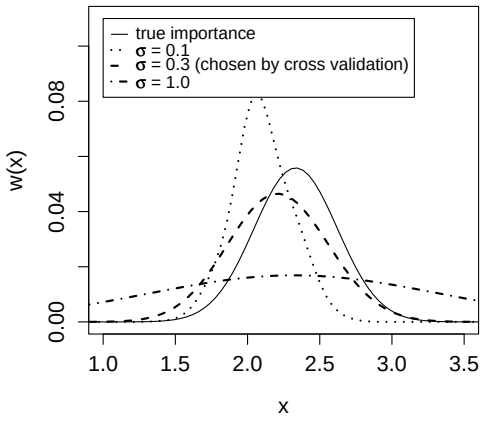


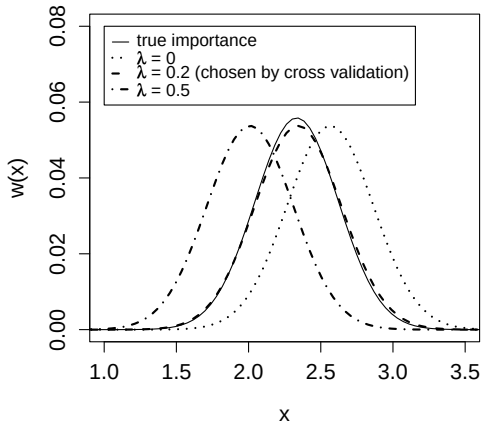
Figure 3: The solid line is the probability density of training data, and the dashed line is the probability density of test data.



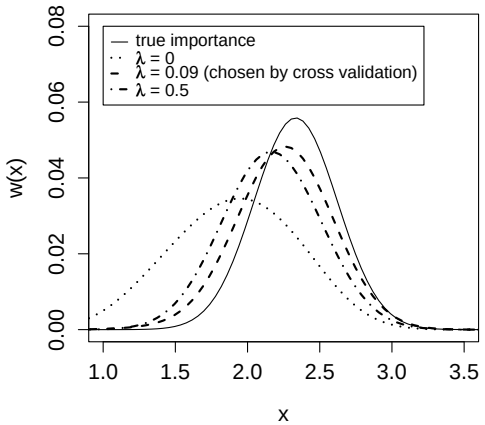
(a) LSIF for $\lambda = 0.2$, $\sigma = 0.1, 0.4, 1.0$.



(b) uLSIF for $\lambda = 0.2$, $\sigma = 0.1, 0.3, 1.0$.

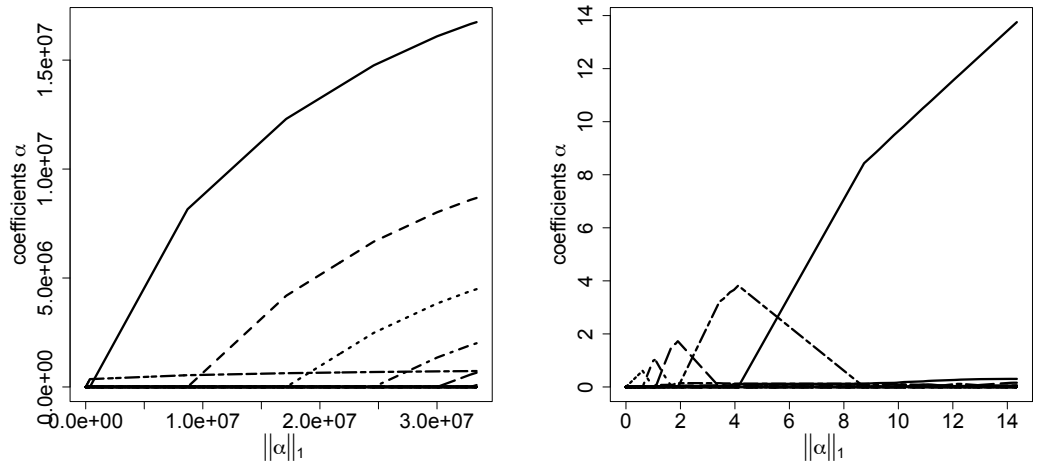


(c) LSIF for $\sigma = 0.3$, $\lambda = 0, 0.2, 0.5$.

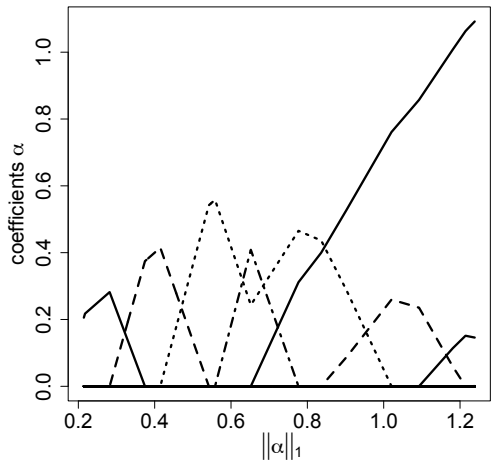


(d) uLSIF for $\sigma = 0.3$, $\lambda = 0, 0.09, 0.5$.

Figure 4: True and estimated importance functions obtained by LSIF and uLSIF for various different Gaussian widths σ and regularization parameters λ.



(a) $\sigma = 0.1$. (b) $\sigma = 0.3$.

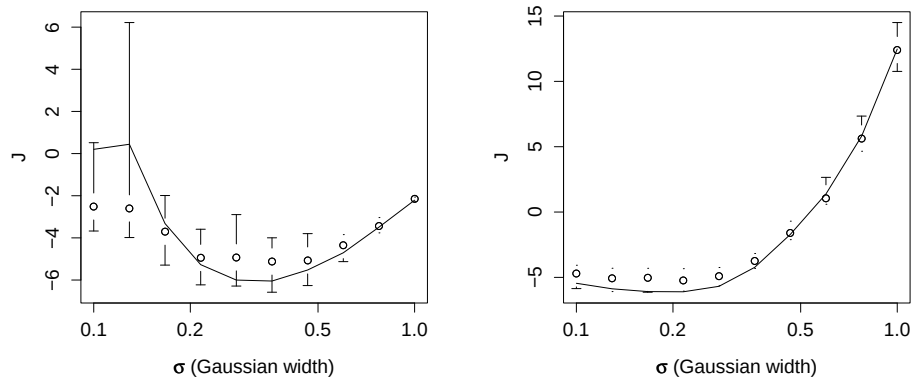


(c) $\sigma = 0.5$.

Figure 5: Regularization path of LSIF: the values of the estimated coefficients $\{\alpha_\ell\}_{\ell=1}^b$ are depicted as functions of the L_1-norm of the estimated parameter vector for $\sigma = 0.1, 0.3$, and 0.5. Small $\|\alpha\|_1$ corresponds to large λ.

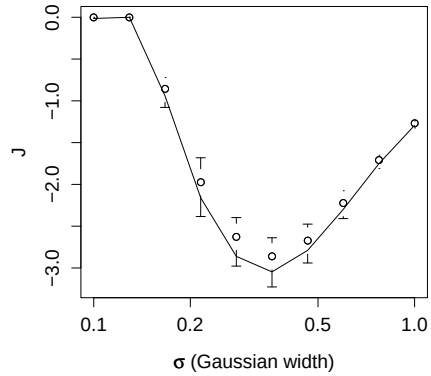
4.4 Cross-validation

Here we illustrate the behavior of the cross-validation scores of LSIF and uLSIF. We set the number of training and test samples at $n_{\mathrm{tr}} = 200$ and $n_{\mathrm{te}} = 1000$, respectively. The number of template points is $b = 100$ and the Gaussian kernel model (22) is used. The centers of the kernel functions are randomly chosen from the test points as described in Section 4.2. The left column of Figure 6 depicts the expectation of the true cost $J(\widehat{\alpha})$ over 50 runs for LSIF and its estimate by 5-fold CV (25, 50, and 75 percentiles are plotted in the figure) as functions of the Gaussian width σ for $\lambda = 0.2$, 0.5, and 0.8. We used the regularization path tracking algorithm for computing the solutions of LSIF.

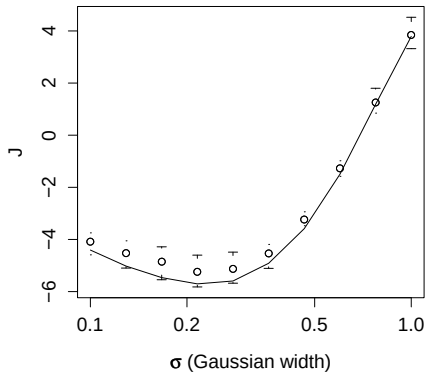


(a) LSIF with 5CV ($\lambda = 0.2$).

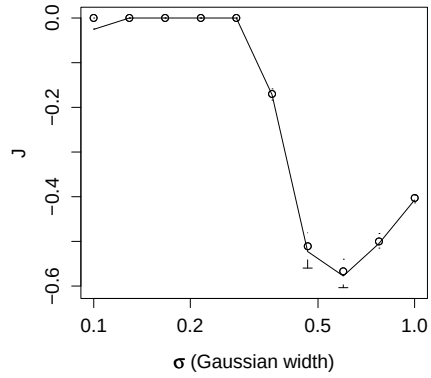
(b) uLSIF with LOOCV ($\lambda = 0.2$).



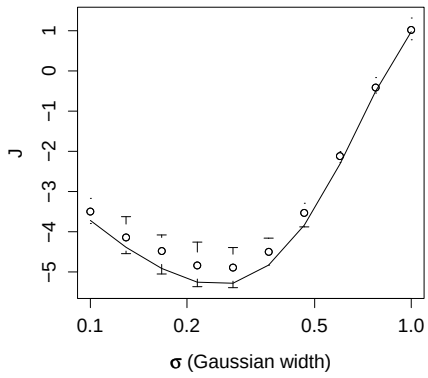
(c) LSIF with 5CV ($\lambda = 0.5$).



(d) uLSIF with LOOCV ($\lambda = 0.5$).



(e) LSIF with 5CV ($\lambda = 0.8$).



(f) uLSIF with LOOCV ($\lambda = 0.8$).

Figure 6: The true cost J and its cross-validation estimate as functions of Gaussian width σ for different values of λ. The solid line denotes the expectation of the true cost J over 50 runs, while '$\circ$' and error bars denote the 25, 50, and 75 percentiles of the cross-validation score.

The right column shows the expected true cost and its LOOCV estimates for uLSIF in the same manner.

The graphs show that overall CV gives reasonably good approximations of the expected cost, although CV for LSIF with small λ and small σ is rather inaccurate due to numerical problems—the solution path of LSIF is computed from $\lambda = \infty$ to $\lambda = 0$, and the numerical error is accumulated as the tracking process approaches to $\lambda = 0$. This phenomenon seems problematic when σ is small.

4.5 Difference between LSIF and uLSIF

In Section 3.3, we analyzed the approximation error of uLSIF against LSIF. Here we numerically investigate the behavior of the approximation error (26) as well as the norm bound (28) and the bridge bound (31). We set the number of training and test samples at $n_{\mathrm{tr}} = 200$ and $n_{\mathrm{te}} = 1000$, respectively. The number of template points in the Gaussian kernel model (22) is set at $b = 100$. The centers of the kernel functions are randomly chosen from the test points (see Section 4.2).

Figure 7 depicts the true approximation error as well as its upper bounds as functions of the regularization parameter λ; λ is varied from 0.001 to 10 and the three Gaussian widths $\sigma = 0.1, 0.5, 1.0$ are tested. The graphs show that when λ and σ are large, the approximation error tends to be small; this is in good agreement with the theoretical analysis given in Section 3.3. The bridge bound is fairly tight in the entire range and is sharper than the norm bound except when σ is small and λ is large.

4.6 Summary

Through the numerical examples, we overall found that LSIF and uLSIF give qualitatively similar results. However, the computation of the solution-path tracking algorithm for LSIF tends to be numerically unstable, which can result in unreliable model selection performance. On the other hand, only a system of linear equations needs to be solved in uLSIF, which turned out to be much more stable than LSIF. Thus, uLSIF would be practically more reliable than LSIF.

Based on the above findings, we will focus on uLSIF in the rest of this paper.

5. Relation to Existing Methods

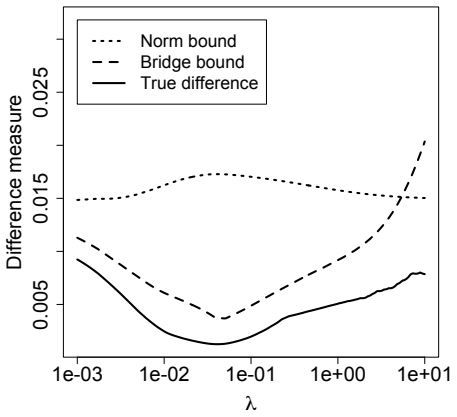
In this section, we discuss the characteristics of existing approaches in comparison with the proposed methods.

5.1 Kernel Density Estimator

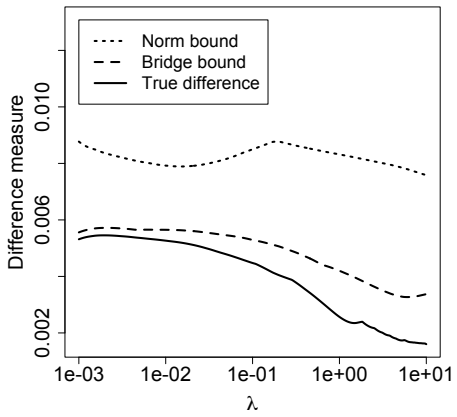
The *kernel density estimator* (KDE) is a non-parametric technique to estimate a probability density function $p(x)$ from its i.i.d. samples $\{x_k\}_{k=1}^{n}$. For the Gaussian kernel (21), KDE is expressed as

$$\widehat{p}(x) = \frac{1}{n_{\mathrm{tr}}(2\pi\sigma^2)^{d/2}} \sum_{k=1}^{n} K_\sigma(x, x_k).$$

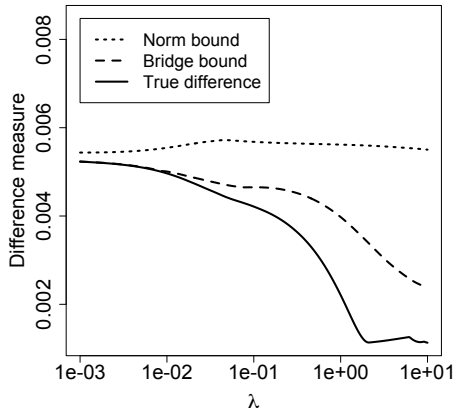
The performance of KDE depends on the choice of the kernel width σ. The kernel width σ can be optimized by *likelihood cross-validation* (LCV) as follows (Härdle et al., 2004): First, divide the samples $\{x_i\}_{i=1}^{n}$ into R disjoint subsets $\{\mathcal{X}_r\}_{r=1}^{R}$. Then obtain a density estimate $\widehat{p}_{\mathcal{X}_k}(x)$ from



(a) $\sigma = 0.1$.



(b) $\sigma = 0.5$.



(c) $\sigma = 1.0$.

Figure 7: The approximation error of uLSIF against LSIF as functions of the regularization parameter λ for different Gaussian width σ. Its upper bounds are also plotted in the graphs.

$\{\mathcal{X}_r\}_{r\neq k}$ (i.e., without $\mathcal{X}_k$) and compute its log-likelihood for $\mathcal{X}_k$:

$$\frac{1}{|\mathcal{X}_k|}\sum_{x\in\mathcal{X}_k}\log\widehat{p}_{\mathcal{X}_k}(x).$$

Repeat this procedure for $r = 1,2,\ldots,R$ and choose the value of σ such that the average of the above hold-out log-likelihood over all r is maximized. Note that the average hold-out log-likelihood is an almost unbiased estimate of the Kullback-Leibler divergence from $p(x)$ to $\widehat{p}(x)$, up to an irrelevant constant.

KDE can be used for importance estimation by first obtaining density estimators $\widehat{p}_{\mathrm{tr}}(x)$ and $\widehat{p}_{\mathrm{te}}(x)$ separately from $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, and then estimating the importance by $\widehat{w}(x) = \widehat{p}_{\mathrm{te}}(x)/\widehat{p}_{\mathrm{tr}}(x)$. A potential limitation of this approach is that KDE suffers from the *curse of dimensionality* (Vapnik, 1998; Härdle et al., 2004), that is, the number of samples needed to maintain the

same approximation quality grows exponentially as the dimension of the domain increases. This is critical when the number of available samples is limited. Therefore, the KDE-based approach may not be reliable in high-dimensional problems.

5.2 Kernel Mean Matching

The *kernel mean matching* (KMM) method allows us to directly obtain an estimate of the importance values at training points without going through density estimation (Huang et al., 2007). The basic idea of KMM is to find $\widehat{w}(x)$ such that the mean discrepancy between nonlinearly transformed samples drawn from $p_{\mathrm{te}}(x)$ and $p_{\mathrm{tr}}(x)$ is minimized in a *universal reproducing kernel Hilbert space* (Steinwart, 2001). The Gaussian kernel (21) is an example of kernels that induce universal reproducing kernel Hilbert spaces and it has been shown that the solution of the following optimization problem agrees with the true importance:

$$\min_{w(x)} \left\| \int K_\sigma(x,\cdot) p_{\mathrm{te}}(x) dx - \int K_\sigma(x,\cdot) w(x) p_{\mathrm{tr}}(x) dx \right\|_{\mathcal{H}}^2$$
$$\text{subject to } \int w(x) p_{\mathrm{tr}}(x) dx = 1 \text{ and } w(x) \geq 0,$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the norm in the Gaussian reproducing kernel Hilbert space and $K_\sigma(x,x')$ is the Gaussian kernel (21).

An empirical version of the above problem is reduced to the following quadratic program:

$$\min_{\{w_i\}_{i=1}^{n_{\mathrm{tr}}}} \left[\frac{1}{2} \sum_{i,i'=1}^{n_{\mathrm{tr}}} w_i w_{i'} K_\sigma(x_i^{\mathrm{tr}}, x_{i'}^{\mathrm{tr}}) - \sum_{i=1}^{n_{\mathrm{tr}}} w_i \kappa_i \right]$$
$$\text{subject to } \left| \sum_{i=1}^{n_{\mathrm{tr}}} w_i - n_{\mathrm{tr}} \right| \leq n_{\mathrm{tr}} \varepsilon \text{ and } 0 \leq w_1, w_2, \ldots, w_{n_{\mathrm{tr}}} \leq B,$$

where

$$\kappa_i = \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} K_\sigma(x_i^{\mathrm{tr}}, x_j^{\mathrm{te}}).$$

B (≥ 0) and ε (≥ 0) are tuning parameters that control the regularization effects. The solution $\{\widehat{w}_i\}_{i=1}^{n_{\mathrm{tr}}}$ is an estimate of the importance at the training points $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$.

Since KMM does not involve density estimation, it is expected to work well even in high dimensional cases. However, the performance is dependent on the tuning parameters B, ε, and σ, and they cannot be simply optimized, for example, by CV since estimates of the importance are available only at the training points. A popular heuristic is to use the median distance between samples as the Gaussian width σ, which is shown to be useful (Schölkopf and Smola, 2002; Song et al., 2007). However, there seems no strong justification for this heuristic. For the choice of ε, a theoretical result given in Huang et al. (2007) could be used as guidance, although it is still hard to determine the best value of ε in practice.

5.3 Logistic Regression

Another approach to directly estimating the importance is to use a probabilistic classifier. Let us assign a selector variable $\eta = -1$ to training samples and $\eta = 1$ to test samples, that is, the training

and test densities are written as

$$p_{\mathrm{tr}}(x) = p(x|\eta = -1),$$
$$p_{\mathrm{te}}(x) = p(x|\eta = 1).$$

Note that η is regarded as a random variable.

Application of the Bayes theorem yields that the importance can be expressed in terms of η as follows (Qin, 1998; Cheng and Chu, 2004; Bickel et al., 2007):

$$w(x) = \frac{p(\eta = -1)}{p(\eta = 1)} \frac{p(\eta = 1|x)}{p(\eta = -1|x)}.$$

The probability ratio of test and training samples may be simply estimated by the ratio of the numbers of samples:

$$\frac{p(\eta = -1)}{p(\eta = 1)} \approx \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}}}.$$

The conditional probability $p(\eta|x)$ could be approximated by discriminating test samples from training samples using a *logistic regression* (LogReg) classifier, where η plays the role of a class variable. Below we briefly explain the LogReg method.

The LogReg classifier employs a parametric model of the following form for expressing the conditional probability $p(\eta|x)$:

$$\widehat{p}(\eta|x) = \frac{1}{1 + \exp\left(-\eta \sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x)\right)},$$

where m is the number of basis functions and $\{\phi_\ell(x)\}_{\ell=1}^{m}$ are fixed basis functions. The parameter ζ is learned so that the negative regularized log-likelihood is minimized:

$$\widehat{\zeta} = \operatorname*{argmin}_{\zeta} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \log\left(1 + \exp\left(\sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x_i^{\mathrm{tr}})\right)\right) + \sum_{j=1}^{n_{\mathrm{te}}} \log\left(1 + \exp\left(-\sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x^{\mathrm{te}})\right)\right) + \lambda \zeta^\top \zeta \right].$$

Since the above objective function is convex, the global optimal solution can be obtained by standard nonlinear optimization methods such as Newton's method, the conjugate gradient method, and the BFGS method (Minka, 2007). Then the importance estimate is given by

$$\widehat{w}(x) = \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}}} \exp\left(\sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x)\right). \tag{34}$$

An advantage of the LogReg method is that model selection (that is, the choice of the basis functions $\{\phi_\ell(x)\}_{\ell=1}^{m}$ as well as the regularization parameter λ) is possible by standard CV since the learning problem involved above is a standard supervised classification problem.

5.4 Kullback-Leibler Importance Estimation Procedure

The *Kullback-Leibler importance estimation procedure* (KLIEP) (Sugiyama et al., 2008a) also directly gives an estimate of the importance function without going through density estimation by matching the two distributions in terms of the Kullback-Leibler divergence (Kullback and Leibler, 1951).

Let us model the importance $w(x)$ by the linear model (1). An estimate of the test density $p_{\mathrm{te}}(x)$ is given by using the model $\widehat{w}(x)$ as

$$\widehat{p}_{\mathrm{te}}(x) = \widehat{w}(x)p_{\mathrm{tr}}(x).$$

In KLIEP, the parameters α are determined so that the Kullback-Leibler divergence from $p_{\mathrm{te}}(x)$ to $\widehat{p}_{\mathrm{te}}(x)$ is minimized:

$$\begin{aligned}\mathrm{KL}[p_{\mathrm{te}}(x)\|\widehat{p}_{\mathrm{te}}(x)] &= \int_{\mathcal{D}} p_{\mathrm{te}}(x)\log\frac{p_{\mathrm{te}}(x)}{\widehat{w}(x)p_{\mathrm{tr}}(x)}dx \\ &= \int_{\mathcal{D}} p_{\mathrm{te}}(x)\log\frac{p_{\mathrm{te}}(x)}{p_{\mathrm{tr}}(x)}dx - \int_{\mathcal{D}} p_{\mathrm{te}}(x)\log\widehat{w}(x)dx. \qquad (35)\end{aligned}$$

The first term is a constant, so it can be safely ignored. Since $\widehat{p}_{\mathrm{te}}(x)$ $(=\widehat{w}(x)p_{\mathrm{tr}}(x))$ is a probability density function, it should satisfy

$$1 = \int_{\mathcal{D}} \widehat{p}_{\mathrm{te}}(x)dx = \int_{\mathcal{D}} \widehat{w}(x)p_{\mathrm{tr}}(x)dx. \qquad (36)$$

Then the KLIEP optimization problem is given by replacing the expectations in Eqs. (35) and (36) with empirical averages as follows:

$$\begin{aligned}&\max_{\{\alpha_\ell\}_{\ell=1}^b}\left[\sum_{j=1}^{n_{\mathrm{te}}}\log\left(\sum_{\ell=1}^{b}\alpha_\ell\varphi_\ell(x_j^{\mathrm{te}})\right)\right] \\ &\qquad\text{subject to } \sum_{\ell=1}^{b}\alpha_\ell\left(\sum_{i=1}^{n_{\mathrm{tr}}}\varphi_\ell(x_i^{\mathrm{tr}})\right) = n_{\mathrm{tr}} \text{ and } \alpha_1,\alpha_2,\ldots,\alpha_b \geq 0.\end{aligned}$$

This is a convex optimization problem and the global solution—which tends to be sparse (Boyd and Vandenberghe, 2004)—can be obtained, for example, by simply performing gradient ascent and feasibility satisfaction iteratively. Model selection of KLIEP is possible by LCV.

Properties of KLIEP-type algorithms have been theoretically investigated in Nguyen et al. (2008) and Sugiyama et al. (2008b) (see also Qin, 1998; Cheng and Chu, 2004). Note that the importance model of KLIEP is the linear model (1), while that of LogReg is the log-linear model (34). A variant of KLIEP for log-linear models has been studied in Tsuboi et al. (2008).

5.5 Discussions

Table 1 summarizes properties of proposed and existing methods.

KDE is efficient in computation since no optimization is involved, and model selection is possible by LCV. However, KDE may suffer from the curse of dimensionality due to the difficulty of density estimation in high dimensions.

Methods	Density estimation	Model selection	Optimization	Out-of-sample prediction
KDE	Necessary	**Available**	**Analytic**	**Possible**
KMM	**Not necessary**	Not available	Convex quadratic program	Not possible
LogReg	**Not necessary**	**Available**	Convex non-linear	**Possible**
KLIEP	**Not necessary**	**Available**	Convex non-linear	**Possible**
LSIF	**Not necessary**	**Available**	Convex quadratic program	**Possible**
uLSIF	**Not necessary**	**Available**	**Analytic**	**Possible**

Table 1: Relation between proposed and existing methods.

KMM can potentially overcome the curse of dimensionality by directly estimating the importance. However, there is no objective model selection method. Therefore, model parameters such as the Gaussian width need to be determined by hand, which is highly unreliable unless we have strong prior knowledge. Furthermore, the computation of KMM is rather demanding since a quadratic programming problem has to be solved.

LogReg and KLIEP also do not involve density estimation, but different from KMM, they give an estimate the entire importance function, not only the values of the importance at training points. Therefore, the values of the importance at unseen points can be estimated by LogReg and KLIEP. This feature is highly useful since it enables us to employ CV for model selection, which is a significant advantage over KMM. However, LogReg and KLIEP are computationally rather expensive since non-linear optimization problems have to be solved. Note that the LogReg method is slightly different in motivation from other methods, but has some similarity in computation and implementation, for example, the LogReg method also involves a kernel smoother.

The proposed LSIF method is qualitatively similar to LogReg and KLIEP, that is, it can avoid density estimation, model selection is possible, and non-linear optimization is involved. LSIF is advantageous over LogReg and KLIEP in that it is equipped with a regularization path tracking algorithm. Thanks to this, model selection of LSIF is computationally much more efficient than LogReg and KLIEP. However, the regularization path tracking algorithm tends to be numerically unstable.

The proposed uLSIF method inherits good properties of existing methods such as no density estimation involved and a build-in model selection method equipped. In addition to these preferable properties, the solution of uLSIF can be computed in an efficient and numerically stable manner. Furthermore, thanks to the availability of the closed-form solution of uLSIF, the LOOCV score can be analytically computed without repeating hold-out loops, which highly contributes to reducing the computation time in the model selection phase.

In the next section, we experimentally show that uLSIF is computationally more efficient than existing direct importance estimation methods, while its estimation accuracy is comparable to the best existing methods.

6. Experiments

In this section, we compare the experimental performance of the proposed and existing methods.

6.1 Importance Estimation

Let the dimension of the domain be d and

$$p_{\mathrm{tr}}(x) = \mathcal{N}(x; (0,0,\ldots,0)^\top, I_d),$$
$$p_{\mathrm{te}}(x) = \mathcal{N}(x; (1,0,\ldots,0)^\top, I_d).$$

The task is to estimate the importance at training points:

$$w_i = w(x_i^{\mathrm{tr}}) = \frac{p_{\mathrm{te}}(x_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(x_i^{\mathrm{tr}})} \quad \text{for } i = 1,2,\ldots,n_{\mathrm{tr}}.$$

We compare the following methods:

KDE(CV): The Gaussian kernel (21) is used, where the kernel widths of the training and test densities are separately optimized based on 5-fold LCV.

KMM(med): The performance of KMM is dependent on B, ε, and σ. We set $B = 1000$ and $\varepsilon = (\sqrt{n_{\mathrm{tr}}} - 1)/\sqrt{n_{\mathrm{tr}}}$ following the original paper (Huang et al., 2007), and the Gaussian width σ is set at the median distance between samples within the training set and the test set (Schölkopf and Smola, 2002; Song et al., 2007).

LogReg(CV): The Gaussian kernel model (22) are used as basis functions. The kernel width σ and the regularization parameter λ are chosen based on 5-fold CV.[1]

KLIEP(CV): The Gaussian kernel model (22) is used. The kernel width σ is selected based on 5-fold LCV.

uLSIF(CV): The Gaussian kernel model (22) is used. The kernel width σ and the regularization parameter λ are determined based on LOOCV.

All the methods are implemented using the *MATLAB*® environment, where the *CPLEX*® optimizer is used for solving quadratic programs in KMM and the *LIBLINEAR* implementation is used for LogReg (Lin et al., 2007).

We fixed the number of test points at $n_{\mathrm{te}} = 1000$ and consider the following two setups for the number n_{tr} of training samples and the input dimensionality d:

(a) n_{tr} is fixed at $n_{\mathrm{tr}} = 100$ and d is changed as $d = 1,2,\ldots,20$,

(b) d is fixed at $d = 10$ and n_{tr} is changed as $n_{\mathrm{tr}} = 50,60,\ldots,150$.

We run the experiments 100 times for each d, each n_{tr}, and each method, and evaluate the quality of the importance estimates $\{\widehat{w}_i\}_{i=1}^{n_{\mathrm{tr}}}$ by the *normalized mean squared error* (NMSE):

$$\mathrm{NMSE} = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\frac{\widehat{w}_i}{\sum_{i'=1}^{n_{\mathrm{tr}}} \widehat{w}_{i'}} - \frac{w_i}{\sum_{i'=1}^{n_{\mathrm{tr}}} w_{i'}} \right)^2.$$

1. In Sugiyama et al. (2008b) where KLIEP has been proposed, the performance of LogReg has been experimentally investigated in the same setup. In that paper, however, LogReg was not regularized since KLIEP was not also regularized. On the other hand, we use a regularized LogReg method and choose the regularization parameter in addition to the Gaussian kernel width by CV here. Thanks to the regularization effect, the results of LogReg in the current paper tends to be better than that reported in Sugiyama et al. (2008b).

In practice, the scale of the importance is not significant and the relative magnitude among w_i is important. Thus the above NMSE would be a suitable error metric for evaluating the performance of each method.

NMSEs averaged over 100 trials (a) as a function of input dimensionality d and (b) as a function of the training sample size n_{tr} are plotted in log scale in Figure 8. Error bars are omitted for clear visibility—instead, the best method in terms of the mean error and comparable ones based on the t-test at the significance level 1% are indicated by '$\circ$'; the methods with significant difference from the best methods are indicated by '$\times$'.

Figure 8(a) shows that the error of KDE(CV) sharply increases as the input dimensionality grows, while LogReg, KLIEP, and uLSIF tend to give much smaller errors than KDE. This would be the fruit of directly estimating the importance without going through density estimation. KMM tends to perform poorly, which is caused by an inappropriate choice of the Gaussian kernel width. On the other hand, model selection in LogReg, KLIEP, and uLSIF seems to work quite well. Figure 8(b) shows that the errors of all methods tend to decrease as the number of training samples grows. Again LogReg, KLIEP, and uLSIF tend to give much smaller errors than KDE and KMM.

Next we investigate the computation time. Each method has a different model selection strategy, that is, KMM does not involve CV, KDE and KLIEP involve CV over the kernel width, and LogReg and uLSIF involve CV over both the kernel width and the regularization parameter. Thus the naive comparison of the total computation time is not so meaningful. For this reason, we first investigate the computation time of each importance estimation method after the model parameters are fixed.

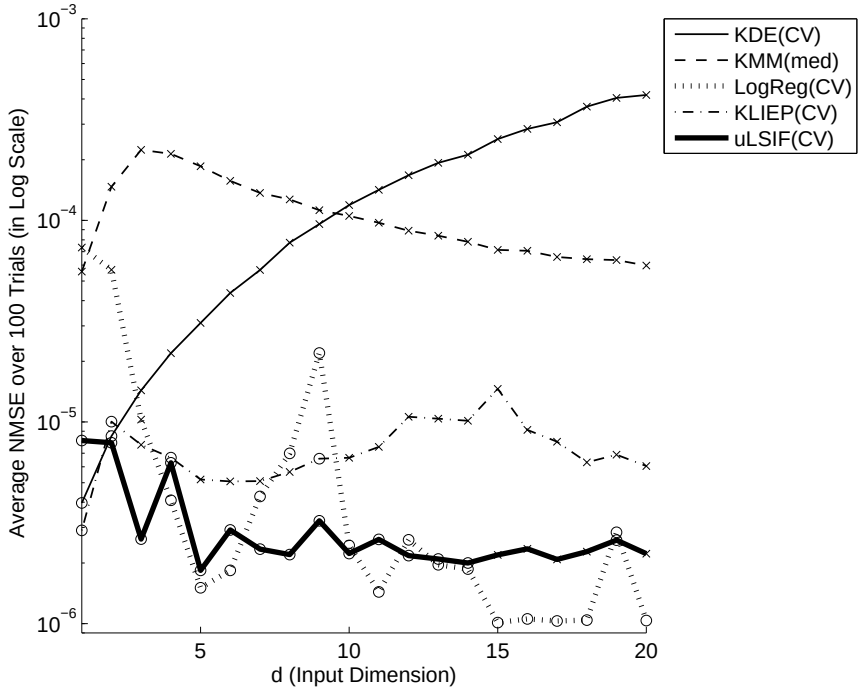
The average CPU computation time over 100 trials are summarized in Figure 9. Figure 9(a) shows that the computation time of KDE, KLIEP, and uLSIF is almost independent of the input dimensionality, while that of KMM and LogReg is rather dependent on the input dimensionality. Note that LogReg for $d \leq 3$ is slow due to some convergence problem of the LIBLINEAR package. Among them, the proposed uLSIF is one of the fastest methods. Figure 9(b) shows that the computation time of LogReg, KLIEP, and uLSIF is nearly independent of the number of training samples, while that of KDE and KMM sharply increase as the number of training samples increases.

Both LogReg and uLSIF have high accuracy and their computation time after model selection is comparable. Finally, we compare the entire computation time of LogReg and uLSIF including CV, which is summarized in Figure 10. We note that the Gaussian width σ and the regularization parameter λ are chosen over the 9×9 grid in this experiment for both LogReg and uLSIF. Therefore, the comparison of the entire computation time is fair. Figures 10(a) and 10(b) show that uLSIF is approximately 5 times faster than LogReg.

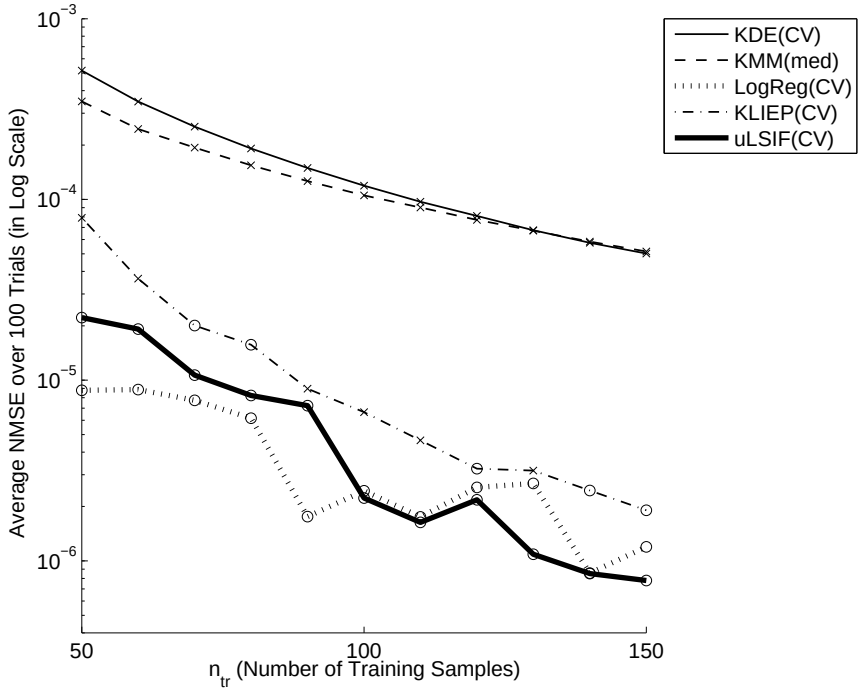
Overall, uLSIF is shown to be comparable to the best existing method (LogReg) in terms of the accuracy, but is computationally more efficient than LogReg.

6.2 Covariate Shift Adaptation in Regression and Classification

Next, we illustrate how the importance estimation methods could be used in *covariate shift adaptation* (Shimodaira, 2000; Zadrozny, 2004; Sugiyama and Müller, 2005; Huang et al., 2007; Bickel and Scheffer, 2007; Bickel et al., 2007; Sugiyama et al., 2007). Covariate shift is a situation in supervised learning where the input distributions change between the training and test phase but the conditional distribution of outputs given inputs remains unchanged. Under covariate shift, standard learning techniques such as maximum likelihood estimation or cross-validation are biased—the bias

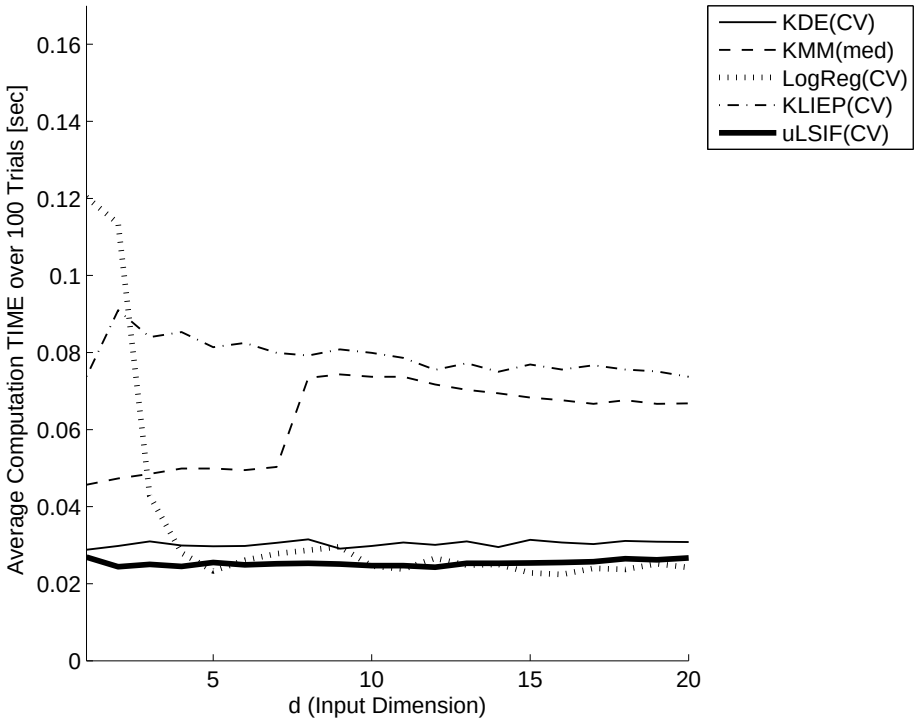


(a) When input dimensionality is changed

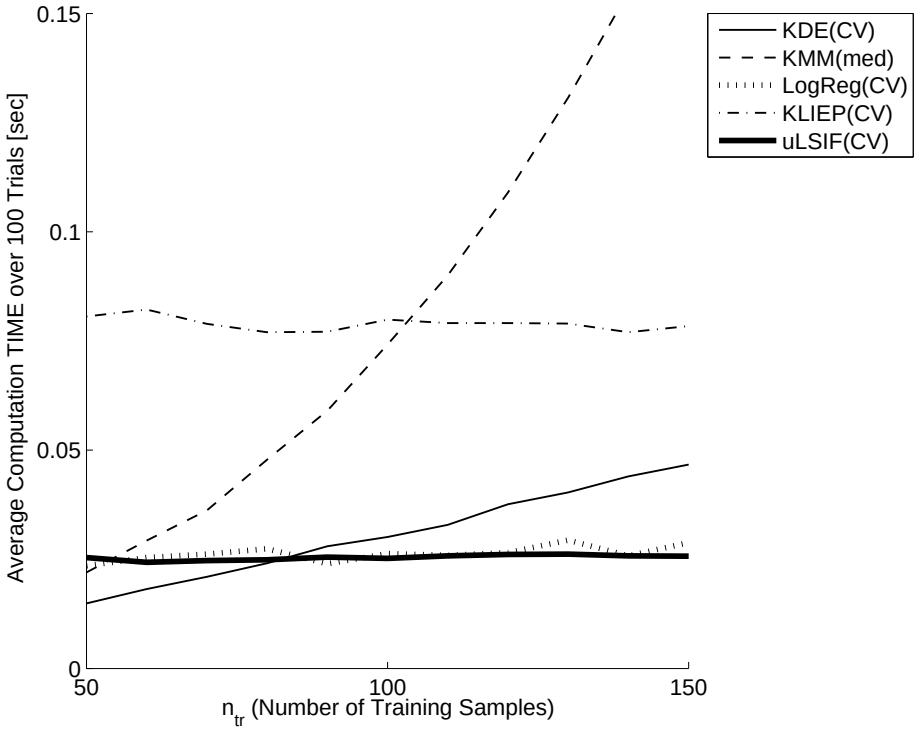


(b) When training sample size is changed

Figure 8: NMSEs averaged over 100 trials in log scale for the artificial data set. Error bars are omitted for clear visibility. Instead, the best method in terms of the mean error and comparable ones based on the *t-test* at the significance level 1% are indicated by '$\circ$'; the methods with significant difference from the best methods are indicated by '$\times$'.

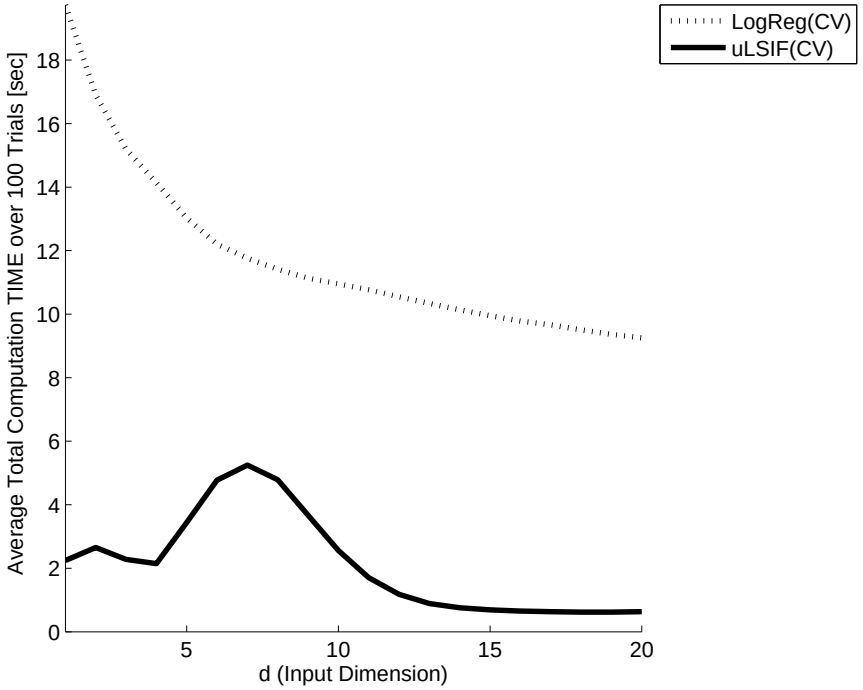


(a) When input dimensionality is changed

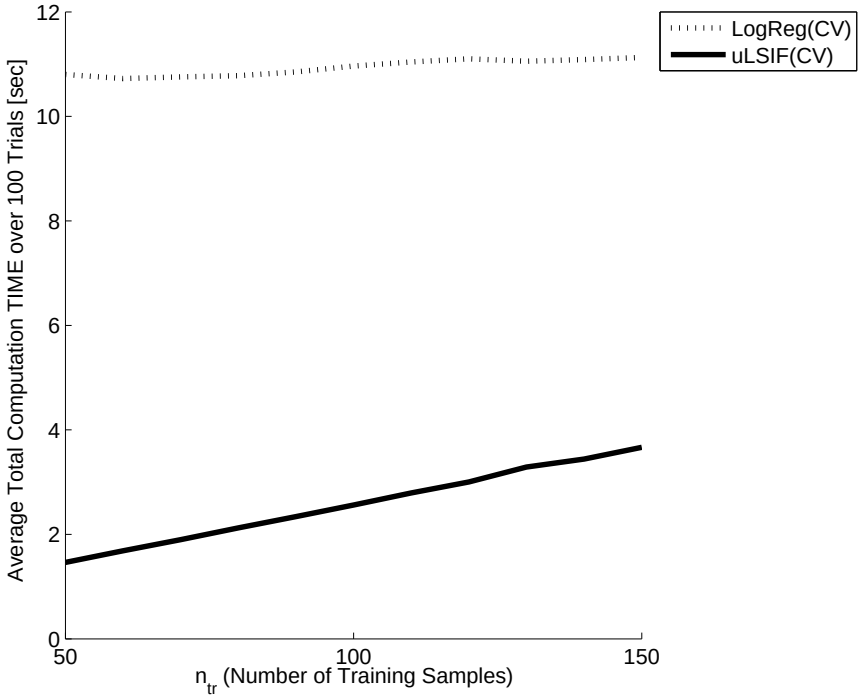


(b) When training sample size is changed

Figure 9: Average computation time (after model selection) over 100 trials for the artificial data set.



(a) When input dimensionality is changed



(b) When training sample size is changed

Figure 10: Average computation time over 100 trials for the artificial data set (including model selection of the Gaussian width σ and the regularization parameter λ over the 9×9 grid).

caused by covariate shift can be asymptotically canceled by weighting the loss function according to the importance.

In addition to training input samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ drawn from a training input density $p_{\mathrm{tr}}(x)$ and test input samples $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ drawn from a test input density $p_{\mathrm{te}}(x)$, suppose that we are given training *output* samples $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ at the training input points $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. The task is to predict the outputs for test inputs $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ based on the input-output training samples $\{(x_i^{\mathrm{tr}},y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$.

We use the following kernel model for function learning:

$$\widehat{f}(x;\theta)=\sum_{\ell=1}^{t}\theta_\ell K_h(x,m_\ell),$$

where $K_h(x,x')$ is the Gaussian kernel (21) and m_ℓ is a template point randomly chosen from $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ without replacement. We set the number of kernels at $t=50$. We learn the parameter θ by *importance weighted regularized least-squares* (IWRLS) (Evgeniou et al., 2000; Sugiyama and Müller, 2005):

$$\widehat{\theta}_{\mathrm{IWRLS}}\equiv\operatorname*{argmin}_{\theta}\left[\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}})\left(\widehat{f}(x_i^{\mathrm{tr}};\theta)-y_i^{\mathrm{tr}}\right)^2+\gamma\|\theta\|^2\right]. \tag{37}$$

It is known that IWRLS is consistent when the true importance $w(x_i^{\mathrm{tr}})$ is used as weights—unweighted RLS is not consistent due to covariate shift, given that the true learning target function $f(x)$ is not realizable by the model $\widehat{f}(x)$ (Shimodaira, 2000).

The solution $\widehat{\theta}_{\mathrm{IWRLS}}$ is analytically given by

$$\widehat{\theta}_{\mathrm{IWRLS}}=(K^\top\widehat{W}K+\gamma I_b)^{-1}K^\top\widehat{W}y^{\mathrm{tr}},$$

where

$$\begin{aligned}K_{i,\ell}&=K_h(x_i^{\mathrm{tr}},m_\ell),\\ \widehat{W}&=\mathrm{diag}\left(\widehat{w}(x_1^{\mathrm{tr}}),\widehat{w}(x_2^{\mathrm{tr}}),\ldots,\widehat{w}(x_{n_{\mathrm{tr}}}^{\mathrm{tr}})\right),\\ y^{\mathrm{tr}}&=(y_1^{\mathrm{tr}},y_2^{\mathrm{tr}},\ldots,y_{n_{\mathrm{tr}}}^{\mathrm{tr}})^\top.\end{aligned}$$

$\mathrm{diag}(a,b,\ldots,c)$ denotes the diagonal matrix with the diagonal elements $a,b,\ldots,c$.

The kernel width h and the regularization parameter γ in IWRLS (37) are chosen by *importance weighted CV* (IWCV) (Sugiyama et al., 2007). More specifically, we first divide the training samples $\{z_i^{\mathrm{tr}}\mid z_i^{\mathrm{tr}}=(x_i^{\mathrm{tr}},y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ into R disjoint subsets $\{\mathcal{Z}_r^{\mathrm{tr}}\}_{r=1}^{R}$. Then a function $\widehat{f}_r(x)$ is learned using $\{\mathcal{Z}_j^{\mathrm{tr}}\}_{j\neq r}$ by IWRLS and its mean test error for the remaining samples $\mathcal{Z}_r^{\mathrm{tr}}$ is computed:

$$\frac{1}{|\mathcal{Z}_r^{\mathrm{tr}}|}\sum_{(x,y)\in\mathcal{Z}_r^{\mathrm{tr}}}\widehat{w}(x)\mathrm{loss}\left(\widehat{f}_r(x),y\right),$$

where

$$\mathrm{loss}(\widehat{y},y)=\begin{cases}(\widehat{y}-y)^2 & \text{(Regression)},\\ \frac{1}{2}(1-\mathrm{sign}\{\widehat{y}y\}) & \text{(Classification)}.\end{cases}$$

We repeat this procedure for $r=1,2,\ldots,R$ and choose the kernel width h and the regularization parameter γ so that the average of the above mean test error over all r is minimized. We set the number of folds in IWCV at $R=5$. IWCV is shown to be an (almost) unbiased estimator of the

Data	Uniform	KDE (CV)	KMM (med)	LogReg (CV)	KLIEP (CV)	uLSIF (CV)
kin-8fh	1.00(0.34)	1.22(0.52)	1.55(0.39)	1.31(0.39)	**0.95**(**0.31**)	**1.02**(**0.33**)
kin-8fm	1.00(0.39)	1.12(0.57)	1.84(0.58)	1.38(0.57)	**0.86**(**0.35**)	**0.88**(**0.39**)
kin-8nh	**1.00**(**0.26**)	1.09(0.20)	1.19(0.29)	1.09(0.19)	**0.99**(**0.22**)	**1.02**(**0.18**)
kin-8nm	**1.00**(**0.30**)	1.14(0.26)	1.20(0.20)	1.12(0.21)	**0.97**(**0.25**)	1.04(0.25)
abalone	**1.00**(**0.50**)	1.02(0.41)	**0.91**(**0.38**)	**0.97**(**0.49**)	**0.94**(**0.67**)	**0.96**(**0.61**)
image	**1.00**(**0.51**)	0.98(0.45)	1.08(0.54)	**0.98**(**0.46**)	**0.94**(**0.44**)	**0.98**(**0.47**)
ringnorm	1.00(0.04)	0.87(0.04)	**0.87**(**0.04**)	0.95(0.08)	0.99(0.06)	0.91(0.08)
twonorm	1.00(0.58)	1.16(0.71)	**0.94**(**0.57**)	**0.91**(**0.61**)	**0.91**(**0.52**)	**0.88**(**0.57**)
waveform	1.00(0.45)	1.05(0.47)	0.98(0.31)	**0.93**(**0.32**)	**0.93**(**0.34**)	**0.92**(**0.32**)
Average	1.00(0.38)	1.07(0.40)	1.17(0.37)	1.07(0.37)	0.94(0.35)	0.96(0.36)
Comp. time	—	0.82	3.50	3.27	2.23	1.00

Table 2: Mean test error averaged over 100 trials for covariate shift adaptation in regression and classification. The numbers in the brackets are the standard deviation. All the error values are normalized by that of 'Uniform' (uniform weighting, or equivalently no importance weighting). For each data set, the best method in terms of the mean error and comparable ones based on the *Wilcoxon signed rank test* at the significance level 1% are described in bold face. The upper half corresponds to regression data sets taken from DELVE (Rasmussen et al., 1996), while the lower half correspond to classification data sets taken from IDA (Rätsch et al., 2001). All the methods are implemented using the *MATLAB*® environment, where the *CPLEX*® optimizer is used for solving quadratic programs in KMM and the *LIBLINEAR* implementation is used for LogReg (Lin et al., 2007).

generalization error, while unweighted CV with misspecified models is biased due to covariate shift (Zadrozny, 2004; Sugiyama et al., 2007).

The data sets provided by DELVE (Rasmussen et al., 1996) and IDA (Rätsch et al., 2001) are used for performance evaluation. Each data set consists of input/output samples $\{(x_k, y_k)\}_{k=1}^{n}$. We normalize all the input samples $\{x_k\}_{k=1}^{n}$ into $[0,1]^d$ and choose the test samples $\{(x_j^{\mathrm{te}}, y_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ from the pool $\{(x_k, y_k)\}_{k=1}^{n}$ as follows. We randomly choose one sample (x_k, y_k) from the pool and accept this with probability $\min(1, 4(x_k^{(c)})^2)$, where $x_k^{(c)}$ is the c-th element of x_k and c is randomly determined and fixed in each trial of the experiments. Then we remove x_k from the pool regardless of its rejection or acceptance, and repeat this procedure until n_{te} samples are accepted. We choose the training samples $\{(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ uniformly from the rest. Thus, in this experiment, the test input density tends to be lower than the training input density when $x_k^{(c)}$ is small. We set the number of samples at $n_{\mathrm{tr}} = 100$ and $n_{\mathrm{te}} = 500$ for all data sets. Note that we only use $\{(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ for training regressors or classifiers; the test output values $\{y_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ are used only for evaluating the generalization performance.

We run the experiments 100 times for each data set and evaluate the *mean test error*:

$$\frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \mathrm{loss}\left(\widehat{f}(x_j^{\mathrm{te}}), y_j^{\mathrm{te}}\right).$$

The results are summarized in Table 2, where 'Uniform' denotes uniform weights (or equivalently, no importance weight). The numbers in the brackets are the standard deviation. All the error values are normalized so that the mean error of Uniform is one. For each data set, the best method in terms of the mean error and comparable ones based on the *Wilcoxon signed rank test* at the significance level 1% are described in bold face. The upper half of the table corresponds to regression data sets taken from DELVE (Rasmussen et al., 1996), while the lower half correspond to classification data sets taken from IDA (Rätsch et al., 2001). All the methods are implemented using the *MATLAB*® environment, where the *CPLEX*® optimizer is used for solving quadratic programs in KMM and the *LIBLINEAR* implementation is used for LogReg (Lin et al., 2007).

The table shows that the generalization performance of uLSIF tends to be better than that of Uniform, KDE, KMM, and LogReg, while it is comparable to the best existing method (KLIEP). The mean computation time over 100 trials is described in the bottom row of the table, where the value is normalized so that the computation time of uLSIF is one. This shows that the computation time of uLSIF is much shorter than KLIEP. Thus, uLSIF is overall shown to be useful in covariate shift adaptation.

6.3 Outlier Detection

Finally, we apply importance estimation methods in outlier detection.

Here, we consider an outlier detection problem of finding irregular samples in a data set ("evaluation data set") based on another data set ("model data set") that only contains regular samples. Defining the importance over two sets of samples, we can see that the importance values for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus the importance values could be used as an index of the degree of outlyingness in this scenario. Since the evaluation data set has wider support than the model data set, we regard the evaluation data set as the training set $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ (that is, the denominator in the importance) and the model data set as the test set $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ (that is, the numerator in the importance). Then outliers tend to have smaller importance values (that is, close to zero).

We again test KMM(med), LogReg(CV), KLIEP(CV), and uLSIF(CV) for importance estimation; in addition, we include native outlier detection methods for comparison purposes. The outlier detection problem that the native methods used below solve is to find outliers in a single data set $\{x_k\}_{k=1}^{n}$—the native methods can be employed in the current scenario just by finding outliers from all samples:

$$\{x_k\}_{k=1}^{n} = \{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}} \cup \{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}.$$

One-class support vector machine (OSVM): The *support vector machine* (SVM) (Vapnik, 1998; Schölkopf and Smola, 2002) is one of the most successful classification algorithms in machine learning. The core idea of SVM is to separate samples in different classes by the maximum margin hyperplane in a kernel-induced feature space.

OSVM is an extension of SVM to outlier detection (Schölkopf et al., 2001). The basic idea of OSVM is to separate data samples $\{x_k\}_{k=1}^{n}$ into outliers and inliers by a hyperplane in a Gaussian reproducing kernel Hilbert space. More specifically, the solution of OSVM is given

as the solution of the following convex quadratic programming problem:

$$\min_{\{w_k\}_{k=1}^n} \frac{1}{2}\sum_{k,k'=1}^{n} w_k w_{k'} K_\sigma(x_k, x_{k'})$$
$$\text{subject to } \sum_{k=1}^{n} w_k = 1 \text{ and } 0 \le w_1, w_2, \ldots, w_n \le \frac{1}{\nu n},$$

where ν ($0 \le \nu \le 1$) is the maximum fraction of outliers.

We use the inverse distance of a sample from the separating hyperplane as an outlier score. The OSVM solution is dependent on the outlier ratio ν and the Gaussian kernel width σ, and there seems to be no systematic method to determine the values of these tuning parameters. Here we use the median distance between samples as the Gaussian width, which is a popular heuristic (Schölkopf and Smola, 2002; Song et al., 2007). The value of ν is fixed at the true output ratio, that is, the ideal optimal value. Thus the simulation results below should be slightly in favor of OSVM.

Local outlier factor (LOF): LOF is the score to detect a local outlier which lies relatively far from the nearest dense region (Breunig et al., 2000). For a prefixed natural number k, the LOF value of a sample x is defined by

$$\mathrm{LOF}_R(x) = \frac{1}{k}\sum_{i=1}^{k} \frac{\mathrm{imd}_k(\mathrm{nearest}_i(x))}{\mathrm{imd}_k(x)},$$

where $\mathrm{nearest}_i(x)$ denotes the i-th nearest neighbor of x and $\mathrm{imd}_k(x)$ denotes the inverse mean distance from x to its k nearest neighbors:

$$\mathrm{imd}_k(x) = \frac{1}{\frac{1}{k}\sum_{i=1}^{k} \|x - \mathrm{nearest}_i(x)\|}.$$

If x alone is apart from a cloud of points, $\mathrm{imd}_k(x)$ tends to become smaller than than $\mathrm{imd}_k(\mathrm{nearest}_i(x))$ for all i. Then the LOF value gets large and therefore such a point is regarded as an outlier. The performance of LOF depends on the choice of the parameter k and there seems no systematic way to find an appropriate value of k. Here we test several different values of k.

Kernel density estimator (KDE'): A naive density estimation of all data samples $\{x_k\}_{k=1}^n$ can also be used for outlier detection since the density value itself could be regarded as an outlier score. We use KDE with the Gaussian kernel (21) for density estimation, where the kernel width is determined based on 5-fold LCV.

All the methods are implemented using the R environment—we use the *ksvm* routine in the *kernlab* package for OSVM (Karatzoglou et al., 2004) and the *lofactor* routine in the *dprep* package for LOF (Fernandez, 2005).

The data sets provided by IDA (Rätsch et al., 2001) are used for performance evaluation. These data sets are binary classification data sets consisting of positive/negative and training/test samples. We allocate all positive training samples for the "model" set, while all positive test samples and a fraction ρ $(= 0.01, 0.02, 0.05)$ of negative test samples are assigned in the "evaluation" set. Thus, we regard the positive samples as regular and the negative samples as irregular.

In the evaluation of the performance of outlier detection methods, it is important to take into account both the detection rate (the amount of true outliers an outlier detection algorithm can find) and the detection accuracy (the amount of true inliers that an outlier detection algorithm misjudges as outliers). Since there is a trade-off between the detection rate and the detection accuracy, we adopt the area under the ROC curve (AUC) as our error metric (Bradley, 1997).

The mean AUC values over 20 trials as well as the computation time are summarized in Table 3, showing that uLSIF works fairly well. KLIEP works slightly better than uLSIF, but uLSIF is computationally much more efficient. LogReg overall works reasonably well, but it performs poorly for some data sets and the average AUC performance is not as good as uLSIF or KLIEP. KMM and OSVM are not comparable to uLSIF in AUC and they are computationally inefficient. Note that we also tested KMM and OSVM with several different Gaussian widths and experimentally found that the heuristic of using the median sample distance as the Gaussian kernel width works reasonably well in this experiment. Thus the AUC values of KMM and OSVM are close to optimal. LOF with large k is shown to work well, although it is not clear whether the heuristic of simply using large k is always appropriate or not. The computational cost of LOF is high since nearest neighbor search is computationally expensive. KDE' works reasonably well, but its performance is not as good as uLSIF and KLIEP.

Overall, uLSIF is shown to work well with low computational costs.

7. Conclusions

The importance is useful in various machine learning scenarios such as covariate shift adaptation and outlier detection. In this paper, we proposed a new method of importance estimation that can avoid solving a substantially more difficult task of density estimation. We formulated the importance estimation problem as least-squares function fitting and casted the optimization problem as a convex quadratic program (we referred to it as LSIF). We theoretically elucidated the convergence property of LSIF and showed that the entire regularization path of LSIF can be efficiently computed based on a parametric optimization technique. We further developed an approximation algorithm (we called it uLSIF), which allows us to obtain the closed-form solution. We showed that the leave-one-out cross-validation score can be computed analytically for uLSIF—this makes the computation of uLSIF highly efficient. We carried out extensive simulations in covariate shift adaptation and outlier detection, and experimentally confirmed that the proposed uLSIF is computationally more efficient than existing approaches, while the accuracy of uLSIF is comparable to the best existing methods. Thanks to the low computational cost, uLSIF would be highly scalability to large data sets, which is very important in practical applications.

We have given convergence proofs for LSIF and uLSIF. A possible future direction to pursue along this line is to show the convergence of LSIF and uLSIF in non-parametric cases, for example, following Nguyen et al. (2008) and Sugiyama et al. (2008b). We are currently exploring various possible applications of important estimation methods beyond covariate shift adaptation or outlier detection, for example, feature selection, conditional distribution estimation, independent component analysis, and dimensionality reduction—we believe that importance estimation could be used as a new versatile tool in statistical machine learning.

Data		uLSIF	KLIEP	LogReg	KMM	OSVM	LOF			KDE'
Name	ρ	(CV)	(CV)	(CV)	(med)	(med)	$k=5$	$k=30$	$k=50$	(CV)
banana	0.01	0.851	0.815	0.447	0.578	0.360	0.838	0.915	0.919	0.934
	0.02	0.858	0.824	0.428	0.644	0.412	0.813	0.918	0.920	0.927
	0.05	0.869	0.851	0.435	0.761	0.467	0.786	0.907	0.909	0.923
b-cancer	0.01	0.463	0.480	0.627	0.576	0.508	0.546	0.488	0.463	0.400
	0.02	0.463	0.480	0.627	0.576	0.506	0.521	0.445	0.428	0.400
	0.05	0.463	0.480	0.627	0.576	0.498	0.549	0.480	0.452	0.400
diabetes	0.01	0.558	0.615	0.599	0.574	0.563	0.513	0.403	0.390	0.425
	0.02	0.558	0.615	0.599	0.574	0.563	0.526	0.453	0.434	0.425
	0.05	0.532	0.590	0.636	0.547	0.545	0.536	0.461	0.447	0.435
f-solar	0.01	0.416	0.485	0.438	0.494	0.522	0.480	0.441	0.385	0.378
	0.02	0.426	0.456	0.432	0.480	0.550	0.442	0.406	0.343	0.374
	0.05	0.442	0.479	0.432	0.532	0.576	0.455	0.417	0.370	0.346
german	0.01	0.574	0.572	0.556	0.529	0.535	0.526	0.559	0.552	0.561
	0.02	0.574	0.572	0.556	0.529	0.535	0.553	0.549	0.544	0.561
	0.05	0.564	0.555	0.540	0.532	0.530	0.548	0.571	0.555	0.547
heart	0.01	0.659	0.647	0.833	0.623	0.681	0.407	0.659	0.739	0.638
	0.02	0.659	0.647	0.833	0.623	0.678	0.428	0.668	0.746	0.638
	0.05	0.659	0.647	0.833	0.623	0.681	0.440	0.666	0.749	0.638
satimage	0.01	0.812	0.828	0.600	0.813	0.540	0.909	0.930	0.896	0.916
	0.02	0.829	0.847	0.632	0.861	0.548	0.785	0.919	0.880	0.898
	0.05	0.841	0.858	0.715	0.893	0.536	0.712	0.895	0.868	0.892
splice	0.01	0.713	0.748	0.368	0.541	0.737	0.765	0.778	0.768	0.845
	0.02	0.754	0.765	0.343	0.588	0.744	0.761	0.793	0.783	0.848
	0.05	0.734	0.764	0.377	0.643	0.723	0.764	0.785	0.777	0.849
thyroid	0.01	0.534	0.720	0.745	0.681	0.504	0.259	0.111	0.071	0.256
	0.02	0.534	0.720	0.745	0.681	0.505	0.259	0.111	0.071	0.256
	0.05	0.534	0.720	0.745	0.681	0.485	0.259	0.111	0.071	0.256
titanic	0.01	0.525	0.534	0.602	0.502	0.456	0.520	0.525	0.525	0.461
	0.02	0.496	0.498	0.659	0.513	0.526	0.492	0.503	0.503	0.472
	0.05	0.526	0.521	0.644	0.538	0.505	0.499	0.512	0.512	0.433
twonorm	0.01	0.905	0.902	0.161	0.439	0.846	0.812	0.889	0.897	0.875
	0.02	0.896	0.889	0.197	0.572	0.821	0.803	0.892	0.901	0.858
	0.05	0.905	0.903	0.396	0.754	0.781	0.765	0.858	0.874	0.807
waveform	0.01	0.890	0.881	0.243	0.477	0.861	0.724	0.887	0.889	0.861
	0.02	0.901	0.890	0.181	0.602	0.817	0.690	0.887	0.890	0.861
	0.05	0.885	0.873	0.236	0.757	0.798	0.705	0.847	0.874	0.831
Average		0.661	0.685	0.530	0.608	0.596	0.594	0.629	0.622	0.623
Comp. time		1.00	11.7	5.35	751	12.4	85.5			8.70

Table 3: Mean AUC values for outlier detection over 20 trials for the benchmark data sets. All the methods are implemented using the R environment, where quadratic programs in KMM are solved by the *ipop* optimizer (Karatzoglou et al., 2004), the *ksvm* routine is used for OSVM (Karatzoglou et al., 2004), and the *lofactor* routine is used for LOF (Fernandez, 2005).

Acknowledgments

The authors wish to thank Issei Sato for fruitful discussion and helpful comments. The authors would also like to thank the anonymous referees whose comments helped to improve the paper further. This work was supported by MEXT (20680007), SCAT, and AOARD.

Appendix A. Existence of the Inverse Matrix of $\widehat{G}$

Here we prove Lemma 1.

Let us consider the following system of linear equations:

$$\begin{pmatrix} \widehat{H} & -\widehat{E}^\top \\ -\widehat{E} & O_{|\widehat{\mathcal{A}}|\times|\widehat{\mathcal{A}}|} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0_b \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}, \tag{38}$$

where x and y are b- and $|\widehat{\mathcal{A}}|$-dimensional vectors, respectively. From the upper half of Eq. (38), we have

$$x = \widehat{H}^{-1}\widehat{E}^\top y.$$

Substituting this into the lower half of Eq. (38), we have

$$\widehat{E}\widehat{H}^{-1}\widehat{E}^\top y = 0_{|\widehat{\mathcal{A}}|}.$$

From the definition, the rank of the matrix $\widehat{E}$ is $|\widehat{\mathcal{A}}|$, that is, $\widehat{E}$ is a row-full rank matrix. As a result, the matrix $\widehat{E}\widehat{H}^{-1}\widehat{E}^\top$ is invertible. Therefore, Eq. (38) has the unique solution $x = 0_b$ and $y = 0_{|\widehat{\mathcal{A}}|}$. This implies that $\widehat{G}$ is invertible.

Appendix B. Active Set of LSIF

Here, we prove Theorem 2.

We prove that the active set $\mathcal{A}$ does not change under the infinitesimal shift of H and h if the strict complementarity condition is satisfied. We regard the pair of a symmetric matrix and a vector (H', h') as an element in the $(\frac{b(b+1)}{2} + b)$-dimensional Euclidean space. We consider the following linear equation:

$$\begin{pmatrix} \alpha' \\ \xi' \end{pmatrix} = \begin{pmatrix} H' & -E^\top \\ -E & O_{|\mathcal{A}|\times|\mathcal{A}|} \end{pmatrix}^{-1} \begin{pmatrix} h' - \lambda 1_b \\ 0_{|\mathcal{A}|} \end{pmatrix},$$

where E is the $|\mathcal{A}| \times b$ indicator matrix determined from the active set $\mathcal{A}$ (see Section 2.3 for the detailed definition). If $H' = H$ and $h' = h$ hold, the solution $(\alpha', \xi') = (\alpha^*(\lambda), \xi^*(\lambda))$ satisfies

$$\begin{aligned} &\alpha'_\ell = 0,\ \xi'_\ell > 0, \quad {}^\forall \ell \in \mathcal{A}, \\ &\alpha'_\ell > 0,\ \xi'_\ell = 0, \quad {}^\forall \ell \notin \mathcal{A}, \end{aligned} \tag{39}$$

because of the strict complementarity condition. On the other hand, if the norm of $(H', h') - (H, h)$ is infinitesimal, the solution (α', ξ') also satisfies Eq. (39) because of the continuity of the relation between (H', h') and (α', ξ').

As a result, there exists an ε-ball B_ε in $\mathbb{R}^{\frac{b(b+1)}{2}+b}$ such that the equality $\mathcal{A} = \{\ell \mid \alpha'_\ell = 0\}$ holds for any $(H', h') \in B_\varepsilon$. Therefore, we have $P(\mathcal{A} \neq \widehat{\mathcal{A}}) \leq P((\widehat{H}, \widehat{h}) \notin B_\varepsilon)$. Due to the large deviation principle (Dembo and Zeitouni, 1998), there is a positive constant c such that

$$-\frac{1}{\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}} \log P((\widehat{H}, \widehat{h}) \notin B_\varepsilon) > c > 0,$$

if $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}$ is large enough. Thus, asymptotically $P(\widehat{\mathcal{A}} \neq \mathcal{A}) < e^{-c \min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}}$ holds.

Appendix C. Learning Curve of LSIF

Here, we prove Theorem 3.

Let us consider the ideal problem (7). Let $\alpha^*(\lambda)$ and $\xi^*(\lambda)$ be the optimal parameter and Lagrange multiplier (that is, the KKT conditions are fulfilled; see Section 2.3) and let $\xi^{*\prime}(\lambda)$ be the vector of non-zero elements of $\xi^*(\lambda)$ defined in the same way as Eq. (11). Then $\alpha^*(\lambda)$ and $\xi^{*\prime}(\lambda)$ satisfy

$$G \begin{pmatrix} \alpha^*(\lambda) \\ \xi^{*\prime}(\lambda) \end{pmatrix} = \begin{pmatrix} h - \lambda 1_b \\ 0_{|\mathcal{A}|} \end{pmatrix}, \tag{40}$$

where

$$G = \begin{pmatrix} H & -E^\top \\ -E & O_{|\mathcal{A}| \times |\mathcal{A}|} \end{pmatrix}.$$

From the central limit theorem and the assumption (18), we have

$$\widehat{h} = h + O_p\left(\frac{1}{\sqrt{n_{\mathrm{te}}}}\right) = h + o_p\left(\frac{1}{n_{\mathrm{tr}}}\right), \tag{41}$$

where O_p and o_p denote the asymptotic order in probability. The assumption (a) implies that the equality

$$\widehat{E} = E \tag{42}$$

holds with exponentially high probability due to Theorem 2. Note that $\widehat{G}$ is the same size as G if $\widehat{E} = E$. Thus we have

$$\widehat{G} = G + \delta G,$$

where

$$\delta G = \begin{pmatrix} \delta H & O_{b \times |\mathcal{A}|} \\ O_{|\mathcal{A}| \times b} & O_{|\mathcal{A}| \times |\mathcal{A}|} \end{pmatrix},$$
$$\delta H = \widehat{H} - H. \tag{43}$$

Combining Eqs. (12), (40), (41), and (42), we have

$$\begin{pmatrix} \widehat{\alpha}(\lambda) \\ \widehat{\xi}'(\lambda) \end{pmatrix} = \widehat{G}^{-1} G \begin{pmatrix} \alpha^*(\lambda) \\ \xi^{*\prime}(\lambda) \end{pmatrix} + o_p\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{44}$$

The matrix Taylor expansion (Petersen and Pedersen, 2007) yields

$$\widehat{G}^{-1} = G^{-1} - G^{-1}\delta G G^{-1} + G^{-1}\delta G G^{-1}\delta G G^{-1} - \cdots, \tag{45}$$

and the central limit theorem asserts that

$$\delta H = O_p\left(\frac{1}{\sqrt{n_{\mathrm{tr}}}}\right). \tag{46}$$

Combining Eqs. (44), (45), (14), and (46), we have

$$\begin{aligned}\delta\alpha &= \widehat{\alpha}(\lambda) - \alpha^*(\lambda) &(47)\\ &= -A\delta H\alpha^*(\lambda) + A\delta HA\delta H\alpha^*(\lambda) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). &(48)\end{aligned}$$

Through direct calculation, we can confirm that

$$AHA = A. \tag{49}$$

Similar to Eq. (15), it holds that

$$\alpha^*(\lambda) = A(h - \lambda 1_b). \tag{50}$$

From Eqs. (49) and (50), we have

$$A(H\alpha^*(\lambda) - h) = -\lambda A 1_b. \tag{51}$$

Eqs. (43), (4), and (3) imply

$$\mathbb{E}[\delta H] = O_{b\times b}. \tag{52}$$

From Eqs. (2) and (47), we have

$$J(\widehat{\alpha}(\lambda)) = J(\alpha^*(\lambda)) + \frac{1}{2}\delta\alpha^\top H\delta\alpha + (H\alpha^*(\lambda) - h)^\top\delta\alpha. \tag{53}$$

From Eqs. (46), (48), and (49), we have

$$\begin{aligned}\mathbb{E}\left[\delta\alpha^\top H\delta\alpha\right] &= \mathrm{tr}(H\,\mathbb{E}\left[\delta\alpha\delta\alpha^\top\right])\\ &= \mathrm{tr}(AHA\,\mathbb{E}\left[(\delta H\alpha^*(\lambda))(\delta H\alpha^*(\lambda))^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ &= \mathrm{tr}(A\,\mathbb{E}\left[(\delta H\alpha^*(\lambda))(\delta H\alpha^*(\lambda))^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). &(54)\end{aligned}$$

From Eqs. (48), (51), and (52), we have

$$\begin{aligned}\mathbb{E}\left[\delta\alpha^\top(H\alpha^*(\lambda) - h)\right] &= -\,\mathbb{E}\left[(\delta H\alpha^*(\lambda) - \delta HA\delta H\alpha^*(\lambda))^\top A(H\alpha^*(\lambda) - h)\right] + o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ &= \mathbb{E}\left[(\delta H\alpha^*(\lambda) - \delta HA\delta H\alpha^*(\lambda))^\top\lambda A 1_b\right] + o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ &= -\,\lambda\,\mathrm{tr}(A\,\mathbb{E}\left[(\delta H\alpha^*(\lambda))(\delta HA1_b)^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). &(55)\end{aligned}$$

Combining Eqs. (53), (54), and (55), we have

$$\begin{aligned}\mathbb{E}\left[J(\widehat{\alpha}(\lambda))\right] =& J(\alpha^*(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(A\,\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))^\top\right])\\ &- \frac{\lambda}{n_{\mathrm{tr}}}\mathrm{tr}(A\,\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HA1_b)^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).\end{aligned}$$

According to the central limit theorem, $\sqrt{n_{\mathrm{tr}}}\delta H_{i,j}$ asymptotically follows the normal distribution with mean zero and variance

$$\int \varphi_i^2(x)\varphi_j^2(x)p_{\mathrm{tr}}(x)dx - H_{i,j}^2,$$

and the asymptotic covariance between $\sqrt{n_{\mathrm{tr}}}\delta H_{i,j}$ and $\sqrt{n_{\mathrm{tr}}}\delta H_{i',j'}$ is given by

$$\int \varphi_i(x)\varphi_j(x)\varphi_{i'}(x)\varphi_{j'}(x)p_{\mathrm{tr}}(x)dx - H_{i,j}H_{i',j'}.$$

Then we have

$$\begin{aligned}\lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))^\top\right] &= C_{w^*,w^*},\\ \lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HA1_b)^\top\right] &= C_{w^*,v},\end{aligned}$$

where $C_{w,w'}$ is the $b\times b$ covariance matrix with the (ℓ,ℓ')-th element being the covariance between $w(x)\varphi_\ell(x)$ and $w'(x)\varphi_{\ell'}(x)$ under $p_{\mathrm{tr}}(x)$. Then we obtain Eq. (19).

Appendix D. Regularization Path of LSIF

Here, we derive the regularization path tracking algorithm given in Figure 1.

When λ is greater than or equal to $\max_k \widehat{h}_k$, the solution of the KKT conditions (9)–(10) is provided as $\alpha = 0_b$, $\xi = \lambda 1_b - \widehat{h} \geq 0_b$. Therefore, the initial value of λ_0 is $\max_k \widehat{h}_k$, and the corresponding optimal solution is $\widehat{\alpha}(\lambda_0) = 0_b$.

Since $\widehat{\xi}'(\lambda)$ corresponds to non-zero elements of $\widehat{\xi}(\lambda)$ as shown in Eq. (11), we have

$$\widehat{\xi}_j(\lambda) = \begin{cases}\widehat{\xi}'_i(\lambda) & \text{if } j = \widehat{j}_i,\\ 0 & \text{otherwise.}\end{cases} \tag{56}$$

When λ is decreased, the solutions $\widehat{\alpha}(\lambda)$ and $\widehat{\xi}(\lambda)$ still satisfy Eqs. (12) and (56) as long as the active set $\widehat{\mathcal{A}}$ remains unchanged. Change points of the active set can be found by examining the non-negativity conditions of $\widehat{\alpha}(\lambda)$ and $\widehat{\xi}(\lambda)$ as follows. Suppose λ is decreased and the non-negativity condition

$$\begin{pmatrix}\widehat{\alpha}(\lambda)\\ \widehat{\xi}(\lambda)\end{pmatrix} \geq 0_{2b}$$

is violated at $\lambda = \lambda'$. That is, both $\widehat{\alpha}(\lambda') \geq 0_b$ and $\widehat{\xi}(\lambda') \geq 0_b$ hold, and either $\widehat{\alpha}(\lambda'-\varepsilon) \geq 0_b$ or $\widehat{\xi}(\lambda'-\varepsilon) \geq 0_b$ is violated for any $\varepsilon > 0$. If $\widehat{\alpha}_j(\lambda') = 0$ for $j \notin \widehat{\mathcal{A}}$, j should be added to the active set

$\widehat{\mathcal{A}}$; on the other hand, if $\widehat{\xi}_j(\lambda') = 0$ for some $j \in \widehat{\mathcal{A}}$, $\widehat{\alpha}_j(\lambda')$ will take a positive value and therefore j should be removed from the active set $\widehat{\mathcal{A}}$. Then, for the updated active set, we compute the solutions by Eqs. (12) and (56). Iterating this procedure until λ reaches zero, we can obtain the entire regularization path.

Note that we omitted some minor exceptional cases for the sake of simplicity—treatments for all possible exceptions and the rigorous convergence property are exhaustively studied in Best (1982).

Appendix E. Negative Index Set of $\beta^\circ(\lambda)$

Here we prove Theorem 4.

As explained in Appendix B, we regard the pair of a symmetric matrix and a vector (H', h') as an element in the $(\frac{b(b+1)}{2} + b)$-dimensional Euclidean space.

We consider the linear equation

$$\beta' = (H' + \lambda I_b)^{-1} h'.$$

Due to the assumption, for $H' = H$ and $h' = h$, we have

$$\beta'_\ell \neq 0,\ \ell = 1, 2, \ldots, b. \tag{57}$$

On the other hand, if the norm of $(H', h') - (H, h)$ is infinitesimal, the solution β' also satisfies Eq. (57), and the sign of β'_ℓ is same as that of β_ℓ for $\ell = 1, 2, \ldots, b$, because of the continuity of the relation between (H', h') and β'.

As a result, there exists an ε-ball B_ε in $\mathbb{R}^{\frac{b(b+1)}{2}+b}$ such that the equality $\mathcal{B} = \widetilde{\mathcal{B}}$ holds for any $(H', h') \in B_\varepsilon$. Therefore, we have $P(\mathcal{B} \neq \widetilde{\mathcal{B}}) \leq P((\widehat{H}, \widehat{h}) \notin B_\varepsilon)$. Due to the large deviation principle (Dembo and Zeitouni, 1998), there is a positive constant c such that

$$-\frac{1}{\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}} \log P((\widehat{H}, \widehat{h}) \notin B_\varepsilon) > c > 0,$$

if $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}$ is large enough. Thus, asymptotically $P(\mathcal{B} \neq \widetilde{\mathcal{B}}) < e^{-c \min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}}$ holds.

Appendix F. Learning Curve of uLSIF

Here, we prove Theorem 5.

Let

$$\widehat{B}_\lambda = \widehat{H} + \lambda I_b.$$

The matrix Taylor expansion (Petersen and Pedersen, 2007) yields

$$\widehat{B}_\lambda^{-1} = B_\lambda^{-1} - B_\lambda^{-1} \delta H B_\lambda^{-1} + B_\lambda^{-1} \delta H B_\lambda^{-1} \delta H B_\lambda^{-1} - \cdots. \tag{58}$$

Let $\widetilde{\mathcal{B}} \subset \{1, 2, \ldots, b\}$ be the set of negative indices of $\widetilde{\beta}(\lambda)$, that is,

$$\widetilde{\mathcal{B}} = \{\ell \mid \widetilde{\beta}_\ell(\lambda) < 0,\ \ell = 1, 2, \ldots, b\}.$$

Let $\widehat{D}$ be the b-dimensional diagonal matrix with the ℓ-th diagonal element

$$\widehat{D}_{\ell,\ell} = \begin{cases} 0 & \ell \in \widetilde{\mathcal{B}}, \\ 1 & \text{otherwise}. \end{cases}$$

The assumption (a) implies that the equality

$$\widehat{D} = D \tag{59}$$

holds with exponentially high probability due to Theorem 4. Combining Eqs. (59), (41), (58), and (24), we have

$$\begin{aligned}\delta\beta &= \widehat{\beta}(\lambda) - \beta^*(\lambda)\\ &= \widehat{D}\widehat{B}_\lambda^{-1}\widehat{h} - DB_\lambda^{-1}h\\ &= -DB_\lambda^{-1}\delta H\beta^\circ(\lambda) + DB_\lambda^{-1}\delta HB_\lambda^{-1}\delta H\beta^\circ(\lambda) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).\end{aligned} \tag{60}$$

From Eqs. (46) and (60), we have

$$\mathbb{E}\left[\delta\beta^\top H\delta\beta\right] = \mathrm{tr}(B_\lambda^{-1}DHDB_\lambda^{-1}\,\mathbb{E}\left[(\delta H\beta^\circ(\lambda))(\delta H\beta^\circ(\lambda))^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{61}$$

From Eqs. (52) and (24), we have

$$\begin{aligned}\mathbb{E}\left[\delta\beta^\top(H\beta^*(\lambda)-h)\right] =& \mathbb{E}\left[(-\delta H\beta^\circ(\lambda) + \delta HB_\lambda^{-1}\delta H\beta^\circ(\lambda))^\top B_\lambda^{-1}D(H\beta^*(\lambda)-h)\right]\\ &+ o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ =& \mathbb{E}\left[\mathrm{tr}(B_\lambda^{-1}(\delta H\beta^\circ(\lambda))(\delta HB_\lambda^{-1}D(H\beta^*(\lambda)-h))^\top)\right]\\ &+ o\left(\frac{1}{n_{\mathrm{tr}}}\right).\end{aligned} \tag{62}$$

Combining Eqs. (53), (61), and (62), we have

$$\begin{aligned}\mathbb{E}\left[J(\widehat{\beta}(\lambda))\right] =& J(\beta^*(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(B_\lambda^{-1}DHDB_\lambda^{-1}\,\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))^\top)\\ &+ \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(B_\lambda^{-1}\,\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HB_\lambda^{-1}D(H\beta^*(\lambda)-h))^\top) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).\end{aligned}$$

According to the central limit theorem, we have

$$\begin{aligned}\lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))^\top] &= C_{w^\circ,w^\circ},\\ \lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HB_\lambda^{-1}D(H\beta^*(\lambda)-h))^\top] &= C_{w^\circ,u}.\end{aligned}$$

Then we obtain Eq. (25).

Appendix G. 'Norm' Upper Bound of Approximation Error for uLSIF

Here we prove Theorem 6.

Using the weighted norm (27), we can express $\mathrm{diff}(\lambda)$ as

$$\mathrm{diff}(\lambda) = \frac{\inf_{\lambda'\geq 0}\|\widehat{\alpha}(\lambda') - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda))}.$$

As shown in Appendix D, $\widehat{\alpha}(\lambda') = 0_b$ holds for some large λ'. Then we immediately have

$$\mathrm{diff}(\lambda) \leq \frac{\|\widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))},$$

which proves Eq. (28). Let $\kappa_{\max}$ be the largest eigenvalue of $\widehat{H}$. Then $\|\widehat{\beta}(\lambda)\|_{\widehat{H}}$ can be upper bounded as

$$\|\widehat{\beta}(\lambda)\|_{\widehat{H}} \leq \sqrt{\kappa_{\max}}\|\widehat{\beta}(\lambda)\|_2 \leq \sqrt{\kappa_{\max}}\|\widetilde{\beta}(\lambda)\|_2,$$

where the first inequality may be confirmed by eigen-decomposing $\widehat{H}$ and the second inequality is clear from the definitions of $\widehat{\beta}(\lambda)$ and $\widetilde{\beta}(\lambda)$. Let $\kappa_{\min}$ be the smallest eigenvalue of $\widehat{H}$. Then an upper bound of $\|\widetilde{\beta}(\lambda)\|_2^2$ is given as

$$\|\widetilde{\beta}(\lambda)\|_2^2 = \widehat{h}^\top (\widehat{H} + \lambda I_b)^{-2}\widehat{h} \leq \frac{1}{(\kappa_{\min} + \lambda)^2}\|\widehat{h}\|_2^2 \leq \frac{1}{\lambda^2}\|\widehat{h}\|_2^2,$$

where the last inequality follows from $\kappa_{\min} > 0$.

Now we have

$$\begin{aligned}
\frac{\|\widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} w(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} &\leq \frac{1}{\sum_{i=1}^{n_{\mathrm{tr}}} w(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} \frac{\sqrt{\kappa_{\max}}\|\widehat{h}\|_2}{\lambda} \\
&= \frac{1}{\sum_{i=1}^{n_{\mathrm{tr}}} \sum_{\ell=1}^{b} \varphi_\ell(x_i^{\mathrm{tr}})\widehat{\beta}_\ell(\lambda)/\|\widehat{\beta}(\lambda)\|_1} \frac{\sqrt{\kappa_{\max}}\|\widehat{h}\|_2}{\lambda\|\widehat{\beta}(\lambda)\|_1}.
\end{aligned}$$

For the denominator of the above expression, we have

$$\sum_{i=1}^{n_{\mathrm{tr}}} \sum_{\ell=1}^{b} \varphi_\ell(x_i^{\mathrm{tr}}) \frac{\widehat{\beta}_\ell(\lambda)}{\|\widehat{\beta}(\lambda)\|_1} \geq \min_{\ell'}\Big(\sum_{i=1}^{n_{\mathrm{tr}}} \varphi_{\ell'}(x_i^{\mathrm{tr}})\Big) \cdot \sum_{\ell=1}^{b} \frac{\widehat{\beta}_\ell(\lambda)}{\|\widehat{\beta}(\lambda)\|_1} = \min_{\ell} \sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}}),$$

where the last equality follows from the non-negativity of $\widehat{\beta}_\ell(\lambda)$. The reciprocal of $\|\widehat{h}\|_2/\|\widehat{\beta}(\lambda)\|_1$ is lower bounded as follows:

$$\frac{\|\widehat{\beta}(\lambda)\|_1}{\|\widehat{h}\|_2} = \left\|\max\left\{\frac{\widetilde{\beta}(\lambda)}{\|\widehat{h}\|_2}, 0\right\}\right\|_1 \geq \left\|\max\left\{\frac{\widetilde{\beta}(\lambda)}{\|\widehat{h}\|_2}, 0\right\}\right\|_\infty = \max_{\ell} \frac{\widetilde{\beta}_\ell(\lambda)}{\|\widehat{h}\|_2},$$

where the last equality follows from the fact that there is an ℓ such that $\widetilde{\beta}_\ell(\lambda) > 0$; otherwise, we have $\sum_{i=1}^{n_{\mathrm{tr}}} w(x_i^{\mathrm{tr}}; \widehat{\beta}) = 0$ which contradicts to the assumption of the theorem. Let us put

$$\kappa e = \frac{\widetilde{\beta}(\lambda)}{\|\widehat{h}\|_2},$$

where $\kappa > 0$ and $e \in \mathbb{R}^b$ such that $\|e\|_2 = 1$. Then we have

$$(\kappa_{\max} + \lambda)^{-1} \leq \kappa \text{ and } e^\top \widehat{h} > 0.$$

Note that there exists an ℓ such that $e_\ell > 0$. Then, we have

$$\max_\ell \frac{\widetilde{\beta}_\ell(\lambda)}{\|\widehat{h}\|_2} = \max_\ell \kappa e_\ell = \kappa \max_\ell e_\ell \geq \frac{1}{\kappa_{\max}+\lambda} \max_\ell e_\ell$$
$$\geq \frac{1}{\kappa_{\max}+\lambda} \min_e \{\max_\ell e_\ell \mid e^\top e = 1, e^\top \widehat{h}/\|\widehat{h}\|_1 > 0\}.$$

Now we prove the following lemma.

Lemma 8 *Let $p_1, p_2, \ldots, p_b$ $(b \geq 2)$ be positive numbers such that*

$$\sum_{\ell=1}^{b} p_\ell = 1,$$

and let

$$\varepsilon = \frac{1}{\sqrt{b}} \min_\ell \frac{p_\ell}{1-p_\ell}.$$

Then, there exists no $e = (e_1, e_2, \ldots, e_b) \in \mathbb{R}^b$ such that the three conditions,

$$\sum_{\ell=1}^{b} e_\ell^2 = 1, \ \sum_{\ell=1}^{b} p_\ell e_\ell > 0, \text{ and } e_\ell < \varepsilon \text{ for } \ell = 1, 2, \ldots, b$$

are satisfied at the same time.

Proof We suppose that $e \in \mathbb{R}^b$ satisfies the three conditions. If $\min_\ell p_\ell/(1-p_\ell) > 1$, we have $p_\ell > 1/2$ for all ℓ. However, this is contradictory to $\sum_{\ell=1}^b p_\ell = 1$. Therefore, we have

$$\min_\ell p_\ell/(1-p_\ell) \leq 1,$$

from which we have

$$\varepsilon \leq 1/\sqrt{b}.$$

The equality constraint $\sum_{\ell=1}^b e_\ell^2 = 1$ implies the condition that there exists an e_i such that $|e_i| \geq 1/\sqrt{b}$. Moreover, we have $e_1, e_2, \ldots, e_b < \varepsilon \leq 1/\sqrt{b}$, and thus there is an e_i such that $e_i \leq -1/\sqrt{b}$. Hence, we have

$$\frac{p_i}{\sqrt{b}} \leq -p_i e_i < \sum_{\ell \neq i} p_\ell e_\ell < \sum_{\ell \neq i} p_\ell \frac{1}{\sqrt{b}} \min_k \frac{p_k}{1-p_k} = (1-p_i)\frac{1}{\sqrt{b}} \min_k \frac{p_k}{1-p_k} \leq \frac{p_i}{\sqrt{b}}.$$

This results in contradiction. ■

Let $p_\ell = \widehat{h}_\ell/\|\widehat{h}\|_1$ and we use Lemma 8. Note that any element of $\widehat{h}$ is positive. Then, we have

$$\frac{\|\widehat{\beta}(\lambda)\|_1}{\|\widehat{h}\|_2} \geq \frac{1}{\kappa_{\max}+\lambda} \cdot \frac{1}{\sqrt{b}} \min_\ell \frac{p_\ell}{\sum_{i \neq \ell} p_i}.$$

Moreover, we have

$$\min_\ell \frac{p_\ell}{\sum_{i \neq \ell} p_i} \geq \frac{\min_\ell \widehat{h}_\ell}{\sum_{\ell'=1}^b \widehat{h}_{\ell'}} = \frac{\min_\ell \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})}{\sum_{\ell'=1}^b \sum_{j=1}^{n_{\mathrm{te}}} \varphi_{\ell'}(x_j^{\mathrm{te}})} \geq \frac{\min_\ell \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})}{n_{\mathrm{te}} b},$$

where the last inequality follows from the assumption $0 < \varphi_\ell(x) \leq 1$. Therefore, we have the inequality

$$\begin{aligned}
&\frac{1}{\sum_{i=1}^{n} w(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} \frac{\sqrt{\kappa_{\max}} \|\widehat{h}\|_2}{\lambda} \\
&\leq b\sqrt{b\,\kappa_{\max}} \left(1 + \frac{\kappa_{\max}}{\lambda}\right) \frac{1}{\min_\ell \sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})} \cdot \frac{n_{\mathrm{te}}}{\min_{\ell'} \sum_{j=1}^{n_{\mathrm{te}}} \varphi_{\ell'}(x_j^{\mathrm{te}})}. \qquad (63)
\end{aligned}$$

An upper bound of $\kappa_{\max}$ is given as follows. For all $a \in \mathbb{R}^b$, the inequality

$$-\sum_{\ell=1}^{b} |a_\ell| \varphi_\ell(x) \leq \sum_{\ell=1}^{b} a_\ell \varphi_\ell(x) \leq \sum_{\ell=1}^{b} |a_\ell| \varphi_\ell(x) \qquad (64)$$

holds because of the positivity of $\varphi_\ell(x)$. Let us define $\overline{a} \in \mathbb{R}^b$ for given $a \in \mathbb{R}^b$ as

$$\overline{a} \;=\; (|a_1|, |a_2|, \ldots, |a_b|)^\top.$$

Note that $\|\overline{a}\|_2 = \|a\|_2$ holds. Then, using Eq. (64), we obtain the inequality

$$a^\top \widehat{H} a = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2 \leq \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} |a_\ell| \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2 = \overline{a}^\top \widehat{H} \overline{a},$$

for any $a \in \mathbb{R}^b$. Therefore, we obtain

$$\max_{\|a\|_2=1} a^\top \widehat{H} a \leq \max_{\|a\|_2=1} \overline{a}^\top \widehat{H} \overline{a} = \max_{\|a\|_2=1,\, a \geq 0_b} a^\top \widehat{H} a, \qquad (65)$$

where the last equality is derived from the relation,

$$\{\overline{a} \mid \|a\|_2 = 1,\; a \in \mathbb{R}^b\} \;=\; \{a \mid \|a\|_2 = 1,\; a \geq 0_b,\; a \in \mathbb{R}^b\}.$$

On the other hand, due to the additional constraint $a \geq 0_b$, the inequality

$$\max_{\|a\|_2=1,\, a \geq 0_b} a^\top \widehat{H} a \;\leq\; \max_{\|a\|_2=1} a^\top \widehat{H} a \qquad (66)$$

holds. From Eqs. (65) and (66), we have

$$\kappa_{\max} = \max_{\|a\|_2=1} a^\top \widehat{H} a = \max_{\|a\|_2=1, a \geq 0_b} a^\top \widehat{H} a = \max_{\|a\|_2=1,\, a \geq 0_b} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2.$$

Using the assumption $0 < \varphi_\ell(x) \leq 1$, we have

$$\begin{aligned}
\kappa_{\max} \;&=\; \max_{\|a\|_2=1,\, a \geq 0_b} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2 \;\leq\; \max_{\|a\|_2=1,\, a \geq 0_b} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \right)^2 \\
&=\; \max_{\|a\|_2=1,\, a \geq 0_b} \left(\sum_{\ell=1}^{b} a_\ell \right)^2 \;\leq\; \max_{\|a\|_2=1,\, a \geq 0_b} b \cdot \sum_{\ell=1}^{b} a_\ell^2 \\
&=\; b, \qquad (67)
\end{aligned}$$

where the Schwarz inequality for a and 1_b is used in the last inequality. The inequalities (63) and (67) lead to the inequality (29).

It is clear that the upper bound (29) is a decreasing function of λ (> 0). For the Gaussian basis function, $\varphi_\ell(x)$ is an increasing function with respect to the Gaussian width σ. Thus, Eq. (29) is a decreasing function of σ.

Appendix H. 'Bridge' Upper Bound of Approximation Error for uLSIF

Here we prove Theorem 7.

From the triangle inequality, we obtain

$$\mathrm{diff}(\lambda) \leq \frac{\inf_{\lambda' \geq 0} \|\widehat{\alpha}(\lambda') - \widehat{\gamma}(\lambda)\|_{\widehat{H}} + \|\widehat{\gamma}(\lambda) - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))}. \tag{68}$$

We derive an upper bound of the first term.

First, we show that the LSIF optimization problem (6) is equivalently expressed as

$$\min_{\alpha \in \mathbb{R}^b} \left[\frac{1}{2} \alpha^\top \widehat{H} \alpha - \widehat{h}^\top \alpha \right]$$
$$\text{subject to } \alpha \geq 0_b, \ \ 1_b^\top \alpha \leq c,$$

which we refer to as LSIF$'$. The KKT conditions of LSIF (6) are given as

$$\begin{cases} \widehat{H}\alpha - \widehat{h} + \lambda 1_b - \mu = 0_b, \\ \alpha \geq 0_b, \ \ \mu \geq 0_b, \ \ \alpha^\top \mu = 0, \end{cases}$$

where μ is the Lagrange multiplier vector. Similarly, the KKT conditions of LSIF$'$ are given as

$$\begin{cases} \widehat{H}\alpha - \widehat{h} + \mu_0 1_b - \mu = 0_b, \\ \alpha \geq 0_b, \ \ \mu \geq 0_b, \ \ \alpha^\top \mu = 0, \\ 1_b^\top \alpha - c \leq 0, \ \mu_0 \geq 0, \ \ (1_b^\top \alpha - c)\mu_0 = 0, \end{cases} \tag{69}$$

where μ and μ_0 are the Lagrange multipliers. Let $(\widehat{\alpha}(\lambda), \widehat{\mu}(\lambda))$ be the solution of the KKT conditions of LSIF. Then, we find that $(\alpha, \mu, \mu_0) = (\widehat{\alpha}(\lambda), \widehat{\mu}(\lambda), \lambda)$ is the solution of Eq. (69) with $c = 1_b^\top \widehat{\alpha}(\lambda)$. Note that LSIF$'$ is a strictly convex optimization problem, and thus $\widehat{\alpha}(\lambda)$ is the unique optimal solution. Conversely, when the solution of Eq. (69) is provided as $(\widehat{\alpha}, \widehat{\mu}, \mu_0)$, LSIF with $\lambda = \mu_0$ has the same optimal solution $\widehat{\alpha}$.

When the optimal solution of LSIFq is $\widehat{\gamma}(\lambda)$, the KKT conditions of LSIFq (30) are given as

$$\widehat{H}\widehat{\gamma}(\lambda) - \widehat{h} + \lambda \widehat{\gamma}(\lambda) - \widehat{\eta} = 0_b, \tag{70}$$
$$\widehat{\gamma}(\lambda) \geq 0_b, \ \ \widehat{\eta} \geq 0_b, \ \ \widehat{\gamma}(\lambda)^\top \widehat{\eta} = 0, \tag{71}$$

where $\widehat{\eta}$ is the Lagrange multiplier vector.

Let $\widehat{\alpha}(\lambda_1)$ be the optimal solution of LSIF$'$ with $c = 1_b^\top \widehat{\gamma}(\lambda)$, and suppose that the solution $\widehat{\alpha}(\lambda_1)$ coincides with that of LSIF with $\lambda = \lambda_1$. Then, from Eq. (69), we have

$$\widehat{H}\widehat{\alpha}(\lambda_1) - \widehat{h} + \lambda_1 1_b - \widehat{\mu}(\lambda_1) = 0_b, \tag{72}$$
$$\widehat{\alpha}(\lambda_1) \geq 0_b, \ \ \widehat{\mu}(\lambda_1) \geq 0_b, \ \ \widehat{\alpha}(\lambda_1)^\top \widehat{\mu}(\lambda_1) = 0, \tag{73}$$
$$1_b^\top \widehat{\alpha}(\lambda_1) - 1_b^\top \widehat{\gamma}(\lambda) \leq 0, \ \ \lambda_1 \geq 0, \ \ (1_b^\top \widehat{\alpha}(\lambda_1) - 1_b^\top \widehat{\gamma}(\lambda))\lambda_1 = 0. \tag{74}$$

From Eqs. (70) and (72), we obtain

$$\widehat{H}(\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda)) = -\lambda_1 1_b + \lambda \widehat{\gamma}(\lambda) + \widehat{\mu}(\lambda_1) - \widehat{\eta}. \tag{75}$$

Applying Eqs. (71), (73), (74), and (75), we have

$$\begin{aligned}\inf_{\lambda' \geq 0} \|\widehat{\alpha}(\lambda') - \widehat{\gamma}(\lambda)\|_{\widehat{H}}^2 &\leq (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top \widehat{H} (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda)) \\ &= -\lambda_1 (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top 1_b + \lambda (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top \widehat{\gamma}(\lambda) \\ &\quad + (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top (\widehat{\mu}(\lambda_1) - \widehat{\eta}) \\ &= \lambda (\widehat{\alpha}(\lambda_1)^\top \widehat{\gamma}(\lambda) - \|\widehat{\gamma}(\lambda)\|_2^2) - \widehat{\alpha}(\lambda_1)^\top \widehat{\eta} - \widehat{\gamma}(\lambda)^\top \widehat{\mu}(\lambda_1) \\ &\leq \lambda (\widehat{\alpha}(\lambda_1)^\top \widehat{\gamma}(\lambda) - \|\widehat{\gamma}(\lambda)\|_2^2). \end{aligned} \tag{76}$$

From $\widehat{\alpha}(\lambda_1) \geq 0_b$, $\widehat{\gamma}(\lambda) \geq 0_b$, and $1_b^\top \widehat{\alpha}(\lambda_1) \leq 1_b^\top \widehat{\gamma}(\lambda)$, we have

$$\|\widehat{\alpha}(\lambda_1)\|_1 = 1_b^\top \widehat{\alpha}(\lambda_1) \leq 1_b^\top \widehat{\gamma}(\lambda) \leq \|\widehat{\gamma}(\lambda)\|_1.$$

Then we have the following inequality:

$$\begin{aligned}\widehat{\alpha}(\lambda_1)^\top \widehat{\gamma}(\lambda) &\leq \widehat{\alpha}(\lambda_1)^\top (\|\widehat{\gamma}(\lambda)\|_\infty 1_b) \\ &= \|\widehat{\alpha}(\lambda_1)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty \leq \|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty. \end{aligned} \tag{77}$$

For p and q such that $1/p + 1/q = 1$ and $1 \leq p, q \leq \infty$, Hölder's inequality states that

$$\|\alpha * \beta\|_1 \leq \|\alpha\|_p \cdot \|\beta\|_q,$$

where $\alpha * \beta$ denotes the element-wise product of α and β. Setting $p = 1$, $q = \infty$, and $\alpha = \beta = \widehat{\gamma}(\lambda)$ in Hölder's inequality, we have

$$\|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty - \|\widehat{\gamma}(\lambda)\|_2^2 \geq 0. \tag{78}$$

Combining Eqs. (68), (76), (77), and (78), we obtain

$$\mathrm{diff}(\lambda) \leq \frac{\sqrt{\lambda(\|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty - \|\widehat{\gamma}(\lambda)\|_2^2)} + \|\widehat{\gamma}(\lambda) - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))}.$$

Appendix I. Closed Form of LOOCV Score for uLSIF

Here we derive a closed form expression of the LOOCV score for uLSIF (see Figure 2 for the pseudo code).

Let

$$\varphi(x) = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_b(x))^\top.$$

Then the matrix $\widehat{H}$ and the vector $\widehat{h}$ are expressed as

$$\widehat{H} = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \varphi(x_i^{\mathrm{tr}}) \varphi(x_i^{\mathrm{tr}})^\top,$$

$$\widehat{h} = \frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \varphi(x_j^{\mathrm{te}}),$$

and the coefficients $\widetilde{\beta}(\lambda)$ can be computed by

$$\widetilde{\beta}(\lambda) = \widehat{B}_{\lambda}^{-1}\widehat{h}.$$

Let $\widehat{\beta}^{(i)}$ be the estimator obtained without the i-th training sample x_i^{tr} and the i-th test sample x_i^{te}. Then the estimator has the following closed form:

$$\begin{aligned}
\widehat{\beta}^{(i)}(\lambda) &= \max(0_b, \widetilde{\beta}^{(i)}(\lambda)), \\
\widetilde{\beta}^{(i)}(\lambda) &= \left(\frac{1}{n_{\mathrm{tr}}-1}(n_{\mathrm{tr}}\widehat{H} - \varphi(x_i^{\mathrm{tr}})\varphi(x_i^{\mathrm{tr}})^{\top}) + \lambda I_b\right)^{-1} \frac{1}{n_{\mathrm{te}}-1}(n_{\mathrm{te}}\widehat{h} - \varphi(x_j^{\mathrm{te}})).
\end{aligned}$$

Let $\widehat{B} = \widehat{H} + \frac{\lambda(n_{\mathrm{tr}}-1)}{n_{\mathrm{tr}}} I_b$ and $\widetilde{\beta} = \widehat{B}^{-1}\widehat{h}$ in the following calculation. Using the Sherman-Woodbury-Morrison formula (33), we can simplify the expression of $\widetilde{\beta}^{(i)}(\lambda)$ as follows:

$$\begin{aligned}
\widetilde{\beta}^{(i)}(\lambda) =& \frac{n_{\mathrm{tr}}-1}{n_{\mathrm{tr}}}\left(\widehat{B} - \frac{1}{n_{\mathrm{tr}}}\varphi(x_i^{\mathrm{tr}})\varphi(x_i^{\mathrm{tr}})^{\top}\right)^{-1}\left(\frac{n_{\mathrm{te}}}{n_{\mathrm{te}}-1}\widehat{h} - \frac{1}{n_{\mathrm{te}}-1}\varphi(x_i^{\mathrm{te}})\right) \\
=& \frac{n_{\mathrm{tr}}-1}{n_{\mathrm{tr}}}\left(\widehat{B}^{-1} + \frac{1}{n_{\mathrm{tr}} - \varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})\varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\right) \\
& \times \left(\frac{n_{\mathrm{te}}}{n_{\mathrm{te}}-1}\widehat{h} - \frac{1}{n_{\mathrm{te}}-1}\varphi(x_i^{\mathrm{te}})\right) \\
=& \frac{(n_{\mathrm{tr}}-1)n_{\mathrm{te}}}{n_{\mathrm{tr}}(n_{\mathrm{te}}-1)}\left(\widetilde{\beta} + \frac{\varphi(x_i^{\mathrm{tr}})^{\top}\widetilde{\beta}}{n_{\mathrm{tr}} - \varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})\right) \\
& - \frac{(n_{\mathrm{tr}}-1)}{n_{\mathrm{tr}}(n_{\mathrm{te}}-1)}\left(\widehat{B}^{-1}\varphi(x_i^{\mathrm{te}}) + \frac{\varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{te}})}{n_{\mathrm{tr}} - \varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})\right).
\end{aligned}$$

Thus the matrix inversion required for computing $\widetilde{\beta}^{(i)}(\lambda)$ for all $i = 1, 2, \ldots, n_{\mathrm{tr}}$ is only $\widehat{B}$. Applying this to Eq. (32) and rearrange the formula, we can compute the LOOCV score analytically.

References

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, 1974.

S. Amari, N. Fujita, and S. Shinomoto. Four types of learning curves. *Neural Computation*, 4(4): 605–618, 1992.

P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, 1998.

D. Bertsekas, A. Nedic, and A. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 2003.

M. J. Best. An algorithm for the solution of the parametric quadratic programming problem. CORR Report 82-24, Faculty of Mathematics, University of Waterloo, 1982.

S. Bickel and T. Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2000.

G. C. Cawley and N. L. C. Talbot. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks*, 17(10):1467–75, 2004.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

K. F. Cheng and C. K. Chu. Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, 10(4):583–604, 2004.

A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Springer-Verlag, New York, 1998.

B. Efron, T. Hastie, R. Tibshirani, and I. Johnstone. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

E. A. Fernandez. *The dprep Package*, 2005. URL `http://math.uprm.edu/ edgar/dprep.pdf`.

G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, Berlin, 1996.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 1996.

H. Hachiya, T. Akiyama, M. Sugiyama, and J. Peters. Adaptive importance sampling with automatic model selection in value function approximation. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI2008)*, pages 1351–1356, Chicago, USA, Jul. 13–17 2008. The AAAI Press.

L. K. Hansen and J. Larsen. Linear unlearning for crossvalidation. *Advances in Computational Mathematics*, 5:269–280, 1996.

W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer, Berlin, 2004.

T. Hastie, S. Rosset, R. Tibshirani, and J. ZHu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–162, 1979.

S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-based outlier detection via direct density ratio estimation. In *Proceedings of IEEE International Conference on Data Mining (ICDM2008)*, Pisa, Italy, Dec. 15–19 2008.

V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, Cambridge, MA, 2007.

A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis. An S4 package for kernel methods in R. *Journal of Statistical Planning and Inference*, 11(9):1–20, 2004.

S. Konishi and G. Kitagawa. Generalized information criteria in model selection. *Biometrika*, 83: 875–890, 1996.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region Newton method for large-scale logistic regression. Technical report, Department of Computer Science, National Taiwan University, 2007. URL `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`.

A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetica*, 3, 1969. in Russian.

T. P. Minka. A comparison of numerical optimizers for logistic regression. Technical report, Microsoft Research, 2007. URL `http://research.microsoft.com/~minka/papers/logreg/minka-logreg.pdf`.

J. E. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 847–854. Morgan Kaufmann Publishers, Inc., 1992.

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functions and the likelihood ratio by penalized convex risk minimization. In *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2008. MIT Press.

K. B. Petersen and M. S. Pedersen. The matrix cookbook. Technical report, Technical University of Denmark, 2007. URL `http://www2.imm.dtu.dk/pubdb/p.php?3274`.

J. Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–639, 1998.

J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors. *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2008.

C. E. Rasmussen, R. M. Neal, G. E. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and R. Tibshirani. The DELVE manual, 1996. URL `http://www.cs.toronto.edu/~delve/`.

G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. *Machine Learning*, 42(3): 287–320, 2001.

K. Scheinberg. An efficient implementation of an active set method for SVMs. *Journal of Machine Learning Research*, 7:2237–2257, 2006.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine learning*, pages 823–830, New York, NY, USA, 2007. ACM.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

M. Stone. Cross-validatory choice and assessment of statistical predictors. *Journal of the Royal Statistical Society B*, 32(2):111–147, 1974.

M. Sugiyama and K.-R. Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4):249–279, 2005.

M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, May 2007.

M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1433–1440, Cambridge, MA, 2008a. MIT Press.

M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008b.

R. S. Sutton and G. A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. In *Proceedings of 2008 SIAM International Conference on Data Mining (SDM2008)*, Atlanta, Georgia, USA, 2008.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

G. Wahba. *Spline Model for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.

P. M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7 (1):117–143, 1995.

J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, NY, 2004. ACM Press.

Direct Density-ratio Estimation with Dimensionality Reduction via Least-squares Hetero-distributional Subspace Search

Masashi Sugiyama
Tokyo Institute of Technology
and PRESTO, Japan Science and Technology Agency,
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan.
sugi@cs.titech.ac.jp http://sugiyama-www.cs.titech.ac.jp/~sugi

Makoto Yamada
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan.
yamada@sg.cs.titech.ac.jp

Paul von Bünau
Technical University of Berlin
Franklinstr. 28/29, 10587 Berlin, Germany.
buenau@cs.tu-berlin.de

Taiji Suzuki
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.
s-taiji@stat.t.u-tokyo.ac.jp

Takafumi Kanamori
Nagoya University
Furocho, Chikusaku, Nagoya 464-8603, Japan.
kanamori@is.nagoya-u.ac.jp

Motoaki Kawanabe
Fraunhofer FIRST.IDA
Kekuléstr. 7, 12489 Berlin, Germany.
motoaki.kawanabe@first.fraunhofer.de

Abstract

Methods for directly estimating the ratio of two probability density functions have been actively explored recently since they can be used for various data processing tasks such as *non-stationarity adaptation*, *outlier detection*, and *feature selection*. In this paper, we develop a new method which incorporates dimensionality reduction into a direct density-ratio estimation procedure. Our key idea is to find a low-dimensional subspace in which densities are significantly different and perform density ratio estimation only in this subspace. The proposed method, D^3-LHSS (Direct Density-ratio estimation with Dimensionality reduction via Least-squares Hetero-distributional Subspace Search), is shown to overcome the limitation of baseline methods.

Keywords

density ratio estimation, dimensionality reduction, unconstrained least-squares importance fitting

1 Introduction

Recently, it has been demonstrated that various machine learning and data mining tasks can be formulated in terms of the ratio of two probability density functions (Sugiyama et al., 2009; Sugiyama et al., 2011). Examples of such tasks include *covariate shift adaptation* (Shimodaira, 2000; Zadrozny, 2004; Sugiyama et al., 2007; Sugiyama & Kawanabe, 2010), *transfer learning* (Storkey & Sugiyama, 2007), *multi-task learning* (Bickel et al., 2008), *outlier detection* (Hido et al., 2008; Smola et al., 2009; Hido et al., 2010), *conditional density estimation* (Sugiyama et al., 2010c), *probabilistic classification* (Sugiyama, 2010), *variable selection* (Suzuki et al., 2009a), *independent component analysis* (Suzuki & Sugiyama, 2009), *supervised dimensionality reduction* (Suzuki & Sugiyama, 2010), and *causal inference* (Yamada & Sugiyama, 2010), For this reason, estimating the density ratio has been attracting a great deal of attention, and various approaches have been explored (Silverman, 1978; Ćwik & Mielniczuk, 1989; Gijbels & Mielniczuk, 1995; Sun & Woodroofe, 1997; Jacob & Oliveira, 1997; Qin, 1998; Cheng & Chu, 2004; Huang et al., 2007; Bensaid & Fabre, 2007; Bickel et al., 2007; Sugiyama et al., 2008; Kanamori et al., 2009a; Chen et al., 2009; Sugiyama et al., 2010b; Nguyen et al., 2010).

A naive approach to density ratio estimation is to approximate the two densities in the ratio (i.e., the numerator and the denominator) separately using a flexible technique such as non-parametric *kernel density estimation* (Silverman, 1986; Härdle et al., 2004), and then take the ratio of the estimated densities. However, this naive two-step approach is not reliable in practical situations since kernel density estimation performs poorly in high-dimensional cases; furthermore, division by an estimated density tends to magnify the estimation error. To improve the estimation accuracy, various methods have been developed for directly estimating the density ratio without going through density estimation, e.g., the moment matching method using reproducing kernels (Aronszajn, 1950;

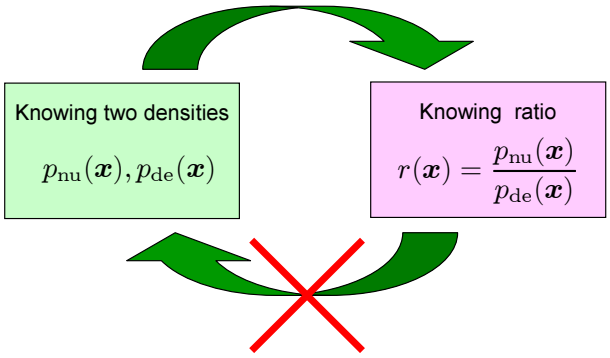


Figure 1: Density ratio estimation is substantially easier than density estimation. The density ratio $r(\boldsymbol{x})$ can be computed if two densities $p_{\mathrm{nu}}(\boldsymbol{x})$ and $p_{\mathrm{de}}(\boldsymbol{x})$ are known. However, even if the density ratio is known, the two densities cannot be computed in general.

Steinwart, 2001) called *kernel mean matching* (KMM) (Huang et al., 2007; Quiñonero-Candela et al., 2009), the method based on *logistic regression* (LR) (Qin, 1998; Cheng & Chu, 2004; Bickel et al., 2007), the distribution matching method under the *Kullback-Leibler (KL) divergence* (Kullback & Leibler, 1951) called the *KL importance estimation procedure* (KLIEP) (Sugiyama et al., 2008; Nguyen et al., 2010), and the density-ratio matching methods under the squared-loss called *least-squares importance fitting* (LSIF) and *unconstrained LSIF* (uLSIF) (Kanamori et al., 2009a). These methods have been shown to compare favorably with naive kernel density estimation through extensive experiments.

The success of these direct density-ratio estimation methods could be intuitively understood through *Vapnik's principle* (Vapnik, 1998): "When solving a problem of interest, one should not solve a more general problem as an intermediate step". The *support vector machine* would be a successful example following this principle—instead of estimating the data generation model, it directly models the decision boundary which is simpler and sufficient for pattern recognition. In the current context, estimating the densities is more general than estimating the density ratio since knowing the two densities implies knowing the ratio, but not vice versa (Figure 1). Thus directly estimating the density ratio would be more promising than density ratio estimation via density estimation.

However, density ratio estimation in high-dimensional cases is still challenging even when the ratio is estimated directly without going through density estimation. Recently, an approach called *Direct Density-ratio estimation with Dimensionality reduction* (D^3) has been proposed (Sugiyama et al., 2010a). The basic idea of D^3 is the following two-step procedure: First a subspace in which the numerator and denominator densities are significantly different (called the *hetero-distributional subspace*) are identified, and then density ratio estimation is performed in this subspace. The rationale behind this approach is that, in practice, the distribution change does not occur in the entire space, but is often confined in a subspace. For example, in non-stationarity adaptation scenarios, the distribution change often occurs only for some attributes and other variables are stable; in

outlier detection scenarios, only a small number of attributes would cause a data sample to be an outlier.

In the D^3 algorithm, the hetero-distributional subspace is identified by searching a subspace in which samples drawn from the two distributions (i.e., the numerator and the denominator of the ratio) are separated from each other—this search is carried out in a computationally efficient manner using a supervised dimensionality reduction method called *local Fisher discriminant analysis* (LFDA) (Sugiyama, 2007). Then, within the identified hetero-distributional subspace, a direct density-ratio estimation method called *unconstrained least-squares importance Fitting* (uLSIF)—which was shown to be computationally efficient (Kanamori et al., 2009a) and numerically stable (Kanamori et al., 2009b)—is employed for obtaining the final density-ratio estimator. Through experiments, this D^3 procedure (which we refer to as D^3-LFDA/uLSIF) was shown to improve the performance in high-dimensional cases.

Although the framework of D^3 is promising, the above D^3-LFDA/uLSIF method possesses two fundamental weaknesses: the restrictive definition of the hetero-distributional subspace and the limiting ability of its search method. More specifically, the component inside the hetero-distributional subspace and its complementary component are assumed to be statistically independent in the original formulation (Sugiyama et al., 2010a). However, this assumption is rather restrictive and may not be fulfilled in practice. Also, in the above D^3 procedure, the hetero-distributional subspace is identified by searching a subspace in which samples drawn from the numerator and denominator distributions are separated from each other. If samples from the two distributions are separable, the two distributions would be significantly different. However, the opposite may not be always true, i.e., non-separability does not necessarily imply that the two distributions are different (consider two similar distributions with the common support). Thus LFDA (and any other supervised dimensionality reduction methods) does not necessarily identify the correct hetero-distributional subspace.

The goal of this paper is to give a new procedure of D^3 that can overcome the above weaknesses. First, we adopt a more general definition of the hetero-distributional subspace. More precisely, we remove the independence assumption between the component inside the hetero-distributional subspace and its complementary component. This allows us to apply the concept of D^3 to a wider class of problems. However, this general definition in turn makes the problem of searching the hetero-distributional subspace more challenging—supervised dimensionality reduction methods for separating samples drawn from the two distributions cannot be used anymore, but we need an alternative method that identifies the largest subspace such that the two *conditional* distributions are equivalent in its complementary subspace.

We prove that the hetero-distributional subspace can be identified by finding a subspace in which two *marginal* distributions are maximally different under the *Pearson divergence*, which is a squared-loss variant of the *Kullback-Leibler divergence* and is an instance of the *f-divergences* (Ali & Silvey, 1966; Csiszár, 1967). Then we propose a new method, which we call *Least-squares Hetero-distributional Subspace Search* (LHSS), for searching a subspace such that the Pearson divergence between two marginal distri-

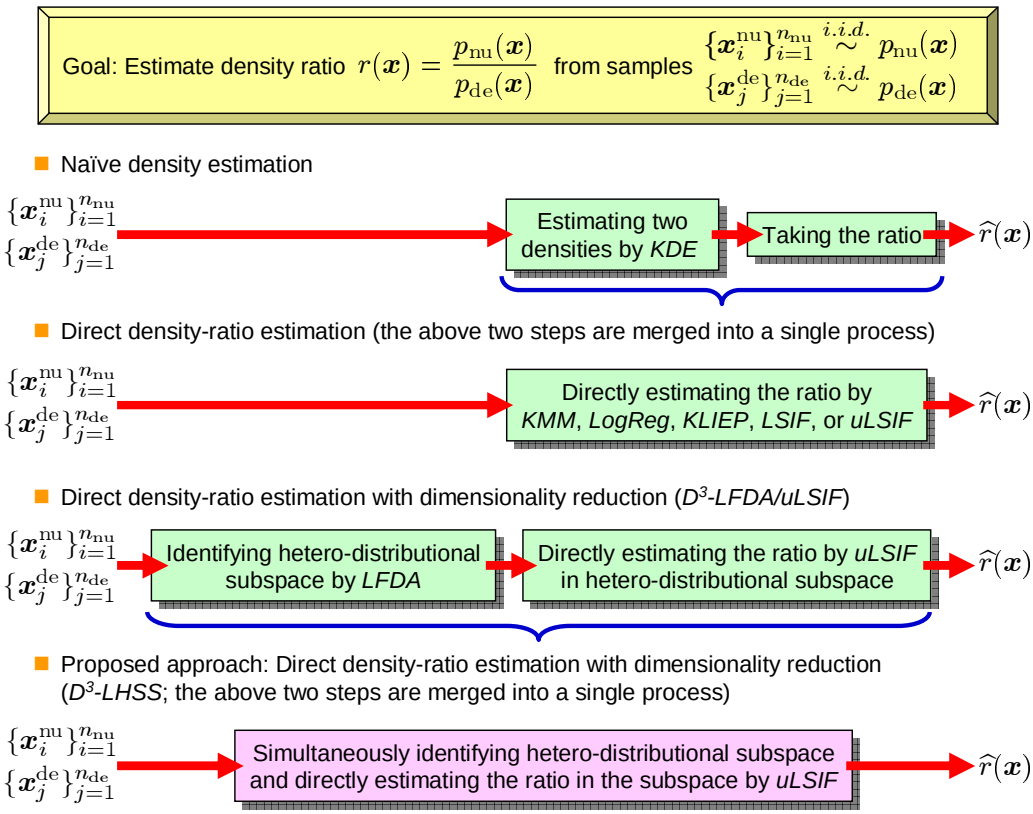


Figure 2: Existing and proposed density-ratio estimation approaches.

butions are maximized. An advantage of the LHSS method is that the subspace search (divergence estimation within a subspace) is carried out also using the density-ratio estimation method uLSIF. Thus the two steps in the D^3 procedure (first identifying the hetero-distributional subspace and then estimating the density ratio within the subspace) are merged into a single step. Thanks to this, the final density-ratio estimator can be automatically obtained without additional computation. We call the combined single-shot density-ratio estimation procedure *D^3 via LHSS* (D^3-LHSS). Through experiments, we show that the weaknesses of the existing approach can be successfully overcome by the D^3-LHSS approach.

Relation among the existing and proposed density-ratio estimation methods is summarized in Figure 2.

2 Formulation of Density-ratio Estimation Problem

In this section, we formulate the problem of density ratio estimation and review a relevant density-ratio estimation method. We briefly summarize possible usage of density ratios in various data processing tasks in Appendix A.

2.1 Problem Formulation

Let $\mathcal{D}$ ($\subset \mathbb{R}^d$) be the data domain and suppose we are given independent and identically distributed (i.i.d.) samples $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ from a distribution with density $p_{\mathrm{nu}}(\boldsymbol{x})$ and i.i.d. samples $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^{n_{\mathrm{de}}}$ from another distribution with density $p_{\mathrm{de}}(\boldsymbol{x})$. We assume that the latter density $p_{\mathrm{de}}(\boldsymbol{x})$ is strictly positive, i.e.,

$$p_{\mathrm{de}}(\boldsymbol{x}) > 0 \text{ for all } \boldsymbol{x} \in \mathcal{D}.$$

The problem we address in this paper is to estimate the density ratio

$$r(\boldsymbol{x}) := \frac{p_{\mathrm{nu}}(\boldsymbol{x})}{p_{\mathrm{de}}(\boldsymbol{x})}$$

from samples $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ and $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^{n_{\mathrm{de}}}$. The subscripts 'nu' and 'de' denote 'numerator' and 'denominator', respectively.

2.2 Directly Estimating Density Ratios by Unconstrained Least-squares Importance Fitting (uLSIF)

As described in Appendix A, density ratios are useful in various data processing tasks. Since the density ratio is usually unknown and needs to be estimated from data, methods of estimating the density ratio have been actively explored recently (Qin, 1998; Cheng & Chu, 2004; Huang et al., 2007; Bickel et al., 2007; Sugiyama et al., 2008; Kanamori et al., 2009a). Here, we briefly review a direct density-ratio estimation method called *unconstrained least-squares importance fitting* (uLSIF) proposed by Kanamori et al. (2009a). For convenience in later sections, we replace the symbol $\boldsymbol{x}$ with $\boldsymbol{u}$, i.e., let us consider the problem of estimating the density ratio

$$r(\boldsymbol{u}) := \frac{p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{u})}$$

from the i.i.d. samples $\{\boldsymbol{u}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ and $\{\boldsymbol{u}_j^{\mathrm{de}}\}_{j=1}^{n_{\mathrm{de}}}$.

2.2.1 Linear Least-squares Estimation of Density Ratios

Let us model the density ratio $r(\boldsymbol{u})$ by the following linear model:

$$\widehat{r}(\boldsymbol{u}) := \sum_{\ell=1}^{b} \alpha_\ell \psi_\ell(\boldsymbol{u}),$$

where

$$\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \ldots, \alpha_b)^\top$$

are parameters to be learned from data samples, b denotes the number of parameters, $^\top$ denotes the transpose of a matrix or a vector, and $\{\psi_\ell(\boldsymbol{u})\}_{\ell=1}^b$ are basis functions such that

$$\psi_\ell(\boldsymbol{u}) \geq 0 \text{ for all } \boldsymbol{u} \text{ and for } \ell = 1, 2, \ldots, b.$$

Note that b and $\{\psi_\ell(\boldsymbol{u})\}_{\ell=1}^b$ could be dependent on the samples $\{\boldsymbol{u}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ and $\{\boldsymbol{u}_j^{\mathrm{de}}\}_{j=1}^{n_{\mathrm{de}}}$, meaning that kernel models are also allowed. We explain how the basis functions $\{\psi_\ell(\boldsymbol{u})\}_{\ell=1}^b$ are designed in Section 2.2.2.

The parameters $\{\alpha_\ell\}_{\ell=1}^b$ in the model $\widehat{r}(\boldsymbol{u})$ are determined so that the following squared error J_0 is minimized:

$$\begin{aligned} J_0(\boldsymbol{\alpha}) &:= \frac{1}{2}\int \left(\widehat{r}(\boldsymbol{u}) - r(\boldsymbol{u})\right)^2 p_{\mathrm{de}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \\ &= \frac{1}{2}\int \widehat{r}(\boldsymbol{u})^2 p_{\mathrm{de}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} - \int \widehat{r}(\boldsymbol{u}) p_{\mathrm{nu}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} + \frac{1}{2}\int r(\boldsymbol{u}) p_{\mathrm{nu}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}, \end{aligned}$$

where the last term is a constant and therefore can be safely ignored. Let us denote the first two terms by J:

$$J(\boldsymbol{\alpha}) := \frac{1}{2}\int \widehat{r}(\boldsymbol{u})^2 p_{\mathrm{de}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} - \int \widehat{r}(\boldsymbol{u}) p_{\mathrm{nu}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}. \tag{1}$$

Note that the same objective function can be obtained via the *Legendre-Fenchel duality* of a divergence (Nguyen et al., 2010).

Approximating the expectations in J by empirical averages, we obtain

$$\begin{aligned} \widehat{J}(\boldsymbol{\alpha}) &:= \frac{1}{2n_{\mathrm{de}}}\sum_{j=1}^{n_{\mathrm{de}}} \widehat{r}(\boldsymbol{u}_j^{\mathrm{de}})^2 - \frac{1}{n_{\mathrm{nu}}}\sum_{i=1}^{n_{\mathrm{nu}}} \widehat{r}(\boldsymbol{u}_i^{\mathrm{nu}}) \\ &= \frac{1}{2}\boldsymbol{\alpha}^\top \widehat{\boldsymbol{H}}\boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top \boldsymbol{\alpha}, \end{aligned}$$

where $\widehat{\boldsymbol{H}}$ is the $b \times b$ matrix with the (ℓ, ℓ')-th element

$$\widehat{H}_{\ell,\ell'} := \frac{1}{n_{\mathrm{de}}}\sum_{j=1}^{n_{\mathrm{de}}} \psi_\ell(\boldsymbol{u}_j^{\mathrm{de}})\psi_{\ell'}(\boldsymbol{u}_j^{\mathrm{de}}), \tag{2}$$

and $\widehat{\boldsymbol{h}}$ is the b-dimensional vector with the ℓ-th element

$$\widehat{h}_\ell := \frac{1}{n_{\mathrm{nu}}}\sum_{i=1}^{n_{\mathrm{nu}}} \psi_\ell(\boldsymbol{u}_i^{\mathrm{nu}}). \tag{3}$$

Now the optimization problem is formulated as follows.

$$\widehat{\boldsymbol{\alpha}} := \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^b} \left[\frac{1}{2}\boldsymbol{\alpha}^\top \widehat{\boldsymbol{H}}\boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top \boldsymbol{\alpha} + \frac{\lambda}{2}\boldsymbol{\alpha}^\top \boldsymbol{\alpha}\right], \tag{4}$$

where a penalty term $\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha}/2$ is included for regularization purposes, and λ (≥ 0) is a regularization parameter that controls the strength of regularization. It is easy to confirm that the solution $\widehat{\boldsymbol{\alpha}}$ can be analytically computed as

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_b)^{-1} \widehat{\boldsymbol{h}}, \tag{5}$$

where $\boldsymbol{I}_b$ is the b-dimensional identity matrix. Thanks to this analytic-form expression, uLSIF is computationally efficient compared with other density-ratio estimators which involve non-linear optimization (Qin, 1998; Cheng & Chu, 2004; Huang et al., 2007; Bickel et al., 2007; Sugiyama et al., 2008; Nguyen et al., 2010).

In the original uLSIF paper (Kanamori et al., 2009a), the above solution is further modified as

$$\widehat{\alpha}_\ell \longleftarrow \max(0, \widehat{\alpha}_\ell).$$

This modification may improve the estimation accuracy in finite sample cases since the true density ratio is non-negative. Even so, we still use Eq.(5) as it is since it is differentiable with respect to $\boldsymbol{U}$, where $\boldsymbol{u} = \boldsymbol{U}\boldsymbol{x}$. This differentiability will play a crucial role in the next section. Note that, even without the above round-up modification, the solution is guaranteed to converge to the optimal vector asymptotically both in parametric and non-parametric cases (Kanamori et al., 2009a; Kanamori et al., 2009b). Thus omitting the above modification step may not have a strong effect.

It was theoretically shown that uLSIF possesses superior theoretical properties in statistical convergence and numerical stability (Kanamori et al., 2009a; Kanamori et al., 2009b).

2.2.2 Basis Function Design

The performance of uLSIF depends on the choice of the basis functions $\{\psi_\ell(\boldsymbol{u})\}_{\ell=1}^b$. As explained below, the use of Gaussian basis functions would be reasonable:

$$\widehat{r}(\boldsymbol{u}) = \sum_{\ell=1}^{n_{\mathrm{nu}}} \alpha_\ell K(\boldsymbol{u}, \boldsymbol{u}_\ell^{\mathrm{nu}}),$$

where $K(\boldsymbol{u}, \boldsymbol{u}')$ is the Gaussian kernel with kernel width σ (> 0):

$$K(\boldsymbol{u}, \boldsymbol{u}') = \exp\left(-\frac{\|\boldsymbol{u} - \boldsymbol{u}'\|^2}{2\sigma^2}\right).$$

By definition, the density ratio $r(\boldsymbol{u})$ tends to take large values if $p_{\mathrm{nu}}(\boldsymbol{u})$ is large and $p_{\mathrm{de}}(\boldsymbol{u})$ is small; conversely, $r(\boldsymbol{u})$ tends to be small (i.e., close to zero) if $p_{\mathrm{nu}}(\boldsymbol{u})$ is small and $p_{\mathrm{de}}(\boldsymbol{u})$ is large. When a non-negative function is approximated by a Gaussian kernel model, many kernels may be needed in the region where the output of the target function is large; on the other hand, only a small number of kernels would be enough in the region where the output of the target function is close to zero (see Figure 3). Following this

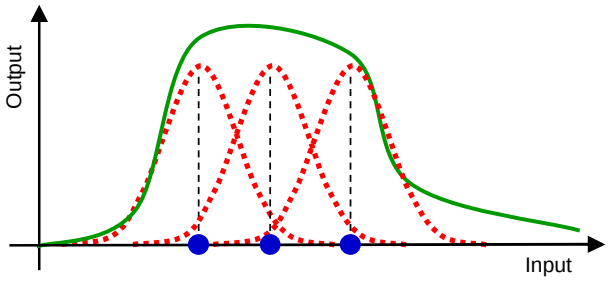


Figure 3: Heuristic of Gaussian kernel allocation.

heuristic, we allocate many kernels in the region where $p_{\mathrm{nu}}(\boldsymbol{u})$ takes large values, which may be approximately achieved by setting the Gaussian centers at $\{\boldsymbol{u}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$.

Alternatively, we may locate $(n_{\mathrm{nu}} + n_{\mathrm{de}})$ Gaussian kernels at both $\{\boldsymbol{u}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ and $\{\boldsymbol{u}_j^{\mathrm{de}}\}_{j=1}^{n_{\mathrm{de}}}$. However, in our preliminary experiments, this did not further improve the performance, but slightly increased the computational cost. When n_{nu} is very large, just using all the test input points $\{\boldsymbol{u}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ as Gaussian centers is already computationally rather demanding. To ease this problem, a subset of $\{\boldsymbol{u}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ may be used as Gaussian centers for computational efficiency, i.e., for a prefixed b $(\in \{1, 2, \ldots, n_{\mathrm{nu}}\})$, we use

$$\widehat{r}(\boldsymbol{u}) = \sum_{\ell=1}^{b} \alpha_\ell K(\boldsymbol{u}, \boldsymbol{c}_\ell),$$

where $\{\boldsymbol{c}_\ell\}_{\ell=1}^{b}$ are template points randomly chosen from $\{\boldsymbol{u}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ without replacement.

The performance of uLSIF depends on the kernel width σ and the regularization parameter λ. Model selection of uLSIF is possible based on cross-validation (CV) with respect to the error criterion (1) (Kanamori et al., 2009a).

3 Direct Density-ratio Estimation with Dimensionality Reduction

Although uLSIF was shown to be a useful density ratio estimation method (Kanamori et al., 2009a), estimating the density ratio in high-dimensional spaces is still challenging. In this section, we propose a new method of direct density-ratio estimation that involves dimensionality reduction.

3.1 Hetero-distributional Subspace

Our basic idea is to first find a low-dimensional subspace in which the two densities are significantly different from each other, and then perform density ratio estimation only in this subspace. Although a similar framework has been explored in Sugiyama et al. (2010a), the current formulation is substantially more general than the previous approach, as explained below.

Let $\boldsymbol{u}$ be an m-dimensional vector ($1 \leq m \leq d$) and $\boldsymbol{v}$ be a $(d-m)$-dimensional vector defined as

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} := \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix} \boldsymbol{x},$$

where $\boldsymbol{U}$ is an $m \times d$ matrix and $\boldsymbol{V}$ is a $(d-m) \times d$ matrix. In order to ensure the uniqueness of the decomposition, we assume (without loss of generality) that the row vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$ form an orthonormal basis, i.e., $\boldsymbol{U}$ and $\boldsymbol{V}$ correspond to "projection" matrices that are orthogonally complementary to each other (see Figure 4). Then the two densities $p_{\mathrm{nu}}(\boldsymbol{x})$ and $p_{\mathrm{de}}(\boldsymbol{x})$ can be decomposed as

$$p_{\mathrm{nu}}(\boldsymbol{x}) = p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u})p_{\mathrm{nu}}(\boldsymbol{u}),$$
$$p_{\mathrm{de}}(\boldsymbol{x}) = p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u})p_{\mathrm{de}}(\boldsymbol{u}).$$

The key theoretical assumption which forms the basis of our proposed algorithm is that the conditional densities $p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u})$ and $p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u})$ agree with each other, i.e., the two densities $p_{\mathrm{nu}}(\boldsymbol{x})$ and $p_{\mathrm{de}}(\boldsymbol{x})$ are decomposed as

$$p_{\mathrm{nu}}(\boldsymbol{x}) = p(\boldsymbol{v}|\boldsymbol{u})p_{\mathrm{nu}}(\boldsymbol{u}),$$
$$p_{\mathrm{de}}(\boldsymbol{x}) = p(\boldsymbol{v}|\boldsymbol{u})p_{\mathrm{de}}(\boldsymbol{u}),$$

where $p(\boldsymbol{v}|\boldsymbol{u})$ is the common conditional density. This assumption implies that the marginal densities of $\boldsymbol{u}$ are different, but the conditional density of $\boldsymbol{v}$ given $\boldsymbol{u}$ is common to $p_{\mathrm{nu}}(\boldsymbol{x})$ and $p_{\mathrm{de}}(\boldsymbol{x})$. Then the density ratio is simplified as

$$r(\boldsymbol{x}) = \frac{p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{u})} =: r(\boldsymbol{u}).$$

Thus, the density ratio does not have to be estimated in the entire d-dimensional space, but it is sufficient to estimate the ratio only in the m-dimensional subspace specified by $\boldsymbol{U}$.

Below, we will use the term, the *hetero-distributional subspace*, for indicating the subspace specified by $\boldsymbol{U}$ in which $p_{\mathrm{nu}}(\boldsymbol{u})$ and $p_{\mathrm{de}}(\boldsymbol{u})$ are different. More precisely, let $\mathcal{S}$ be a subspace specified by $\boldsymbol{U}$ and $\boldsymbol{V}$ such that

$$\mathcal{S} = \{\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{x} \mid p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u}) = p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u}),\ \ \boldsymbol{u} = \boldsymbol{U}\boldsymbol{x},\ \ \boldsymbol{v} = \boldsymbol{V}\boldsymbol{x}\}.$$

Then the hetero-distributional subspace is defined as the *intersection* of all subspaces $\mathcal{S}$. Intuitively, the hetero-distributional subspace is the 'smallest' subspace specified by $\boldsymbol{U}$ such that $p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u})$ and $p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u})$ agree with each other. We refer to the orthogonal complement of the hetero-distributional subspace as the *homo-distributional subspace* (see Figure 4).

This formulation is a generalization of the one proposed in Sugiyama et al. (2010a) in which the components in the hetero-distributional subspace and its complimentary subspace are assumed to be independent of each other. On the other hand, we do not impose

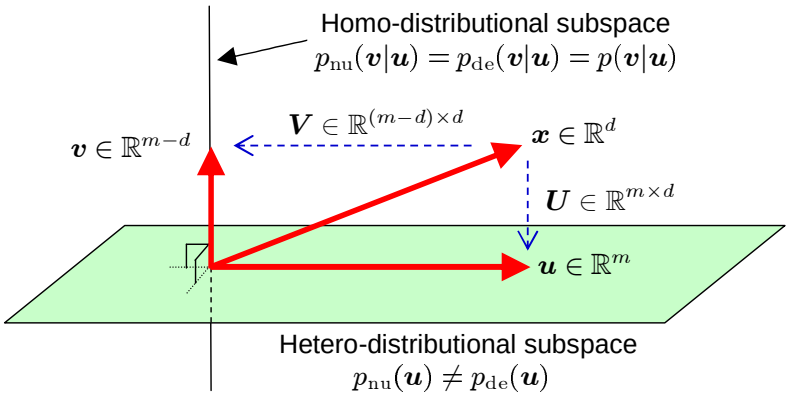


Figure 4: Hetero-distributional subspace.

such an independence assumption in the current paper. As will be demonstrated in Section 4.1, this generalization has a remarkable effect in extending the range of applications of direct density-ratio estimation with dimensionality reduction.

For the moment, we assume that the true dimensionality m of the hetero-distributional subspace is known. Later, we explain how m is estimated from data.

3.2 Estimating Pearson Divergence Using uLSIF

Here, we introduce a criterion for hetero-distributional subspace search and how it is estimated from data.

We use the *Pearson divergence* (PD) as our criterion for evaluating the discrepancy between two distributions. PD is a squared-loss variant of the *Kullback-Leibler divergence* (Kullback & Leibler, 1951), and is an instance of the *f-divergences*, which are also known as the Csiszár f-divergences (Csiszár, 1967) or the Ali-Silvey distances (Ali & Silvey, 1966). PD from $p_{\mathrm{nu}}(\boldsymbol{x})$ to $p_{\mathrm{de}}(\boldsymbol{x})$ is defined and expressed as

$$\begin{aligned}\mathrm{PD}[p_{\mathrm{nu}}(\boldsymbol{x}), p_{\mathrm{de}}(\boldsymbol{x})] &:= \frac{1}{2}\int \left(\frac{p_{\mathrm{nu}}(\boldsymbol{x})}{p_{\mathrm{de}}(\boldsymbol{x})} - 1\right)^2 p_{\mathrm{de}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2}\int \frac{p_{\mathrm{nu}}(\boldsymbol{x})}{p_{\mathrm{de}}(\boldsymbol{x})} p_{\mathrm{nu}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \frac{1}{2}.\end{aligned}$$

$\mathrm{PD}[p_{\mathrm{nu}}(\boldsymbol{x}), p_{\mathrm{de}}(\boldsymbol{x})]$ vanishes if and only if $p_{\mathrm{nu}}(\boldsymbol{x}) = p_{\mathrm{de}}(\boldsymbol{x})$.

The following lemma (called the "*data processing*" inequality) characterizes the hetero-distributional subspace in terms of PD.

Lemma 1 *Let*

$$\begin{aligned}\mathrm{PD}[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})] &= \frac{1}{2}\int \left(\frac{p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{u})} - 1\right)^2 p_{\mathrm{de}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \\ &= \frac{1}{2}\int \frac{p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{u})} p_{\mathrm{nu}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} - \frac{1}{2}. \qquad (6)\end{aligned}$$

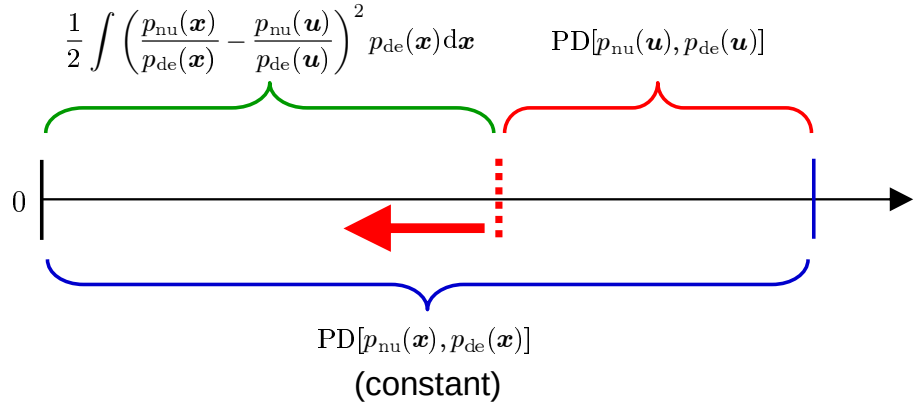


Figure 5: Since $\mathrm{PD}[p_{\rm nu}(\boldsymbol{x}),p_{\rm de}(\boldsymbol{x})]$ is constant, minimizing $\frac{1}{2}\int\left(\frac{p_{\rm nu}(\boldsymbol{x})}{p_{\rm de}(\boldsymbol{x})}-\frac{p_{\rm nu}(\boldsymbol{u})}{p_{\rm de}(\boldsymbol{u})}\right)^2 p_{\rm de}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ is equivalent to maximizing $\mathrm{PD}[p_{\rm nu}(\boldsymbol{u}),p_{\rm de}(\boldsymbol{u})]$.

Then we have

$$\begin{aligned}\mathrm{PD}[p_{\rm nu}(\boldsymbol{x}),p_{\rm de}(\boldsymbol{x})]-\mathrm{PD}[p_{\rm nu}(\boldsymbol{u}),p_{\rm de}(\boldsymbol{u})]&=\frac{1}{2}\int\left(\frac{p_{\rm nu}(\boldsymbol{x})}{p_{\rm de}(\boldsymbol{x})}-\frac{p_{\rm nu}(\boldsymbol{u})}{p_{\rm de}(\boldsymbol{u})}\right)^2 p_{\rm de}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\&\geq 0.\end{aligned}\qquad(7)$$

A proof of the above lemma (for a class of f-divergences) is provided in Appendix B. The right-hand side of Eq.(7) is non-negative, and it vanishes if and only if $p_{\rm nu}(\boldsymbol{v}|\boldsymbol{u})=p_{\rm de}(\boldsymbol{v}|\boldsymbol{u})$. Since $\mathrm{PD}[p_{\rm nu}(\boldsymbol{x}),p_{\rm de}(\boldsymbol{x})]$ is a constant with respect to $\boldsymbol{U}$, maximizing $\mathrm{PD}[p_{\rm nu}(\boldsymbol{u}),p_{\rm de}(\boldsymbol{u})]$ with respect to $\boldsymbol{U}$ leads to $p_{\rm nu}(\boldsymbol{v}|\boldsymbol{u})=p_{\rm de}(\boldsymbol{v}|\boldsymbol{u})$ (Figure 5). That is, the hetero-distributional subspace can be characterized as the maximizer[1] of $\mathrm{PD}[p_{\rm nu}(\boldsymbol{u}),p_{\rm de}(\boldsymbol{u})]$.

Although the hetero-distributional subspace can be characterized as the maximizer of $\mathrm{PD}[p_{\rm nu}(\boldsymbol{u}),p_{\rm de}(\boldsymbol{u})]$, we cannot directly find the maximizer since $p_{\rm nu}(\boldsymbol{u})$ and $p_{\rm de}(\boldsymbol{u})$ are unknown. Here, we utilize a direct density-ratio estimator uLSIF (see Section 2.2) for approximating $\mathrm{PD}[p_{\rm nu}(\boldsymbol{u}),p_{\rm de}(\boldsymbol{u})]$ from samples. Let us replace the density ratio $p_{\rm nu}(\boldsymbol{u})/p_{\rm de}(\boldsymbol{u})$ in Eq.(6) by a density ratio estimator $\widehat{r}(\boldsymbol{u})$. Approximating the expectation over $p_{\rm nu}(\boldsymbol{u})$ by an empirical average over $\{\boldsymbol{u}_i^{\rm nu}\}_{i=1}^{n_{\rm nu}}$, we have the following PD estimator.

$$\widehat{\mathrm{PD}}[p_{\rm nu}(\boldsymbol{u}),p_{\rm de}(\boldsymbol{u})]:=\frac{1}{2n_{\rm nu}}\sum_{i=1}^{n_{\rm nu}}\widehat{r}(\boldsymbol{u}_i^{\rm nu})-\frac{1}{2}.$$

Since uLSIF was shown to be *consistent* (i.e., the solution converges to the optimal value) both in parametric and non-parametric cases (Kanamori et al., 2009a; Kanamori et al., 2009b), $\widehat{\mathrm{PD}}$ would be a consistent estimator of the true PD.

[1] As shown in Appendix B, the data processing inequality holds not only for PD, but also for *any* f-divergences. Thus the characterization of the hetero-distributional subspace is not limited to PD, but is applicable to all f-divergences.

3.3 Least-squares Hetero-distributional Subspace Search (LHSS)

Given the uLSIF-based PD estimator $\widehat{\mathrm{PD}}[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})]$, our next task is to find a maximizer of $\widehat{\mathrm{PD}}[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})]$ with respect to $\boldsymbol{U}$, and identify the hetero-distributional subspace (cf. the data processing inequality given in Lemma 1). We call this procedure *Least-squares Hetero-distributional Subspace Search* (LHSS).

We may employ various optimization techniques to find a maximizer of $\widehat{\mathrm{PD}}[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})]$. Here we describe several possibilities.

3.3.1 Plain Gradient Algorithm

A gradient ascent algorithm would be a fundamental approach to non-linear smooth optimization. We utilize the following lemma.

Lemma 2 *The gradient of* $\widehat{\mathrm{PD}}[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})]$ *with respect to* $\boldsymbol{U}$ *is expressed as*

$$\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}} = \sum_{\ell=1}^{b} \widehat{\alpha}_\ell \frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{U}} - \frac{1}{2} \sum_{\ell,\ell'=1}^{b} \widehat{\alpha}_\ell \widehat{\alpha}_{\ell'} \frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{U}}, \tag{8}$$

where $\widehat{\boldsymbol{\alpha}}$ *is given by Eq.*(5) *and*

$$\frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{U}} = \frac{1}{n_{\mathrm{nu}}} \sum_{i=1}^{n_{\mathrm{nu}}} \frac{\partial \psi_\ell(\boldsymbol{u}_i^{\mathrm{nu}})}{\partial \boldsymbol{U}}, \tag{9}$$

$$\frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{U}} = \frac{1}{n_{\mathrm{de}}} \sum_{j=1}^{n_{\mathrm{de}}} \left(\frac{\partial \psi_\ell(\boldsymbol{u}_j^{\mathrm{de}})}{\partial \boldsymbol{U}} \psi_{\ell'}(\boldsymbol{u}_j^{\mathrm{de}}) + \psi_\ell(\boldsymbol{u}_j^{\mathrm{de}}) \frac{\partial \psi_{\ell'}(\boldsymbol{u}_j^{\mathrm{de}})}{\partial \boldsymbol{U}} \right), \tag{10}$$

$$\frac{\partial \psi_\ell(\boldsymbol{u})}{\partial \boldsymbol{U}} = -\frac{1}{\sigma^2} (\boldsymbol{u} - \boldsymbol{c}_\ell)(\boldsymbol{x} - \boldsymbol{c}'_\ell)^\top \psi_\ell(\boldsymbol{u}). \tag{11}$$

$\boldsymbol{c}'_\ell$ $(\in \mathbb{R}^d)$ *is a pre-image of* $\boldsymbol{c}_\ell$ $(\in \mathbb{R}^m)$*:*

$$\boldsymbol{c}_\ell = \boldsymbol{U} \boldsymbol{c}'_\ell.$$

A proof of the above lemma is provided in Appendix C. Note that $\{\widehat{\alpha}_\ell\}_{\ell=1}^b$ in Eq.(8) depend on $\widehat{\boldsymbol{U}}$ through $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{h}}$ in Eq.(5), which was taken into account when deriving the gradient (see Appendix C). A plain gradient update rule is then given as

$$\boldsymbol{U} \longleftarrow \boldsymbol{U} + t \frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}},$$

where t (> 0) is a learning rate. t may be chosen in practice by some approximate line search method such as *Armijo's rule* (Patriksson, 1999) or *backtracking line search* (Boyd & Vandenberghe, 2004).

A naive gradient update does not necessarily fulfill the orthonormality $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_m$, where $\boldsymbol{I}_m$ is the m-dimensional identity matrix. Thus, after every gradient step, we need to orthonormalize $\boldsymbol{U}$ by, e.g., the *Gram-Schmidt process* (Golub & Loan, 1996) to guarantee its orthonormality. However, this may be rather time-consuming.

3.3.2 Natural Gradient Algorithm

In the Euclidean space, the ordinary gradient $\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}$ gives the steepest direction. On the other hand, in the current setup, the matrix $\boldsymbol{U}$ is restricted to be a member of the *Stiefel manifold* $\mathbb{S}_m^d(\mathbb{R})$:

$$\mathbb{S}_m^d(\mathbb{R}) := \{\boldsymbol{U} \in \mathbb{R}^{m \times d} \mid \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_m\}.$$

On a manifold, it is known that, not the ordinary gradient, but the *natural gradient* (Amari, 1998) gives the steepest direction. The natural gradient $\nabla\widehat{\mathrm{PD}}(\boldsymbol{U})$ at $\boldsymbol{U}$ is the projection of the ordinary gradient $\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}$ onto the tangent space of $\mathbb{S}_m^d(\mathbb{R})$ at $\boldsymbol{U}$.

If the tangent space is equipped with the canonical metric, i.e., for any $\boldsymbol{G}$ and $\boldsymbol{G}'$ in the tangent space,

$$\langle \boldsymbol{G}, \boldsymbol{G}' \rangle = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{G}^\top \boldsymbol{G}'\right), \tag{12}$$

the natural gradient is given by

$$\nabla\widehat{\mathrm{PD}}(\boldsymbol{U}) = \frac{1}{2}\left(\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}} - \boldsymbol{U}\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}^\top \boldsymbol{U}\right).$$

Then the *geodesic* from $\boldsymbol{U}$ to the direction of the natural gradient $\nabla\widehat{\mathrm{PD}}(\boldsymbol{U})$ over $\mathbb{S}_m^d(\mathbb{R})$ can be expressed using $t \in \mathbb{R}$ as

$$\boldsymbol{U}_t := \boldsymbol{U}\exp\left\{t\left(\boldsymbol{U}^\top \frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}} - \frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}^\top \boldsymbol{U}\right)\right\},$$

where 'exp' for a matrix denotes the *matrix exponential*, i.e., for a square matrix $\boldsymbol{T}$,

$$\exp(\boldsymbol{T}) := \sum_{k=0}^{\infty} \frac{1}{k!}\boldsymbol{T}^k. \tag{13}$$

Thus, line search along the geodesic in the natural gradient direction is equivalent to finding a maximizer from

$$\{\boldsymbol{U}_t \mid t \geq 0\}.$$

More details of geometric structure of the Stiefel manifold can be found in Nishimori and Akaho (2005).

A natural gradient update rule is then given as

$$\boldsymbol{U} \longleftarrow \boldsymbol{U}_t,$$

where t (> 0) is the learning rate. Since the orthonormality of $\boldsymbol{U}$ is automatically satisfied in the natural gradient method, it would be computationally more efficient than the plain gradient method. However, optimizing the $m \times d$ matrix $\boldsymbol{U}$ is still computationally expensive.

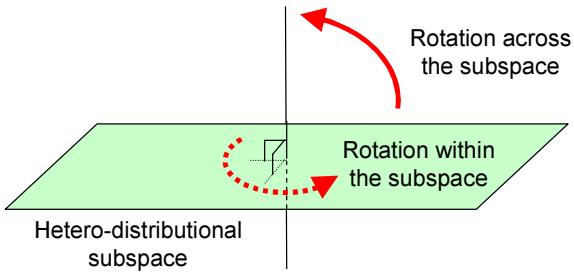


Figure 6: In the hetero-distributional subspace search, rotation which changes the subspace only matters (the solid arrow); rotation within the subspace (dotted arrow) can be ignored since this does not change the subspace. Similarly, rotation within the orthogonal complement of the hetero-distributional subspace can also be ignored (not depicted in the figure).

3.3.3 Givens Rotation

Another simple strategy for optimizing $\boldsymbol{U}$ is to rotate the matrix in the plane spanned by two coordinate axes (which is called the *Givens rotations*; see Golub & Loan, 1996). That is, we randomly choose a two-dimensional subspace spanned by the i-th and j-th variables, and rotate the matrix $\boldsymbol{U}$ within this subspace:

$$\boldsymbol{U} \longleftarrow \boldsymbol{R}_\theta^{(i,j)} \boldsymbol{U},$$

where $\boldsymbol{R}_\theta^{(i,j)}$ is the rotation matrix by angle θ within the subspace spanned by the i-th and j-th variables. $\boldsymbol{R}_\theta^{(i,j)}$ is equal to the identity matrix except that its elements (i,i), (i,j), (j,i), and (j,j) form a two-dimensional rotation matrix:

$$\begin{bmatrix} [\boldsymbol{R}_\theta^{(i,j)}]_{i,i} & [\boldsymbol{R}_\theta^{(i,j)}]_{i,j} \\ [\boldsymbol{R}_\theta^{(i,j)}]_{j,i} & [\boldsymbol{R}_\theta^{(i,j)}]_{j,j} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

The rotation angle θ ($0 \leq \theta \leq \pi$) may be optimized by some secant method (Press et al., 1992).

As shown above, the update rule of the Givens rotations is computationally very efficient. However, since the update direction is not optimized as in the plain/natural gradient methods, the Givens-rotation method could be potentially less efficient as an optimization strategy.

3.3.4 Subspace Rotation

Since we are searching for a subspace, rotation *within* the subspace does not have any influence on the objective value $\widehat{\mathrm{PD}}$ (see Figure 6). This implies that the number of parameters to be optimized in the gradient algorithm can be reduced.

For a *skew-symmetric* matrix $\boldsymbol{M}$ ($\in \mathbb{R}^{d\times d}$), i.e., $\boldsymbol{M}^\top = -\boldsymbol{M}$, rotation of $\boldsymbol{U}$ can be expressed as follows (Plumbley, 2005):

$$\begin{bmatrix}\boldsymbol{I}_m & \boldsymbol{O}_{m,(d-m)}\end{bmatrix} \exp(\boldsymbol{M}) \begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{V}\end{bmatrix},$$

where $\boldsymbol{O}_{d,d'}$ is the $d \times d'$ matrix with all zeros, and $\exp(\boldsymbol{M})$ is the matrix exponential of $\boldsymbol{M}$ (see Eq.(13)). $\boldsymbol{M} = \boldsymbol{O}_{d,d}$ (i.e., $\exp(\boldsymbol{O}_{d,d}) = \boldsymbol{I}_d$) corresponds to no rotation. Here we update $\boldsymbol{U}$ through the matrix $\boldsymbol{M}$.

Let us adopt Eq.(12) as the inner product in the space of skew-symmetric matrices. Then we have the following lemma.

Lemma 3 *The derivative of* $\widehat{\mathrm{PD}}$ *with respect to* $\boldsymbol{M}$ *at* $\boldsymbol{M} = \boldsymbol{O}_{d,d}$ *is given by*

$$\left.\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{M}}\right|_{\boldsymbol{M}=\boldsymbol{O}_{d,d}} = \begin{bmatrix}\boldsymbol{O}_{m,m} & \frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{V}^\top \\ -(\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{V}^\top)^\top & \boldsymbol{O}_{(d-m),(d-m)}\end{bmatrix}. \tag{14}$$

A proof of the above lemma is provided in Appendix D. The block structure of Eq.(14) has an intuitive explanation: the non-zero off-diagonal blocks correspond to the rotation angles *between* the hetero-distributional subspace and its orthogonal complement which do affect the objective function $\widehat{\mathrm{PD}}$. On the other hand, the derivative of rotation *within* the two subspaces vanishes because this does not change the objective value. Thus the variables to be optimized are only the angles corresponding to the non-zero off-diagonal blocks $\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{V}^\top$, which includes only $m(d-m)$ variables. In contrast, the plain/natural gradient algorithms optimize the matrix $\boldsymbol{U}$, which contains md variables. Thus, when m is large, the subspace rotation approach may be computationally more efficient than the plain/natural gradient algorithms.

The gradient ascent update rule of $\boldsymbol{M}$ is given by

$$\boldsymbol{M} \longleftarrow t \left.\frac{\partial \widehat{\mathrm{PD}}}{\partial \boldsymbol{M}}\right|_{\boldsymbol{M}=\boldsymbol{O}_{d,d}},$$

where t is a step-size. Then $\boldsymbol{U}$ is updated as

$$\boldsymbol{U} \longleftarrow \begin{bmatrix}\boldsymbol{I}_m & \boldsymbol{O}_{m,(d-m)}\end{bmatrix} \exp(\boldsymbol{M}) \begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{V}\end{bmatrix}.$$

The *conjugate gradient* method (Golub & Loan, 1996) may be used for the update of $\boldsymbol{M}$.

Following the update of $\boldsymbol{U}$, its counterpart $\boldsymbol{V}$ also needs to be updated accordingly since the hetero-distributional subspace and its complement specified by $\boldsymbol{U}$ and $\boldsymbol{V}$ should be orthogonal to each other (see Figure 4). This can be achieved by setting

$$\boldsymbol{V} \longleftarrow \begin{bmatrix}\boldsymbol{\varphi}_1 & \boldsymbol{\varphi}_2 & \cdots & \boldsymbol{\varphi}_{d-m}\end{bmatrix}^\top,$$

where $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{d-m}$ are orthonormal basis vectors in the orthogonal complement of the hetero-distributional subspace.

3.4 Proposed Algorithm: D^3-LHSS

Finally, we estimate the density ratio in the hetero-distributional subspace detected by the above LHSS method.

A notable fact of the LHSS algorithm is that the density ratio estimator in the hetero-distributional subspace has already been obtained during the hetero-distributional subspace search procedure. Thus, we do not need an additional estimation procedure—our final solution is simply given by

$$\widehat{r}(\boldsymbol{x}) = \sum_{\ell=1}^{b} \widehat{\alpha}_\ell \psi_\ell(\widehat{\boldsymbol{U}}\boldsymbol{x}),$$

where $\widehat{\boldsymbol{U}}$ is a projection matrix obtained by the LHSS algorithm. $\{\widehat{\alpha}_\ell\}_{\ell=1}^b$ are the learned parameters for $\widehat{\boldsymbol{U}}$, which have been obtained and used when computing the gradient (see Lemma 2).

This expression implies that if the dimensionality is not reduced (i.e., $m = d$), the proposed method agrees with the original uLSIF (see Section 2.2). Thus, the proposed method could be regarded as a natural extension of uLSIF to high-dimensional data.

Given the true dimensionality m of the hetero-distributional subspace, we can estimate the hetero-distributional subspace by the LHSS algorithm. When m is unknown, we may choose the best dimensionality based on the CV score of the uLSIF estimator. We refer to our proposed procedure *D^3-LHSS* (D-cube LHSS; Direct Density-ratio estimation with Dimensionality reduction via Least-squares Hetero-distributional Subspace Search).

The complete procedure of D^3-LHSS is summarized in Figure 7. A MATLAB® implementation of D^3-LHSS is available from

'`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/D3LHSS/`'.

4 Experiments

In this section, we investigate the experimental performance of the proposed method. We employ the subspace rotation algorithm explained in Section 3.3.4 in our D^3-LHSS implementation. In uLSIF, the number of parameters is fixed to $b = 100$; the Gaussian width σ and the regularization parameter λ are chosen based on cross-validation.

4.1 Illustrative Examples

First, we illustrate how the D^3-LHSS algorithm behaves.

As explained in Section 1, the previous D^3 method, D^3-LFDA/uLSIF (Sugiyama et al., 2010a), has two potential weaknesses:

- The component $\boldsymbol{u}$ inside the hetero-distributional subspace and its complementary component $\boldsymbol{v}$ are assumed to be statistically independent (cf. Section 3.1).

Input: Two sets of samples $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ and $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^{n_{\mathrm{de}}}$ on $\mathbb{R}^d$
Output: Density ratio estimator $\widehat{r}(\boldsymbol{x})$

For each reduced dimension $m = 1, 2, \ldots, d$
  Initialize embedding matrix $\boldsymbol{U}_m$ ($\in \mathbb{R}^{m \times d}$);
  Repeat until $\boldsymbol{U}_m$ converges
    Choose Gaussian width σ and regularization parameter λ by CV;
    Update $\boldsymbol{U}$ by some optimization method (see Section 3.3);
  end
  Obtain embedding matrix $\widehat{\boldsymbol{U}}_m$ and corresponding density-ratio estimator $\widehat{r}_m(\boldsymbol{x})$;
  Compute its CV value as a function of m;
end
Choose the best reduced dimensionality $\widehat{m}$ that minimizes the CV score;
Set $\widehat{r}(\boldsymbol{x}) = \widehat{r}_{\widehat{m}}(\boldsymbol{x})$;

Figure 7: Pseudo code of D^3-LHSS.

- Separability of samples drawn from two distributions implies that the two distributions are different, but non-separability does not necessarily imply that the two distributions are equivalent. Thus, D^3-LFDA/uLSIF may not be able to detect the subspace in which the two distributions are different, but samples are not really separable.

Here, through numerical examples, we illustrate these weaknesses of D^3-LFDA/uLSIF, and show these problems can be overcome by D^3-LHSS. Let us consider two-dimensional examples (i.e., $d = 2$), and suppose that the two densities $p_{\mathrm{nu}}(\boldsymbol{x})$ and $p_{\mathrm{de}}(\boldsymbol{x})$ are different only in the one-dimensional subspace (i.e., $m = 1$) spanned by $(1, 0)^\top$:

$$\begin{aligned} \boldsymbol{x} &= (x^{(1)}, x^{(2)})^\top = (u, v)^\top, \\ p_{\mathrm{nu}}(\boldsymbol{x}) &= p(v|u)p_{\mathrm{nu}}(u), \\ p_{\mathrm{de}}(\boldsymbol{x}) &= p(v|u)p_{\mathrm{de}}(u). \end{aligned}$$

Let $n_{\mathrm{nu}} = n_{\mathrm{de}} = 1000$. We use the following three datasets:

"Rather-separate" dataset (Figure 8):

$$\begin{aligned} p(v|u) &= p(v) = N(v; 0, 1^2), \\ p_{\mathrm{nu}}(u) &= N(u; 0, 0.5^2), \\ p_{\mathrm{de}}(u) &= 0.5N(u; -1, 1^2) + 0.5N(u; 1, 1^2), \end{aligned}$$

where $N(u; \mu, \sigma^2)$ denotes the Gaussian density with mean μ and variance σ^2 with respect to u. This is an easy and simple dataset for the purpose of illustrating the usefulness of dimensionality reduction in density ratio estimation.

"Highly-overlapped" dataset (Figure 9):

$$p(v|u) = p(v) = N(v; 0, 1^2),$$
$$p_{\mathrm{nu}}(u) = N(u; 0, 0.6^2),$$
$$p_{\mathrm{de}}(u) = N(u; 0, 1.2^2).$$

Since v is independent of u, D^3-LFDA/uLSIF is still applicable in principle. However, u^{nu} and u^{de} are highly overlapped and are not clearly separable. Thus this dataset would be hard for D^3-LFDA/uLSIF.

"Dependent" dataset (Figure 10):

$$p(v|u) = N(v; u, 1^2),$$
$$p_{\mathrm{nu}}(u) = N(u; 0, 0.5^2),$$
$$p_{\mathrm{de}}(u) = 0.5N(u; -1, 1^2) + 0.5N(u; 1, 1^2).$$

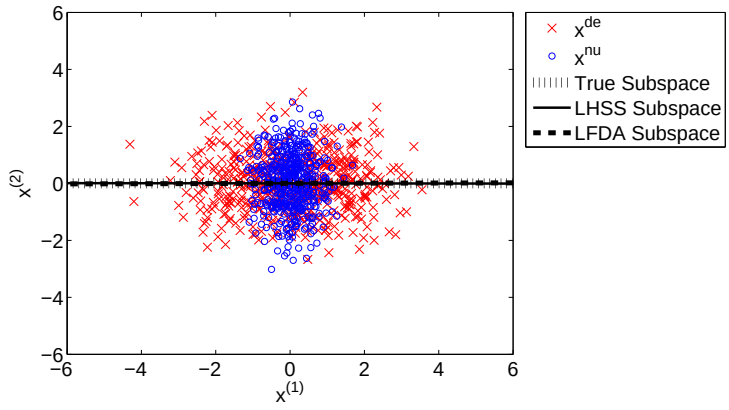
In this dataset, the *conditional* distribution $p(v|u)$ is common, but the *marginal* distributions $p_{\mathrm{nu}}(v)$ and $p_{\mathrm{de}}(v)$ are different. Since v is not independent of u, this dataset would be out of scope for D^3-LFDA/uLSIF.

The true hetero-distributional subspace for the "rather-separate" dataset is depicted by the dotted line in Figure 8(a); the solid line and the dashed line depict the hetero-distributional subspace found by LHSS and LFDA with reduced dimensionality $m = 1$, respectively. This graph shows that LHSS and LFDA both give very good estimates of the true hetero-distributional subspace. In Figure 8(c), Figure 8(d), and Figure 8(e), density ratio functions estimated by the plain uLSIF without dimensionality reduction, D^3-LFDA/uLSIF, and D^3-LHSS for the "rather-separate" dataset are depicted. These graphs show that both D^3-LHSS and D^3-LFDA/uLSIF give much better estimates of the density ratio function (see Figure 8(b) for the profile of the true density ratio function) than the plain uLSIF without dimensionality reduction. Thus, the usefulness of dimensionality reduction in density ratio estimation was illustrated.

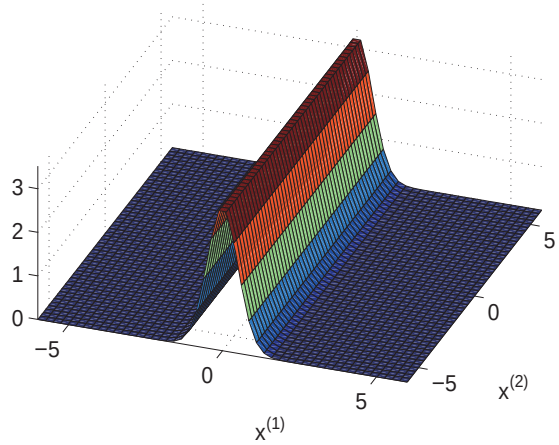
For the "highly-overlapped" dataset (Figure 9), LHSS gives a reasonable estimate of the hetero-distributional subspace, while LFDA is highly erroneous due to less separability. As a result, the density ratio function obtained by D^3-LFDA/uLSIF does not reflect the true redundant structure appropriately. On the other hand, D^3-LHSS still works well.

Finally, for the "dependent" dataset (Figure 10), LHSS gives an accurate estimate of the hetero-distributional subspace. However, LFDA gives a highly biased solution since the marginal distributions $p_{\mathrm{nu}}(v)$ and $p_{\mathrm{de}}(v)$ are no longer common in the "dependent" dataset. Consequently, the density ratio function obtained by D^3-LFDA/uLSIF is highly erroneous. In contrast, D^3-LHSS still works very well for the "dependent" dataset.

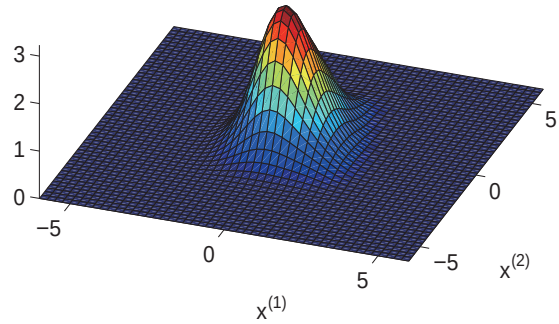
The experimental results for the "highly-overlapped" and "dependent" datasets illustrated typical failure modes of LFDA, and LHSS was shown to be able to successfully overcome these weaknesses of LFDA.



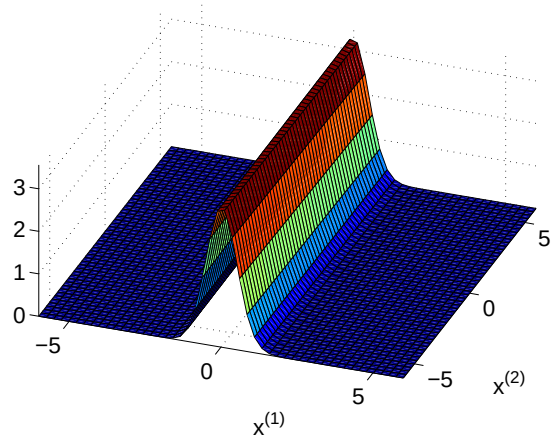
(a) Hetero-distributional subspace



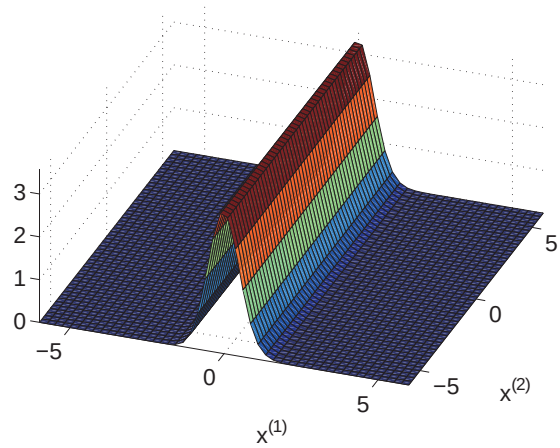
(b) $r(\boldsymbol{x})$



(c) $\widehat{r}(\boldsymbol{x})$ by plain uLSIF.

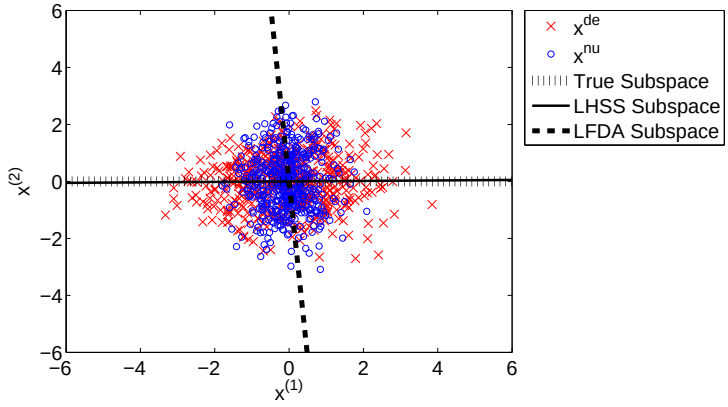


(d) $\widehat{r}(\boldsymbol{x})$ by D^3-LFDA/uLSIF.

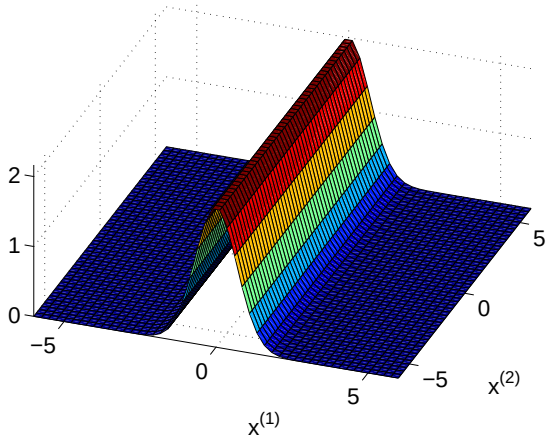


(e) $\widehat{r}(\boldsymbol{x})$ by D^3-LHSS.

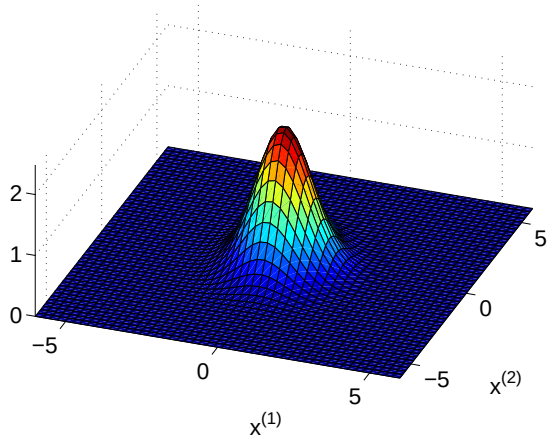
Figure 8: "Rather-separate" dataset.



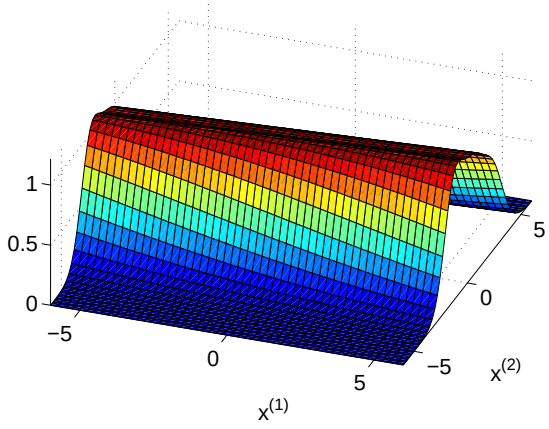
(a) Hetero-distributional subspace



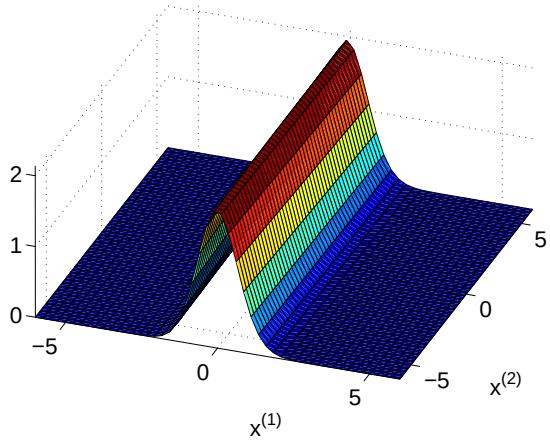
(b) $r(\boldsymbol{x})$



(c) $\widehat{r}(\boldsymbol{x})$ by plain uLSIF.

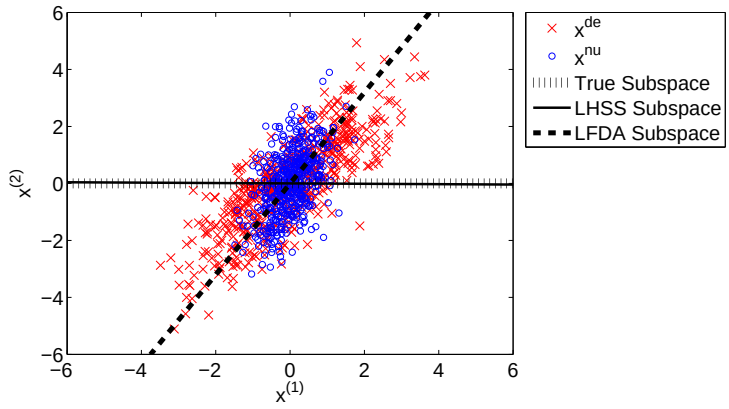


(d) $\widehat{r}(\boldsymbol{x})$ by D^3-LFDA/uLSIF.

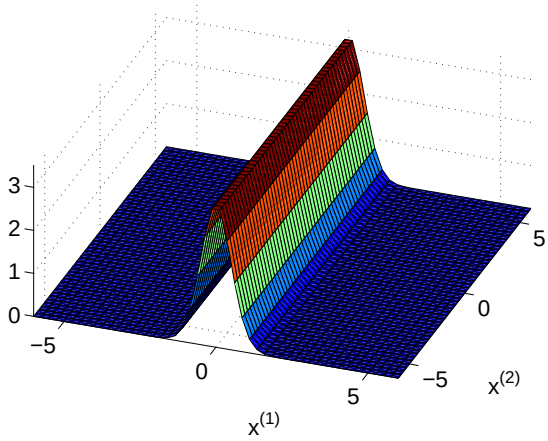


(e) $\widehat{r}(\boldsymbol{x})$ by D^3-LHSS.

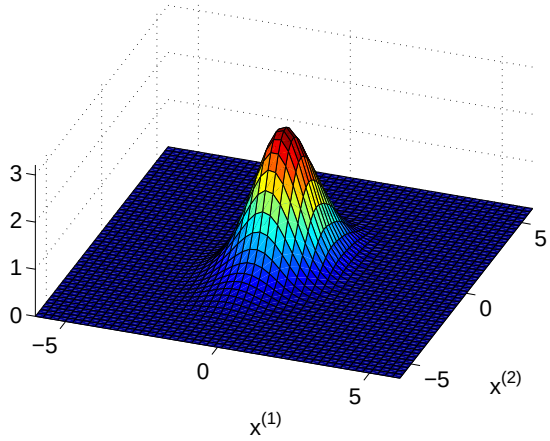
Figure 9: "Highly-overlapped" dataset.



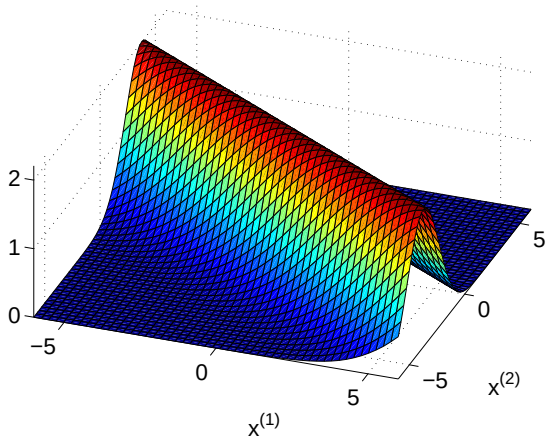
(a) Hetero-distributional subspace



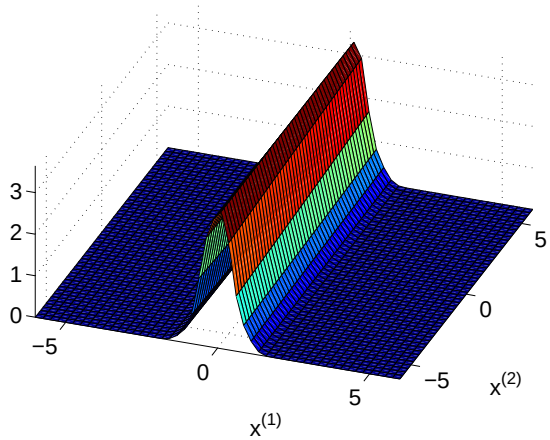
(b) $r(\boldsymbol{x})$



(c) $\widehat{r}(\boldsymbol{x})$ by plain uLSIF.



(d) $\widehat{r}(\boldsymbol{x})$ by D^3-LFDA/uLSIF.



(e) $\widehat{r}(\boldsymbol{x})$ by D^3-LHSS.

Figure 10: "Dependent" dataset.

4.2 Evaluation on Artificial Data

Next, we systematically compare the performance of the proposed D^3-LHSS with that of the plain uLSIF and D^3-LFDA/uLSIF for high-dimensional artificial data.

For the three datasets used in the previous experiments, we increase the entire dimensionality as $d = 2, 3, \ldots, 10$ by adding dimensions consisting of standard normal noise. The dimensionality of the hetero-distributional subspace is estimated based on the CV score of uLSIF. We evaluate the error of a density ratio estimator $\widehat{r}(\boldsymbol{x})$ by

$$\text{Error} := \frac{1}{2}\int \left(\widehat{r}(\boldsymbol{x}) - r(\boldsymbol{x})\right)^2 p_{\text{de}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{15}$$

which uLSIF tries to minimize (see Section 2.2).

The left graphs in Figure 11 show the density-ratio estimation error averaged over 100 runs as functions of the entire input dimensionality d. The best method in terms of the mean error and comparable methods according to the *t-test* (Henkel, 1979) at the significance level 1% are specified by '$\circ$'; otherwise methods are specified by '$\times$'.

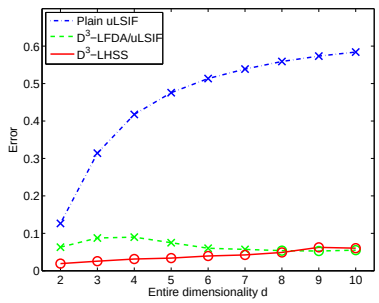
These plots show that, while the error of the plain uLSIF increases rapidly as the entire dimensionality d increases, that of the proposed D^3-LHSS is kept moderate. Consequently, the proposed method consistently outperforms the plain uLSIF. D^3-LHSS is comparable to D^3-LFDA/uLSIF for the "rather-separate" dataset, and D^3-LHSS significantly outperforms D^3-LFDA/uLSIF for the "highly-overlapped" and "dependent" datasets. Thus, D^3-LHSS was overall shown to compare favorably with the other approaches.

The choice of the dimensionality of the hetero-distributional subspace in D^3-LHSS and D^3-LFDA/uLSIF is illustrated in the middle and right columns of Figure 11; the darker the color is, the more frequently the corresponding dimensionality is chosen. The plots show that D^3-LHSS reasonably identifies the true dimensionality ($m = 1$ in the current setup) for all the three datasets, while D^3-LFDA/uLSIF performs well only for the "rather-separate" dataset. This happened because D^3-LFDA/uLSIF cannot find appropriate low-dimensional subspaces for the "highly-overlapped" and "dependent" datasets, and therefore the CV scores misled the choice of subspace dimensionality.

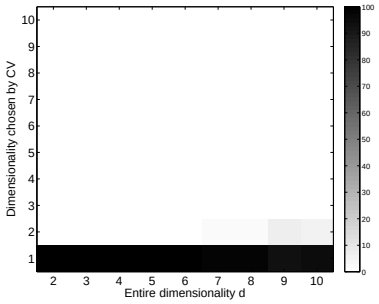
4.3 Inlier-based Outlier Detection for Benchmark Data

Finally, we apply the proposed method to inlier-based outlier detection, i.e., finding outliers in an evaluation dataset based on another "model" dataset that only contains inliers (see Section A.2 for details).

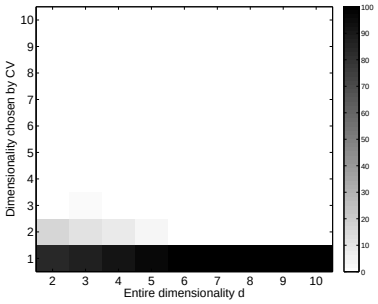
We use the *USPS hand-written digit dataset* taken from the *UCI Machine Learning Repository* (Asuncion & Newman, 2007). We regard samples in the class '1' as inliers and samples in other classes as outliers. We randomly take 500 samples from the class '1', and assign them to the model dataset. Then we randomly take 500 samples from the class '1' without overlap, and 25 samples from one of the other classes. From these samples, density ratio estimation is performed and the outlier score is computed. Since the USPS hand-written digit dataset contains 10 classes (i.e., from '0' to '9'), we have 9



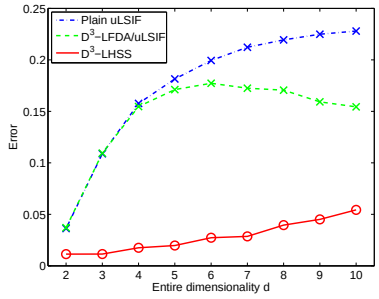
(a) Density-ratio estimation error



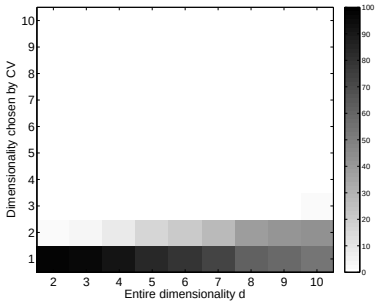
(b) Choice of Dimensionality by D^3-LHSS



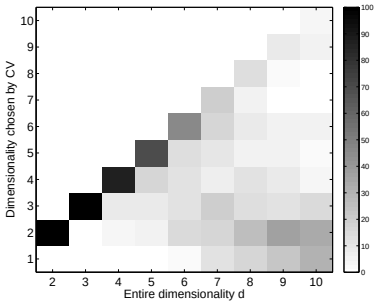
(c) Choice of Dimensionality by D^3-LFDA/uLSIF



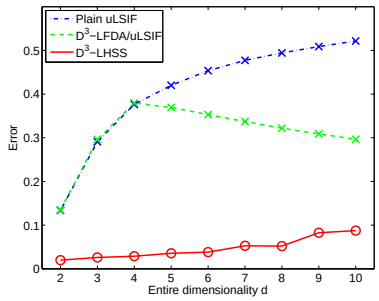
(d) Density-ratio estimation error



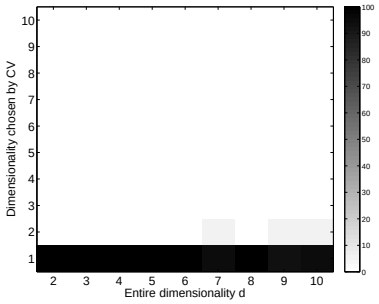
(e) Choice of Dimensionality by D^3-LHSS



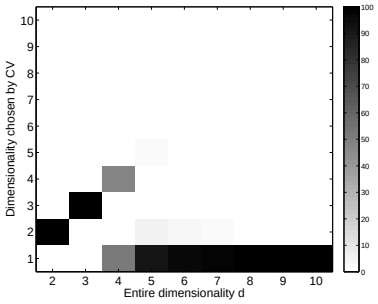
(f) Choice of Dimensionality by D^3-LFDA/uLSIF



(g) Density-ratio estimation error



(h) Choice of Dimensionality by D^3-LHSS



(i) Choice of Dimensionality by D^3-LFDA/uLSIF

Figure 11: Top: "Rather-separate" dataset. Middle: "Highly-overlapped" dataset. Bottom: "Dependent" dataset. Left: Density-ratio estimation error (15) averaged over 100 runs as a function of the entire data dimensionality d. The best method in terms of the mean error and comparable methods according to the *t-test* at the significance level 1% are specified by '$\circ$'; otherwise methods are specified by '$\times$'. Center: The dimensionality of the hetero-distributional subspace chosen by CV in LHSS. Right: The dimensionality of the hetero-distributional subspace chosen by CV in LFDA.

different tasks in total. The dimensionality of the samples is $d = 256$. For the D^3-LHSS and D^3-LFDA/uLSIF methods, we choose the dimensionality of the hetero-distributional subspace from $m = 1, 2, \ldots, 5$ by cross-validation.

When evaluating the performance of outlier detection methods, it is important to take into account both the *detection rate* (i.e., the amount of true outliers an outlier detection algorithm can find) and the *detection accuracy* (i.e., the amount of true inliers an outlier detection algorithm misjudges as outliers). Since there is a trade-off between the detection rate and the detection accuracy, we adopt the *area under the ROC curve* (AUC) as our error metric (Bradley, 1997).

The mean and standard deviation of AUC scores over 100 runs with different random seeds are summarized in Table 1, where the best method in terms of the mean AUC and comparable methods according to the *t-test* at the significance level 1% are specified by '$\circ$'. The table shows that the proposed D^3-LHSS tends to outperform the plain uLSIF and D^3-LFDA/uLSIF. It is also note worthy that D^3-LFDA/uLSIF is actually outperformed by the plain uLSIF—the baseline method. This is perhaps because the numerator and denominator datasets are highly overlapped in outlier detection scenarios, so D^3-LFDA/uLSIF performs rather poorly (cf. Figure 9)

We also evaluate the performance of each method for an additional test dataset which is not used for density ratio estimation. The test dataset consists of 100 randomly chosen samples from the class '1' and 5 randomly chosen samples from the outlier class (which is the same as the evaluation dataset). The results are summarized in Table 2, showing that the advantage of the proposed method is still valid in this more challenging scenario.

5 Conclusions

Density ratios are becoming quantities of interest in the machine learning and data mining communities since it can be used for solving various important data processing tasks such as non-stationarity adaptation, outlier detection, and feature selection (Sugiyama et al., 2009; Sugiyama et al., 2011). In this paper, we tackled a challenging problem of estimating density ratios in high-dimensional spaces, and gave a new procedure in the framework of *Direct Density-ratio estimation with Dimensionality reduction* (D^3; D-cube). The basic idea of D^3 is to identify a subspace called the *hetero-distributional subspace*, in which two distributions (corresponding to the numerator and denominator of the density ratio) are different.

In the existing approach of D^3 (Sugiyama et al., 2010a), the hetero-distributional subspace is identified by finding a subspace in which samples drawn from the two distributions are maximally separated from each other. To this end, supervised dimensionality reduction methods such as *local Fisher discriminant analysis* (LFDA) (Sugiyama, 2007) are utilized. This approach was shown to work well when the components inside and outside the hetero-distributional subspace are statistically independent, and samples drawn from the two distributions are highly separable from each other in the hetero-distributional subspace.

Table 1: Outlier detection for the USPS hand-written digit dataset ($d = 256$). The means (and standard deviations in the bracket) of AUC scores over 100 runs for the evaluation dataset are summarized. The best method in terms of the mean AUC value and comparable methods according to the t-test at the significance level 1% are specified by '°'. The means (and standard deviations in the bracket) of the chosen dimensionality by cross-validation are also included in the table.

	D^3-LHSS		D^3-LFDA/uLSIF		Plain uLSIF
Data	AUC	$\widehat{m}$	AUC	$\widehat{m}$	AUC
Digit 2	°0.956 (0.035)	4.3 (0.8)	0.889 (0.104)	1.7 (1.1)	0.902 (0.038)
Digit 3	°0.967 (0.032)	4.4 (0.8)	0.868 (0.136)	1.8 (1.1)	0.921 (0.039)
Digit 4	°0.907 (0.061)	4.4 (0.9)	0.825 (0.104)	1.4 (0.6)	0.870 (0.036)
Digit 5	°0.965 (0.037)	4.3 (0.9)	0.882 (0.109)	1.6 (0.9)	0.906 (0.037)
Digit 6	°0.974 (0.022)	4.4 (0.8)	0.891 (0.090)	1.7 (1.1)	0.941 (0.029)
Digit 7	°0.924 (0.072)	4.4 (0.9)	0.642 (0.139)	2.3 (1.4)	0.878 (0.035)
Digit 8	°0.929 (0.051)	4.2 (1.0)	0.804 (0.147)	1.8 (1.1)	0.860 (0.033)
Digit 9	°0.942 (0.048)	4.6 (0.7)	0.790 (0.136)	1.8 (1.1)	0.892 (0.035)
Digit 0	°0.986 (0.019)	4.2 (0.9)	0.920 (0.071)	1.9 (0.8)	°0.979 (0.019)
Average	0.950 (0.051)	4.4 (0.9)	0.835 (0.142)	1.8 (1.1)	0.905 (0.049)

Table 2: Outlier detection for the USPS hand-written digit dataset ($d = 256$). The means (and standard deviations in the bracket) of AUC scores over 100 runs for unlearned test dataset are summarized.

	D^3-LHSS		D^3-LFDA/uLSIF		Plain uLSIF
Data	AUC	$\widehat{m}$	AUC	$\widehat{m}$	AUC
Digit 2	°0.946 (0.047)	4.3 (0.8)	0.817 (0.132)	1.7 (1.1)	0.905 (0.044)
Digit 3	°0.953 (0.061)	4.4 (0.8)	0.780 (0.161)	1.8 (1.1)	0.924 (0.045)
Digit 4	°0.880 (0.094)	4.4 (0.9)	0.767 (0.121)	1.4 (0.6)	°0.870 (0.063)
Digit 5	°0.954 (0.057)	4.3 (0,9)	0.813 (0.142)	1.6 (0.9)	0.906 (0.047)
Digit 6	°0.959 (0.052)	4.4 (0.8)	0.806 (0.141)	1.7 (1.1)	0.939 (0.040)
Digit 7	°0.909 (0.079)	4.4 (0.9)	0.689 (0.173)	2.3 (1.4)	0.877 (0.056)
Digit 8	°0.903 (0.078)	4.2 (1.0)	0.741 (0.173)	1.8 (1.1)	0.861 (0.049)
Digit 9	°0.932 (0.072)	4.6 (0.7)	0.793 (0.128)	1.8 (1.1)	0.894 (0.054)
Digit 0	°0.982 (0.039)	4.2 (0.9)	0.859 (0.098)	1.9 (0.8)	°0.982 (0.022)
Average	0.935 (0.073)	4.4 (0.9)	0.785 (0.150)	1.8 (1.1)	0.906 (0.060)

However, as illustrated in Section 4.1, violation of these conditions can cause significant performance degradation. This problem can be overcome in principle by finding a subspace such that two *conditional* distributions are similar to each other in its complementary subspace. However, comparing conditional distributions is a cumbersome task. To cope with this problem, we first proved that the hetero-distributional subspace can be characterized as the subspace in which two *marginal* distributions are maximally different under the *Pearson divergence* (Lemma 1). Based on this lemma, we proposed a new algorithm for finding the hetero-distributional subspace called *Least-squares Hetero-distributional Subspace Search* (LHSS). Since a density-ratio estimation method is utilized during hetero-distributional subspace search in the LHSS procedure, an additional density-ratio estimation step is not needed after hetero-distributional subspace search. Thus, two steps in the previous method (hetero-distributional subspace search followed by density ratio estimation in the identified subspace) were merged into a single step (see Figure 2). The proposed single-shot procedure, *D*^{3}*-LHSS* (D-cube LHSS), was shown to be able to overcome the limitations of the D^3-LFDA/uLSIF approach through experiments.

In the experiments in Section 4, we employed the subspace rotation algorithm explained in Section 3.3.4 in our D^3-LHSS implementation. Although we experimentally found that the subspace rotation algorithm is useful, this does not necessarily mean that subspace rotation is always the best performing algorithm. Other approaches explained in Section 3.3 may also be useful in some situations. Further investigating the optimization issue is an important future work.

We gave a general proof of the data processing inequality (Lemma 1) for a class of f-divergences (Ali & Silvey, 1966; Csiszár, 1967). Thus, the hetero-distributional subspace is characterized not only by the Pearson divergence, but also by *any* f-divergences. Since a framework of density ratio estimation for f-divergences has been provided in Nguyen et al. (2010), an interesting future direction is to develop hetero-distributional subspace search methods for general f-divergences.

Acknowledgments

MS was supported by SCAT, AOARD, and the JST PRESTO program. MY was supported by the JST PRESTO program. We thank Satoshi Hara for having performed preliminary experiments using an earlier version of the proposed method. Our special thanks also go to anonymous reviewers for their comments.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *AC-19*, 716–723.

Akiyama, T., Hachiya, H., & Sugiyama, M. (2010). Efficient exploration through active

learning for value function approximation in reinforcement learning. *Neural Networks*, *23*, 639–648.

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, *28*, 131–142.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10*, 251–276.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*, 337–404.

Asuncion, A., & Newman, D. (2007). UCI machine learning repository.

Bensaid, N., & Fabre, J. P. (2007). Optimal asymptotic quadratic error of kernel estimators of Radon-Nikodym derivatives for strong mixing data. *Journal of Nonparametric Statistics*, *19*, 77–88.

Bickel, S., Bogojeska, J., Lengauer, T., & Scheffer, T. (2008). Multi-task learning for HIV therapy screening. *Proceedings of 25th Annual International Conference on Machine Learning (ICML2008)* (pp. 56–63).

Bickel, S., Brückner, M., & Scheffer, T. (2007). Discriminative learning for differing training and test distributions. *Proceedings of the 24th International Conference on Machine Learning (ICML2007)* (pp. 81–88).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge, UK: Cambridge University Press.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*, 1145–1159.

Chen, S.-M., Hsu, Y.-S., & Liaw, J.-T. (2009). On kernel estimators of density ratio. *Statistics*, *43*, 463–479.

Cheng, K. F., & Chu, C. K. (2004). Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, *10*, 583–604.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. Hoboken, NJ, USA: John Wiley & Sons, Inc. 2nd edition.

Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, *2*, 229–318.

Ćwik, J., & Mielniczuk, J. (1989). Estimating density ratio with application to discriminant analysis. *Communications in Statistics: Theory and Methods*, *18*, 3057–3069.

Fishman, G. S. (1996). *Monte Carlo: Concepts, algorithms, and applications*. Berlin, Germany: Springer-Verlag.

Gijbels, I., & Mielniczuk, J. (1995). Asymptotic properties of kernel estimators of the Radon-Nikodym derivative with applications to discriminant analysis. *Statistica Sinica*, *5*, 261–278.

Golub, G. H., & Loan, C. F. V. (1996). *Matrix computations*. Baltimore, MD, USA: Johns Hopkins University Press.

Hachiya, H., Akiyama, T., Sugiyama, M., & Peters, J. (2009a). Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Networks*, *22*, 1399–1410.

Hachiya, H., Peters, J., & Sugiyama, M. (2009b). Efficient sample reuse in EM-based policy search. *Machine Learning and Knowledge Discovery in Databases* (pp. 469–484). Berlin: Springer.

Härdle, W., Müller, M., Sperlich, S., & Werwatz, A. (2004). *Nonparametric and semiparametric models*. Berlin, Germany: Springer.

Henkel, R. E. (1979). *Tests of significance*. Beverly Hills, CA, USA.: SAGE Publication.

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., & Kanamori, T. (2008). Inlier-based outlier detection via direct density ratio estimation. *Proceedings of IEEE International Conference on Data Mining (ICDM2008)* (pp. 223–232). Pisa, Italy.

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., & Kanamori, T. (2010). Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*. to appear.

Huang, J., Smola, A., Gretton, A., Borgwardt, K. M., & Schölkopf, B. (2007). Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt and T. Hoffman (Eds.), *Advances in neural information processing systems 19*, 601–608. Cambridge, MA, USA: MIT Press.

Hulle, M. M. V. (2005). Edgeworth approximation of multivariate differential entropy. *Neural Computation*, *17*, 1903–1910.

Jacob, P., & Oliveira, P. E. (1997). Kernel estimators of general Radon-Nikodym derivatives. *Statistics*, *30*, 25–46.

Kanamori, T., Hido, S., & Sugiyama, M. (2009a). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, *10*, 1391–1445.

Kanamori, T., & Shimodaira, H. (2003). Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference*, *116*, 149–162.

Kanamori, T., Suzuki, T., & Sugiyama, M. (2009b). *Condition number analysis of kernel-based density ratio estimation* (Technical Report). arXiv.

Kawahara, Y., & Sugiyama, M. (2009). Change-point detection in time-series data by direct density-ratio estimation. *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)* (pp. 389–400). Sparks, Nevada, USA.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, *69*, 066138.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*, 79–86.

Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*. to appear.

Nishimori, Y., & Akaho, S. (2005). Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing*, *67*, 106–135.

Patriksson, M. (1999). *Nonlinear programming and variational inequality problems*. Dordrecht, the Netherlands: Kluwer Academic.

Petersen, K. B., & Pedersen, M. S. (2008). *The matrix cookbook* (Technical Report). Technical University of Denmark.

Plumbley, M. D. (2005). Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras. *Neurocomputing*, *67*, 161–197.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1992). *Numerical recipes in C*. Cambridge, UK: Cambridge University Press. 2nd edition.

Qin, J. (1998). Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, *85*, 619–639.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. (Eds.). (2009). *Dataset shift in machine learning*. Cambridge, Massachusetts, USA: MIT Press.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, *90*, 227–244.

Silverman, B. W. (1978). Density ratios, empirical likelihood and cot death. *Journal of the Royal Statistical Society, Series C*, *27*, 26–33.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London, UK: Chapman and Hall.

Smola, A., Song, L., & Teo, C. H. (2009). Relative novelty detection. *Proceedings of Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS2009)* (pp. 536–543). Clearwater Beach, FL, USA.

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, *2*, 67–93.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, *36*, 111–147.

Storkey, A., & Sugiyama, M. (2007). Mixture regression for covariate shift. *Advances in Neural Information Processing Systems 19* (pp. 1337–1344). Cambridge, Massachusetts, USA: MIT Press.

Sugiyama, M. (2006). Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, *7*, 141–166.

Sugiyama, M. (2007). Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of Machine Learning Research*, *8*, 1027–1061.

Sugiyama, M. (2010). Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting. *IEICE Transactions on Information and Systems*, *E93-D*, 2690–2701.

Sugiyama, M., Kanamori, T., Suzuki, T., Hido, S., Sese, J., Takeuchi, I., & Wang, L. (2009). A density-ratio framework for statistical data processing. *IPSJ Transactions on Computer Vision and Applications*, *1*, 183–208.

Sugiyama, M., & Kawanabe, M. (2010). *Covariate shift adaptation: Towards machine learning in non-stationary environment*. Cambridge, Massachusetts, USA: MIT Press. to appear.

Sugiyama, M., Kawanabe, M., & Chui, P. L. (2010a). Dimensionality reduction for density ratio estimation in high-dimensional spaces. *Neural Networks*, *23*, 44–59.

Sugiyama, M., Krauledat, M., & Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, *8*, 985–1005.

Sugiyama, M., & Müller, K.-R. (2005). Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, *23*, 249–279.

Sugiyama, M., & Nakajima, S. (2009). Pool-based active learning in approximate linear regression. *Machine Learning*, *75*, 249–274.

Sugiyama, M., Suzuki, T., & Kanamori, T. (2010b). Density ratio estimation: A comprehensive review. *RIMS Kokyuroku*, 10–31.

Sugiyama, M., Suzuki, T., & Kanamori, T. (2011). *Density ratio estimation in machine learning: A versatile tool for statistical data processing*. Cambridge, UK: Cambridge University Press.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., & Kawanabe, M. (2008). Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, *60*, 699–746.

Sugiyama, M., Takeuchi, I., Suzuki, T., Kanamori, T., Hachiya, H., & Okanohara, D. (2010c). Least-squares conditional density estimation. *IEICE Transactions on Information and Systems*, *E93-D*, 583–594.

Sun, J., & Woodroofe, M. (1997). Semi-parametric estimates under biased sampling. *Statistica Sinica*, *7*, 545–575.

Suzuki, T., & Sugiyama, M. (2009). Estimating squared-loss mutual information for independent component analysis. *Independent Component Analysis and Signal Separation* (pp. 130–137). Berlin, Germany: Springer.

Suzuki, T., & Sugiyama, M. (2010). Sufficient dimension reduction via squared-loss mutual information estimation. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS2010)* (pp. 804–811). Sardinia, Italy.

Suzuki, T., Sugiyama, M., Kanamori, T., & Sese, J. (2009a). Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, *10*, S52.

Suzuki, T., Sugiyama, M., Sese, J., & Kanamori, T. (2008). Approximating mutual information by maximum likelihood density ratio estimation. *Proceedings of ECML-PKDD2008 Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery 2008 (FSDM2008)* (pp. 5–20). Antwerp, Belgium.

Suzuki, T., Sugiyama, M., & Tanaka, T. (2009b). Mutual information approximation via maximum likelihood estimation of density ratio. *Proceedings of 2009 IEEE International Symposium on Information Theory (ISIT2009)* (pp. 463–467). Seoul, Korea.

Takeuchi, I., Nomura, K., & Kanamori, T. (2009). Nonparametric conditional density estimation using piecewise-linear solution path of kernel quantile regression. *Neural Computation*, *21*, 533–559.

Tsuboi, Y., Kashima, H., Hido, S., Bickel, S., & Sugiyama, M. (2009). Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, *17*, 138–155.

Ueki, K., Sugiyama, M., & Ihara, Y. (2010). Perceived age estimation under lighting condition change by covariate shift adaptation. *20th International Conference on Pattern Recognition (ICPR2010)* (pp. 3400–3403). Istanbul, Turkey.

Vapnik, V. N. (1998). *Statistical learning theory*. New York, NY, USA: Wiley.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Wiens, D. P. (2000). Robust weights and designs for biased regression models: Least squares and generalized M-estimation. *Journal of Statistical Planning and Inference*, *83*, 395–412.

Yamada, M., & Sugiyama, M. (2010). Dependence minimizing regression with model selection for non-linear causal inference under non-Gaussian noise. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI2010)* (pp. 643–648). Atlanta, Georgia, USA: The AAAI Press.

Yamada, M., Sugiyama, M., & Matsui, T. (2010). Semi-supervised speaker identification under covariate shift. *Signal Processing*, *90*, 2353–2361.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. *Proceedings of the Twenty-First International Conference on Machine Learning (ICML2004)* (pp. 903–910). New York, NY, USA: ACM Press.

A Usage of Density Ratios in Data Processing

We are interested in estimating density ratios since they are useful in various data processing tasks. Here, we briefly review possible usage of density ratios (Sugiyama et al., 2009; Sugiyama et al., 2011).

A.1 Covariate Shift Adaptation

Covariate shift (Shimodaira, 2000) is a situation in supervised learning where input distributions change between the training and test phases, but the conditional distribution of outputs given inputs remains unchanged. *Extrapolation* (i.e., prediction is made outside the training region) would be a typical example of covariate shift. Standard learning techniques such as maximum likelihood estimation are biased under covariate shift; the bias caused by covariate shift can be asymptotically canceled by weighting the loss function according to the *importance*[2] (Shimodaira, 2000; Zadrozny, 2004; Sugiyama & Müller, 2005; Sugiyama et al., 2007; Quiñonero-Candela et al., 2009; Sugiyama & Kawanabe, 2010).

[2]The test input density over the training input density is referred to as the importance in the context of *importance sampling* (Fishman, 1996).

The basic idea of covariate shift adaptation is summarized in the following importance sampling identity:

$$\begin{aligned}\mathop{\mathbb{E}}_{p_{\mathrm{nu}}(\boldsymbol{x})}[\mathrm{loss}(\boldsymbol{x})] &= \int \mathrm{loss}(\boldsymbol{x}) p_{\mathrm{nu}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \int \mathrm{loss}(\boldsymbol{x}) r(\boldsymbol{x}) p_{\mathrm{de}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \mathop{\mathbb{E}}_{p_{\mathrm{de}}(\boldsymbol{x})}[\mathrm{loss}(\boldsymbol{x}) r(\boldsymbol{x})].\end{aligned}$$

That is, the expectation of a function $\mathrm{loss}(\boldsymbol{x})$ over $p_{\mathrm{nu}}(\boldsymbol{x})$ can be computed by the importance-weighted expectation of $\mathrm{loss}(\boldsymbol{x})$ over $p_{\mathrm{de}}(\boldsymbol{x})$. Similarly, standard model selection criteria such as *cross-validation* (Stone, 1974; Wahba, 1990) or *Akaike's information criterion* (Akaike, 1974) lose their unbiasedness under covariate shift; proper unbiasedness can be recovered by modifying the methods based on importance weighting (Shimodaira, 2000; Zadrozny, 2004; Sugiyama & Müller, 2005; Sugiyama et al., 2007). Furthermore, the performance of *active learning* or the *experiment design*, i.e., the training input distribution is designed by the user to enhance the generalization performance, could also be improved by the use of the importance (Wiens, 2000; Kanamori & Shimodaira, 2003; Sugiyama, 2006; Sugiyama & Nakajima, 2009).

Thus, the importance plays a central role in covariate shift adaptation, and density-ratio estimation methods could be used for reducing the estimation bias under covariate shift. Examples of successful real-world applications include brain-computer interface (Sugiyama et al., 2007), robot control (Hachiya et al., 2009a; Akiyama et al., 2010; Hachiya et al., 2009b), speaker identification (Yamada et al., 2010), age prediction from face images (Ueki et al., 2010), wafer alignment in semiconductor exposure apparatus (Sugiyama & Nakajima, 2009), and natural language processing (Tsuboi et al., 2009). A similar importance-weighting idea also plays a central role in domain adaptation (Storkey & Sugiyama, 2007) and multi-task learning (Bickel et al., 2008).

A.2 Inlier-based Outlier Detection

Let us consider an outlier detection problem of finding irregular samples in a dataset ("evaluation dataset") based on another dataset ("model dataset") that only contains regular samples. Defining the density ratio over the two sets of samples, we can see that the density ratio values for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus, the density ratio value could be used as an index of the degree of outlyingness (Hido et al., 2008; Smola et al., 2009; Hido et al., 2010). Since the evaluation dataset usually has a wider support than the model dataset, we regard the evaluation dataset as samples corresponding to $p_{\mathrm{de}}(\boldsymbol{x})$ and the model dataset as samples corresponding to $p_{\mathrm{nu}}(\boldsymbol{x})$. Then outliers tend to have smaller density-ratio values (i.e., close to zero). As such, density-ratio estimation methods could be employed in outlier detection scenarios.

A similar idea could be used for change-point detection in time-series (Kawahara & Sugiyama, 2009).

A.3 Conditional Density Estimation

Suppose we are given n i.i.d. paired samples $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^n$ drawn from a joint distribution with density $q(\boldsymbol{x}, \boldsymbol{y})$. The goal is to estimate the conditional density $q(\boldsymbol{y}|\boldsymbol{x})$. When the domain of $\boldsymbol{x}$ is continuous, conditional density estimation is not straightforward since a naive empirical approximation cannot be used (Bishop, 2006; Takeuchi et al., 2009).

In the context of density ratio estimation, let us regard $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^n$ as samples corresponding to the numerator of the density ratio and $\{\boldsymbol{x}_k\}_{k=1}^n$ as samples corresponding to the denominator of the density ratio, i.e., we consider the density ratio defined by

$$r(\boldsymbol{x}, \boldsymbol{y}) := \frac{q(\boldsymbol{x}, \boldsymbol{y})}{q(\boldsymbol{x})} = q(\boldsymbol{y}|\boldsymbol{x}),$$

where $q(\boldsymbol{x})$ is the marginal density of $\boldsymbol{x}$. Then a density-ratio estimation method directly gives an estimate of the conditional density (Sugiyama et al., 2010c).

When $\boldsymbol{y}$ is categorical, the same method can be used for probabilistic classification (Sugiyama, 2010).

A.4 Mutual Information Estimation

Suppose we are given n i.i.d. paired samples $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^n$ drawn from a joint distribution with density $q(\boldsymbol{x}, \boldsymbol{y})$. Let us denote the marginal densities of $\boldsymbol{x}$ and $\boldsymbol{y}$ by $q(\boldsymbol{x})$ and $q(\boldsymbol{y})$, respectively. Then *mutual information* $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y})$ between random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined by

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) := \iint q(\boldsymbol{x}, \boldsymbol{y}) \log \frac{q(\boldsymbol{x}, \boldsymbol{y})}{q(\boldsymbol{x})q(\boldsymbol{y})} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y},$$

which plays a central role in *information theory* (Cover & Thomas, 2006).

Let us regard $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^n$ as samples corresponding to the numerator of the density ratio and $\{(\boldsymbol{x}_k, \boldsymbol{y}_{k'})\}_{k,k'=1}^n$ as samples corresponding to the denominator of the density ratio, i.e.,

$$r(\boldsymbol{x}, \boldsymbol{y}) := \frac{q(\boldsymbol{x}, \boldsymbol{y})}{q(\boldsymbol{x})q(\boldsymbol{y})}.$$

Then mutual information can be directly estimated using a density-ratio estimation method (Suzuki et al., 2008; Suzuki et al., 2009b). General divergence functionals can also be estimated in a similar way (Nguyen et al., 2010).

Mutual information can be used for measuring independence between random variables (Kraskov et al., 2004; Hulle, 2005) since it vanishes if and only if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are statistically independent. Thus density-ratio estimation methods are applicable, e.g., to variable selection (Suzuki et al., 2009a), independent component analysis (Suzuki & Sugiyama, 2009), supervised dimensionality reduction (Suzuki & Sugiyama, 2010), and causal inference (Yamada & Sugiyama, 2010).

B Proof of Lemma 1

Here, let us consider the *f-divergences* (Ali & Silvey, 1966; Csiszár, 1967) and prove a similar inequality for a broader class of divergences. An f-divergence is defined using a convex function f such that $f(1) = 0$ as

$$I_f[p_{\mathrm{nu}}(\boldsymbol{x}), p_{\mathrm{de}}(\boldsymbol{x})] := \int p_{\mathrm{de}}(\boldsymbol{x}) f\left(\frac{p_{\mathrm{nu}}(\boldsymbol{x})}{p_{\mathrm{de}}(\boldsymbol{x})}\right) \mathrm{d}\boldsymbol{x}.$$

The f-divergence is reduced to the Kullback-Leibler divergence if

$$f(t) = -\log t,$$

and the Pearson divergence if

$$f(t) = \frac{1}{2}(t-1)^2.$$

Using *Jensen's inequality* (Bishop, 2006), we have

$$\begin{aligned}
I_f[p_{\mathrm{nu}}(\boldsymbol{x}), p_{\mathrm{de}}(\boldsymbol{x})] &= \iint p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u}) p_{\mathrm{de}}(\boldsymbol{u}) f\left(\frac{p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u}) p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u}) p_{\mathrm{de}}(\boldsymbol{u})}\right) \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v} \\
&\geq \int p_{\mathrm{de}}(\boldsymbol{u}) f\left(\int p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u}) \frac{p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u}) p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u}) p_{\mathrm{de}}(\boldsymbol{u})} \mathrm{d}\boldsymbol{v}\right) \mathrm{d}\boldsymbol{u} \\
&= \int p_{\mathrm{de}}(\boldsymbol{u}) f\left(\frac{p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{u})} \int p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u}) \mathrm{d}\boldsymbol{v}\right) \mathrm{d}\boldsymbol{u} \\
&= \int p_{\mathrm{de}}(\boldsymbol{u}) f\left(\frac{p_{\mathrm{nu}}(\boldsymbol{u})}{p_{\mathrm{de}}(\boldsymbol{u})}\right) \mathrm{d}\boldsymbol{u} \\
&= I_f[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})].
\end{aligned}$$

Thus, we have

$$I_f[p_{\mathrm{nu}}(\boldsymbol{x}), p_{\mathrm{de}}(\boldsymbol{x})] - I_f[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})] \geq 0,$$

and the equality holds if and only if $p_{\mathrm{nu}}(\boldsymbol{v}|\boldsymbol{u}) = p_{\mathrm{de}}(\boldsymbol{v}|\boldsymbol{u})$. ■

C Proof of Lemma 2

For

$$\boldsymbol{F} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_b)^{-1},$$

$\widehat{\mathrm{PD}}[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})]$ can be expressed as

$$\begin{aligned}
\widehat{\mathrm{PD}}[p_{\mathrm{nu}}(\boldsymbol{u}), p_{\mathrm{de}}(\boldsymbol{u})] &= \frac{1}{2} \sum_{\ell=1}^{b} \widehat{\alpha}_\ell \widehat{h}_\ell - \frac{1}{2} \\
&= \frac{1}{2} \sum_{\ell,\ell'=1}^{b} \widehat{h}_\ell \widehat{h}_{\ell'} F_{\ell,\ell'} - \frac{1}{2}.
\end{aligned}$$

Thus, its partial derivative with respect to $\boldsymbol{U}$ is given by

$$\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{U}} = \sum_{\ell=1}^{b} \widehat{\alpha}_\ell \frac{\partial\widehat{h}_\ell}{\partial\boldsymbol{U}} + \frac{1}{2}\sum_{\ell,\ell'=1}^{b} \widehat{h}_\ell \widehat{h}_{\ell'} \frac{\partial F_{\ell,\ell'}}{\partial\boldsymbol{U}}. \tag{16}$$

Since

$$\frac{\partial\boldsymbol{B}^{-1}}{\partial t} = -\boldsymbol{B}^{-1}\frac{\partial\boldsymbol{B}}{\partial t}\boldsymbol{B}^{-1}$$

holds for a square invertible matrix $\boldsymbol{B}$ (Petersen & Pedersen, 2008), it holds that

$$\frac{\partial\boldsymbol{F}}{\partial U_{k,k'}} = -(\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}_b)^{-1}\frac{\partial\widehat{\boldsymbol{H}}}{\partial U_{k,k'}}(\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}_b)^{-1}.$$

Then we have

$$\begin{aligned}\sum_{\ell,\ell'=1}^{b} \widehat{h}_\ell\widehat{h}_{\ell'}\left[\frac{\partial\boldsymbol{F}}{\partial U_{k,k'}}\right]_{\ell,\ell'} &= -\widehat{\boldsymbol{h}}^\top(\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}_b)^{-1}\frac{\partial\widehat{\boldsymbol{H}}}{\partial U_{k,k'}}(\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}_b)^{-1}\widehat{\boldsymbol{h}} \\ &= -\sum_{\ell,\ell'=1}^{b}\widehat{\alpha}_\ell\widehat{\alpha}_{\ell'}\left[\frac{\partial\widehat{\boldsymbol{H}}}{\partial U_{k,k'}}\right]_{\ell,\ell'}.\end{aligned}$$

Substituting this into Eq.(16), we obtain Eq.(8). Eqs.(9) and (10) are clear from Eqs.(3) and (2). Finally, we prove Eq.(11). The basis function $\psi_\ell(\boldsymbol{u})$ can be expressed as

$$\psi_\ell(\boldsymbol{u}) = \psi_\ell(\boldsymbol{U}\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{U}(\boldsymbol{x}-\boldsymbol{c}'_\ell)\|^2}{2\sigma^2}\right).$$

Since $\frac{\partial\boldsymbol{a}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{a}}{\partial\boldsymbol{A}} = 2\boldsymbol{A}\boldsymbol{a}^\top\boldsymbol{a}$ (Petersen & Pedersen, 2008), we have

$$\frac{\partial\psi_\ell(\boldsymbol{u})}{\partial\boldsymbol{U}} = -\frac{1}{\sigma^2}\boldsymbol{U}(\boldsymbol{x}-\boldsymbol{c}'_\ell)(\boldsymbol{x}-\boldsymbol{c}'_\ell)^\top\exp\left(-\frac{\|\boldsymbol{U}(\boldsymbol{x}-\boldsymbol{c}'_\ell)\|^2}{2\sigma^2}\right),$$

from which we obtain Eq.(11). ■

D Proof of Lemma 3

The proof we provide here essentially follows the argument in Plumbley (2005).

For

$$\boldsymbol{W} = \begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{V}\end{bmatrix}, \quad \boldsymbol{W}_0 = \begin{bmatrix}\boldsymbol{U}_0\\ \boldsymbol{V}_0\end{bmatrix},$$

rotation $\boldsymbol{W}$ from some $\boldsymbol{W}_0$ can be expressed as follows (Plumbley, 2005):

$$\boldsymbol{W} = \exp(\boldsymbol{M})\boldsymbol{W}_0, \tag{17}$$

where $\boldsymbol{M}$ is some skew-symmetric matrix. Let us consider the space of skew-symmetric matrices, and let $\boldsymbol{E}$ be an element in that space with unit length. Then the gradient of a function $\widehat{\mathrm{PD}}(\boldsymbol{M})$ with respect to $\boldsymbol{M}$, $\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{M}}$, in this space is given as an element whose inner product $\left\langle \frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{M}}, \boldsymbol{E} \right\rangle$ is equal to the derivative of $\widehat{\mathrm{PD}}(\boldsymbol{M})$ in the direction $\boldsymbol{E}$ (Plumbley, 2005). Thus, for $\boldsymbol{M} = t\boldsymbol{E}$ with t being a scalar, we have

$$\frac{\partial\widehat{\mathrm{PD}}(t\boldsymbol{E})}{\partial t} = \left\langle \frac{\partial\widehat{\mathrm{PD}}(\boldsymbol{M})}{\partial\boldsymbol{M}}, \boldsymbol{E} \right\rangle.$$

If we adopt Eq.(12) as the inner product of the space of skew-symmetric matrices, we have

$$\frac{\partial\widehat{\mathrm{PD}}(t\boldsymbol{E})}{\partial t} = \frac{1}{2}\mathrm{tr}\left(\frac{\partial\widehat{\mathrm{PD}}(\boldsymbol{M})}{\partial\boldsymbol{M}}\boldsymbol{E}^\top\right). \tag{18}$$

On the other hand, from Eq.(17) with $\boldsymbol{M} = t\boldsymbol{E}$, $\frac{\partial\boldsymbol{W}}{\partial t}$ can be expressed as follows (Petersen & Pedersen, 2008):

$$\frac{\partial\boldsymbol{W}}{\partial t} = \boldsymbol{E}\exp(t\boldsymbol{E})\boldsymbol{W}_0 = \boldsymbol{E}\boldsymbol{W}.$$

Then $\frac{\partial\widehat{\mathrm{PD}}(t\boldsymbol{E})}{\partial t}$ can be expressed as

$$\frac{\partial\widehat{\mathrm{PD}}(t\boldsymbol{E})}{\partial t} = \mathrm{tr}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}\frac{\partial\boldsymbol{W}}{\partial t}^\top\right) = \mathrm{tr}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}\boldsymbol{W}^\top\boldsymbol{E}^\top\right). \tag{19}$$

Since $\boldsymbol{E}$ is skew-symmetric, it can be expressed as

$$\boldsymbol{E} = \frac{1}{2}\boldsymbol{E} + \frac{1}{2}\boldsymbol{E} = \frac{1}{2}\boldsymbol{E} - \frac{1}{2}\boldsymbol{E}^\top.$$

Substituting this into Eq.(19), we have

$$\begin{aligned}
\frac{\partial\widehat{\mathrm{PD}}(t\boldsymbol{E})}{\partial t} &= \frac{1}{2}\mathrm{tr}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}\boldsymbol{W}^\top\boldsymbol{E}^\top\right) - \frac{1}{2}\mathrm{tr}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}\boldsymbol{W}^\top\boldsymbol{E}\right) \\
&= \frac{1}{2}\mathrm{tr}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}\boldsymbol{W}^\top\boldsymbol{E}^\top\right) - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{W}\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}^\top\boldsymbol{E}^\top\right) \\
&= \frac{1}{2}\mathrm{tr}\left(\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}\boldsymbol{W}^\top - \boldsymbol{W}\frac{\partial\widehat{\mathrm{PD}}}{\partial\boldsymbol{W}}^\top\right)\boldsymbol{E}^\top\right).
\end{aligned} \tag{20}$$

Combining Eqs.(18) and (20), we have

$$\begin{aligned}\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{M}} &= \frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{W}}\boldsymbol{W}^\top - \boldsymbol{W}\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{W}}^\top \\ &= \begin{bmatrix} \frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{U}^\top - \boldsymbol{U}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\right)^\top & \frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{V}^\top - \boldsymbol{U}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{V}}\right)^\top \\ \frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{V}}\boldsymbol{U}^\top - \boldsymbol{V}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\right)^\top & \frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{V}}\boldsymbol{V}^\top - \boldsymbol{V}\left(\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{V}}\right)^\top \end{bmatrix}. \end{aligned} \tag{21}$$

Eq.(11) implies that $\frac{\partial\psi_\ell(\boldsymbol{u})}{\partial \boldsymbol{U}}\boldsymbol{U}^\top$ is symmetric. Then Eqs.(8) and (9) imply that $\frac{\partial\widehat{h}_\ell}{\partial \boldsymbol{U}}\boldsymbol{U}^\top$ and $\frac{\partial\widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{U}}\boldsymbol{U}^\top$ are also symmetric. Consequently, Eq.(10) imply that $\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{U}^\top$ is symmetric:

$$\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{U}^\top = \left(\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{U}^\top\right)^\top = \boldsymbol{U}\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}^\top.$$

Since the range of $\boldsymbol{V}$ is assumed to be orthogonal to the range of $\boldsymbol{U}$ (see Section 3.1), $\widehat{\mathrm{PD}}$ is independent of $\boldsymbol{V}$, and thus we have

$$\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{V}} = \boldsymbol{O}_{(d-m),d},$$

where $\boldsymbol{O}_{d,d'}$ is the $d \times d'$ matrix with all zeros. Then Eq.(21) yields

$$\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{M}} = \begin{bmatrix} \boldsymbol{O}_{m,m} & \frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{V}^\top \\ -(\frac{\partial\widehat{\mathrm{PD}}}{\partial \boldsymbol{U}}\boldsymbol{V}^\top)^\top & \boldsymbol{O}_{(d-m),(d-m)} \end{bmatrix},$$

which concludes the proof. ■

IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E93-A, no.4, pp.787–798, 2010.

Theoretical Analysis of Density Ratio Estimation

Takafumi Kanamori (kanamori@is.nagoya-u.ac.jp)
Nagoya University

Taiji Suzuki (s-taiji@stat.t.u-tokyo.ac.jp)
The University of Tokyo

Masashi Sugiyama (sugi@cs.titech.ac.jp)
Tokyo Institute of Technology
and
Japan Science and Technology Agency

Abstract

Density ratio estimation has gathered a great deal of attention recently since it can be used for various data processing tasks. In this paper, we consider three methods of density ratio estimation: (A) the numerator and denominator densities are separately estimated and then the ratio of the estimated densities is computed, (B) a logistic regression classifier discriminating denominator samples from numerator samples is learned and then the ratio of the posterior probabilities is computed, and (C) the density ratio function is directly modeled and learned by minimizing the empirical Kullback-Leibler divergence. We first prove that when the numerator and denominator densities are known to be members of the exponential family, (A) is better than (B) and (B) is better than (C). Then we show that once the model assumption is violated, (C) is better than (A) and (B). Thus in practical situations where no exact model is available, (C) would be the most promising approach to density ratio estimation.

Keywords

density ratio estimation, density estimation, logistic regression, asymptotic analysis, Gaussian assumption.

1 Introduction

The ratio of two probability density functions has been demonstrated to be useful in various data processing tasks [21], such as non-stationarity adaptation [18, 35, 23, 22, 17, 27], outlier detection [7, 19], conditional density estimation [26], feature selection [31, 30], feature extraction [29], and independent component analysis [28]. Thus accurately estimating the density ratio is an important and challenging research topic in the machine learning and data mining communities.

A naive approach to density ratio estimation is (A) density ratio estimation by separate maximum likelihood density estimation—first the numerator and denominator densities are separately estimated and then the ratio of the estimated densities is computed. However, density estimation is substantially more difficult than density ratio estimation and the above two-shot process of first estimating the densities and then taking their ratio is thought to be less accurate. To cope with this problem, various alternative methods have been developed recently, which allow one to estimate the density ratio without going through density estimation [16, 8, 15, 25, 10].

In this paper, we consider the following two methods in addition to the method (A): (B) density ratio estimation by logistic regression [16, 3, 1]—a logistic regression classifier discriminating numerator samples from denominator samples is used for density ratio estimation, and (C) direct density ratio estimation by empirical Kullback-Leibler divergence minimization [15, 25]—the density ratio function is directly modeled and learned. The goal of this paper is to theoretically compare the accuracy of these three density ratio estimation schemes.

We first prove that when the numerator and denominator densities are known to be members of the exponential family, (A) is better than (B) and (B) is better than (C). The fact that (A) is better than (B) could be regarded as an extension of the existing result for binary classification [5]—estimating data generating densities by maximum likelihood estimation has higher statistical efficiency than logistic regression in classification scenarios. On the other hand, the fact that (B) is better than (C) follows from the fact that (B) has the smallest asymptotic variance in a class of semi-parametric estimators [16].

We then show that when the model assumption is violated, (C) is better than (A) and (B). Our statement is that the estimator obtained by (C) converges to the projection of the true density ratio function onto the target parametric model (i.e., the optimal approximation in the model), while the estimators obtained by (A) and (B) do not generally converge to the projection.

Since model misspecification would be a usual situation in practice, (C) is the most promising approach in density ratio estimation. In a regression framework, an asymptotic analysis with a similar spirit exists [14].

2 Density Ratio Estimation

In this section, we formulate the problem of density ratio estimation and review three density ratio estimators.

2.1 Problem Formulation

Let $\mathcal{X}$ ($\subset \mathbb{R}^d$) be the data domain and suppose we are given independent and identically distributed (i.i.d.) samples $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^n$ drawn from a distribution with density $p_{\mathrm{nu}}^*(\boldsymbol{x})$ and i.i.d. samples $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^n$ drawn from another distribution with density $p_{\mathrm{de}}^*(\boldsymbol{x})$:

$$\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^n \overset{i.i.d.}{\sim} p_{\mathrm{nu}}^*(\boldsymbol{x}),$$
$$\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^n \overset{i.i.d.}{\sim} p_{\mathrm{de}}^*(\boldsymbol{x}).$$

The subscripts 'nu' and 'de' denote 'numerator' and 'denominator', respectively. We assume that the latter density $p_{\mathrm{de}}^*(\boldsymbol{x})$ is strictly positive, i.e.,

$$p_{\mathrm{de}}^*(\boldsymbol{x}) > 0, \quad \forall \boldsymbol{x} \in \mathcal{X}.$$

The problem we address in this paper is to estimate the density ratio

$$r^*(\boldsymbol{x}) := \frac{p_{\mathrm{nu}}^*(\boldsymbol{x})}{p_{\mathrm{de}}^*(\boldsymbol{x})}$$

from samples $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^n$ and $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^n$.

The goal of this paper is to theoretically compare the performance of the following three density ratio estimators:

(A) Density ratio estimation by separate maximum likelihood density estimation (see Section 2.3 for details),

(B) Density ratio estimation by logistic regression [16, 3, 1] (see Section 2.4 for details),

(C) Direct density ratio estimation by empirical Kullback-Leibler divergence minimization [15, 25] (see Section 2.5 for details).

2.2 Measure of Accuracy

Let us consider the *unnormalized Kullback-Leibler divergence* [2] from the true density $p_{\mathrm{nu}}^*(\boldsymbol{x})$ to its estimator $\widehat{r}(\boldsymbol{x})p_{\mathrm{de}}^*(\boldsymbol{x})$:

$$\mathrm{UKL}(p_{\mathrm{nu}}^* \| \widehat{r} \cdot p_{\mathrm{de}}^*) := \int p_{\mathrm{nu}}^*(\boldsymbol{x}) \log \frac{p_{\mathrm{nu}}^*(\boldsymbol{x})}{\widehat{r}(\boldsymbol{x})p_{\mathrm{de}}^*(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} - 1 + \int \widehat{r}(\boldsymbol{x})p_{\mathrm{de}}^*(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{1}$$

$\mathrm{UKL}(p_{\mathrm{nu}}^*(\boldsymbol{x}) \| \widehat{r}(\boldsymbol{x})p_{\mathrm{de}}^*(\boldsymbol{x}))$ is non-negative for all $\widehat{r}$ and vanishes if and only if $\widehat{r} = r^*$. If $\widehat{r}(\boldsymbol{x})p_{\mathrm{de}}^*(\boldsymbol{x})$ is normalized to be a probability density function, i.e.,

$$\int \widehat{r}(\boldsymbol{x})p_{\mathrm{de}}^*(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1,$$

then the unnormalized Kullback-Leibler divergence is reduced to the ordinary Kullback-Leibler divergence [12]:

$$\mathrm{KL}(p^*_{\mathrm{nu}} \| \widehat{r} \cdot p^*_{\mathrm{de}}) := \int p^*_{\mathrm{nu}}(\boldsymbol{x}) \log \frac{p^*_{\mathrm{nu}}(\boldsymbol{x})}{\widehat{r}(\boldsymbol{x}) p^*_{\mathrm{de}}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}. \tag{2}$$

In our theoretical analysis, we use the expectation of $\mathrm{UKL}(p^*_{\mathrm{nu}} \| \widehat{r} \cdot p^*_{\mathrm{de}})$ over $\{\boldsymbol{x}^{\mathrm{nu}}_i\}_{i=1}^n$ and $\{\boldsymbol{x}^{\mathrm{de}}_j\}_{j=1}^n$ as the measure of accuracy of a density ratio estimator $\widehat{r}(\boldsymbol{x})$:

$$J(\widehat{r}) := \mathbb{E}\left[\mathrm{UKL}(p^*_{\mathrm{nu}} \| \widehat{r} \cdot p^*_{\mathrm{de}})\right], \tag{3}$$

where $\mathbb{E}$ denotes the expectation over $\{\boldsymbol{x}^{\mathrm{nu}}_i\}_{i=1}^n$ and $\{\boldsymbol{x}^{\mathrm{de}}_j\}_{j=1}^n$.

In the rest of this section, the three methods of density ratio estimation we are dealing with are described in detail.

2.3 Method (A): Density Ratio Estimation by Separate Maximum Likelihood Density Estimation

For $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$, two parametric models $p_{\mathrm{nu}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{nu}})$ and $p_{\mathrm{de}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{de}})$ such that

$$\begin{aligned}
&\int p_{\mathrm{nu}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{nu}}) \mathrm{d}\boldsymbol{x} = 1, \quad \forall \boldsymbol{\theta}_{\mathrm{nu}} \in \Theta_{\mathrm{nu}}, \\
&p_{\mathrm{nu}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{nu}}) \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{X}, \ \forall \boldsymbol{\theta}_{\mathrm{nu}} \in \Theta_{\mathrm{nu}}, \\
&\int p_{\mathrm{de}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{de}}) \mathrm{d}\boldsymbol{x} = 1 \quad \forall \boldsymbol{\theta}_{\mathrm{de}} \in \Theta_{\mathrm{de}}, \\
&p_{\mathrm{de}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{de}}) \geq 0 \quad \forall \boldsymbol{x} \in \mathcal{X}, \ \forall \boldsymbol{\theta}_{\mathrm{de}} \in \Theta_{\mathrm{de}},
\end{aligned}$$

are prepared. Then the maximum likelihood estimators $\widehat{\boldsymbol{\theta}}_{\mathrm{nu}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{de}}$ are computed separately from $\{\boldsymbol{x}^{\mathrm{nu}}_i\}_{i=1}^n$ and $\{\boldsymbol{x}^{\mathrm{de}}_j\}_{j=1}^n$:

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}_{\mathrm{nu}} &:= \operatorname*{argmax}_{\boldsymbol{\theta}_{\mathrm{nu}} \in \Theta_{\mathrm{nu}}} \left[\sum_{i=1}^n \log p_{\mathrm{nu}}(\boldsymbol{x}^{\mathrm{nu}}_i; \boldsymbol{\theta}_{\mathrm{nu}}) \right], \\
\widehat{\boldsymbol{\theta}}_{\mathrm{de}} &:= \operatorname*{argmax}_{\boldsymbol{\theta}_{\mathrm{de}} \in \Theta_{\mathrm{de}}} \left[\sum_{j=1}^n \log p_{\mathrm{de}}(\boldsymbol{x}^{\mathrm{de}}_j; \boldsymbol{\theta}_{\mathrm{de}}) \right].
\end{aligned}$$

Note that the maximum likelihood estimators $\widehat{\boldsymbol{\theta}}_{\mathrm{nu}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{de}}$ minimize the empirical Kullback-Leibler divergences from the true densities $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$ to their models $p_{\mathrm{nu}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{nu}})$ and $p_{\mathrm{de}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{de}})$, respectively:

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}_{\mathrm{nu}} &= \operatorname*{argmin}_{\boldsymbol{\theta}_{\mathrm{nu}} \in \Theta_{\mathrm{nu}}} \left[\frac{1}{n} \sum_{i=1}^n \log \frac{p^*_{\mathrm{nu}}(\boldsymbol{x}^{\mathrm{nu}}_i)}{p_{\mathrm{nu}}(\boldsymbol{x}^{\mathrm{nu}}_i; \boldsymbol{\theta}_{\mathrm{nu}})} \right], \\
\widehat{\boldsymbol{\theta}}_{\mathrm{de}} &= \operatorname*{argmin}_{\boldsymbol{\theta}_{\mathrm{de}} \in \Theta_{\mathrm{de}}} \left[\frac{1}{n} \sum_{j=1}^n \log \frac{p^*_{\mathrm{de}}(\boldsymbol{x}^{\mathrm{de}}_j)}{p_{\mathrm{nu}}(\boldsymbol{x}^{\mathrm{de}}_j; \boldsymbol{\theta}_{\mathrm{de}})} \right].
\end{aligned}$$

Finally, a density ratio estimator is constructed by taking the ratio of the estimated densities:

$$\widehat{r}_{\mathrm{A}}(\boldsymbol{x}) := \frac{p_{\mathrm{nu}}(\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{nu}})}{p_{\mathrm{de}}(\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{de}})} \left(\frac{1}{n}\sum_{j=1}^{n} \frac{p_{\mathrm{nu}}(\boldsymbol{x}_j^{\mathrm{de}};\widehat{\boldsymbol{\theta}}_{\mathrm{nu}})}{p_{\mathrm{de}}(\boldsymbol{x}_j^{\mathrm{de}};\widehat{\boldsymbol{\theta}}_{\mathrm{de}})} \right)^{-1},$$

where the estimator is normalized so that

$$\frac{1}{n}\sum_{j=1}^{n} \widehat{r}_{\mathrm{A}}(\boldsymbol{x}_j^{\mathrm{de}}) = 1.$$

2.4 Method (B): Density Ratio Estimation by Logistic Regression

Let us assign a selector variable $y =$ 'nu' to samples drawn from $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $y =$ 'de' to samples drawn from $p^*_{\mathrm{de}}(\boldsymbol{x})$, i.e., the two densities are written as

$$p^*_{\mathrm{nu}}(\boldsymbol{x}) = q^*(\boldsymbol{x}|y = \text{'nu'}),$$
$$p^*_{\mathrm{de}}(\boldsymbol{x}) = q^*(\boldsymbol{x}|y = \text{'de'}).$$

Since

$$q^*(\boldsymbol{x}|y = \text{'nu'}) = \frac{q^*(y = \text{'nu'}|\boldsymbol{x})q^*(\boldsymbol{x})}{q^*(y = \text{'nu'})},$$
$$q^*(\boldsymbol{x}|y = \text{'de'}) = \frac{q^*(y = \text{'de'}|\boldsymbol{x})q^*(\boldsymbol{x})}{q^*(y = \text{'de'})},$$

the density ratio can be expressed in terms of y as

$$r^*(\boldsymbol{x}) = \frac{q^*(y = \text{'nu'}|\boldsymbol{x})}{q^*(y = \text{'nu'})} \frac{q^*(y = \text{'de'})}{q^*(y = \text{'de'}|\boldsymbol{x})} = \frac{q^*(y = \text{'nu'}|\boldsymbol{x})}{q^*(y = \text{'de'}|\boldsymbol{x})},$$

where we used the fact that

$$q^*(y = \text{'nu'}) = q^*(y = \text{'de'}) = \frac{1}{2}$$

in the current setup.

The conditional probability $q^*(y|\boldsymbol{x})$ could be approximated by discriminating $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^n$ from $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^n$ using a *logistic regression* classifier, i.e., for a parametric function $r(\boldsymbol{x};\boldsymbol{\theta})$ such that

$$r(\boldsymbol{x};\boldsymbol{\theta}) \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{X}, \ \forall \boldsymbol{\theta} \in \Theta, \tag{4}$$

the conditional probabilities $q^*(y = \text{'nu'}|\boldsymbol{x})$ and $q^*(y = \text{'de'}|\boldsymbol{x})$ are modeled by

$$q(y = \text{'nu'}|\boldsymbol{x};\boldsymbol{\theta}) = \frac{r(\boldsymbol{x};\boldsymbol{\theta})}{1 + r(\boldsymbol{x};\boldsymbol{\theta})},$$
$$q(y = \text{'de'}|\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{1 + r(\boldsymbol{x};\boldsymbol{\theta})}.$$

Then the maximum likelihood estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{B}}$ is computed from $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^n$ and $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^n$:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{B}} := \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \left[\sum_{i=1}^n \log \frac{r(\boldsymbol{x}_i^{\mathrm{nu}};\boldsymbol{\theta})}{1 + r(\boldsymbol{x}_i^{\mathrm{nu}};\boldsymbol{\theta})} + \sum_{j=1}^n \log \frac{1}{1 + r(\boldsymbol{x}_j^{\mathrm{de}};\boldsymbol{\theta})} \right]. \tag{5}$$

Note that the maximum likelihood estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{B}}$ minimizes the empirical Kullback-Leibler divergences from the true density $q^*(\boldsymbol{x}, y)$ to its estimator $q(y|\boldsymbol{x};\boldsymbol{\theta})q^*(\boldsymbol{x})$:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{B}} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta} \left[\frac{1}{2n} \sum_{i=1}^n \log \frac{q^*(\boldsymbol{x}_i^{\mathrm{nu}}, y = \text{'nu'})}{q(y = \text{'nu'}|\boldsymbol{x}_i^{\mathrm{nu}};\boldsymbol{\theta})q^*(\boldsymbol{x}_i^{\mathrm{nu}})} + \frac{1}{2n} \sum_{j=1}^n \log \frac{q^*(\boldsymbol{x}_j^{\mathrm{de}}, y = \text{'de'})}{q(y = \text{'de'}|\boldsymbol{x}_j^{\mathrm{de}};\boldsymbol{\theta})q^*(\boldsymbol{x}_j^{\mathrm{de}})} \right].$$

Finally, a density ratio estimator is constructed by taking the ratio of $q(y = \text{'nu'}|\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{B}})$ and $q(y = \text{'de'}|\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{B}})$ with proper normalization:

$$\begin{aligned}\widehat{r}_{\mathrm{B}}(\boldsymbol{x}) &:= \frac{q(y = \text{'nu'}|\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{B}})}{q(y = \text{'de'}|\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{B}})} \left(\frac{1}{n} \sum_{j=1}^n \frac{q(y = \text{'nu'}|\boldsymbol{x}_j^{\mathrm{de}};\widehat{\boldsymbol{\theta}}_{\mathrm{B}})}{q(y = \text{'de'}|\boldsymbol{x}_j^{\mathrm{de}};\widehat{\boldsymbol{\theta}}_{\mathrm{B}})} \right)^{-1} \\ &= r(\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{B}}) \left(\frac{1}{n} \sum_{j=1}^n r(\boldsymbol{x}_j^{\mathrm{de}};\widehat{\boldsymbol{\theta}}_{\mathrm{B}}) \right)^{-1}.\end{aligned}$$

2.5 Method (C): Direct Density Ratio Estimation by Empirical Unnormalized Kullback-Leibler Divergence Minimization

For the density ratio function $r^*(\boldsymbol{x})$, a parametric model $r(\boldsymbol{x};\boldsymbol{\theta})$ such that Eq.(4) is fulfilled is prepared. Then the following estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{C}}$ is computed from $\{\boldsymbol{x}_i^{\mathrm{nu}}\}_{i=1}^n$ and $\{\boldsymbol{x}_j^{\mathrm{de}}\}_{j=1}^n$:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{C}} := \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \left[\sum_{i=1}^n \log r(\boldsymbol{x}_i^{\mathrm{nu}};\boldsymbol{\theta}) - \sum_{j=1}^n r(\boldsymbol{x}_j^{\mathrm{de}};\boldsymbol{\theta}) \right]. \tag{6}$$

Note that $\widehat{\boldsymbol{\theta}}_{\mathrm{C}}$ minimizes the empirical unnormalized Kullback-Leibler divergence from the true density $p_{\mathrm{nu}}^*(\boldsymbol{x})$ to its estimator $\widehat{r}(\boldsymbol{x})p_{\mathrm{de}}^*(\boldsymbol{x})$:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{C}} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta} \left[\frac{1}{n} \sum_{i=1}^n \log \frac{p_{\mathrm{nu}}^*(\boldsymbol{x}_i^{\mathrm{nu}})}{\widehat{r}(\boldsymbol{x}_i^{\mathrm{nu}})p_{\mathrm{de}}^*(\boldsymbol{x}_i^{\mathrm{nu}})} - 1 + \frac{1}{n} \sum_{j=1}^n \widehat{r}(\boldsymbol{x}_j^{\mathrm{de}}) \right].$$

Finally, a density ratio estimator is obtained by

$$\widehat{r}_{\mathrm{C}}(\boldsymbol{x}) := r(\boldsymbol{x}; \widehat{\boldsymbol{\theta}}_{\mathrm{C}}) \left(\frac{1}{n} \sum_{j=1}^{n} r(\boldsymbol{x}_j^{\mathrm{de}}; \widehat{\boldsymbol{\theta}}_{\mathrm{C}}) \right)^{-1}.$$

3 Exponential Models

In our theoretical analysis, we employ the *exponential model*, which is explained in this section.

3.1 Exponential Models for Densities and Ratios

We use the following exponential model for the densities $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$.

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left\{ \boldsymbol{\theta}^\top \boldsymbol{\xi}(\boldsymbol{x}) - \varphi(\boldsymbol{\theta}) \right\}, \quad \theta \in \Theta, \tag{7}$$

where $h(\boldsymbol{x})$ is a *base measure*, $\boldsymbol{\xi}(\boldsymbol{x})$ is a *sufficient statistic*, $\varphi(\boldsymbol{\theta})$ is a *normalization factor*, and $^\top$ denotes the transpose of a vector [13]. The exponential model includes various popular models as special cases, e.g., the normal, exponential, gamma, chi-square, and beta distributions.

Correspondingly, we use the following exponential model for the ratio $r^*(\boldsymbol{x})$.

$$r(\boldsymbol{x}; \boldsymbol{\theta}, \theta_0) = \exp\left\{ \theta_0 + \boldsymbol{\theta}^\top \boldsymbol{\xi}(\boldsymbol{x}) \right\}, \quad \theta \in \Theta, \quad \theta_0 \in \mathbb{R}. \tag{8}$$

3.2 Method (A)

For the exponential model (7), the maximum likelihood estimators $\widehat{\boldsymbol{\theta}}_{\mathrm{nu}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{de}}$ are given by

$$\widehat{\boldsymbol{\theta}}_{\mathrm{nu}} = \operatorname*{argmax}_{\boldsymbol{\theta}_{\mathrm{nu}} \in \Theta} \left[\sum_{i=1}^{n} \boldsymbol{\theta}^\top \boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}}) - n\varphi(\boldsymbol{\theta}) \right],$$
$$\widehat{\boldsymbol{\theta}}_{\mathrm{de}} = \operatorname*{argmax}_{\boldsymbol{\theta}_{\mathrm{de}} \in \Theta} \left[\sum_{j=1}^{n} \boldsymbol{\theta}^\top \boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}}) - n\varphi(\boldsymbol{\theta}) \right],$$

where irrelevant constants are ignored. The density ratio estimator $\widehat{r}_{\mathrm{A}}(\boldsymbol{x})$ for the exponential density model is expressed as

$$\widehat{r}_{\mathrm{A}}(\boldsymbol{x}) = \exp\left\{ \widehat{\boldsymbol{\theta}}_{\mathrm{A}}^\top \boldsymbol{\xi}(\boldsymbol{x}) \right\} \left(\frac{1}{n} \sum_{j=1}^{n} \exp\left\{ \widehat{\boldsymbol{\theta}}_{\mathrm{A}}^\top \boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}}) \right\} \right)^{-1},$$

where

$$\widehat{\boldsymbol{\theta}}_{\mathrm{A}} := \widehat{\boldsymbol{\theta}}_{\mathrm{nu}} - \widehat{\boldsymbol{\theta}}_{\mathrm{de}}.$$

One can use the other estimator such as

$$\widetilde{r}_{\mathrm{A}}(\boldsymbol{x}) = \exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{A}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}) - \varphi(\widehat{\boldsymbol{\theta}}_{\mathrm{nu}}) + \varphi(\widehat{\boldsymbol{\theta}}_{\mathrm{de}})\right\}$$

instead of $\widehat{r}_{\mathrm{A}}(\boldsymbol{x})$. We compare $\widehat{r}_{\mathrm{A}}(\boldsymbol{x})$ to Method (B) and Method (C), since the same normalization factor as $\widehat{r}_{\mathrm{A}}(\boldsymbol{x})$ appears in the other methods as shown below. This fact facilitates the theoretical analysis.

3.3 Method (B)

For the exponential model (8), the optimization problem (5) is expressed as

$$\begin{aligned}(\widehat{\boldsymbol{\theta}}_{\mathrm{B}}, \widehat{\theta}_{\mathrm{B},0}) &= \underset{(\boldsymbol{\theta},\theta_0)\in\Theta\times\mathbb{R}}{\operatorname{argmax}} \left[\sum_{i=1}^{n}\log\frac{r(\boldsymbol{x}_i^{\mathrm{nu}};\boldsymbol{\theta},\theta_0)}{1+r(\boldsymbol{x}_i^{\mathrm{nu}};\boldsymbol{\theta},\theta_0)} + \sum_{j=1}^{n}\log\frac{1}{1+r(\boldsymbol{x}_j^{\mathrm{de}};\boldsymbol{\theta},\theta_0)}\right] \\ &= \underset{(\boldsymbol{\theta},\theta_0)\in\Theta\times\mathbb{R}}{\operatorname{argmax}} \left[\sum_{i=1}^{n}\log\frac{\exp\left\{\theta_0+\boldsymbol{\theta}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}})\right\}}{1+\exp\left\{\theta_0+\boldsymbol{\theta}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}})\right\}} \right. \\ &\qquad \left. + \sum_{j=1}^{n}\log\frac{1}{1+\exp\left\{\theta_0+\boldsymbol{\theta}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}}\right].\end{aligned}$$

The density ratio estimator $\widehat{r}_{\mathrm{B}}(\boldsymbol{x})$ for the exponential ratio model is expressed as

$$\widehat{r}_{\mathrm{B}}(\boldsymbol{x}) = \exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{B}}^{\top}\boldsymbol{\xi}(\boldsymbol{x})\right\}\left(\frac{1}{n}\sum_{j=1}^{n}\exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{B}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}\right)^{-1}.$$

3.4 Method (C)

For the exponential model (8), the optimization problem (6) is expressed as

$$\begin{aligned}(\widehat{\boldsymbol{\theta}}_{\mathrm{C}}, \widehat{\theta}_{\mathrm{C},0}) &= \underset{(\boldsymbol{\theta},\theta_0)\in\Theta\times\mathbb{R}}{\operatorname{argmax}} \left[\frac{1}{n}\sum_{i=1}^{n}\log r(\boldsymbol{x}_i^{\mathrm{nu}};\boldsymbol{\theta},\theta_0) - \frac{1}{n}\sum_{j=1}^{n} r(\boldsymbol{x}_i^{\mathrm{de}};\boldsymbol{\theta},\theta_0)\right] \\ &= \underset{(\boldsymbol{\theta},\theta_0)\in\Theta\times\mathbb{R}}{\operatorname{argmax}} \left[\frac{1}{n}\sum_{i=1}^{n}(\theta_0+\boldsymbol{\theta}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}})) - \frac{1}{n}\sum_{j=1}^{n}\exp\left\{\theta_0+\boldsymbol{\theta}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}\right]. \qquad (9)\end{aligned}$$

The density ratio estimator $\widehat{r}_{\mathrm{C}}(\boldsymbol{x})$ for the exponential ratio model is expressed as

$$\widehat{r}_{\mathrm{C}}(\boldsymbol{x}) = \exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^{\top}\boldsymbol{\xi}(\boldsymbol{x})\right\}\left(\frac{1}{n}\sum_{j=1}^{n}\exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}\right)^{-1}.$$

4 Accuracy Analysis for Correctly Specified Exponential Models

In this section, we theoretically analyze the accuracy of the above three density ratio estimators under the assumption that the true densities $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$ both belong to the exponential family, i.e., there exist $\boldsymbol{\theta}^*_{\mathrm{nu}} \in \Theta$ and $\boldsymbol{\theta}^*_{\mathrm{de}} \in \Theta$ such that

$$p^*_{\mathrm{nu}}(\boldsymbol{x}) = p(\boldsymbol{x}; \boldsymbol{\theta}^*_{\mathrm{nu}}),$$
$$p^*_{\mathrm{de}}(\boldsymbol{x}) = p(\boldsymbol{x}; \boldsymbol{\theta}^*_{\mathrm{de}}).$$

Since the ratio of two exponential densities also belongs to the exponential model, the above assumption implies that there exist $\boldsymbol{\theta}^* \in \Theta$ and $\theta^*_0 \in \mathbb{R}$ such that

$$r^*(\boldsymbol{x}) = r(\boldsymbol{x}; \boldsymbol{\theta}^*, \theta^*_0). \tag{10}$$

It is straightforward to extended the results in this section to general parametric models, since we focus on the first-order asymptotics of the estimators. An arbitrary parametric model $p(\boldsymbol{x}; \boldsymbol{\theta})$ has the same first-order asymptotics as the exponential model of the form

$$p_{\exp}(\boldsymbol{x}; \boldsymbol{\theta}) \propto \exp\{\log p(\boldsymbol{x}; \boldsymbol{\theta}^*) + (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \nabla \log p(\boldsymbol{x}; \boldsymbol{\theta}^*)\}$$

around the parameter $\boldsymbol{\theta}^*$. Thus the same theoretical property holds.

First, we analyze the asymptotic behavior of $J(\widehat{r}_{\mathrm{A}})$. Then we have the following lemma (proofs of all lemmas, theorems, and corollaries are provided in Appendix).

Lemma 1 *$J(\widehat{r}_{\mathrm{A}})$ can be asymptotically expressed as*

$$J(\widehat{r}_{\mathrm{A}}) = \frac{1}{2n}\left[\dim \Theta + \mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{de}})^{-1}\right) + \mathrm{PE}(p^*_{\mathrm{de}} \| p^*_{\mathrm{nu}})\right] + \mathcal{O}(n^{-3/2}),$$

where $\mathcal{O}(\cdot)$ denotes the asymptotic order. $\boldsymbol{F}(\boldsymbol{\theta})$ denote the Fisher information matrix of the exponential model $p(\boldsymbol{x}; \boldsymbol{\theta})$:

$$\boldsymbol{F}(\boldsymbol{\theta}) := \int \nabla \log p(\boldsymbol{x}; \boldsymbol{\theta}) \nabla \log p(\boldsymbol{x}; \boldsymbol{\theta})^\top p(\boldsymbol{x}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{x},$$

where ∇ denotes the partial differential operator with respect to $\boldsymbol{\theta}$. $\mathrm{PE}(p\|q)$ denotes the Pearson divergence of two densities p and q defined as

$$\mathrm{PE}(p\|q) := \frac{1}{2}\int \frac{(p(\boldsymbol{x}) - q(\boldsymbol{x}))^2}{p(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}. \tag{11}$$

Next, we investigate the asymptotic behavior of $J(\widehat{r}_{\mathrm{B}})$ and $J(\widehat{r}_{\mathrm{C}})$. Let y be the selector variable taking 'nu' or 'de' as defined in Section 2.4. The statistical model of the joint probability for $\boldsymbol{z} = (\boldsymbol{x}, y)$ is defined as

$$q(\boldsymbol{z}; \boldsymbol{\theta}, \theta_0) = q(y|\boldsymbol{x}; \boldsymbol{\theta}, \theta_0) \times \frac{p^*_{\mathrm{nu}}(\boldsymbol{x}) + p^*_{\mathrm{de}}(\boldsymbol{x})}{2}, \tag{12}$$

where $q(y|\boldsymbol{x};\boldsymbol{\theta},\theta_0)$ is the conditional probability of y such that

$$\begin{aligned} q(y=\text{'nu'}|\boldsymbol{x};\boldsymbol{\theta},\theta_0) &= \frac{r(\boldsymbol{x};\boldsymbol{\theta},\theta_0)}{1+r(\boldsymbol{x};\boldsymbol{\theta},\theta_0)} \\ &= \frac{\exp\left\{\theta_0+\boldsymbol{\theta}^\top\boldsymbol{\xi}(\boldsymbol{x})\right\}}{1+\exp\left\{\theta_0+\boldsymbol{\theta}^\top\boldsymbol{\xi}(\boldsymbol{x})\right\}}, \\ q(y=\text{'de'}|\boldsymbol{x};\boldsymbol{\theta},\theta_0) &= \frac{1}{1+r(\boldsymbol{x};\boldsymbol{\theta},\theta_0)} \\ &= \frac{1}{1+\exp\left\{\theta_0+\boldsymbol{\theta}^\top\boldsymbol{\xi}(\boldsymbol{x})\right\}}. \end{aligned}$$

The Fisher information matrix of the model (12) is denoted as

$$\widetilde{\boldsymbol{F}}(\boldsymbol{\theta},\theta_0)\in\mathbb{R}^{(\dim\Theta+1)\times(\dim\Theta+1)}.$$

The submatrix of $\widetilde{\boldsymbol{F}}(\boldsymbol{\theta},\theta_0)$ formed by the first $(\dim\Theta)$ rows and the first $(\dim\Theta)$ columns is defined as

$$\int\nabla\log q(\boldsymbol{z};\boldsymbol{\theta},\theta_0)\nabla\log q(\boldsymbol{z};\boldsymbol{\theta},\theta_0)^\top q(\boldsymbol{z};\boldsymbol{\theta},\theta_0)\mathrm{d}\boldsymbol{z}.$$

The inverse matrix of $\widetilde{\boldsymbol{F}}(\boldsymbol{\theta},\theta_0)$ is expressed as

$$\widetilde{\boldsymbol{F}}(\boldsymbol{\theta},\theta_0)^{-1}=\begin{pmatrix}\boldsymbol{H}_{11}(\boldsymbol{\theta},\theta_0) & \boldsymbol{h}_{12}(\boldsymbol{\theta},\theta_0)\\ \boldsymbol{h}_{12}(\boldsymbol{\theta},\theta_0)^\top & h_{22}(\boldsymbol{\theta},\theta_0)\end{pmatrix}, \tag{13}$$

where $\boldsymbol{H}_{11}(\boldsymbol{\theta},\theta_0)$ is a $(\dim\Theta)\times(\dim\Theta)$ matrix. Then we have the following lemmas.

Lemma 2 *$J(\widehat{r}_{\mathrm{B}})$ can be asymptotically expressed as*

$$J(\widehat{r}_{\mathrm{B}})=\frac{1}{2n}\left[\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\boldsymbol{H}_{11}(\boldsymbol{\theta}^*,\theta_0^*)\right)+\mathrm{PE}(p^*_{\mathrm{de}}\|p^*_{\mathrm{nu}})\right]+\mathcal{O}(n^{-3/2}),$$

where $(\boldsymbol{\theta}^*,\theta_0^*)$ is defined in Eq.(10).

Lemma 3 *$J(\widehat{r}_{\mathrm{C}})$ can be asymptotically expressed as*

$$J(\widehat{r}_{\mathrm{C}})=\frac{1}{2n}\left[\dim\Theta+\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})^{-1}\boldsymbol{G}\right)+\mathrm{PE}(p^*_{\mathrm{de}}\|p^*_{\mathrm{nu}})\right]+\mathcal{O}(n^{-3/2}),$$

where

$$\boldsymbol{G}:=\int r^*(\boldsymbol{x})(\boldsymbol{\xi}(\boldsymbol{x})-\boldsymbol{\eta}_{\mathrm{nu}})(\boldsymbol{\xi}(\boldsymbol{x})-\boldsymbol{\eta}_{\mathrm{nu}})^\top p^*_{\mathrm{nu}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}.$$

Based on the above lemmas, we compare the accuracy of the three methods. For the accuracy of (A) and (B), we have the following theorem.

Theorem 4 *Asymptotically, the inequality*

$$J(\widehat{r}_{\mathrm{A}}) \leq J(\widehat{r}_{\mathrm{B}})$$

holds.

Thus the method (A) is more accurate than the method (B) in terms of the expected unnormalized Kullback-Leibler divergence (3). Theorem 4 may be regarded as an extension of the result for binary classification [5]: estimating data generating Gaussian densities by maximum likelihood estimation has high statistical efficiency than logistic regression in the sense of classification error rate.

Next, we compare the accuracy of (B) and (C).

Theorem 5 *Asymptotically, the inequality*

$$J(\widehat{r}_{\mathrm{B}}) \leq J(\widehat{r}_{\mathrm{C}})$$

holds.

Thus the method (B) is more accurate than the method (C) in terms of the expected unnormalized Kullback-Leibler divergence (3). This inequality is a direct consequence of the paper by Qin [16]. In that paper, it was shown that the method (B) has the smallest asymptotic variance in a class of semi-parametric estimators. It is easy to see the method (C) is included in the class.

Finally, we compare the accuracy of (A) and (C). From Theorem 4 and Theorem 5, we immediately have the following corollary.

Corollary 6 *The inequality*

$$J(\widehat{r}_{\mathrm{A}}) \leq J(\widehat{r}_{\mathrm{C}})$$

holds.

It was advocated that one should avoid solving more difficult intermediate problems when solving a target problem [33]. This statement is sometimes referred to as "Vapnik's principle", and the *support vector machine* [4] would be a successful example of this principle—instead of estimating a data generation model, it directly models the decision boundary which is sufficient for pattern recognition.

If we followed Vapnik's principle, directly estimating the ratio $r^*(\boldsymbol{x})$ would be more promising than estimating the two densities $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$ since knowing $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$ implies knowing $r^*(\boldsymbol{x})$ but not vice versa; indeed, $r^*(\boldsymbol{x})$ cannot be uniquely decomposed into $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$. Thus Corollary 6 is at a glance counter-intuitive. However, Corollary 6 would be reasonable since the method (C) does not make use of the knowledge that *each* density is exponential, but only the knowledge that the ratio is exponential. Thus the method (A) can utilize the a priori model information more effectively. Thanks to the additional knowledge that the both densities belong to the exponential model, the intermediate problems (i.e., density estimation) were actually made easier in terms of Vapnik's principle.

5 Accuracy Analysis for Misspecified Exponential Models

In this section, we theoretically analyze the approximation error of the three density ratio estimators for misspecified exponential models, i.e., the true densities and ratio are not necessarily included in the exponential models. The unnormalized Kullback-Leibler divergence is employed to measure the approximation error.

First, we study the convergence of the method (A). Let $\overline{p}_{\mathrm{nu}}(\boldsymbol{x})$ and $\overline{p}_{\mathrm{de}}(\boldsymbol{x})$ be the projections of the true densities $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$ onto the model $p(\boldsymbol{x};\boldsymbol{\theta})$ in terms of the Kullback-Leibler divergence (2):

$$\overline{p}_{\mathrm{nu}}(\boldsymbol{x}) := p(\boldsymbol{x};\overline{\boldsymbol{\theta}}_{\mathrm{nu}}),$$
$$\overline{p}_{\mathrm{de}}(\boldsymbol{x}) := p(\boldsymbol{x};\overline{\boldsymbol{\theta}}_{\mathrm{de}}),$$

where

$$\overline{\boldsymbol{\theta}}_{\mathrm{nu}} := \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta}\left[\int p^*_{\mathrm{nu}}(\boldsymbol{x})\log\frac{p^*_{\mathrm{nu}}(\boldsymbol{x})}{p(\boldsymbol{x};\boldsymbol{\theta})}\mathrm{d}\boldsymbol{x}\right],$$
$$\overline{\boldsymbol{\theta}}_{\mathrm{de}} := \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta}\left[\int p^*_{\mathrm{de}}(\boldsymbol{x})\log\frac{p^*_{\mathrm{de}}(\boldsymbol{x})}{p(\boldsymbol{x};\boldsymbol{\theta})}\mathrm{d}\boldsymbol{x}\right].$$

This means that $\overline{p}_{\mathrm{nu}}(\boldsymbol{x})$ and $\overline{p}_{\mathrm{de}}(\boldsymbol{x})$ are the optimal approximations to $p^*_{\mathrm{nu}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})$ in the model $p(\boldsymbol{x};\boldsymbol{\theta})$ in terms of the Kullback-Leibler divergence. Let

$$\overline{r}_{\mathrm{A}}(\boldsymbol{x}) := \frac{\overline{p}_{\mathrm{nu}}(\boldsymbol{x})}{\overline{p}_{\mathrm{de}}(\boldsymbol{x})}.$$

Since the ratio of two exponential densities also belongs to the exponential model, there exists $\overline{\boldsymbol{\theta}}_{\mathrm{A}} \in \Theta$ such that

$$\overline{r}_{\mathrm{A}}(\boldsymbol{x}) = r(\boldsymbol{x};\overline{\boldsymbol{\theta}}_{\mathrm{A}},\bar{\theta}_{\mathrm{A},0}).$$

Then we have the following lemma.

Lemma 7 *$\widehat{r}_{\mathrm{A}}$ converges in probability to $\overline{r}_{\mathrm{A}}$ as $n\to\infty$.*

Next, we investigate the convergence of the method (B). Let $q^*(\boldsymbol{x},y)$ be the joint probability defined as

$$q^*(\boldsymbol{x},y) = q^*(y|\boldsymbol{x}) \times \frac{p^*_{\mathrm{nu}}(\boldsymbol{x}) + p^*_{\mathrm{de}}(\boldsymbol{x})}{2}, \tag{14}$$

where $q^*(y|\boldsymbol{x})$ is the conditional probability of y such that

$$q^*(y = \text{`nu'}|\boldsymbol{x}) = \frac{r^*(\boldsymbol{x})}{1+r^*(\boldsymbol{x})},$$
$$q^*(y = \text{`de'}|\boldsymbol{x}) = \frac{1}{1+r^*(\boldsymbol{x})}.$$

The model (12) is used to estimate $q^*(\boldsymbol{x}, y)$, and let $\overline{q}(\boldsymbol{x}, y)$ be the projection of the true density $q^*(\boldsymbol{x}, y)$ onto the model (12) in terms of the Kullback-Leibler divergence (2):

$$\overline{q}(\boldsymbol{x}, y) := q(\boldsymbol{x}, y; \overline{\boldsymbol{\theta}}_{\mathrm{B}}, \bar{\theta}_{\mathrm{B},0}), \tag{15}$$

where

$$(\overline{\boldsymbol{\theta}}_{\mathrm{B}}, \bar{\theta}_{\mathrm{B},0}) := \operatorname*{argmin}_{(\boldsymbol{\theta},\theta_0)\in\Theta\times\mathbb{R}} \left[\int \sum_{y\in\{\text{`nu'},\text{`de'}\}} q^*(\boldsymbol{x}, y) \log \frac{q^*(y|\boldsymbol{x})}{q(y|\boldsymbol{x}; \boldsymbol{\theta}, \theta_0)} \mathrm{d}\boldsymbol{x} \right].$$

This means that $\overline{q}(\boldsymbol{x}, y)$ is the optimal approximation to $q^*(\boldsymbol{x}, y)$ in the model

$$q(y|\boldsymbol{x}; \boldsymbol{\theta}, \theta_0) \frac{p^*_{\mathrm{nu}}(\boldsymbol{x}) + p^*_{\mathrm{de}}(\boldsymbol{x})}{2}$$

in terms of the Kullback-Leibler divergence. Let

$$\overline{r}_{\mathrm{B}}(\boldsymbol{x}) := r(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_{\mathrm{B}}, \bar{\theta}_{\mathrm{B},0}).$$

Then we have the following lemma.

Lemma 8 *$\widehat{r}_{\mathrm{B}}$ converges in probability to $\overline{r}_{\mathrm{B}}$ as $n \to \infty$.*

Finally, we study the convergence of the method (C). Suppose that the model $r(\boldsymbol{x}; \boldsymbol{\theta}, \theta_0)$ in Eq.(8) is employed. Let $\overline{r}_{\mathrm{C}}(\boldsymbol{x})$ be the projection of the true ratio function $r^*(\boldsymbol{x})$ onto the model $r(\boldsymbol{x}; \boldsymbol{\theta}, \theta_0)$ in terms of the unnormalized Kullback-Leibler divergence (1):

$$\overline{r}_{\mathrm{C}}(\boldsymbol{x}) := r(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_{\mathrm{C}}, \bar{\theta}_{\mathrm{C},0}),$$

where

$$(\overline{\boldsymbol{\theta}}_{\mathrm{C}}, \bar{\theta}_{\mathrm{C},0}) := \operatorname*{argmin}_{(\boldsymbol{\theta},\theta_0)\in\Theta\times\mathbb{R}} \left[\int p^*_{\mathrm{nu}}(\boldsymbol{x}) \log \frac{r^*(\boldsymbol{x})}{r(\boldsymbol{x}; \boldsymbol{\theta}, \theta_0)} \mathrm{d}\boldsymbol{x} - 1 + \int p^*_{\mathrm{de}}(\boldsymbol{x}) r(\boldsymbol{x}; \boldsymbol{\theta}, \theta_0) \mathrm{d}\boldsymbol{x} \right]. \tag{16}$$

This means that $\overline{r}_{\mathrm{C}}(\boldsymbol{x})$ is the optimal approximation to $r^*(\boldsymbol{x})$ in the model $r(\boldsymbol{x}; \boldsymbol{\theta})$ in terms of the unnormalized Kullback-Leibler divergence. Then we have the following lemma.

Lemma 9 *$\widehat{r}_{\mathrm{C}}$ converges in probability to $\overline{r}_{\mathrm{C}}$ as $n \to \infty$.*

Based on the above lemmas, we investigate the relation among the three methods. Lemma 9 implies that the method (C) is consistent to the optimal approximation $\overline{r}_{\mathrm{C}}$. However, as we will show below, the methods (A) and (B) are not consistent to the optimal approximation $\overline{r}_{\mathrm{C}}$ in general. Let us measure the deviation of a density ratio function $\overline{r}'$ from $\overline{r}$ by

$$\mathrm{D}(\overline{r}', \overline{r}) := \int p^*_{\mathrm{de}}(\boldsymbol{x}) \left(\overline{r}'(\boldsymbol{x}) - \overline{r}(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x}.$$

Then we have the following theorem.

Theorem 10 *The inequalities*

$$\mathrm{D}(\overline{r}_{\mathrm{A}}, \overline{r}_{\mathrm{C}}) \geq \left| \int p^*_{\mathrm{de}}(\boldsymbol{x})\, \overline{r}_{\mathrm{A}}(\boldsymbol{x}) d\boldsymbol{x} - 1 \right|^2,$$
$$\mathrm{D}(\overline{r}_{\mathrm{B}}, \overline{r}_{\mathrm{C}}) \geq \left| \int p^*_{\mathrm{de}}(\boldsymbol{x})\, \overline{r}_{\mathrm{B}}(\boldsymbol{x}) d\boldsymbol{x} - 1 \right|^2$$

hold. More generally, for any $\overline{r}$ *in the exponential model,*

$$\mathrm{D}(\overline{r}, \overline{r}_{\mathrm{C}}) \geq \left| \int p^*_{\mathrm{de}}(\boldsymbol{x})\, \overline{r}(\boldsymbol{x}) d\boldsymbol{x} - 1 \right|^2 \tag{17}$$

holds.

When the model is misspecified, $p^*_{\mathrm{de}}(\boldsymbol{x})\, \overline{r}_{\mathrm{A}}(\boldsymbol{x})$ and $p^*_{\mathrm{de}}(\boldsymbol{x})\, \overline{r}_{\mathrm{B}}(\boldsymbol{x})$ are not probability densities in general. Then Theorem 10 implies that the method (A) and the method (B) are not consistent to the optimal approximation $\overline{r}_{\mathrm{C}}$.

Since model misspecification would be a usual situation in practice, the method (C) is the most promising approach in density ratio estimation.

Finally, for the consistency of the method (A), we also have the following additional result.

Corollary 11 *If* $p^*_{\mathrm{de}}(\boldsymbol{x})$ *belongs to the exponential model* (7)*, i.e., there exists* $\overline{\boldsymbol{\theta}}_{\mathrm{de}} \in \Theta$ *such that*

$$p^*_{\mathrm{de}}(\boldsymbol{x}) = p(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_{\mathrm{de}}),$$

then

$$\overline{r}_{\mathrm{A}} = \overline{r}_{\mathrm{C}}$$

holds even when $p^*_{\mathrm{nu}}(\boldsymbol{x})$ *does not belong to the exponential model* (7)*.*

This corollary means that, as long as $p^*_{\mathrm{de}}(\boldsymbol{x})$ is correctly specified, the method (A) is still consistent.

6 Conclusions

In this paper, we theoretically investigated the accuracy of three density ratio estimation approaches: (A) density ratio estimation by separate maximum likelihood density estimation, (B) density ratio estimation by logistic regression, and (C) direct density ratio estimation by empirical Kullback-Leibler divergence minimization. Intuitively, the method (C) seems to be better than the other approaches due to "Vapnik's principle"—one should not solve more difficult intermediate problems (density estimation in the current context) when solving a target problem (density ratio estimation in the current context).

However, as we proved in Section 4, the method (A) is more accurate than the other approaches when the numerator and denominator densities are known to be members of the exponential family. This result is at first sight counter-intuitive, but it would be reasonable because the methods (B) and (C) do not make use of the knowledge that *each* density is exponential, but only the knowledge that their ratio is exponential. Thus the method (A) can utilize the a priori model information more effectively than the other methods. We note that this result is not contradictory to Vapnik's principle since the additional knowledge that the densities belong to the exponential model is utilized to make the intermediate problems (density estimation) substantially easier.

On the other hand, once the correct model assumption is not fulfilled, the method (C) was shown to be consistent to the optimal approximation in the model, while the methods (A) and (B) are not consistent in general (see Section 5). The fact that the direct method outperforms the other approaches in the absence of the additional knowledge would follow Vapnik's principle.

It seems to be a common phenomenon in various situations that a method which works optimally for correctly specified models performs poorly for misspecified models and conversely a method which works well for misspecified models performs poorly for correctly specified models. For example, in active learning (or the experiment design), the traditional variance-only approach works optimally for correctly specified models [6]. However, it was shown that the traditional method works poorly once the correct model assumption is slightly violated [20]. To cope with this problem, various active learning methods which do not require the correct model assumption have been developed and shown to work better than the traditional method for misspecified models [34, 11, 20, 9, 24]. However, these methods cannot outperform the traditional method when the model is correctly specified. Thus the performance loss for correctly specified models would be the price one has to pay for acquiring robustness against model misspecification.

Model misspecification would almost always occur in practice, so developing methods for misspecified models is crucial. Based on these observations, we conclude that the direct density ratio approach (C) is the most promising density ratio estimation method.

Acknowledgments

MS was supported by SCAT, AOARD, and the JST PRESTO program.

A Asymptotic Expansion of Measure of Accuracy

First, we show some fundamental results used for proving Lemma 1, Lemma 2, and Lemma 3.

Using the Taylor expansion

$$\log(1+t) = t - \frac{t^2}{2} + \mathcal{O}(t^3),$$

we have the following expansion:

$$\begin{aligned}\log \frac{p^*_{\mathrm{nu}}(\boldsymbol{x})}{\widehat{r}(\boldsymbol{x})p^*_{\mathrm{de}}(\boldsymbol{x})} &= \log \frac{r^*(\boldsymbol{x})}{\widehat{r}(\boldsymbol{x})} \\ &= -\log \frac{\widehat{r}(\boldsymbol{x})}{r^*(\boldsymbol{x})} \\ &= -\left(\frac{\widehat{r}(\boldsymbol{x})}{r^*(\boldsymbol{x})} - 1\right) + \frac{1}{2}\left(\frac{\widehat{r}(\boldsymbol{x})}{r^*(\boldsymbol{x})} - 1\right)^2 + \mathcal{O}_p\left(\left|\frac{\widehat{r}(\boldsymbol{x})}{r^*(\boldsymbol{x})} - 1\right|^3\right),\end{aligned}$$

where $\mathcal{O}_p(\cdot)$ denotes the stochastic order. Substituting this expansion into the unnormalized Kullback-Leibler divergence $\mathrm{UKL}(p^*_{\mathrm{nu}}\|\widehat{r}\cdot p^*_{\mathrm{de}})$, we obtain

$$\mathrm{UKL}(p^*_{\mathrm{nu}}\|\widehat{r}\cdot p^*_{\mathrm{de}}) = \mathrm{PE}(p^*_{\mathrm{nu}}\|\widehat{r}\cdot p^*_{\mathrm{de}}) + \mathcal{O}(\|\widehat{r}/r^* - 1\|^3), \tag{18}$$

where 'PE' denotes the Pearson divergence defined by Eq.(11) and $\|\widehat{r}/r^* - 1\|$ is defined as

$$\|\widehat{r}/r^* - 1\| := \left(\int p^*_{\mathrm{nu}}(\boldsymbol{x})|\widehat{r}(\boldsymbol{x})/r^*(\boldsymbol{x}) - 1|^2 \mathrm{d}\boldsymbol{x}\right)^{1/2}.$$

Under a regularity condition of asymptotic statistics, the expectation $\mathbb{E}\left[\|\widehat{r}/r^* - 1\|^3\right]$ is of order $\mathcal{O}(n^{-3/2})$:

$$\mathbb{E}\left[\|\widehat{r}/r^* - 1\|^3\right] = \mathcal{O}(n^{-3/2}).$$

See Theorem 5.23 in [32] for the details of the regularity condition on general M-estimators. Hence, the measure of accuracy $J(\widehat{r})$ can be represented as

$$J(\widehat{r}) = \mathbb{E}\left[\mathrm{PE}(p^*_{\mathrm{nu}}\|\widehat{r}\cdot p^*_{\mathrm{de}})\right] + \mathcal{O}(n^{-3/2}). \tag{19}$$

Then we have the following lemma.

Lemma 12 (Asymptotics of measure of accuracy) *Let $\widehat{\boldsymbol{\theta}}$ be an estimator of the parameter $\boldsymbol{\theta}^*$ in r^*, and $\widehat{r}(\boldsymbol{x})$ be the estimator defined as*

$$\widehat{r}(\boldsymbol{x}) := \exp\left\{\widehat{\boldsymbol{\theta}}^{\top}\boldsymbol{\xi}(\boldsymbol{x})\right\}\left(\frac{1}{n}\sum_{j=1}^{n}\exp\left\{\widehat{\boldsymbol{\theta}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}^{\mathrm{de}}_j)\right\}\right)^{-1}.$$

Then, the measure of accuracy of $\widehat{r}$ is asymptotically given as

$$J(\widehat{r}) = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\cdot\mathbb{E}\left[\delta\boldsymbol{\theta}\delta\boldsymbol{\theta}^{\top}\right]\right) + \frac{1}{2n}\mathrm{PE}(p^*_{\mathrm{de}}\|p^*_{\mathrm{nu}}) + \mathcal{O}(n^{-3/2}), \tag{20}$$

where $\delta\boldsymbol{\theta}$ denotes the deviation of $\widehat{\boldsymbol{\theta}}$ from the parameter $\boldsymbol{\theta}^*$:

$$\delta\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*.$$

Proof: The probabilistic order of $\delta\boldsymbol{\theta}$ is $\mathcal{O}_p(n^{-1/2})$. Let $\boldsymbol{\eta}_{\mathrm{nu}}$ be

$$\boldsymbol{\eta}_{\mathrm{nu}} := \int \boldsymbol{\xi}(\boldsymbol{x}) p^*_{\mathrm{nu}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

Using the Taylor expansion

$$\begin{aligned} \exp(t) &= 1 + t + \mathcal{O}(t^2), \\ \log(1+t) &= t + \mathcal{O}(t^2), \end{aligned}$$

we have the following the asymptotic expansion of $\widehat{r}$:

$$\begin{aligned} \log \widehat{r}(\boldsymbol{x}) &= \log \frac{r^*(\boldsymbol{x}) \exp\left\{\delta\boldsymbol{\theta}^\top \boldsymbol{\xi}(\boldsymbol{x})\right\}}{\frac{1}{n}\sum_{j=1}^n r^*(\boldsymbol{x}^{\mathrm{de}}_j) \exp\left\{\delta\boldsymbol{\theta}^\top \boldsymbol{\xi}(\boldsymbol{x}^{\mathrm{de}}_j)\right\}} \\ &= \log r^*(\boldsymbol{x}) + \delta\boldsymbol{\theta}^\top \boldsymbol{\xi}(\boldsymbol{x}) \\ &\quad - \log\left\{1 + \left(\frac{1}{n}\sum_{j=1}^n r^*(\boldsymbol{x}^{\mathrm{de}}_j) - 1\right) + \delta\boldsymbol{\theta}^\top \cdot \frac{1}{n}\sum_{j=1}^n r^*(\boldsymbol{x}^{\mathrm{de}}_j)\boldsymbol{\xi}(\boldsymbol{x}^{\mathrm{de}}_j) + \mathcal{O}_p(n^{-1})\right\} \\ &= \log r^*(\boldsymbol{x}) + \delta\boldsymbol{\theta}^\top(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{nu}}) - \left(\frac{1}{n}\sum_{j=1}^n r^*(\boldsymbol{x}^{\mathrm{de}}_j) - 1\right) + \mathcal{O}_p(n^{-1}). \end{aligned}$$

Therefore, we have

$$\frac{\widehat{r}(\boldsymbol{x})}{r^*(\boldsymbol{x})} - 1 = \delta\boldsymbol{\theta}^\top(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{nu}}) - \left(\frac{1}{n}\sum_{j=1}^n r^*(\boldsymbol{x}^{\mathrm{de}}_j) - 1\right) + \mathcal{O}_p(n^{-1}).$$

Substituting the above expression into the Pearson divergence in Eq.(19), we obtain

$$\begin{aligned} \mathrm{PE}(p^*_{\mathrm{nu}} \| \widehat{r} \cdot p^*_{\mathrm{de}}) &= \frac{1}{2}\int p^*_{\mathrm{nu}}(\boldsymbol{x}) \left(\frac{\widehat{r}(\boldsymbol{x})}{r^*(\boldsymbol{x})} - 1\right)^2 \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\delta\boldsymbol{\theta}\delta\boldsymbol{\theta}^\top\right) + \frac{1}{2}\left(\frac{1}{n}\sum_{j=1}^n (r^*(\boldsymbol{x}^{\mathrm{de}}_j) - 1)\right)^2 + \mathcal{O}_p(n^{-3/2}). \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}\left[\mathrm{PE}(p^*_{\mathrm{nu}} \| \widehat{r} \cdot p^*_{\mathrm{de}})\right] &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}}) \cdot \mathbb{E}\left[\delta\boldsymbol{\theta}\delta\boldsymbol{\theta}^\top\right]\right) + \frac{1}{2n}\int p^*_{\mathrm{de}}(\boldsymbol{x})(r^*(\boldsymbol{x}) - 1)^2 \mathrm{d}\boldsymbol{x} \\ &\quad + \mathcal{O}(n^{-3/2}) \\ &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}}) \cdot \mathbb{E}\left[\delta\boldsymbol{\theta}\delta\boldsymbol{\theta}^\top\right]\right) + \frac{1}{2n}\mathrm{PE}(p^*_{\mathrm{de}} \| p^*_{\mathrm{nu}}) + \mathcal{O}(n^{-3/2}). \end{aligned}$$

Applying Eq (18) to the above equation, we obtain Eq.(20). ■

B Proof of Lemma 1

According to Lemma 12, we need to compute the asymptotic variance of estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{A}}$ in order to compute the measure of accuracy of $\widehat{r}_{\mathrm{A}}$. Based on the standard asymptotic statistics, the asymptotic variance of the maximum likelihood estimator for the exponential family is given as

$$\begin{aligned}\sqrt{n}(\widehat{\boldsymbol{\theta}}_{\mathrm{nu}} - \boldsymbol{\theta}^*_{\mathrm{nu}}) &\sim N(\mathbf{0}, \boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})^{-1}),\\ \sqrt{n}(\widehat{\boldsymbol{\theta}}_{\mathrm{de}} - \boldsymbol{\theta}^*_{\mathrm{de}}) &\sim N(\mathbf{0}, \boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{de}})^{-1}),\end{aligned}$$

when the sample size n goes to infinity. Under the regularity condition of parametric estimation, the bias of estimator is given as

$$\begin{aligned}\mathbb{E}\left[\widehat{\boldsymbol{\theta}}_{\mathrm{nu}} - \boldsymbol{\theta}^*_{\mathrm{nu}}\right] &= \mathcal{O}(n^{-1}),\\ \mathbb{E}\left[\widehat{\boldsymbol{\theta}}_{\mathrm{de}} - \boldsymbol{\theta}^*_{\mathrm{de}}\right] &= \mathcal{O}(n^{-1}).\end{aligned}$$

Then, for

$$\begin{aligned}\delta\widehat{\boldsymbol{\theta}}_{\mathrm{A}} &:= \widehat{\boldsymbol{\theta}}_{\mathrm{A}} - (\boldsymbol{\theta}^*_{\mathrm{nu}} - \boldsymbol{\theta}^*_{\mathrm{de}})\\ &= (\widehat{\boldsymbol{\theta}}_{\mathrm{nu}} - \boldsymbol{\theta}^*_{\mathrm{nu}}) - (\widehat{\boldsymbol{\theta}}_{\mathrm{de}} - \boldsymbol{\theta}^*_{\mathrm{de}}),\end{aligned}$$

we have

$$\mathbb{E}\left[\delta\widehat{\boldsymbol{\theta}}_{\mathrm{A}}\delta\widehat{\boldsymbol{\theta}}_{\mathrm{A}}^{\top}\right] = \frac{1}{n}\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})^{-1} + \frac{1}{n}\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{de}})^{-1} + \mathcal{O}(n^{-3/2}),$$

where we used the fact that $\widehat{\boldsymbol{\theta}}_{\mathrm{nu}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{de}}$ are independent. Substituting the above asymptotic variance of $\delta\widehat{\boldsymbol{\theta}}_{\mathrm{A}}$ into the first term of Eq.(20), we obtain

$$\begin{aligned}J(\widehat{r}_{\mathrm{A}}) &= \frac{1}{2n}\left[\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})^{-1} + \boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{de}})^{-1}\right)) + \mathrm{PE}(p^*_{\mathrm{de}}\|p^*_{\mathrm{nu}})\right] + \mathcal{O}(n^{-3/2}),\\ &= \frac{1}{2n}\left[\dim\Theta + \mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{de}})^{-1}\right) + \mathrm{PE}(p^*_{\mathrm{de}}\|p^*_{\mathrm{nu}})\right] + \mathcal{O}(n^{-3/2}),\end{aligned}$$

which concludes the proof. ■

C Proof of Lemma 2

Let $(\widehat{\boldsymbol{\theta}}_{\mathrm{B}}, \widehat{\theta}_{\mathrm{B},0})$ be the maximum likelihood estimator with the model (12). Let

$$\begin{aligned}\delta\widehat{\boldsymbol{\theta}}_{\mathrm{B}} &:= \widehat{\boldsymbol{\theta}}_{\mathrm{B}} - \boldsymbol{\theta}^*\\ &= \widehat{\boldsymbol{\theta}}_{\mathrm{B}} - (\boldsymbol{\theta}^*_{\mathrm{nu}} - \boldsymbol{\theta}^*_{\mathrm{de}}).\end{aligned}$$

Based on the standard asymptotic statistics, the asymptotic variance of the maximum likelihood estimator for the exponential family is given as

$$\sqrt{n}\delta\boldsymbol{\theta}_{\mathrm{B}} \sim N\left(\mathbf{0}, \boldsymbol{H}_{11}(\boldsymbol{\theta}^*, \theta_0^*)\right),$$

when the sample size n goes to infinity. $\boldsymbol{H}_{11}(\boldsymbol{\theta}, \theta_0)$ is the submatrix of the inverse matrix of the Fisher information matrix as defined in Eq.(13) and $(\boldsymbol{\theta}^*, \theta_0^*)$ is the parameter corresponding to the density ratio $r^*(\boldsymbol{x})$. Hence, the asymptotic variance of $\delta\boldsymbol{\theta}_{\mathrm{B}}$ is given as

$$\mathbb{E}\left[\delta\boldsymbol{\theta}_{\mathrm{B}}\delta\boldsymbol{\theta}_{\mathrm{B}}^{\top}\right] = \frac{1}{n}\boldsymbol{H}_{11}(\boldsymbol{\theta}^*, \theta_0^*) + \mathcal{O}(n^{-3/2}).$$

Substituting the asymptotic variance of $\delta\widehat{\boldsymbol{\theta}}_{\mathrm{B}}$ into the first term of Eq.(20), we establish the lemma. ■

D Proof of Lemma 3

By simple calculation, we find that the optimal solution $(\widehat{\boldsymbol{\theta}}_{\mathrm{C}}, \widehat{\theta}_{\mathrm{C},0})$ satisfies

$$\widehat{\theta}_{\mathrm{C},0} = -\log\left(\frac{1}{n}\sum_{j=1}^{n}\exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}\right).$$

The extremal condition for Eq.(9) with the above expression provides the following equation:

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}}) = \frac{\sum_{j=1}^{n}\exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})}{\sum_{j=1}^{n}\exp\left\{\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}}. \tag{21}$$

Let $\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}$ be

$$\begin{aligned}\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}} &:= \widehat{\boldsymbol{\theta}}_{\mathrm{C}} - \boldsymbol{\theta}^* \\ &= \widehat{\boldsymbol{\theta}}_{\mathrm{C}} - (\boldsymbol{\theta}_{\mathrm{nu}}^* - \boldsymbol{\theta}_{\mathrm{de}}^*).\end{aligned}$$

Then, Eq.(21) is represented as

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}}) = \frac{\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\exp\left\{\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})}{\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\exp\left\{\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^{\top}\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\right\}}.$$

Using the Taylor expansion

$$\begin{aligned}\exp(t) &= 1 + t + \mathcal{O}(t^2), \\ \frac{1}{1-t} &= 1 + t + \mathcal{O}(t^2),\end{aligned}$$

the asymptotic expansion of the right-hand side of the above equation yields

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}}) &= \left\{\frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}}) + \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})^\top \delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}\right\} \\
&\quad \times \left\{1 - \left(\frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}}) - 1\right) - \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})^\top \delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}\right\} + \mathcal{O}_p(n^{-1}) \\
&= \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}}) - \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\left(\frac{1}{n}\sum_{j'=1}^{n} r^*(\boldsymbol{x}_{j'}^{\mathrm{de}}) - 1\right) \\
&\quad + \left\{\frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})^\top \right. \\
&\qquad \left. - \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}})\frac{1}{n}\sum_{j'=1}^{n} r^*(\boldsymbol{x}_{j'}^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_{j'}^{\mathrm{de}})^\top\right\}\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}} + \mathcal{O}_p(n^{-1}) \\
&= \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}}) - (\boldsymbol{\eta}_{\mathrm{nu}} + \mathcal{O}_p(n^{-1/2}))\left(\frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}}) - 1\right) \\
&\quad + \boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}} + \mathcal{O}_p(n^{-1}) \\
&= \boldsymbol{\eta}_{\mathrm{nu}} + \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})\left(\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}}) - \boldsymbol{\eta}_{\mathrm{nu}}\right) + \boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}} + \mathcal{O}_p(n^{-1}).
\end{aligned}
$$

Hence, we obtain

$$
\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}}) - \boldsymbol{\eta}_{\mathrm{nu}}) - \frac{1}{n}\sum_{j=1}^{n} r^*(\boldsymbol{x}_i^{\mathrm{de}})(\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{de}}) - \boldsymbol{\eta}_{\mathrm{nu}}) + \mathcal{O}_p(n^{-1}). \tag{22}
$$

When the sample size goes to infinity, the central limit theorem provides

$$
\begin{aligned}
\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(\boldsymbol{\xi}(\boldsymbol{x}_i^{\mathrm{nu}}) - \boldsymbol{\eta}_{\mathrm{nu}}) &\sim N(\boldsymbol{0}, \boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})), \\
\frac{1}{\sqrt{n}}\sum_{j=1}^{n} r^*(\boldsymbol{x}_j^{\mathrm{de}})(\boldsymbol{\xi}(\boldsymbol{x}_j^{\mathrm{de}}) - \boldsymbol{\eta}_{\mathrm{nu}}) &\sim N(\boldsymbol{0}, \boldsymbol{G}),
\end{aligned} \tag{23}
$$

where $\boldsymbol{G}$ is the matrix defined in Lemma 3. Combining Eqs.(22) and (23), we obtain the following expression of the asymptotic variance of $\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}$:

$$
\mathbb{E}\left[\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}^\top\right] = \frac{1}{n}\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})^{-1} + \frac{1}{n}\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})^{-1}\boldsymbol{G}\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})^{-1} + \mathcal{O}(n^{-3/2}).
$$

Substituting the asymptotic variance of $\delta\widehat{\boldsymbol{\theta}}_{\mathrm{C}}$ into the first term of Eq.(20), we establish the lemma. ■

E Proof of Theorem 4

We compare the coefficients of order $\mathcal{O}(n^{-1})$ in $J(\widehat{r}_{\mathrm{A}})$ and $J(\widehat{r}_{\mathrm{B}})$, and prove the following inequality:

$$\dim\Theta + \mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{de}})^{-1}\right) \leq \mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{\theta}^*_{\mathrm{nu}})\boldsymbol{H}_{11}(\boldsymbol{\theta}^*, \theta^*_0)\right). \tag{24}$$

Let $\widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \theta_0)$ be the Fisher information matrix of the logistic model

$$\begin{aligned} q_{\boldsymbol{\eta}}(y = \text{`nu'}|\boldsymbol{x}; \boldsymbol{\theta}, \theta_0) &= \frac{\exp\left\{\theta_0 + \boldsymbol{\theta}^\top(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta})\right\}}{1 + \exp\left\{\theta_0 + \boldsymbol{\theta}^\top(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta})\right\}}, \\ q_{\boldsymbol{\eta}}(y = \text{`de'}|\boldsymbol{x}; \boldsymbol{\theta}, \theta_0) &= \frac{1}{1 + \exp\left\{\theta_0 + \boldsymbol{\theta}^\top(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta})\right\}}, \end{aligned} \tag{25}$$

where $\boldsymbol{\eta}$ is a fixed vector. Let us represent $\widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \theta_0)^{-1}$ in a block form as

$$\widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \theta_0)^{-1} = \begin{pmatrix} \boldsymbol{H}_{\boldsymbol{\eta},11}(\boldsymbol{\theta}, \theta_0) & \boldsymbol{h}_{\boldsymbol{\eta},12}(\boldsymbol{\theta}, \theta_0) \\ \boldsymbol{h}_{\boldsymbol{\eta},12}(\boldsymbol{\theta}, \theta_0)^\top & h_{\boldsymbol{\eta},22}(\boldsymbol{\theta}, \theta_0) \end{pmatrix}.$$

When the functions $1, \xi_1(\boldsymbol{x}), \ldots, \xi_k(\boldsymbol{x})$ are linearly independent, the maximum likelihood estimator (mle) of $\boldsymbol{\theta}$ for model (25) is given by $\widehat{\boldsymbol{\theta}}_{\mathrm{B}}$. The equality

$$\theta_0 + \boldsymbol{\theta}^\top\boldsymbol{\xi}(\boldsymbol{x}) = \widetilde{\theta}_0 + \widetilde{\boldsymbol{\theta}}^\top(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta})$$

implies $\boldsymbol{\theta} = \widetilde{\boldsymbol{\theta}}$ and $\theta_0 = \widetilde{\theta}_0 - \widetilde{\boldsymbol{\theta}}^\top\boldsymbol{\eta} = \widetilde{\theta}_0 - \boldsymbol{\theta}^\top\boldsymbol{\eta}$. Due to the equality $\boldsymbol{\theta} = \widetilde{\boldsymbol{\theta}}$, we see that the mle of $\boldsymbol{\theta}$ is equal to that of $\widetilde{\boldsymbol{\theta}}$, and hence, the variance is unchanged under the parameter transformation, that is,

$$\boldsymbol{H}_{\boldsymbol{\eta},11}(\widetilde{\boldsymbol{\theta}}, \widetilde{\theta}_0) = \boldsymbol{H}_{\boldsymbol{\eta},11}(\boldsymbol{\theta}, \theta_0 + \boldsymbol{\theta}^\top\boldsymbol{\eta}) = \boldsymbol{H}_{11}(\boldsymbol{\theta}, \theta_0)$$

holds for any $\boldsymbol{\eta}$. The Fisher information matrix $\widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}}$ can be represented as

$$\begin{aligned} \widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}}(\boldsymbol{\theta}^*, \theta^*_0 + \boldsymbol{\theta}^{*\top}\boldsymbol{\eta}) &= \frac{1}{2}\int \frac{p^*_{\mathrm{nu}}(\boldsymbol{x})p^*_{\mathrm{de}}(\boldsymbol{x})}{p^*_{\mathrm{nu}}(\boldsymbol{x}) + p^*_{\mathrm{de}}(\boldsymbol{x})} \begin{pmatrix} \boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta} \\ 1 \end{pmatrix} \begin{pmatrix} (\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta})^\top & 1 \end{pmatrix} \mathrm{d}\boldsymbol{x} \\ &= \begin{pmatrix} \widetilde{\boldsymbol{F}}_{\boldsymbol{\eta},11} & \widetilde{\boldsymbol{f}}_{\boldsymbol{\eta},12} \\ \widetilde{\boldsymbol{f}}_{\boldsymbol{\eta},12}^\top & \widetilde{f}_{\boldsymbol{\eta},22} \end{pmatrix}. \end{aligned}$$

The first equality is obtained by the straightforward calculation of the Fisher information matrix. Applying the matrix inversion formula to the block form, we obtain

$$\begin{aligned} \boldsymbol{H}_{11}(\boldsymbol{\theta}^*, \theta^*_0) &= \boldsymbol{H}_{\boldsymbol{\eta},11}(\boldsymbol{\theta}^*, \theta^*_0 + \boldsymbol{\theta}^{*\top}\boldsymbol{\eta}) \\ &= \widetilde{\boldsymbol{F}}^{-1}_{\boldsymbol{\eta},11} + \frac{\widetilde{\boldsymbol{F}}^{-1}_{\boldsymbol{\eta},11}\widetilde{\boldsymbol{f}}_{\boldsymbol{\eta},12}\widetilde{\boldsymbol{f}}^\top_{\boldsymbol{\eta},12}\widetilde{\boldsymbol{F}}^{-1}_{\boldsymbol{\eta},11}}{\widetilde{f}_{\boldsymbol{\eta},22} - \widetilde{\boldsymbol{f}}^\top_{\boldsymbol{\eta},12}\widetilde{\boldsymbol{F}}^{-1}_{\boldsymbol{\eta},11}\widetilde{\boldsymbol{f}}_{\boldsymbol{\eta},12}}. \end{aligned}$$

Since $\widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}}$ is positive definite, we have

$$\widetilde{f}_{\boldsymbol{\eta},22} - \widetilde{\boldsymbol{f}}_{\boldsymbol{\eta},12}^{\top} \widetilde{\boldsymbol{F}}_{\boldsymbol{\eta},11}^{-1} \widetilde{\boldsymbol{f}}_{\boldsymbol{\eta},12} > 0,$$

and hence, we obtain the inequality

$$\boldsymbol{H}_{11}(\boldsymbol{\theta}^*, \theta_0^*) \succeq \widetilde{\boldsymbol{F}}_{\boldsymbol{\eta},11}^{-1},$$

for any $\boldsymbol{\eta}$. In the above formula, $\boldsymbol{A} \succeq \boldsymbol{B}$ indicates the fact that the matrix $\boldsymbol{A} - \boldsymbol{B}$ is positive semidefinite. On the other hand, the inequalities

$$\begin{aligned}
\widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}_{\mathrm{nu}},11} &= \frac{1}{2} \int \frac{p_{\mathrm{nu}}^*(\boldsymbol{x}) p_{\mathrm{de}}^*(\boldsymbol{x})}{p_{\mathrm{nu}}^*(\boldsymbol{x}) + p_{\mathrm{de}}^*(\boldsymbol{x})} (\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{nu}})(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{nu}})^{\top} \mathrm{d}\boldsymbol{x} \\
&\preceq \frac{1}{2} \int p_{\mathrm{nu}}^*(\boldsymbol{x}) (\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{nu}})(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{nu}})^{\top} \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2} \boldsymbol{F}(\boldsymbol{\theta}_{\mathrm{nu}}^*), \\
\widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}_{\mathrm{de}},11} &= \frac{1}{2} \int \frac{p_{\mathrm{nu}}^*(\boldsymbol{x}) p_{\mathrm{de}}^*(\boldsymbol{x})}{p_{\mathrm{nu}}^*(\boldsymbol{x}) + p_{\mathrm{de}}^*(\boldsymbol{x})} (\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{de}})(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{de}})^{\top} \mathrm{d}\boldsymbol{x} \\
&\preceq \frac{1}{2} \int p_{\mathrm{de}}^*(\boldsymbol{x}) (\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{de}})(\boldsymbol{\xi}(\boldsymbol{x}) - \boldsymbol{\eta}_{\mathrm{de}})^{\top} \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2} \boldsymbol{F}(\boldsymbol{\theta}_{\mathrm{de}}^*)
\end{aligned}$$

hold. Therefore, we obtain

$$\begin{aligned}
\boldsymbol{H}_{11}(\boldsymbol{\theta}^*, \theta_0^*) &\succeq \frac{1}{2} \widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}_{\mathrm{nu}},11}^{-1} + \frac{1}{2} \widetilde{\boldsymbol{F}}_{\boldsymbol{\eta}_{\mathrm{de}},11}^{-1} \\
&\succeq \boldsymbol{F}(\boldsymbol{\theta}_{\mathrm{nu}}^*)^{-1} + \boldsymbol{F}(\boldsymbol{\theta}_{\mathrm{de}}^*)^{-1}.
\end{aligned}$$

By multiplying $\boldsymbol{F}(\boldsymbol{\theta}_{\mathrm{nu}}^*)$ from the left-hand side and taking the trace of both sides, we obtain the inequality (24). ■

F Proof of Theorem 10

We prove the general expression (17) for any $\overline{r}$ in the exponential model. The optimality condition of Eq.(16) provides the equality

$$\int p_{\mathrm{de}}^*(\boldsymbol{x}) \overline{r}_{\mathrm{C}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int p_{\mathrm{nu}}^*(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1.$$

Hence, we have

$$\int p_{\mathrm{de}}^*(\boldsymbol{x}) (\overline{r}(\boldsymbol{x}) - \overline{r}_{\mathrm{C}}(\boldsymbol{x})) \mathrm{d}\boldsymbol{x} = \int p_{\mathrm{de}}^*(\boldsymbol{x}) \overline{r}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - 1.$$

Applying the Schwarz inequality to the above equality, we obtain

$$\mathrm{D}(\overline{r}, \overline{r}_{\mathrm{C}}) \geq \left| \int p^*_{\mathrm{de}}(\boldsymbol{x})\, \overline{r}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - 1 \right|^2 .$$

Thus, $\overline{r}$ is different from $\overline{r}_{\mathrm{C}}$ unless $p^*_{\mathrm{de}} \cdot \overline{r}$ is a probability density. ■

G Proof of Corollary 11

The optimality condition of the method (A) provides the equality

$$\int p^*_{\mathrm{nu}}(\boldsymbol{x}) \boldsymbol{\xi}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int \overline{p}_{\mathrm{nu}}(\boldsymbol{x}) \boldsymbol{\xi}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

Substituting the equality $\overline{p}_{\mathrm{nu}}(\boldsymbol{x}) = \overline{p}_{\mathrm{de}}(\boldsymbol{x}) \overline{r}_{\mathrm{A}}(\boldsymbol{x})$ into the above expression, we have

$$\int p^*_{\mathrm{nu}}(\boldsymbol{x}) \boldsymbol{\xi}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int \overline{p}_{\mathrm{de}}(\boldsymbol{x}) \overline{r}_{\mathrm{A}}(\boldsymbol{x}) \boldsymbol{\xi}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

When p^*_{de} belongs to the exponential model, we have $p^*_{\mathrm{de}} = \overline{p}_{\mathrm{de}}$ and thus, the equality

$$\int p^*_{\mathrm{nu}}(\boldsymbol{x}) \boldsymbol{\xi}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int p^*_{\mathrm{de}}(\boldsymbol{x}) \overline{r}_{\mathrm{A}}(\boldsymbol{x}) \boldsymbol{\xi}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

holds. The above equation is exactly the same as the optimality condition of Eq.(16) for the method (C). Thus, $\overline{r}_{\mathrm{A}} = \overline{r}_{\mathrm{C}}$ holds. ■

References

[1] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," Proceedings of the 24th International Conference on Machine Learning, pp.81–88, 2007.

[2] N. Cesa-Bianchi and G. Lugosi, Prediction, Learning, and Games, Cambridge University Press, Cambridge, NY, 2006.

[3] K.F. Cheng and C.K. Chu, "Semiparametric density estimation under a two-sample density ratio model," Bernoulli, vol.10, no.4, pp.583–604, 2004.

[4] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol.20, pp.273–297, 1995.

[5] B. Efron, "The efficiency of logistic regression compared to normal discriminant analysis," Journal of the American Statistical Association, vol.70, no.352, pp.892–898, 1975.

[6] V.V. Fedorov, Theory of Optimal Experiments, Academic Press, New York, 1972.

[7] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," Knowledge and Information Systems. to appear.

[8] J. Huang, A. Smola, A. Gretton, K.M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in Advances in Neural Information Processing Systems 19, ed. B. Schölkopf, J. Platt, and T. Hoffman, pp.601–608, MIT Press, Cambridge, MA, 2007.

[9] T. Kanamori, "Pool-based active learning with optimal sampling distribution and its information geometrical interpretation," Neurocomputing, vol.71, no.1–3, pp.353–362, 2007.

[10] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," Journal of Machine Learning Research, vol.10, pp.1391–1445, Jul. 2009.

[11] T. Kanamori and H. Shimodaira, "Active learning algorithm using the maximum weighted log-likelihood estimator," Journal of Statistical Planning and Inference, vol.116, no.1, pp.149–162, 2003.

[12] S. Kullback and R.A. Leibler, "On information and sufficiency," Annals of Mathematical Statistics, vol.22, pp.79–86, 1951.

[13] E.L. Lehmann and G. Casella, Theory of Point Estimation, second ed., Springer, New York, 1998.

[14] P. Liang and M. Jordan, "An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators," Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008), ed. A. McCallum and S. Roweis, pp.584–591, Omnipress, 2008.

[15] X. Nguyen, M. Wainwright, and M. Jordan, "Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization," in Advances in Neural Information Processing Systems 20, ed. J.C. Platt, D. Koller, Y. Singer, and S. Roweis, pp.1089–1096, MIT Press, Cambridge, MA, 2008.

[16] J. Qin, "Inferences for case-control and semiparametric two-sample density ratio models," Biometrika, vol.85, no.3, pp.619–639, 1998.

[17] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, eds., Dataset Shift in Machine Learning, MIT Press, Cambridge, MA, 2009.

[18] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," Journal of Statistical Planning and Inference, vol.90, no.2, pp.227–244, 2000.

[19] A. Smola, L. Song, and C.H. Teo, "Relative novelty detection," JMLR Workshop and Conference Proceedings, ed. D. van Dyk and M. Welling, Twelfth International Conference on Artificial Intelligence and Statistics, vol.5, pp.536–543, 2009.

[20] M. Sugiyama, "Active learning in approximately linear regression based on conditional expectation of generalization error," Journal of Machine Learning Research, vol.7, pp.141–166, Jan. 2006.

[21] M. Sugiyama, T. Kanamori, T. Suzuki, S. Hido, J. Sese, I. Takeuchi, and L. Wang, "A density-ratio framework for statistical data processing," IPSJ Transactions on Computer Vision and Applications, vol.1, pp.183–208, 2009.

[22] M. Sugiyama, M. Krauledat, and K.R. Müller, "Covariate shift adaptation by importance weighted cross validation," Journal of Machine Learning Research, vol.8, pp.985–1005, May 2007.

[23] M. Sugiyama and K.R. Müller, "Input-dependent estimation of generalization error under covariate shift," Statistics & Decisions, vol.23, no.4, pp.249–279, 2005.

[24] M. Sugiyama and S. Nakajima, "Pool-based active learning in approximate linear regression," Machine Learning, vol.75, no.3, pp.249–274, 2009.

[25] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," Annals of the Institute of Statistical Mathematics, vol.60, no.4, pp.699–746, 2008.

[26] M. Sugiyama, I. Takeuchi, T. Suzuki, T. Kanamori, H. Hachiya, and D. Okanohara, "Least-squares conditional density estimation," IEICE Transactions on Information and Systems, vol.E93-D, no.3, 2010. to appear.

[27] M. Sugiyama, P. von Bünau, M. Kawanabe, and K.R. Müller, Covariate Shift Adaptation: Towards Machine Learning in Non-Stationary Environment, MIT Press, Cambridge, MA. to appear.

[28] T. Suzuki and M. Sugiyama, "Estimating squared-loss mutual information for independent component analysis.," Independent Component Analysis and Signal Separation, ed. T. Adali, C. Jutten, J.M.T. Romano, and A.K. Barros, Lecture Notes in Computer Science, vol.5441, Berlin, pp.130–137, Springer, 2009.

[29] T. Suzuki and M. Sugiyama, "Sufficient dimension reduction via squared-loss mutual information estimation," Tech. Rep. TR09-0005, Department of Computer Science, Tokyo Institute of Technology, Feb. 2009.

[30] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, "Mutual information estimation reveals global associations between stimuli and biological processes," BMC Bioinformatics, vol.10, no.1, p.S52, 2009.

[31] T. Suzuki, M. Sugiyama, J. Sese, and T. Kanamori, "Approximating mutual information by maximum likelihood density ratio estimation," JMLR Workshop and Conference Proceedings, ed. Y. Saeys, H. Liu, I. Inza, L. Wehenkel, and Y.V. de Peer, New Challenges for Feature Selection in Data Mining and Knowledge Discovery, vol.4, pp.5–20, 2008.

[32] A.W. van der Vaart, Asymptotic Statistics, Cambridge University Press, 1998.

[33] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[34] D.P. Wiens, "Robust weights and designs for biased regression models: Least squares and generalized M-estimation," Journal of Statistical Planning and Inference, vol.83, no.2, pp.395–412, 2000.

[35] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," Proceedings of the Twenty-First International Conference on Machine Learning, New York, NY, pp.903–910, ACM Press, 2004.

Least-squares Independent Component Analysis*

Taiji Suzuki

Department of Mathematical Informatics, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
s-taiji@stat.t.u-tokyo.ac.jp

Masashi Sugiyama

Department of Computer Science, Tokyo Institute of Technology
and PRESTO, Japan Science and Technology Agency
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
sugi@cs.titech.ac.jp

Abstract

Accurately evaluating statistical independence among random variables is a key element of Independent Component Analysis (ICA). In this paper, we employ a squared-loss variant of mutual information as an independence measure and give its estimation method. Our basic idea is to estimate the *ratio* of probability densities directly without going through density estimation, by which a hard task of density estimation can be avoided. In this density ratio approach, a natural cross-validation procedure is available for hyper-parameter selection. Thus, all tuning parameters such as the kernel width or the regularization parameter can be objectively optimized. This is an advantage over recently developed kernel-based independence measures and is a highly useful property in unsupervised learning problems such as ICA. Based on this novel independence measure, we develop an ICA algorithm named *Least-squares Independent Component Analysis* (LICA).

1 Introduction

The purpose of Independent Component Analysis (ICA) (Hyvärinen et al., 2001) is to obtain a transformation matrix that separates mixed signals into statistically-independent source signals. A direct approach to ICA is to find a transformation matrix such that independence among separated signals is maximized under some independence measure such as *mutual information* (MI).

*A MATLAB® implementation of the proposed algorithm, LICA, is available from 'http://www.simplex.t.u-tokyo.ac.jp/~s-taiji/software/LICA/index.html'.

Various approaches to evaluating the independence among random variables from samples have been explored so far. A naive approach is to estimate probability densities based on parametric or non-parametric density estimation methods. However, finding an appropriate parametric model is not easy without strong prior knowledge and non-parametric estimation is not accurate in high-dimensional problems. Thus this naive approach is not reliable in practice. Another approach is to approximate the *negentropy* (or negative entropy) based on the *Gram-Charlier expansion* (Cardoso & Souloumiac, 1993; Comon, 1994; Amari et al., 1996) or the *Edgeworth expansion* (Hulle, 2008). An advantage of this negentropy-based approach is that a hard task of density estimation is not directly involved. However, these expansion techniques are based on the assumption that the target density is close to normal and violation of this assumption can cause large approximation error.

The above approaches are based on the probability densities of signals. Another line of research that does not explicitly involve probability densities employs *non-linear correlation*—signals are statistically independent if and only if all non-linear correlations among the signals vanish. Following this line, computationally efficient algorithms have been developed based on a *contrast function* (Jutten & Herault, 1991; Hyvärinen, 1999), which is an approximation of negentropy or mutual information. However, these methods require to pre-specify non-linearities in the contrast function, and thus could be inaccurate if the predetermined non-linearities do not match the target distribution. To cope with this problem, the *kernel trick* has been applied in ICA, which allows one to evaluate all non-linear correlations in a computationally efficient manner (Bach & Jordan, 2002). However, its practical performance depends on the choice of kernels (more specifically, the Gaussian kernel width) and there seems no theoretically justified method to determine the kernel width (see also Fukumizu et al., 2009). This is a critical problem in unsupervised learning problems such as ICA.

In this paper, we develop a new ICA algorithm that resolves the problems mentioned above. We adopt a squared-loss variant of MI (which we call *squared-loss MI*; SMI) as an independence measure and approximate it by estimating the ratio of probability densities contained in SMI directly without going through density estimation. This approach—which follows the line of Sugiyama et al. (2008), Kanamori et al. (2009), and Nguyen et al. (2010)—allows us to avoid a hard task of density estimation. Another practical advantage of this density-ratio approach is that a natural cross-validation (CV) procedure is available for hyper-parameter selection. Thus all tuning parameters such as the kernel width or the regularization parameter can be objectively and systematically optimized through CV.

From an algorithmic point of view, our density-ratio approach *analytically* provides a non-parametric estimator of SMI; furthermore its derivative can also be computed analytically and these properties are utilized in deriving a new ICA algorithm. The proposed method is named *Least-squares Independent Component Analysis* (LICA).

Characteristics of existing and proposed ICA methods are summarized in Table 1, highlighting the advantage of the proposed LICA approach.

The structure of this paper is as follows. In Section 2, we formulate our estimator of

Table 1: Summary of existing and proposed ICA methods.

	Hyper-parameter selection	Distribution
Fast ICA (FICA) (Hyvärinen, 1999)	**Not Necessary**	Not Free
Natural-gradient ICA (NICA) (Amari et al., 1996)	**Not Necessary**	Not Free
Kernel ICA (KICA) (Bach & Jordan, 2002)	Not Available	**Free**
Edgeworth-expansion ICA (EICA) (Hulle, 2008)	**Not Necessary**	Nearly normal
Least-squares ICA (LICA) (proposed)	**Available**	**Free**

SMI. In Section 3, we derive the LICA algorithm based on the SMI estimator. Section 4 is devoted to numerical experiments where we show that our method properly estimate the true demixing matrix using toy datasets, and compare the performances of the proposed and existing methods on artificial and real datasets.

2 SMI Estimation for ICA

In this section, we formulate the ICA problem and introduce our independence measure, SMI. Then we give an estimation method of SMI and derive an ICA algorithm.

2.1 Problem Formulation

Suppose there is a d-dimensional random signal

$$\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^\top$$

drawn from a distribution with density $p(\boldsymbol{x})$, where $\{x^{(m)}\}_{m=1}^d$ are statistically independent of each other, and $^\top$ denotes the transpose of a matrix or a vector. Thus, $p(\boldsymbol{x})$ can be factorized as

$$p(\boldsymbol{x}) = \prod_{m=1}^{d} p_m(x^{(m)}).$$

We cannot directly observe the *source* signal $\boldsymbol{x}$, but only a linearly mixed signal $\boldsymbol{y}$:

$$\boldsymbol{y} = (y^{(1)}, \ldots, y^{(d)})^\top := \boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{A}$ is a $d \times d$ invertible matrix called the *mixing matrix*. The goal of ICA is, given samples of the mixed signals $\{\boldsymbol{y}_i\}_{i=1}^n$, to obtain a *demixing matrix* $\boldsymbol{W}$ that recovers the original source signal $\boldsymbol{x}$. We denote the demixed signal by $\boldsymbol{z}$:

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{y}.$$

The ideal solution is $\boldsymbol{W} = \boldsymbol{A}^{-1}$, but we can only recover the source signals up to permutation and scaling of components of $\boldsymbol{x}$ due to non-identifiability of the ICA setup (Hyvärinen et al., 2001).

A direct approach to ICA is to determine $\boldsymbol{W}$ so that components of $\boldsymbol{z}$ are as independent as possible. Here, we adopt SMI as the independence measure:

$$I_s(Z^{(1)}, \ldots, Z^{(d)}) := \frac{1}{2} \int \left(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})} - 1 \right)^2 r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}, \tag{1}$$

where $q(\boldsymbol{z})$ denotes the joint density of $\boldsymbol{z}$ and $r(\boldsymbol{z})$ denotes the product of marginal densities $\{q_m(z^{(m)})\}_{m=1}^d$:

$$r(\boldsymbol{z}) = \prod_{m=1}^{d} q_m(z^{(m)}).$$

Note that SMI is the *Pearson divergence* (Pearson, 1900; Paninsky, 2003; Liese & Vajda, 2006; Cichocki et al., 2009) between $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$, while ordinary MI is the *Kullback-Leibler divergence* (Kullback & Leibler, 1951). Since I_s is non-negative and it vanishes if and only if $q(\boldsymbol{z}) = r(\boldsymbol{z})$, the degree of independence among $\{z^{(m)}\}_{m=1}^d$ may be measured by SMI. Note that Eq.(1) corresponds to the *f-divergence* (Ali & Silvey, 1966; Csiszár, 1967) between $q(\boldsymbol{x})$ and $r(\boldsymbol{z})$ with the squared-loss, while ordinary MI corresponds to the f-divergence with the log-loss. Thus SMI could be regarded as a natural generalization of ordinary MI.

Based on the independence detection property of SMI, we try to find the demixing matrix $\boldsymbol{W}$ that minimizes SMI. Let us denote the demixed samples by

$$\{\boldsymbol{z}_i \mid \boldsymbol{z}_i = (z_i^{(1)}, \ldots, z_i^{(d)})^\top := \boldsymbol{W}\boldsymbol{y}_i\}_{i=1}^n.$$

Our key constraint when estimating SMI is that we want to avoid density estimation since it is a hard task (Vapnik, 1998). Below, we show how this could be accomplished.

2.2 SMI Approximation via Density Ratio Estimation

We approximate SMI via *density ratio estimation*. Let us denote the ratio of the densities $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$ by

$$g^*(\boldsymbol{z}) := \frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}. \tag{2}$$

Then SMI can be written as

$$\begin{aligned} I_s(Z^{(1)}, \ldots, Z^{(d)}) &= \frac{1}{2} \int \left(g^*(\boldsymbol{z}) - 1\right)^2 r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \\ &= \frac{1}{2} \int \left(g^*(\boldsymbol{z})^2 r(\boldsymbol{z}) - 2g^*(\boldsymbol{z}) r(\boldsymbol{z}) + r(\boldsymbol{z}) \right) \mathrm{d}\boldsymbol{z} \\ &= \frac{1}{2} \int \left(g^*(\boldsymbol{z}) q(\boldsymbol{z}) - 2q(\boldsymbol{z}) + r(\boldsymbol{z}) \right) \mathrm{d}\boldsymbol{z} \\ &= \frac{1}{2} \int g^*(\boldsymbol{z}) q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} - \frac{1}{2}. \end{aligned} \tag{3}$$

Therefore, SMI can be approximated through the estimation of $\int g^*(\boldsymbol{z})q(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$, the expectation of $g^*(\boldsymbol{z})$ over $q(\boldsymbol{z})$. This can be achieved by taking the sample average of an estimator of the density ratio $g^*(\boldsymbol{z})$, say $\widehat{g}(\boldsymbol{z})$:

$$\widehat{I}_s = \frac{1}{2n}\sum_{i=1}^{n}\widehat{g}(\boldsymbol{z}_i) - \frac{1}{2}. \tag{4}$$

We take the least-squares approach to estimating the density ratio $g^*(\boldsymbol{z})$:

$$\begin{aligned}&\inf_g\left[\frac{1}{2}\int\Big(g(\boldsymbol{z})-g^*(\boldsymbol{z})\Big)^2 r(\boldsymbol{z})\mathrm{d}\boldsymbol{z}\right]\\ &\quad=\inf_g\left[\int\Big(\frac{1}{2}g(\boldsymbol{z})^2 r(\boldsymbol{z})-g(\boldsymbol{z})q(\boldsymbol{z})\Big)\mathrm{d}\boldsymbol{z}\right]+\text{constant},\end{aligned}$$

where $\inf_g$ is taken over all measurable functions. Obviously the optimal solution is the density ratio g^*. Thus computing I_s is now reduced to solving the following optimization problem:

$$\inf_g\left[\int\Big(\frac{1}{2}g(\boldsymbol{z})^2 r(\boldsymbol{z})-g(\boldsymbol{z})q(\boldsymbol{z})\Big)\mathrm{d}\boldsymbol{z}\right]. \tag{5}$$

However, directly solving the problem (5) is not possible due to the following two reasons. The first reason is that finding the minimizer over all measurable functions is not tractable in practice since the search space is too vast. To overcome this problem, we restrict the search space to some linear subspace $\mathcal{G}$:

$$\mathcal{G} = \{\boldsymbol{\alpha}^\top\boldsymbol{\varphi}(\boldsymbol{z}) \mid \boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_b)^\top \in \mathbb{R}^b\}, \tag{6}$$

where $\boldsymbol{\alpha}$ is a parameter to be learned from samples, and $\boldsymbol{\varphi}(\boldsymbol{z})$ is a basis function vector such that

$$\boldsymbol{\varphi}(\boldsymbol{z}) = (\varphi_1(\boldsymbol{z}),\ldots,\varphi_b(\boldsymbol{z}))^\top \geq \boldsymbol{0}_b \quad \text{for all } \boldsymbol{z}.$$

$\boldsymbol{0}_b$ denotes the b-dimensional vector with all zeros. Note that $\boldsymbol{\varphi}(\boldsymbol{z})$ could be dependent on the samples $\{\boldsymbol{z}_i\}_{i=1}^n$, i.e., *kernel* models are also allowed. We explain how the basis functions $\boldsymbol{\varphi}(\boldsymbol{z})$ are chosen in Section 2.3.

The second reason why directly solving the problem (5) is not possible is that the expectations over the true probability densities $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$ cannot be computed since $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$ are unknown. To cope with this problem, we approximate the expectations by their empirical averages—then the optimization problem is reduced to

$$\widehat{\boldsymbol{\alpha}} := \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^b}\left[\frac{1}{2}\boldsymbol{\alpha}^\top\widehat{\boldsymbol{H}}\boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha}^\top\boldsymbol{R}\boldsymbol{\alpha}\right], \tag{7}$$

where we included $\lambda\boldsymbol{\alpha}^\top\boldsymbol{R}\boldsymbol{\alpha}$ ($\lambda > 0$) for avoiding overfitting. λ is called the *regularization*

parameter, and $\boldsymbol{R}$ is some positive definite matrix. $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{h}}$ are defined as

$$\widehat{\boldsymbol{H}} := \frac{1}{n^d} \sum_{i_1,\ldots,i_d=1}^{n} \boldsymbol{\varphi}(z_{i_1}^{(1)}, \ldots, z_{i_d}^{(d)}) \boldsymbol{\varphi}(z_{i_1}^{(1)}, \ldots, z_{i_d}^{(d)})^\top, \tag{8}$$

$$\widehat{\boldsymbol{h}} := \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\varphi}(z_i^{(1)}, \ldots, z_i^{(d)}). \tag{9}$$

Differentiating the objective function in Eq.(7) with respect to $\boldsymbol{\alpha}$ and equating it to zero, we can obtain an analytic-form solution as

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1} \widehat{\boldsymbol{h}}.$$

Thus, the solution can be computed very efficiently just by solving a system of linear equations.

Once the density ratio (2) has been estimated, SMI can be approximated by plugging the estimated density ratio $\widehat{g}(\boldsymbol{z}) = \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{\varphi}(\boldsymbol{z})$ in Eq.(4):

$$\widehat{I}_s = \frac{1}{2} \widehat{\boldsymbol{\alpha}}^\top \widehat{\boldsymbol{h}} - \frac{1}{2}. \tag{10}$$

Note that we may obtain various expressions of SMI using the following identities:

$$\int g^*(\boldsymbol{z})^2 r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int g^*(\boldsymbol{z}) q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z},$$
$$\int g^*(\boldsymbol{z}) r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = 1.$$

Ordinary MI based on the Kullback-Leibler divergence can also be estimated similarly using the density ratio (Suzuki et al., 2008). However, the use of SMI is more advantageous due to the analytic-form solution, as described in Section 3.

2.3 Design of Basis Functions and Hyper-parameter Selection

As basis functions, we propose to use a Gaussian kernel:

$$\varphi_\ell(\boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{v}_\ell\|^2}{2\sigma^2}\right) = \prod_{m=1}^{d} \exp\left(-\frac{(z^{(m)} - v_\ell^{(m)})^2}{2\sigma^2}\right), \tag{11}$$

where

$$\{\boldsymbol{v}_\ell \mid \boldsymbol{v}_\ell = (v_\ell^{(1)}, \ldots, v_\ell^{(d)})^\top\}_{\ell=1}^{b}$$

are Gaussian centers randomly chosen from $\{\boldsymbol{z}_i\}_{i=1}^n$—more precisely, we set $\boldsymbol{v}_\ell = \boldsymbol{z}_{c(\ell)}$, where $\{c(\ell)\}_{\ell=1}^b$ are randomly chosen from $\{1, \ldots, n\}$ without replacement. An advantage

of the Gaussian kernel lies in the factorizability in Eq.(11), contributing to reducing the computational cost of the matrix $\widehat{\boldsymbol{H}}$ significantly:

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n^d} \prod_{m=1}^{d} \left[\sum_{i=1}^{n} \exp\left(-\frac{(z_i^{(m)} - v_\ell^{(m)})^2 + (z_i^{(m)} - v_{\ell'}^{(m)})^2}{2\sigma^2} \right) \right].$$

We use the RKHS (Reproducing Kernel Hilbert Space) norm of $\boldsymbol{\alpha}^\top \boldsymbol{\varphi}(\boldsymbol{z})$ induced by the Gaussian kernel as the regularization term $\boldsymbol{\alpha}^\top \boldsymbol{R} \boldsymbol{\alpha}$, which is a popular choice in the kernel method community (Schölkopf & Smola, 2002):

$$R_{\ell,\ell'} = \exp\left(-\frac{\|\boldsymbol{v}_\ell - \boldsymbol{v}_{\ell'}\|^2}{2\sigma^2} \right). \tag{12}$$

In the experiments, we fix the number of basis functions to

$$b = \min(300, n),$$

and choose the Gaussian width σ and the regularization parameter λ by CV with grid search as follows. First, the samples $\{\boldsymbol{z}_i\}_{i=1}^n$ are divided into K disjoint subsets $\{\mathcal{Z}_k\}_{k=1}^K$ of (approximately) the same size (we use $K = 5$ in the experiments). Then an estimator $\widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k}$ is obtained using $\mathcal{Z}\backslash\mathcal{Z}_k$ (i.e., $\mathcal{Z}$ without $\mathcal{Z}_k$) and the approximation error for the hold-out samples $\mathcal{Z}_k$ is computed:

$$J_{\mathcal{Z}_k}^{(K\text{-CV})} = \frac{1}{2} \widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k}^\top \widehat{\boldsymbol{H}}_{\mathcal{Z}_k} \widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k} - \widehat{\boldsymbol{h}}_{\mathcal{Z}_k}^\top \widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k},$$

where the matrix $\widehat{\boldsymbol{H}}_{\mathcal{Z}_k}$ and the vector $\widehat{\boldsymbol{h}}_{\mathcal{Z}_k}$ are defined in the same way as $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{h}}$, but computed only using $\mathcal{Z}_k$. This procedure is repeated for $k = 1, \ldots, K$ and its average $J^{(K\text{-CV})}$ is computed:

$$J^{(K\text{-CV})} = \frac{1}{K} \sum_{k=1}^{K} J_{\mathcal{Z}_k}^{(K\text{-CV})}.$$

For parameter selection, we compute $J^{(K\text{-CV})}$ for all hyper-parameter candidates (the Gaussian width σ and the regularization parameter λ in the current setting) and choose the parameter that minimizes $J^{(K\text{-CV})}$. We can show that $J^{(K\text{-CV})}$ is an almost unbiased estimator of the objective function in Eq.(5), where the 'almost'-ness comes from the fact that the number of samples is reduced in the CV procedure due to data splitting (Geisser, 1975; Kohave, 1995).

3 The LICA Algorithms

In this section, we show how the above SMI estimation idea could be employed in the context of ICA. Here, we derive two algorithms, which we call *Least-squares Independent Component Analysis* (LICA), for obtaining a minimizer of $\widehat{I}_s$ with respect to the demixing matrix $\boldsymbol{W}$—one is based on a *plain gradient* method (which we refer to as *PG-LICA*) and the other is based on a *natural gradient* method for whitened samples (which we refer to as *NG-LICA*). A MATLAB® implementation of LICA is available from

`http://www.simplex.t.u-tokyo.ac.jp/~s-taiji/software/LICA/index.html`

3.1 Plain Gradient Algorithm: PG-LICA

Based on the plain gradient technique, an update rule of $\boldsymbol{W}$ is given by

$$\boldsymbol{W} \longleftarrow \boldsymbol{W} - \varepsilon \frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}}, \tag{13}$$

where ε (> 0) is the step size. As shown in Appendix, the gradient is given by

$$\frac{\partial \widehat{I}_s}{\partial W_{\ell,\ell'}} = \frac{\partial \widehat{\boldsymbol{h}}^\top}{\partial W_{\ell,\ell'}} \widehat{\boldsymbol{\alpha}} - \frac{1}{2} \widehat{\boldsymbol{\alpha}}^\top \left(\frac{\partial \widehat{\boldsymbol{H}}}{\partial W_{\ell,\ell'}} + \lambda \frac{\partial \boldsymbol{R}}{\partial W_{\ell,\ell'}} \right) \widehat{\boldsymbol{\alpha}}, \tag{14}$$

where, for $\boldsymbol{u}_\ell = \boldsymbol{y}_{c(\ell)}$ and $\boldsymbol{y}_i = (y_i^{(1)}, \ldots, y_i^{(d)})^\top$,

$$\frac{\partial \widehat{h}_\ell}{\partial W_{k,k'}} = \frac{1}{n\sigma^2} \sum_{i=1}^{n} (z_i^{(k)} - v_\ell^{(k)})(u_\ell^{(k')} - y_i^{(k')}) \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{v}_k\|^2}{2\sigma^2} \right), \tag{15}$$

$$\begin{aligned} \frac{\partial \widehat{H}_{\ell,\ell'}}{\partial W_{k,k'}} = {} & \frac{1}{n^{d-1}} \prod_{m \neq k} \left[\sum_{i=1}^{n} \exp\left(-\frac{(z_i^{(m)} - v_\ell^{(m)})^2 + (z_i^{(m)} - v_{\ell'}^{(m)})^2}{2\sigma^2} \right) \right] \\ & \times \left[\frac{1}{n\sigma^2} \sum_{i=1}^{n} \left((z_i^{(k)} - v_\ell^{(k)})(u_\ell^{(k')} - y_i^{(k')}) + (z_i^{(k)} - v_{\ell'}^{(k)})(u_{\ell'}^{(k')} - y_i^{(k')}) \right) \right. \\ & \left. \times \exp\left(-\frac{(z_i^{(k)} - v_\ell^{(k)})^2 + (z_i^{(k)} - v_{\ell'}^{(k)})^2}{2\sigma^2} \right) \right]. \end{aligned} \tag{16}$$

For the regularization matrix $\boldsymbol{R}$ defined by Eq.(12), the partial derivative is given by

$$\frac{\partial R_{\ell,\ell'}}{\partial W_{k,k'}} = \frac{1}{\sigma^2} (v_\ell^{(k)} - v_{\ell'}^{(k)})(u_{\ell'}^{(k')} - u_\ell^{(k')}) \exp\left(-\frac{\|\boldsymbol{v}_\ell - \boldsymbol{v}_{\ell'}\|^2}{2\sigma^2} \right).$$

In ICA, scaling of components of $\boldsymbol{z}$ can be arbitrary. This implies that the above gradient updating rule can lead to a solution with poor scaling, which is not preferable from a numerical point of view. To avoid possible numerical instability, we normalize $\boldsymbol{W}$ at each gradient iteration as

$$W_{k,k'} \longleftarrow \frac{W_{k,k'}}{\sqrt{\sum_{m=1}^{d} W_{k,m}^2}}. \tag{17}$$

In practice, we may iteratively perform line search along the gradient and optimize the Gaussian width σ and the regularization parameter λ by CV. A pseudo code of the PG-LICA algorithm is summarized in Figure 1.

1. Initialize demixing matrix $\boldsymbol{W}$ and normalize it by Eq.(17).
2. Optimize Gaussian width σ and regularization parameter λ by CV.
3. Compute gradient $\frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}}$ by Eq.(14).
4. Choose step-size ε such that $\widehat{I}_s$ (see Eq.(10)) is minimized (*line-search*).
5. Update $\boldsymbol{W}$ by Eq.(13).
6. Normalize $\boldsymbol{W}$ by Eq.(17).
7. Repeat 2.–6. until $\boldsymbol{W}$ converges.

Figure 1: The LICA algorithm with plain gradient descent (PG-LICA).

3.2 Natural Gradient Algorithm for Whitened Data: NG-LICA

The second algorithm is based on a *natural gradient* technique (Amari, 1998).

Suppose the data samples are *whitened*, i.e., samples $\{\boldsymbol{y}_i\}_{i=1}^n$ are transformed as

$$\boldsymbol{y}_i \longleftarrow \widehat{\boldsymbol{C}}^{-\frac{1}{2}} \boldsymbol{y}_i, \tag{18}$$

where $\widehat{\boldsymbol{C}}$ is the sample covariance matrix:

$$\widehat{\boldsymbol{C}} := \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{y}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{y}_j \right) \left(\boldsymbol{y}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{y}_j \right)^\top.$$

Then it can be shown that a demixing matrix which eliminates the second order correlation is an *orthogonal matrix* (Hyvärinen et al., 2001). Thus, for whitened data, the search space of $\boldsymbol{W}$ can be restricted to the orthogonal group $O(d)$ without loss of generality.

The *tangent space* of $O(d)$ at $\boldsymbol{W}$ is equal to the space of all matrices $\boldsymbol{U}$ such that $\boldsymbol{W}^\top \boldsymbol{U}$ is *skew symmetric*, i.e., $\boldsymbol{U}\boldsymbol{W}^\top = -\boldsymbol{W}\boldsymbol{U}^\top$. The steepest direction on this tangent space, which is called the *natural gradient*, is given as follows (Amari, 1998):

$$\nabla \widehat{I}_s(\boldsymbol{W}) := \frac{1}{2} \left(\frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}} - \boldsymbol{W} \frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}}^\top \boldsymbol{W} \right), \tag{19}$$

where the canonical metric $\langle \boldsymbol{G}_1, \boldsymbol{G}_2 \rangle = \frac{1}{2} \mathrm{tr}(\boldsymbol{G}_1^\top \boldsymbol{G}_2)$ is adopted in the tangent space. Then the *geodesic* from $\boldsymbol{W}$ in the direction of the natural gradient over $O(d)$ can be expressed by

$$\boldsymbol{W} \exp\left(t \boldsymbol{W}^\top \nabla \widehat{I}_s(\boldsymbol{W}) \right),$$

where $t \in \mathbb{R}$ and 'exp' denotes the matrix exponential, i.e., for a square matrix $\boldsymbol{D}$,

$$\exp(\boldsymbol{D}) = \sum_{k=0}^{\infty} \frac{1}{k!} \boldsymbol{D}^k.$$

1. Whiten the data samples by Eq.(18).
2. Initialize demixing matrix $\boldsymbol{W}$ and normalize it by Eq.(17).
3. Optimize Gaussian width σ and regularization parameter λ by CV.
4. Compute the natural gradient $\nabla\widehat{I}_s$ by Eq.(19).
5. Choose step-size t such that $\widehat{I}_s$ (see Eq.(10)) is minimized over the set (20).
6. Update $\boldsymbol{W}$ by Eq.(21).
7. Repeat 3.–6. until $\boldsymbol{W}$ converges.

Figure 2: The LICA algorithm with natural gradient descent (NG-LICA).

Thus when we perform line search along the geodesic in the natural gradient direction, the minimizer may be searched from the set

$$\left\{\boldsymbol{W}\exp\left(-t\boldsymbol{W}^\top\nabla\widehat{I}_s(\boldsymbol{W})\right)\;\middle|\;t\geq 0\right\}, \tag{20}$$

i.e., t is chosen such that $\widehat{I}_s$ (see Eq.(10)) is minimized and $\boldsymbol{W}$ is updated as

$$\boldsymbol{W}\longleftarrow\boldsymbol{W}\exp\left(-t\boldsymbol{W}^\top\nabla\widehat{I}_s(\boldsymbol{W})\right). \tag{21}$$

Geometry and optimization algorithms on more general structure, the *Stiefel manifold*, is discussed in more detail in Nishimori and Akaho (2005).

A pseudo code of the NG-LICA algorithm is summarized in Figure 2.

3.3 Remarks

The proposed LICA algorithms can be regarded as an application of the general unconstrained least-squares density-ratio estimator proposed by Kanamori et al. (2009) to SMI in the context of ICA.

The optimization problem (5) can also be obtained following the line of Nguyen et al. (2010), which addresses a divergence estimation problem utilizing the *Legendre-Fenchel duality*. SMI defined by Eq.(1) can be expressed as

$$I_s(Z^{(1)},\ldots,Z^{(d)})=\int\frac{1}{2}\left(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}\right)^2 r(\boldsymbol{z})\mathrm{d}\boldsymbol{z}-\frac{1}{2}. \tag{22}$$

If the Legendre-Fenchel duality of the convex function $\frac{1}{2}x^2$,

$$\frac{1}{2}x^2=\sup_y\left(yx-\frac{1}{2}y^2\right),$$

is applied to $\frac{1}{2}\left(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}\right)^2$ in Eq.(22) in a pointwise manner, we have

$$\begin{aligned} I_s(Z^{(1)},\ldots,Z^{(d)}) &= \sup_g \left[\int \Big(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}g(\boldsymbol{z}) - \frac{1}{2}g(\boldsymbol{z})^2\Big)r(\boldsymbol{z})\mathrm{d}\boldsymbol{z} - \frac{1}{2}\right] \\ &= -\inf_g \left[\int \Big(\frac{1}{2}g(\boldsymbol{z})^2 q(\boldsymbol{z}) - g(\boldsymbol{z})r(\boldsymbol{z})\Big)\mathrm{d}\boldsymbol{z}\right] - \frac{1}{2}, \end{aligned}$$

where $\sup_g$ and $\inf_g$ are taken over all measurable functions.

SMI is closely related to the kernel independence measures developed recently (Gretton et al., 2005a; Gretton et al., 2005b; Fukumizu et al., 2008). In particular, it has been shown that the *NOrmalized Cross-Covariance Operator* (NOCCO) proposed in Fukumizu et al. (2008) is also an estimator of SMI for $d = 2$. However, there is no reasonable hyper-parameter selection method for this and all other kernel-based independence measures (see also Bach & Jordan, 2002 and Fukumizu et al., 2009). This is a crucial limitation in unsupervised learning scenarios such as ICA. On the other hand, cross-validation can be applied to our method for hyper-parameter selection, as shown in Section 2.3.

4 Experiments

In this section, we investigate the experimental performance of the proposed method.

4.1 Illustrative Examples

First, we illustrate how the proposed method behaves using the following three 2-dimensional datasets:

(a) Sub·Sub-Gaussians: $p(\boldsymbol{x}) = U(x^{(1)}; -0.5, 0.5)U(x^{(2)}; -0.5, 0.5)$,

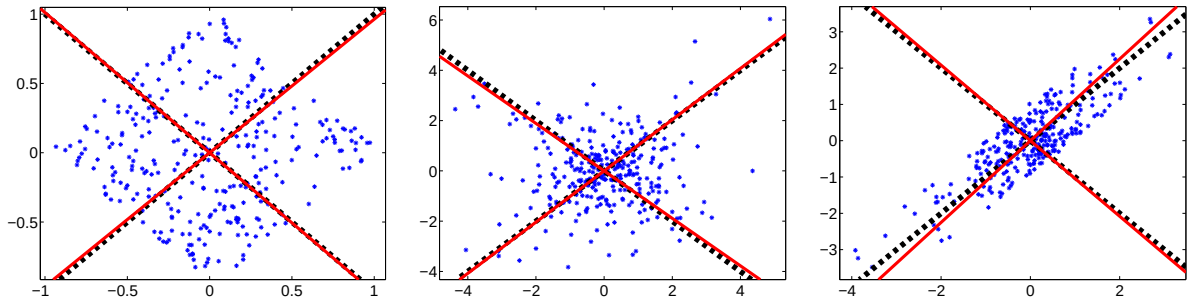
(b) Super·Super-Gaussians: $p(\boldsymbol{x}) = L(x^{(1)}; 0, 1)L(x^{(2)}; 0, 1)$,

(c) Sub·Super-Gaussians: $p(\boldsymbol{x}) = U(x^{(1)}; -0.5, 0.5)L(x^{(2)}; 0, 1)$,

where $U(x; a, b)$ $(a, b \in \mathbb{R}, a < b)$ denotes the uniform density on $[a, b]$ and $L(x; \mu, v)$ $(\mu \in \mathbb{R}, v > 0)$ denotes the Laplace density with mean μ and variance v. Let the number of samples be $n = 300$ and we observe mixed samples $\{\boldsymbol{y}_i\}_{i=1}^n$ through the following mixing matrix:

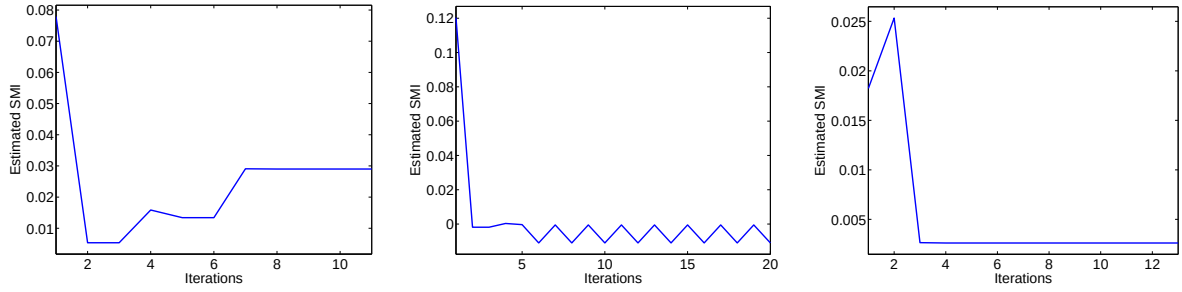
$$\boldsymbol{A} = \begin{pmatrix} \cos(\pi/4) & \sin(\pi/4) \\ -\sin(\pi/4) & \cos(\pi/4) \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

The observed samples are plotted in Figure 3. We employed the NG-LICA algorithm described in Figure 2. Hyper-parameters σ and λ in LICA were chosen by 5-fold CV from the 10 values in $[0.1, 1]$ at regular intervals and the 10 values in $[0.001, 1]$ at regular intervals in log scale, respectively. The regularization term was set to the squared RKHS norm induced by the Gaussian kernel, i.e., we employed $\boldsymbol{R}$ defined by Eq.(12).



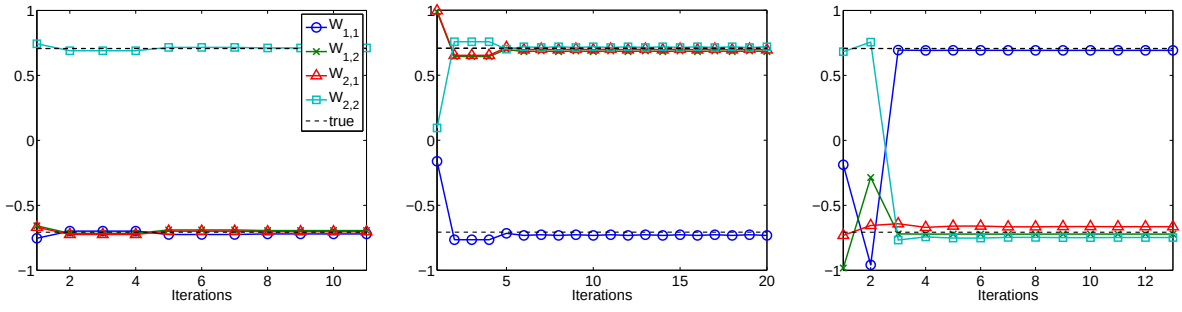
(a) (b) (c)

Figure 3: Observed samples (asterisks), true independent directions (dotted lines) and estimated independent directions (solid lines).



(a) (b) (c)

Figure 4: The value of $\widehat{I}_s$ over iterations.



(a) (b) (c)

Figure 5: The elements of the demixing matrix $\boldsymbol{W}$ over iterations. Solid lines correspond to $W_{1,1}$, $W_{1,2}$, $W_{2,1}$, and $W_{2,2}$, respectively. The dotted lines denote the true values.

The true independent directions as well as the estimated independent directions are plotted in Figure 3. Figure 4 depicts the value of the estimated SMI (10) over iterations and Figure 5 depicts the elements of the demixing matrix $\boldsymbol{W}$ over iterations. The results show that estimated SMI decreases rapidly and good solutions are obtained for all the datasets. The reason why the estimated SMI in Figure 4 does not decrease monotonically is that during the natural gradient optimization procedure, the hyper-parameters (λ and σ) are adjusted by CV (see Figure 2), which possibly causes an increase in the objective values.

4.2 Performance Comparison

Here we compare our method with some existing methods (KICA, FICA, JADE (Cardoso & Souloumiac, 1993)) on artificial and real datasets. We used the datasets (a), (b), and (c) in Section 4.1, the 'demosig' dataset available in the FastICA package[1] for MATLAB®, and '10halo', 'Sergio7', 'Speech4', and 'c5signals' datasets available in the ICALAB signal processing benchmark datasets[2] (Cichocki & Amari, 2003). The datasets (a), (b), (c), 'demosig', Sergio7', and 'c5signals' are artificial datasets. The datasets '10halo' and 'Speech4' are real datasets. We employed the *Amari index* (Amari et al., 1996) as the performance measure (smaller is better):

$$\text{Amari index} := \frac{1}{2d(d-1)} \sum_{m,m'=1}^{d} \left(\frac{|o_{m,m'}|}{\max_{m''} |o_{m,m''}|} + \frac{|o_{m,m'}|}{\max_{m''} |o_{m'',m'}|} \right) - \frac{1}{d-1},$$

where $o_{m,m'} := [\widehat{\boldsymbol{W}}\boldsymbol{A}]_{m,m'}$ for an estimated demixing matrix $\widehat{\boldsymbol{W}}$. We used the publicly available MATLAB® codes for KICA[3], FICA[1] and JADE[4], where default parameter settings were used. Hyper-parameters σ and λ in LICA were chosen by 5-fold CV from the 10 values in $[0.1, 1]$ at regular intervals and the 10 values in $[0.001, 1]$ at regular intervals in log scale, respectively. $\boldsymbol{R}$ was set as Eq.(12).

We randomly generated the mixing matrix $\boldsymbol{A}$ and source signals for artificial datasets, and computed the Amari index between the true $\boldsymbol{A}$ and $\widehat{\boldsymbol{W}}^{-1}$ for $\widehat{\boldsymbol{W}}$ estimated by each method. As training samples, we used the first n samples for Sergio7 and c5signals, and the n samples between the 1001th and (1000+n)-th interval for 10halo and Speech4, where we tested $n = 200$ and 500.

The performance of each method is summarized in Table 2, which depicts the mean and standard deviation of the Amari index over 50 trials. NG-LICA overall shows good performance. KICA tends to work reasonably well for datasets (a), (b), (c) and 'demosig', but it performs poorly for the ICALAB datasets; this seems to be caused by an inappropriate choice of the Gaussian kernel width and local optima. On the other hand, FICA and JADE tend to work reasonably well for the ICALAB datasets, but performs poorly

[1] http://www.cis.hut.fi/projects/ica/fastica

[2] http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/benchmarks/

[3] http://www.di.ens.fr/~fbach/kernel-ica/index.htm

[4] http://perso.telecom-paristech.fr/ cardoso/guidesepsou.html

Table 2: Mean and standard deviation of the Amari index (smaller is better) for the benchmark datasets. The datasets (a), (b), and (c) are taken from Section 4.1. The 'demosig' dataset is taken from the FastICA package. The '10halo', 'Sergio7', 'Speech4', and 'c5signals' datasets are taken from the ICALAB benchmarks datasets. The best method in terms of the mean Amari index and comparable ones based on the one-sided t-test at the significance level 1% are indicated by boldface.

dataset	n	NG-LICA	KICA	FICA	JADE
(a)	200	**0.05(0.03)**	**0.04(0.02)**	0.06(0.03)	**0.04(0.02)**
	500	**0.03(0.01)**	**0.03(0.01)**	**0.03(0.02)**	**0.02(0.01)**
(b)	200	**0.06(0.04)**	0.12(0.15)	0.16(0.20)	0.15(0.17)
	500	**0.04(0.03)**	**0.05(0.04)**	0.11(0.12)	**0.05(0.04)**
(c)	200	**0.08(0.05)**	**0.09(0.06)**	0.14(0.11)	0.13(0.09)
	500	**0.04(0.03)**	**0.04(0.03)**	0.09(0.08)	0.10(0.06)
demosig	200	**0.04(0.01)**	**0.05(0.11)**	0.08(0.05)	0.08(0.08)
	500	**0.02(0.01)**	**0.04(0.09)**	0.04(0.03)	0.04(0.02)
10halo	200	**0.29(0.02)**	0.38(0.03)	0.33(0.07)	0.36(0.00)
	500	**0.22(0.02)**	0.37(0.03)	**0.22(0.03)**	0.28(0.00)
Sergio7	200	**0.04(0.01)**	0.38(0.04)	0.05(0.02)	0.07(0.00)
	500	0.05(0.02)	0.37(0.03)	0.04(0.01)	**0.04(0.00)**
Speech4	200	**0.18(0.03)**	0.29(0.05)	0.20(0.03)	0.22(0.00)
	500	0.07(0.00)	0.10(0.04)	0.10(0.04)	**0.06(0.00)**
c5signals	200	0.12(0.01)	0.25(0.15)	**0.10(0.02)**	0.12(0.00)
	500	**0.06(0.04)**	0.07(0.06)	**0.04(0.02)**	0.07(0.00)

for (a), (b), (c) and 'demosig'; we conjecture that the contrast functions in FICA and the fourth-order statistics in JADE did not appropriately catch the non-Gaussianity of the datasets (a), (b), (c) and 'demosig'. Overall, the proposed LICA algorithm is shown to be a promising ICA method.

5 Conclusions

In this paper, we proposed a new ICA method based on a squared-loss variant of mutual information. The proposed method, named least-squares ICA (LICA), has several preferable properties, e.g., it is distribution-free and hyper-parameter selection by cross-validation is available.

Similarly to other ICA algorithms, the optimization problem involved in LICA is non-convex. Thus it is practically very important to develop good heuristics for initialization and avoiding local optima in the gradient procedures, which is an open research topic to be investigated. Moreover, although our SMI estimator is analytic, the LICA algorithm is still computationally rather expensive due to linear equations and cross-validation. Our

future work will address the computational issue, e.g., by vectorization and parallelization.

Appendix: Derivation of the Gradient of the SMI Estimator

Here we show the derivation of the gradient (14) of the SMI estimator (10). Since $\widehat{I}_s = \frac{1}{2}\widehat{\boldsymbol{h}}^\top \widehat{\boldsymbol{\alpha}} + \frac{1}{2}$ (see Eq.(10)), the derivative of $\widehat{I}_s$ with respect to $W_{k,k'}$ is given as follows:

$$\frac{\partial \widehat{I}_s}{\partial W_{k,k'}} = \frac{1}{2}\widehat{\boldsymbol{h}}^\top \frac{\partial \widehat{\boldsymbol{\alpha}}}{\partial W_{k,k'}} + \frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}}. \tag{23}$$

Remind that $\frac{\mathrm{d}\boldsymbol{B}(x)^{-1}}{\mathrm{d}x} = -\boldsymbol{B}(x)^{-1}\frac{\mathrm{d}\boldsymbol{B}(x)}{\mathrm{d}x}\boldsymbol{B}(x)^{-1}$ for an arbitrary matrix function $\boldsymbol{B}(x)$. Then the partial derivative of $\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\widehat{\boldsymbol{h}}$ with respect to $W_{k,k'}$ is given by

$$\begin{aligned}\frac{\partial \widehat{\boldsymbol{\alpha}}}{\partial W_{k,k'}} &= -(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})}{\partial W_{k,k'}}(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\widehat{\boldsymbol{h}} + (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}} \\ &= -(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} + (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}}.\end{aligned}$$

Substituting this in Eq.(23), we have

$$\begin{aligned}\frac{\partial \widehat{I}_s}{\partial W_{k,k'}} =& \frac{1}{2}\widehat{\boldsymbol{h}}^\top \left(-(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} + (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}} \right) + \frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}} \\ =& -\frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{H}}}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} - \frac{\lambda}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \boldsymbol{R}}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}},\end{aligned}$$

which gives Eq.(14).

Acknowledgments

The authors would like to thank Dr. Takafumi Kanamori for his valuable comments. T.S. was supported in part by the JSPS Research Fellowships for Young Scientists and Global COE Program "The research and training center for new development in mathematics", MEXT, Japan. M.S. acknowledges support from SCAT, AOARD, and the JST PRESTO program.

References

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, *28*, 131–142.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10*, 251–276.

Amari, S., Cichocki, A., & Yang, H. H. (1996). A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems* (pp. 757–763). MIT Press.

Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, *3*, 1–48.

Cardoso, J.-F., & Souloumiac, A. (1993). Blind beamforming for non-Gaussian signals. *Radar and Signal Processing, IEE Proceedings-F*, *140*, 362–370.

Cichocki, A., & Amari, S. (2003). *Adaptive blind signal and image processing: Learning algorithms and applications*. Wiley.

Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. (2009). *Non-negative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. New York: Wiley.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, *36*, 287–314.

Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, *2*, 229–318.

Fukumizu, K., Bach, F. R., & Jordan, M. I. (2009). Kernel dimension reduction in regression. *The Annals of Statistics*, *37*, 1871–1905.

Fukumizu, K., Gretton, A., Sun, X., & Schölkopf, B. (2008). Kernel measures of conditional dependence. *Advances in Neural Information Processing Systems 20* (pp. 489–496). Cambridge, MA: MIT Press.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, *70*, 320–328.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005a). Measuring statistical dependence with Hilbert-Schmidt norms. *Algorithmic Learning Theory* (pp. 63–77). Berlin: Springer-Verlag.

Gretton, A., Herbrich, R., Smola, A., Bousquet, O., & Schölkopf, B. (2005b). Kernel methods for measuring independence. *Journal of Machine Learning Research*, *6*, 2075–2129.

Hulle, M. M. V. (2008). Sequential fixed-point ICA based on mutual information minimization. *Neural Computation*, *20*, 1344–1365.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, *10*, 626–634.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis*. New York: Wiley.

Jutten, C., & Herault, J. (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, *24*, 1–10.

Kanamori, T., Hido, S., & Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, *10*, 1391–1445.

Kohave, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1137–1143). Morgan Kaufmann.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*, 79–86.

Liese, F., & Vajda, I. (2006). On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, *52*, 4394–4412.

Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*. to appear.

Nishimori, Y., & Akaho, S. (2005). Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing*, *67*, 106–135.

Paninsky, L. (2003). Estimation of entropy and mutual information. *Neural Computation*, *15*, 1191–1253.

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, *50*, 157–172.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., & Kawanabe, M. (2008). Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, *60*, 699–746.

Suzuki, T., Sugiyama, M., Sese, J., & Kanamori, T. (2008). Approximating mutual information by maximum likelihood density ratio estimation. *New Challenges for Feature Selection in Data Mining and Knowledge Discovery* (pp. 5–20).

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

IEICE Transactions on Information and Systems,
vol.E93-D, no.10, pp.2690–2701, 2010.
Revised on January 6, 2011.

Superfast-Trainable Multi-Class Probabilistic Classifier by Least-Squares Posterior Fitting

Masashi Sugiyama (sugi@cs.titech.ac.jp)
Tokyo Institute of Technology
and
Japan Science and Technology Agency

Abstract

Kernel logistic regression (KLR) is a powerful and flexible classification algorithm, which possesses an ability to provide the confidence of class prediction. However, its training—typically carried out by (quasi-)Newton methods—is rather time-consuming. In this paper, we propose an alternative probabilistic classification algorithm called *Least-Squares Probabilistic Classifier* (LSPC). KLR models the class-posterior probability by the log-linear combination of kernel functions and its parameters are learned by (regularized) maximum likelihood. In contrast, LSPC employs the linear combination of kernel functions and its parameters are learned by regularized least-squares fitting of the true class-posterior probability. Thanks to this linear regularized least-squares formulation, the solution of LSPC can be computed analytically just by solving a regularized system of linear equations in a class-wise manner. Thus LSPC is computationally very efficient and numerically stable. Through experiments, we show that the computation time of LSPC is faster than that of KLR by orders of magnitude, with comparable classification accuracy.

Keywords

Probabilistic classification, kernel logistic regression, class-posterior probability, squared-loss.

1 Introduction

The *support vector machine* (SVM) [7, 33] is a popular method for classification. Various computationally efficient algorithms for training SVM with massive datasets have been proposed so far (see [24, 16, 5, 6, 29, 26, 32, 13, 11, 30, 17, 31, 12] and many other softwares available online). However, SVM cannot provide the *confidence* of class prediction since it only learns the decision boundaries between different classes. To cope with this problem,

several post-processing methods have been developed for approximately computing the class-posterior probability [25, 34].

On the other hand, *logistic regression* (LR) is a classification algorithm that can naturally give the confidence of class prediction since it directly learns the class-posterior probabilities [15]. Recently, various efficient algorithms for training LR models specialized in *sparse* data have been developed [22, 10].

Applying the *kernel trick* to LR as done in SVM, one can easily obtain a non-linear classifier with probabilistic outputs, called *kernel logistic regression* (KLR). Since the kernel matrix is often *dense* (e.g., Gaussian kernels), the state-of-the-art LR algorithms for sparse data are not applicable to KLR. Thus, in order to train KLR classifiers, standard non-linear optimization techniques such as Newton's method (more specifically, *iteratively reweighted least-squares*) and quasi-Newton methods (for example, the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method) seem to be commonly used in practice [15, 23]. Although the performance of these general-purpose non-linear optimization techniques has been improved together with the evolution of computer environment in the last decade, computing the KLR solution is still challenging when the number of training samples is large. The purpose of this paper is to propose an alternative probabilistic classification method that can be trained very efficiently.

Our proposed method is called the *Least-Squares Probabilistic Classifier (LSPC)*. In LSPC, we use a linear combination of Gaussian kernels centered at training points as a model of class-posterior probabilities. Then we fit this model to the true class-posterior probability by least-squares[1]. An advantage of this linear least-squares formulation is that *consistency* is guaranteed without taking into account the normalization factor. In contrast, normalization is essential in the maximum-likelihood LR formulation; otherwise the likelihood tends to infinity. Thanks to the simplification brought by excluding the normalization factor from the optimization criterion, we can compute the globally optimal solution of LSPC *analytically* just by solving a system of linear equations.

Furthermore, we show that the use of a linear combination of kernel functions in LSPC allows us to learn the parameters in a class-wise manner. This highly contributes to further reducing the computational cost particularly in multi-class classification scenarios. Through experiments, we show that LSPC is computationally much more efficient than KLR with comparable accuracy.

2 Least-squares Approach to Probabilistic Classification

In this section, we formulate the problem of probabilistic classification and give a new method in the least-squares framework.

[1] A least-squares formulation has been employed for improving the computational efficiency of SVMs [29, 26, 13]. However, these approaches deal with deterministic classification, not probabilistic classification.

2.1 Problem Formulation

Let $\mathcal{X}$ ($\subset \mathbb{R}^d$) be the input domain, where d is the dimensionality of the input domain. Let $\mathcal{Y} = \{1, \ldots, c\}$ be the set of labels, where c is the number of classes. Let us consider a joint probability distribution on $\mathcal{X} \times \mathcal{Y}$ with joint probability density $p(\boldsymbol{x}, y)$. Suppose that we are given n independent and identically distributed (i.i.d.) paired samples of input $\boldsymbol{x}$ and output y:

$$\{(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n.$$

The goal is to estimate the class-posterior probability $p(y|\boldsymbol{x})$ from the samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. The class-posterior probability allows us to classify test sample $\boldsymbol{x}$ to class $\widehat{y}$ with confidence $p(\widehat{y}|\boldsymbol{x})$:

$$\widehat{y} := \operatorname*{argmax}_y p(y|\boldsymbol{x}).$$

Let us denote the marginal density of $\boldsymbol{x}$ by $p(\boldsymbol{x})$ and we assume that it is strictly positive:

$$p(\boldsymbol{x}) > 0 \text{ for all } \boldsymbol{x} \in \mathcal{X}.$$

Then, by definition, the class-posterior probability $p(y|\boldsymbol{x})$ can be expressed as

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})}. \tag{1}$$

This expression will be utilized in the derivation of the proposed method below.

2.2 Linear Least-squares Fitting of Class-posterior Probability

Here we introduce our least-squares fitting idea. We begin with the formulation for learning the class-posterior probability $p(y|\boldsymbol{x})$ as a function of both $\boldsymbol{x}$ and y, i.e., the class-posterior probabilities for all classes are learned simultaneously. Then in Section 2.3, we show that this simultaneous learning problem can be decomposed into independent class-wise learning problems, which highly contributes to reducing the computational cost.

We model the class-posterior probability $p(y|\boldsymbol{x})$ by the following linear model:

$$q(y|\boldsymbol{x}; \boldsymbol{\alpha}) := \sum_{\ell=1}^{b} \alpha_\ell \phi_\ell(\boldsymbol{x}, y) = \boldsymbol{\alpha}^\top \boldsymbol{\phi}(\boldsymbol{x}, y),$$

where $^\top$ denotes the transpose of a matrix or a vector,

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_b)^\top$$

are parameters to be learned from samples, and

$$\boldsymbol{\phi}(\boldsymbol{x}, y) = (\phi_1(\boldsymbol{x}, y), \ldots, \phi_b(\boldsymbol{x}, y))^\top$$

are basis functions such that

$$\boldsymbol{\phi}(\boldsymbol{x}, y) \geq \boldsymbol{0}_b \quad \text{for all } (\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}.$$

$\mathbf{0}_b$ denotes the b-dimensional vector with all zeros and the inequality for vectors is applied in the element-wise manner. We explain how the basis functions $\boldsymbol{\phi}(\boldsymbol{x}, y)$ are practically chosen in Section 2.3.

We determine the parameter $\boldsymbol{\alpha}$ in the model $q(y|\boldsymbol{x}; \boldsymbol{\alpha})$ so that the following squared error J is minimized:

$$\begin{aligned} J(\boldsymbol{\alpha}) &:= \frac{1}{2}\sum_{y=1}^{c}\int \left(q(y|\boldsymbol{x};\boldsymbol{\alpha}) - p(y|\boldsymbol{x})\right)^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2}\sum_{y=1}^{c}\int q(y|\boldsymbol{x};\boldsymbol{\alpha})^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \sum_{y=1}^{c}\int q(y|\boldsymbol{x};\boldsymbol{\alpha})p(\boldsymbol{x},y)\mathrm{d}\boldsymbol{x} + \mathrm{Const.} \\ &= \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{h}^\top\boldsymbol{\alpha} + \mathrm{Const.}, \end{aligned}$$

where we used Eq.(1). The $b \times b$ matrix $\boldsymbol{H}$ and the b-dimensional vector $\boldsymbol{h}$ are defined as

$$\begin{aligned} \boldsymbol{H} &:= \sum_{y=1}^{c}\int \boldsymbol{\phi}(\boldsymbol{x},y)\boldsymbol{\phi}(\boldsymbol{x},y)^\top p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \\ \boldsymbol{h} &:= \sum_{y=1}^{c}\int \boldsymbol{\phi}(\boldsymbol{x},y)p(\boldsymbol{x},y)\mathrm{d}\boldsymbol{x}. \end{aligned}$$

$\boldsymbol{H}$ and $\boldsymbol{h}$ contain the expectations over unknown densities $p(\boldsymbol{x})$ and $p(\boldsymbol{x}, y)$, so we approximate the expectations by sample averages. Then we have

$$\begin{aligned} \widehat{\boldsymbol{H}} &:= \frac{1}{n}\sum_{y=1}^{c}\sum_{i=1}^{n}\boldsymbol{\phi}(\boldsymbol{x}_i,y)\boldsymbol{\phi}(\boldsymbol{x}_i,y)^\top, \\ \widehat{\boldsymbol{h}} &:= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\phi}(\boldsymbol{x}_i,y_i). \end{aligned}$$

Now our optimization criterion is formulated as

$$\widehat{\boldsymbol{\alpha}} := \underset{\boldsymbol{\alpha}\in\mathbb{R}^b}{\mathrm{argmin}}\left[\frac{1}{2}\boldsymbol{\alpha}^\top\widehat{\boldsymbol{H}}\boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}\right],$$

where a regularizer $\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}$ ($\lambda > 0$) is included for regularization purposes. Taking the derivative of the above objective function and equating it to zero, we see that the solution $\widehat{\boldsymbol{\alpha}}$ can be obtained just by solving the following system of linear equations.

$$(\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}_b)\boldsymbol{\alpha} = \widehat{\boldsymbol{h}}, \tag{2}$$

where $\boldsymbol{I}_b$ denotes the b-dimensional identity matrix. Thus, the solution $\widehat{\boldsymbol{\alpha}}$ is given analytically as

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}_b)^{-1}\widehat{\boldsymbol{h}}.$$

In order to assure that the solution $q(y|\boldsymbol{x};\widehat{\boldsymbol{\alpha}})$ is a conditional probability, we round up negative outputs to zero [35] and renormalize the solution. Consequently, our final solution is expressed as

$$\widehat{p}(y|\boldsymbol{x}) = \frac{\max(0, \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{\phi}(\boldsymbol{x}, y)}{\sum_{y'=1}^{c} \max(0, \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{\phi}(\boldsymbol{x}, y'))}, \tag{3}$$

We call the above method *Least-Squares Probabilistic Classifier (LSPC)*. LSPC can be regarded as an application of a density ratio estimation method called the *unconstrained Least-Squares Importance Fitting (uLSIF)* [18] to probabilistic classification. Thus all the theoretical properties of uLSIF such as consistency, the rate of convergence, and numerical stability [19, 20] may be directly translated into the current context.

2.3 Basis Function Design

A naive choice of basis functions $\boldsymbol{\phi}(\boldsymbol{x}, y)$ would be a *kernel* model, i.e., for some kernel function K',

$$q(y|\boldsymbol{x};\boldsymbol{\alpha}) = \sum_{y'=1}^{c} \sum_{\ell=1}^{n} \alpha_\ell^{(y')} K'(\boldsymbol{x}, \boldsymbol{x}_\ell, y, y'), \tag{4}$$

which contains cn parameters. For this model, the computational complexity for solving Eq.(2) is $\mathcal{O}(c^3n^3)$.

Here we propose to separate input $\boldsymbol{x}$ and output y, and use the *delta kernel* for y (as in KLR):

$$q(y|\boldsymbol{x};\boldsymbol{\alpha}) = \sum_{y'=1}^{c} \sum_{\ell=1}^{n} \alpha_\ell^{(y')} K(\boldsymbol{x}, \boldsymbol{x}_\ell) \delta_{y,y'},$$

where K is a kernel function for $\boldsymbol{x}$ and $\delta_{y,y'}$ is the *Kronecker delta*:

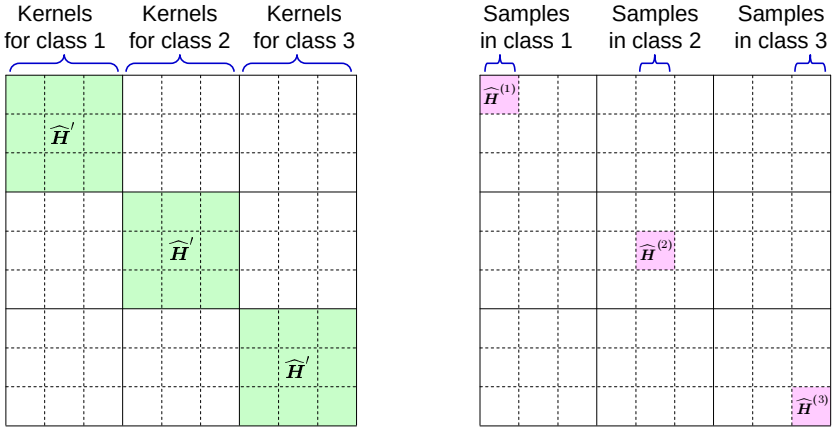
$$\delta_{y,y'} = \begin{cases} 1 & \text{if } y = y', \\ 0 & \text{otherwise.} \end{cases}$$

This model choice actually allows us to speed up the computation of LSPC significantly since all the calculations can be carried out *separately* in a class-wise manner. Indeed, the above model for class y is expressed as

$$q(y|\boldsymbol{x};\boldsymbol{\alpha}) = \sum_{\ell=1}^{n} \alpha_\ell^{(y)} K(\boldsymbol{x}, \boldsymbol{x}_\ell). \tag{5}$$

Then the matrix $\widehat{\boldsymbol{H}}$ becomes block-diagonal, as illustrated in Figure 1(a). Thus we only need to train a model with n parameters separately c times for each class y, by solving the following equation:

$$(\widehat{\boldsymbol{H}}' + \lambda \boldsymbol{I}_n)\boldsymbol{\alpha}^{(y)} = \widetilde{\boldsymbol{h}}^{(y)},$$



(a) Model (5) (b) Model (7)

Figure 1: Structure of matrix $\widehat{\boldsymbol{H}}$ for model (5) and model (7). The number of classes is $c = 3$. Suppose training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are sorted according to label y. Colored blocks are non-zero and others are zeros. For model (5) consisting of c sets of n basis functions, the matrix $\widehat{\boldsymbol{H}}$ becomes block-diagonal (with common block matrix $\widehat{\boldsymbol{H}}'$), and thus training can be carried out separately for each block. For model (7) consisting of c sets of n_y basis functions, the size of the target block is further reduced.

where $\widehat{\boldsymbol{H}}'$ is the $n \times n$ matrix and $\widetilde{\boldsymbol{h}}^{(y)}$ is the n-dimensional vector defined as

$$\widehat{H}'_{\ell,\ell'} := \frac{1}{n}\sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x}_\ell) K(\boldsymbol{x}_i, \boldsymbol{x}_{\ell'}),$$

$$\widetilde{h}^{(y)}_{\ell} := \frac{1}{n}\sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x}_\ell) \delta_{y,y_i}.$$

Since $\widehat{\boldsymbol{H}}'$ is common to all y, we only need to compute $(\widehat{\boldsymbol{H}}' + \lambda \boldsymbol{I}_n)^{-1}$ once. Then the computational complexity for obtaining the solution is $\mathcal{O}(n^3 + cn^2)$, which is smaller than the case with general kernel model (4). Thus this approach would be computationally efficient when the number of classes c is large.

Here, we further propose to reduce the number of kernels in model (5). To this end, we focus on a kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$ that is "localized". Examples of such localized kernels include the popular *Gaussian kernel* [28]:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right). \tag{6}$$

Our idea is to reduce the number of kernels by locating the kernels only at samples belonging to the *target* class:

$$q(y|\boldsymbol{x}; \boldsymbol{\alpha}) = \sum_{\ell=1}^{n_y} \alpha^{(y)}_\ell K(\boldsymbol{x}, \boldsymbol{x}^{(y)}_\ell), \tag{7}$$

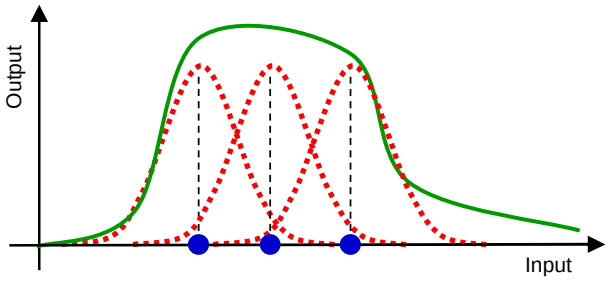


Figure 2: Heuristic of reducing the number of basis functions—locate Gaussian kernels only at the samples of the target class.

where n_y is the number of training samples in class y, and $\{\boldsymbol{x}_i^{(y)}\}_{i=1}^{n_y}$ is the training input samples in class y.

The rationale behind this model simplification is as follows (Figure 2). By definition, the class-posterior probability $p(y|\boldsymbol{x})$ takes large values in the regions where samples in class y are dense; conversely, $p(y|\boldsymbol{x})$ takes smaller values (i.e., close to zero) in the regions where samples in class y are sparse. When a non-negative function is approximated by a Gaussian kernel model, many kernels may be needed in the region where the output of the target function is large; on the other hand, only a small number of kernels would be enough in the region where the output of the target function is close to zero. Following this heuristic, many kernels are allocated in the region where $p(y|\boldsymbol{x})$ takes large values, which can be achieved by Eq.(7).

This model simplification allows us to further reduce the computational cost since the size of the target blocks in matrix $\widehat{\boldsymbol{H}}$ is further reduced, as illustrated in Figure 1(b). In order to learn the n_y-dimensional parameter vector

$$\boldsymbol{\alpha}^{(y)} = (\alpha_1^{(y)}, \ldots, \alpha_{n_y}^{(y)})^\top$$

for each class y, we only need to solve the following system of n_y linear equations:

$$(\widehat{\boldsymbol{H}}^{(y)} + \lambda \boldsymbol{I}_{n_y})\boldsymbol{\alpha}^{(y)} = \widehat{\boldsymbol{h}}^{(y)}, \tag{8}$$

where $\widehat{\boldsymbol{H}}^{(y)}$ is the $n_y \times n_y$ matrix and $\widehat{\boldsymbol{h}}^{(y)}$ is the n_y-dimensional vector defined as

$$\widehat{H}_{\ell,\ell'}^{(y)} := \frac{1}{n}\sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x}_\ell^{(y)})K(\boldsymbol{x}_i, \boldsymbol{x}_{\ell'}^{(y)}), \tag{9}$$

$$\widehat{h}_\ell^{(y)} := \frac{1}{n}\sum_{i=1}^{n_y} K(\boldsymbol{x}_i^{(y)}, \boldsymbol{x}_\ell^{(y)}).$$

Let $\widehat{\boldsymbol{\alpha}}^{(y)}$ be the solution of Eq.(8). Then our final solution is given by

$$\widehat{p}(y|\boldsymbol{x}) = \frac{\max(0, \sum_{\ell=1}^{n_y} \widehat{\alpha}_\ell^{(y)} K(\boldsymbol{x}, \boldsymbol{x}_\ell^{(y)}))}{\sum_{y'=1}^{c} \max(0, \sum_{\ell=1}^{n_{y'}} \widehat{\alpha}_\ell^{(y')} K(\boldsymbol{x}, \boldsymbol{x}_\ell^{(y')}))}. \tag{10}$$

Input: Labeled training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$
(equivalently, $\{\boldsymbol{x}_i^{(y)}\}_{i=1}^{n_y}$ for class $y = 1, \ldots, c$),
Gaussian width σ, and regularization parameter λ;

Output: Class-posterior probability $\widehat{p}(y|\boldsymbol{x})$;

for $y = 1, \ldots, c$

$$\widehat{H}_{\ell,\ell'}^{(y)} \longleftarrow \frac{1}{n}\sum_{i=1}^{n} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_\ell^{(y)}\|^2 + \|\boldsymbol{x}_i - \boldsymbol{x}_{\ell'}^{(y)}\|^2}{2\sigma^2}\right) \quad \text{for } \ell, \ell' = 1, \ldots, n_y;$$

$$\widehat{h}_\ell^{(y)} \longleftarrow \frac{1}{n}\sum_{i=1}^{n_y} \exp\left(-\frac{\|\boldsymbol{x}_i^{(y)} - \boldsymbol{x}_\ell^{(y)}\|^2}{2\sigma^2}\right) \quad \text{for } \ell = 1, \ldots, n_y;$$

Solve linear equation $(\widehat{\boldsymbol{H}}^{(y)} + \lambda \boldsymbol{I}_{n_y})\boldsymbol{\alpha}^{(y)} = \widehat{\boldsymbol{h}}^{(y)}$ and obtain $\widehat{\boldsymbol{\alpha}}^{(y)}$;

end

$$\widehat{p}(y|\boldsymbol{x}) \longleftarrow \frac{\max\left(0, \sum_{\ell=1}^{n_y} \widehat{\alpha}_\ell^{(y)} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_\ell^{(y)}\|^2}{2\sigma^2}\right)\right)}{\sum_{y'=1}^{c} \max\left(0, \sum_{\ell=1}^{n_{y'}} \widehat{\alpha}_\ell^{(y')} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_\ell^{(y')}\|^2}{2\sigma^2}\right)\right)};$$

Figure 3: Pseudo code of LSPC for simplified model (7) with Gaussian kernel (6).

For the simplified model (7), the computational complexity for obtaining the solution is $\mathcal{O}(cn_y^2 n)$—when $n_y = n/c$ for all y, this is equal to $\mathcal{O}(c^{-1}n^3)$. Thus this approach is computationally highly efficient for multi-class problems.

A pseudo code of the simplest LSPC implementation for Gaussian kernels is summarized in Figure 3. Its MATLAB® implementation is available from

`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSPC/`

3 Experiments

In this section, we experimentally compare the performance of the following classification methods:

- **LSPC**: LSPC with model (7).
- **LSPC(full)**: LSPC with model (5).
- **KLR**: ℓ_2-penalized kernel logistic regression with Gaussian kernels. We used a MATLAB® implementation included in the '*minFunc*' package [27], which uses limited-memory BFGS updates with Shanno-Phua scaling in computing the step direction and a bracketing line-search for a point satisfying the strong Wolfe conditions to compute the step direction.

When we fed data to learning algorithms, the input samples were normalized in the element-wise manner so that each element has mean zero and unit variance. The Gaussian width σ and the regularization parameter λ for all the methods are chosen based on 2-fold cross-validation from

$$\sigma \in \{\tfrac{1}{10}m, \tfrac{1}{5}m, \tfrac{1}{2}m, \tfrac{2}{3}m, m, \tfrac{3}{2}m, 2m, 5m, 10m\},$$
$$\lambda \in \{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0}\},$$

where

$$m := \mathrm{median}(\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\}_{i,j=1}^{n}).$$

3.1 Illustrative Examples

First, we illustrate the behavior of each method using a toy dataset.

We set the dimension of the input space to $d = 2$ and the number of classes to $c = 3$. We independently drew samples in each class from the following class-conditional sample densities (see Figure 4):

$$p(\boldsymbol{x}|y=1) = N\left(\boldsymbol{x}; \begin{bmatrix}-2\\0\end{bmatrix}, \begin{bmatrix}1&0\\0&1\end{bmatrix}\right),$$
$$p(\boldsymbol{x}|y=2) = N\left(\boldsymbol{x}; \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}1&0\\0&1\end{bmatrix}\right),$$
$$p(\boldsymbol{x}|y=3) = \frac{1}{2}N\left(\boldsymbol{x}; \begin{bmatrix}0\\-3\end{bmatrix}, \begin{bmatrix}4&0\\0&1\end{bmatrix}\right) + \frac{1}{2}N\left(\boldsymbol{x}; \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}1&0\\0&4\end{bmatrix}\right),$$

where $N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We set the class-prior probabilities $p(y)$ as

$$p(y) = \begin{cases} 1/4 & \text{if } y = 1, 2, \\ 1/2 & \text{if } y = 3, \end{cases}$$

and we set the number of training samples to $n = 200$. Generated samples are plotted in Figure 5.

The true class-posterior probabilities $p(y|\boldsymbol{x})$ ($\propto p(\boldsymbol{x}|y)p(y)$), their estimates obtained by LSPC, LSPC(full), and KLR are depicted in Figure 6. The plots show that all the methods approximate the true class-posterior probabilities well in the training region (say, $[-5, 5]^2$). However, the output outside the training region is substantially different in LSPC and KLR. This is induced by the difference of the models—a linear combination of Gaussian kernels is used in LSPC, while its exponent is used in KLR. Outside the training region, there is no kernel, and thus a linear combination of Gaussian kernels takes values close to zero (note that the values are not exactly zero since Gaussian tails extended from training regions remain everywhere); then typically one of the classes takes a value close to one, and the others tend to zero outside the training regions. On the other

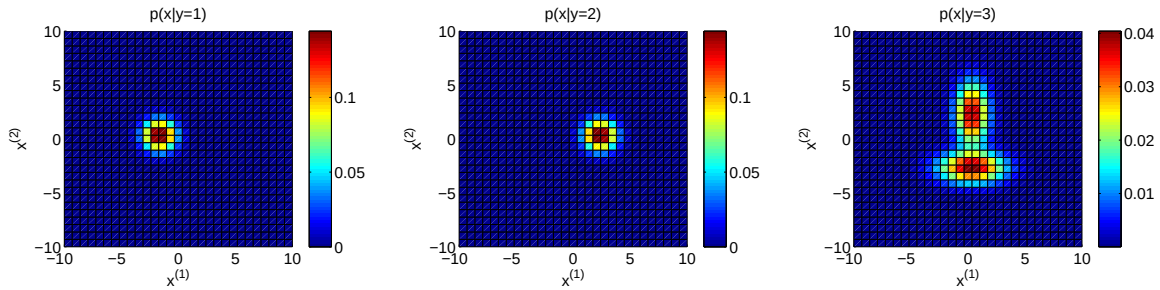


Figure 4: Illustrative examples. Class-conditional sample densities $p(\boldsymbol{x}|y)$.

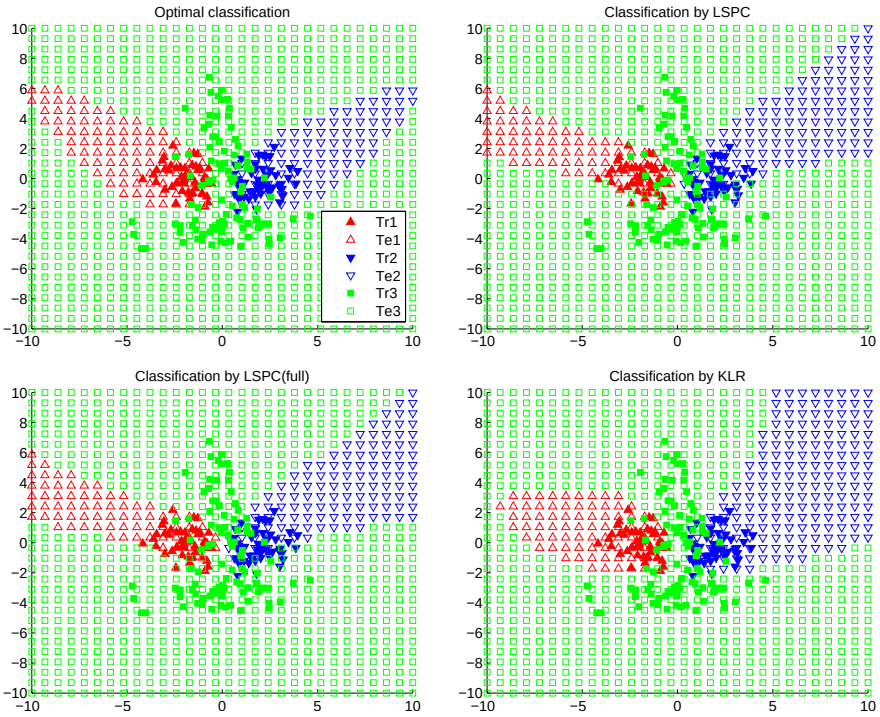


Figure 5: Illustrative examples. Training samples are plotted with filled symbols. Unfilled symbols denote the classification results based on the true class-posterior probabilities and their estimates obtained by LSPC, LSPC(full), and KLR.

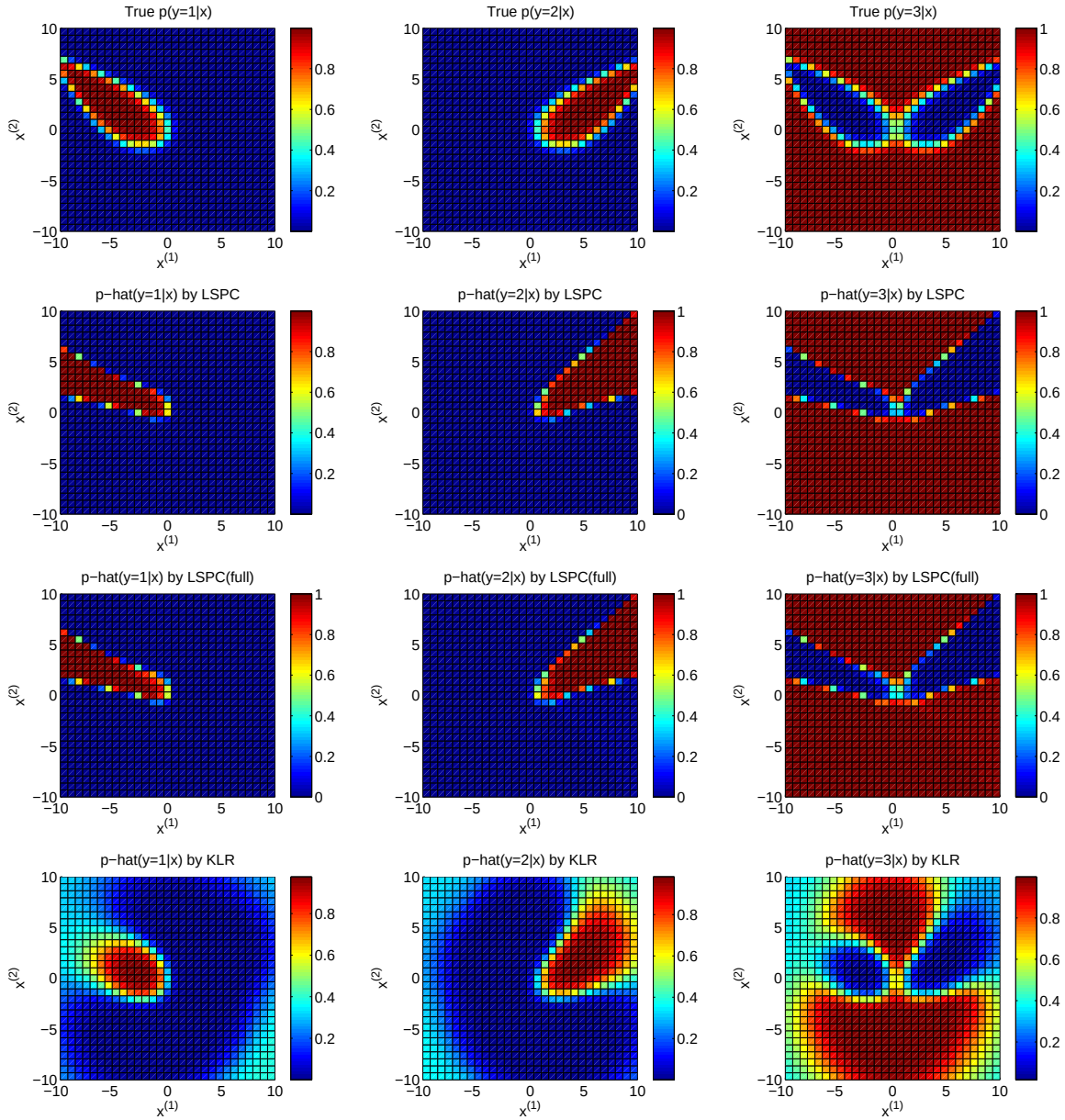


Figure 6: Illustrative examples. The plots show the true class-posterior probabilities $p(y|\boldsymbol{x})$, their estimates by LSPC, LSPC(full), and KLR from top to bottom, and $y = 1, 2, 3$ from left to right.

hand, KLR outputs values close to one outside the training region since $\exp(0) = 1$; then they are normalized and thus are reduced to $1/c$.

The classification results based on the true class-posterior probabilities and their estimates obtained by LSPC, LSPC(full), and KLR are plotted in Figure 5. This shows that all the method gave reasonable classification results.

3.2 Performance Comparison

Next, we evaluate the classification accuracy and computation time of each method using the following multi-class classification datasets taken from the *LIBSVM* web page [5]:

- **mnist**: Input dimensionality is 717 and the number of classes is 10.
- **usps**: Input dimensionality is 256 and the number of classes is 10.
- **satimage**: Input dimensionality is 36 and the number of classes is 6.
- **letter**: Input dimensionality is 16 and the number of classes is 26.

We investigated the classification accuracy and computation time of LSPC, LSPC(full), and KLR. For given n and c, we randomly chose $n_y = \lfloor n/c \rfloor$ training samples from each class y, where $\lfloor t \rfloor$ is the largest integer not greater than t. In the first set of experiments, we fixed the number of classes c to the original number shown above, and changed the number of training samples as $n = 100, 200, 500, 1000, 2000$. In the second set of experiments, we fixed the number of training samples to $n = 1000$, and changed the number of classes c—samples only in the first c classes in the dataset are used. The classification accuracy is evaluated using 100 test samples randomly chosen from each class. The computation time is measured by the CPU computation time required for training each classifier when the Gaussian width and the regularization parameter chosen by cross-validation were used.

The experimental results are summarized in Figure 7 and Figure 8. The left column in Figure 7 shows that when n is increased, the classification error for all the methods tends to decrease, and LSPC, LSPC(full), and KLR performed similarly well. The right column in Figure 7 shows that when n is increased, the computation time tends to grow for all the methods. LSPC is faster than KLR by two orders of magnitude. The left column in Figure 8 shows that when c is increased, the classification error tends to increase for all the methods, and LSPC, LSPC(full), and KLR behaved similarly well. The right column in Figure 8 shows that when c is increased, the computation time of KLR tends to grow, while that of LSPC is kept constant or even it tends to slightly decrease. This happened because the number of samples in each class decreases when c is increased, and the computation time of LSPC is governed by the number of samples in *each* class, not by the *total* number of samples (see Section 2.3).

Overall, the computation of LSPC was shown to be faster than that of KLR by orders of magnitude, while LSPC and KLR were shown to be comparable to each other in terms of the classification accuracy. LSPC and LSPC(full) were shown to possess similar

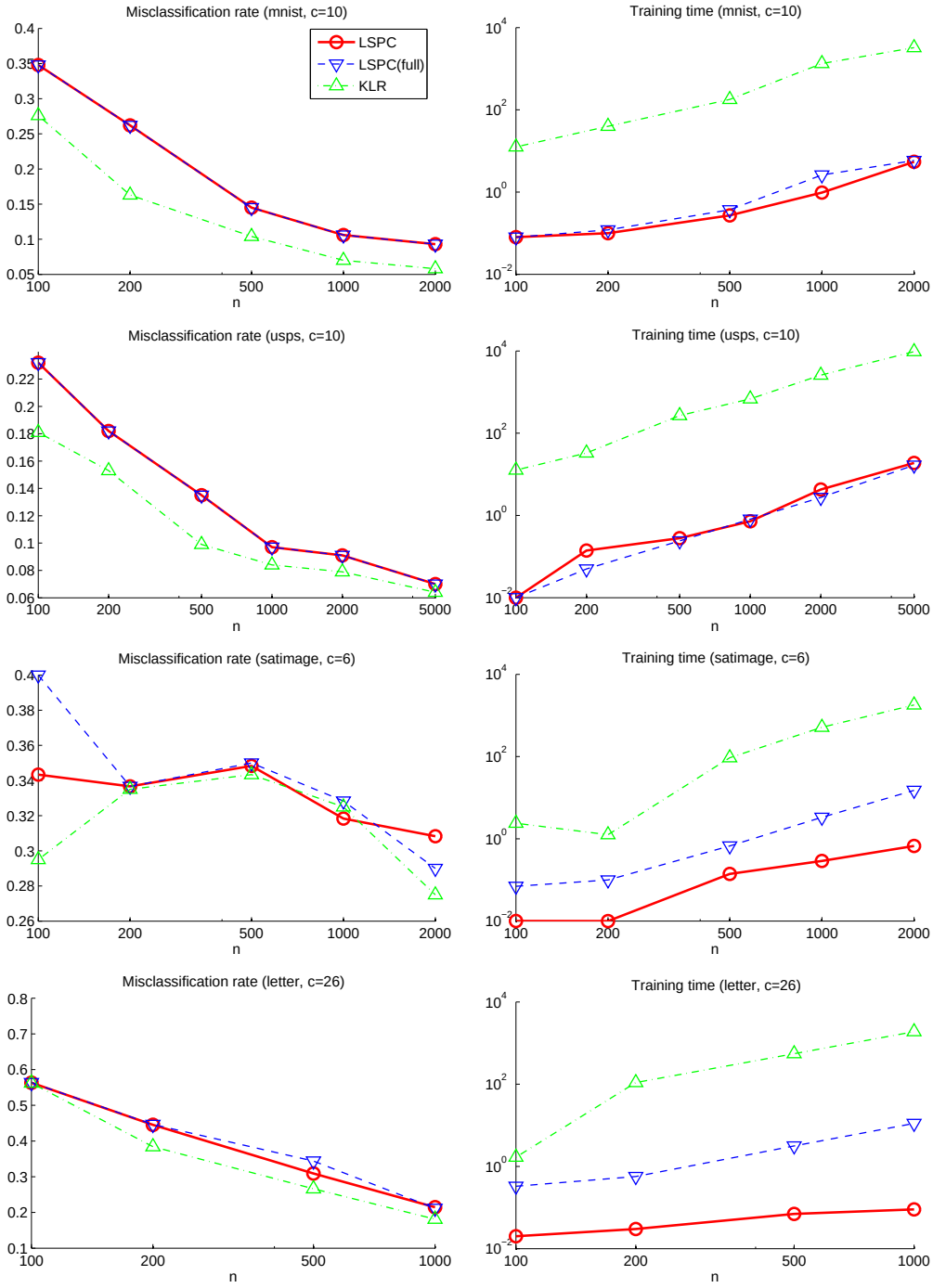


Figure 7: Misclassification rate (in percent, left) and computation time (in second, right) as functions of the number of training samples n. From top to bottom, the graphs correspond to the 'mnist', 'usps', 'satimage', and 'letter' datasets.

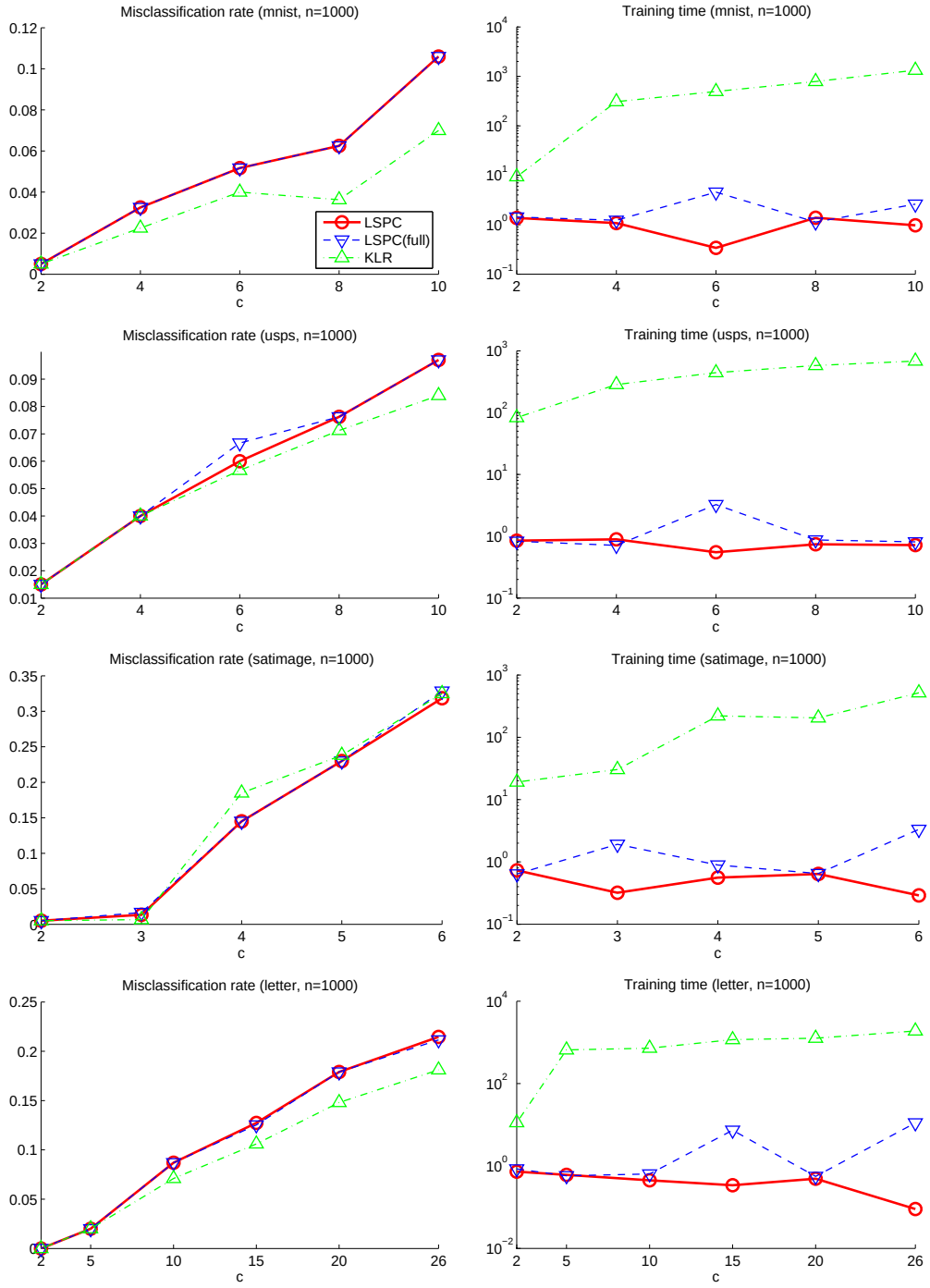


Figure 8: Misclassification rate (in percent, left) and computation time (in second, right) as functions of the number of classes c. From top to bottom, the graphs correspond to the 'mnist', 'usps', 'satimage', and 'letter' datasets.

classification performance, and thus a computationally efficient version, LSPC, would be more preferable in practice.

4 Discussion and Conclusion

Recently, various efficient algorithms for computing the solution of logistic regression have been developed for high-dimensional sparse data [22, 10]. However, for dense data, using standard non-linear optimization techniques such as Newton's method or quasi-Newton methods seem to be a common choice [15, 23]. The performance of these general-purpose non-linear optimizers has been improved in the last decade, but computing the solution of logistic regression for a large number of dense training samples is still a challenge problem.

In this paper, we proposed a simple probabilistic classification algorithm called *Least-Squares Probabilistic Classifier (LSPC)*. LSPC employs a linear combination of Gaussian kernels centered at training points for modeling the class-posterior probability and the parameters are learned by least-squares. Notable advantages of LSPC are that its solution can be computed *analytically* just by solving a system of linear equations and training can be carried out separately in a class-wise manner. In experiments, we showed that LSPC is faster than kernel logistic regression (KLR) in computation time by two orders of magnitude, with comparable accuracy.

The computational efficiency of LSPC was brought by the combination of appropriate model choice and loss function. More specifically, KLR uses a log-linear combination of kernel functions and its parameters are learned by regularized maximum likelihood. In this log-linear maximum likelihood formulation, normalization of the model is essential to avoid the likelihood diverging to infinity. Thus the likelihood function tends to be complicated and numerically solving the optimization problem may be unavoidable. On the other hand, in LSPC, we chose a linear combination of Gaussian kernel functions for modeling the class-posterior probability and its parameters are learned by regularized least-squares. This combination allowed us to obtain the solution analytically. When Newton's method (more specifically, *iteratively reweighted least-squares*) is used for learning the KLR model, a system of linear equations needs to be solved in *every* iteration until convergence [15]. On the other hand, LSPC requires to solve a system of linear equations only once.

We chose to separate the kernel for inputs and outputs, and adopted the delta kernel for outputs (see Eq.(5)). This allowed us to perform the training of LSPC in a class-wise manner. We showed that this contributes to reducing the training time particularly in multi-class classification problems. We note that this model choice is essentially the same as that of KLR[2].

We further proposed to reduce the number of kernels when "localized" kernels such as the Gaussian kernel (6) is used. Through the experimental evaluation in Section 3, we found that this heuristic model simplification does not degrade the classification accuracy,

[2]The number of parameters in LSPC with model (5) is cn, while the number of parameters in KLR is $(c-1)n$ since the normalization ('sum-to-one') constraint is incorporated in the training phase.

but reduces the computation time.

It is straightforward to show that solutions for *all* regularization parameter values (i.e., the *regularization path*, see [9, 14]) can be computed efficiently in LSPC. Let us consider the eigendecomposition of the matrix $\widehat{\boldsymbol{H}}^{(y)}$ (see Eq.(9)):

$$\widehat{\boldsymbol{H}}^{(y)} = \sum_{\ell=1}^{n_y} \gamma_\ell \boldsymbol{\psi}_\ell \boldsymbol{\psi}_\ell^\top,$$

where $\{\boldsymbol{\psi}_\ell\}_{\ell=1}^{n_y}$ are the eigenvectors of $\widehat{\boldsymbol{H}}^{(y)}$ associated with the eigenvalues $\{\gamma_\ell\}_{\ell=1}^{n_y}$. Then, the solution $\widehat{\boldsymbol{\alpha}}^{(y)}$ can be expressed as

$$\widehat{\boldsymbol{\alpha}}^{(y)} = (\widehat{\boldsymbol{H}}^{(y)} + \lambda \boldsymbol{I}_{n_y})^{-1} \widehat{\boldsymbol{h}}^{(y)} = \sum_{\ell=1}^{n_y} \frac{\widehat{\boldsymbol{h}}^\top \boldsymbol{\psi}_\ell}{\gamma_\ell + \lambda} \boldsymbol{\psi}_\ell.$$

Since $(\widehat{\boldsymbol{h}}^\top \boldsymbol{\psi}_\ell)\boldsymbol{\psi}_\ell$ is common to all λ, we can compute the solution $\widehat{\boldsymbol{\alpha}}^{(y)}$ for all λ efficiently by eigendecomposing the matrix $\widehat{\boldsymbol{H}}^{(y)}$ *once* in advance. Although eigendecomposition of $\widehat{\boldsymbol{H}}^{(y)}$ may be computationally slightly more demanding than solving a system of linear equations of the same size, this approach would be useful, e.g., when computing the solutions for various values of λ in the cross-validation procedure.

When n_y is large, we may further reduce the computational cost and memory space by using only a subset of kernels.

$$\begin{aligned} q(y|\boldsymbol{x};\boldsymbol{\alpha}) &= \sum_{\ell=1}^{b_y} \alpha_\ell^{(y)} K(\boldsymbol{x}, \boldsymbol{c}_\ell^{(y)}), \\ \widehat{H}_{\ell,\ell'}^{(y)} &= \frac{1}{n} \sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{c}_\ell^{(y)}) K(\boldsymbol{x}_i, \boldsymbol{c}_{\ell'}^{(y)}), \\ \widehat{h}_\ell^{(y)} &= \frac{1}{n} \sum_{i=1}^{n_y} K(\boldsymbol{x}_i^{(y)}, \boldsymbol{c}_\ell^{(y)}), \end{aligned}$$

where b_y is a constant chosen to be smaller than n_y and $\{\boldsymbol{c}_\ell^{(y)}\}_{\ell=1}^{b_y}$ is a subset of $\{\boldsymbol{x}_\ell^{(y)}\}_{\ell=1}^{n_y}$. This would be a useful heuristic when a huge number of samples are used for training.

Another option for reducing the computation time when the number of samples is very large would be the *stochastic gradient descent* method [1]. That is, starting from some initial parameter value, gradient descent is carried out only for a randomly chosen single sample in each iteration. Since our optimization problem is convex, convergence to the global solution is guaranteed (in a probabilistic sense) by stochastic gradient descent.

We focused on using the delta kernel for class labels (see Section 2.3). We expect that designing appropriate kernel functions for class labels would be useful for improving the classification performance, e.g., in the context of *multi-task learning* [4, 2, 21]. We will pursue this direction in our future work.

Acknowledgments

The author thanks Dr. Ryota Tomioka, Mr. Jaak Simm, and Dr. Hirotaka Hachiya for their valuable comments. This work was supported by AOARD, SCAT, and the JST PRESTO program.

References

[1] S. Amari, "Theory of adaptive pattern classifiers," IEEE Transactions on Electronic Computers, vol.EC-16, no.3, pp.299–307, 1967.

[2] B. Bakker and T. Heskes, "Task clustering and gating for Bayesian multitask learning," Journal of Machine Learning Research, vol.4, pp.83–99, 2003.

[3] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, NY, USA, 2006.

[4] R. Caruana, L. Pratt, and S. Thrun, "Multitask learning," Machine Learning, vol.28, pp.41–75, 1997.

[5] C.C. Chang and C.J. Lin, "LIBSVM: A library for support vector machines," tech. rep., Department of Computer Science, National Taiwan University, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[6] R. Collobert and S. Bengio., "SVMTorch: Support vector machines for large-scale regression problems," Journal of Machine Learning Research, vol.1, pp.143–160, 2001.

[7] C. Cortes and V. Vapnik, "Support vector networks," Machine Learning, vol.20, pp.273–297, 1995.

[8] R.O. Duda, P.E. Hart, and D.G. Stor, Pattern Classification, Wiley, New York, 2001.

[9] B. Efron, T. Hastie, R. Tibshirani, and I. Johnstone, "Least angle regression," The Annals of Statistics, vol.32, no.2, pp.407–499, 2004.

[10] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, "LIBLINEAR: A library for large linear classification," Journal of Machine Learning Research, vol.9, pp.1871–1874, 2008.

[11] R.E. Fan, P.H. Chen, and C.J. Lin, "Working set selection using second order information for training SVM," Journal of Machine Learning Research, vol.6, pp.1889–1918, 2005.

[12] V. Franc and S. Sonnenburg, "Optimized cutting plane algorithm for large-scale risk minimization," Journal of Machine Learning Research, vol.10, pp.2157–2192, 2009.

[13] G.M. Fung and O.L. Mangasarian, "Multicategory proximal support vector machine classifiers," Machine Learning, vol.59, no.1–2, pp.77–97, 2005.

[14] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine," Journal of Machine Learning Research, vol.5, pp.1391–1415, 2004.

[15] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, 2001.

[16] T. Joachims, "Making large-scale SVM learning practical," in Advances in Kernel Methods—Support Vector Learning, ed. B. Schölkopf, C.J.C. Burges, and A.J. Smola, pp.169–184, The MIT Press, Cambridge, MA, 1999.

[17] T. Joachims, "Training linear SVMs in linear time," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2006), pp.217–226, 2006.

[18] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," Journal of Machine Learning Research, vol.10, pp.1391–1445, Jul. 2009.

[19] T. Kanamori, T. Suzuki, and M. Sugiyama, "Condition number analysis of kernel-based density ratio estimation," tech. rep., arXiv, 2009.

[20] T. Kanamori, T. Suzuki, and M. Sugiyama, "Theoretical analysis of density ratio estimation," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E93-A, no.4, pp.787–798, 2010.

[21] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Conic programming for multi-task learning," IEEE Transactions on Knowledge and Data Engineering, vol.22, no.7, pp.957–968, 2010.

[22] K. Koh, S.J. Kim, and S.P. Boyd, "An interior-point method for large-scale l_1-regularized logistic regression," Journal of Machine Learning Research, vol.8, pp.1519–1555, 2007.

[23] T.P. Minka, "A comparison of numerical optimizers for logistic regression," tech. rep., Microsoft Research, 2007.

[24] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in Advances in Kernel Methods—Support Vector Learning, ed. B. Schölkopf, C.J.C. Burges, and A.J. Smola, pp.169–184, The MIT Press, Cambridge, MA, 1999.

[25] J. Platt, "Probabilities for SV machines," in Advances in Large Margin Classifiers, ed. A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, The MIT Press, Cambridge, MA, 2000.

[26] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," Advances in Learning Theory: Methods, Models and Applications, ed. J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, NATO Science Series III: Computer & Systems Sciences, vol.190, Amsterdam, the Netherlands, pp.131–154, IOS Press, 2003.

[27] M. Schmidt, minFunc, 2005. `http://people.cs.ubc.ca/~schmidtm/Software/minFunc.html`.

[28] B. Schölkopf and A.J. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.

[29] J.A.K. Suykens, T.V. Gestel, J.D. Brabanter, B.D. Moor, and J. Vandewalle, Least Squares Support Vector Machines, World Scientific Pub. Co., Singapore, 2002.

[30] Y. Tang and H.H. Zhang, "Multiclass proximal support vector machines," Journal of Computational and Graphical Statistics, vol.15, no.2, pp.339–355, 2006.

[31] C.H. Teo, Q. Le, A. Smola, and S.V.N. Vishwanathan, "A scalable modular convex solver for regularized risk minimization," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2007), pp.727–736, 2007.

[32] I. Tsang, J. Kwok, and P.M. Cheung, "Core vector machines: Fast SVM training on very large data sets," Journal of Machine Learning Research, vol.6, pp.363–392, 2005.

[33] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, NY, USA, 1998.

[34] T.F. Wu, C.J. Lin, and R.C. Weng, "Probability estimates for multi-class classification by pairwise coupling," Journal of Machine Learning Research, vol.5, pp.975–1005, 2004.

[35] M. Yamada, M. Sugiyama, G. Wichern, and J. Simm, "Improving the accuracy of least-squares probabilistic classifiers," IEICE Transactions on Information and Systems, 2011. submitted.

Application of Covariate Shift Adaptation Techniques in Brain Computer Interfaces

Yan Li (soncyme@hi.pi.titech.ac.jp)
Department of Computational Intelligence and Systems Science,
Tokyo Institute of Technology
and JST CREST

Hiroyuki Kambara
Precision and Intelligence Laboratory, Tokyo Institute of Technology
and JST CREST

Yasuharu Koike
Precision and Intelligence Laboratory, Tokyo Institute of Technology
and JST CREST

Masashi Sugiyama (sugi@cs.titech.ac.jp)
http://sugiyama-www.cs.titech.ac.jp/∼sugi/
Department of Computer Science, Tokyo Institute of Technology
JST PRESTO

Abstract

A phenomenon often found in session-to-session transfers of Brain Computer Interfaces (BCIs) is non-stationarity. It can be caused by fatigue and changing attention level of the user, differing electrode placements, varying impedances, among other reasons. Covariate shift adaptation is an effective method which can adapt to the testing sessions without the need for labeling the testing session data. The method was applied on a BCI Competition III dataset. Results showed that covariate shift adaptation compares favorably with methods used in the BCI competition in coping with non-stationarities. Specifically, bagging combined with covariate shift helped to increase stability, when applied to the competition dataset. An online experiment also proved the effectiveness of bagged covariate shift method. Thus, it can be summarized that covariate shift adaptation is helpful to realize adaptive BCI systems.

Keywords

brain-computer interface, covariate shift adaptation, bagging

1 Introduction

A Brain Computer Interface (BCI) is a novel augmentative tool which allows a user to express his or her will without muscle exertion, provided that the brain signals are translated properly. However, it may be difficult to recognize the electroencephalography (EEG) patterns under a fixed algorithm because of high non-stationarity of the EEG signals. The factors causing non-stationarity include changes in user attention level, user fatigue, and small differences in electrode position [1]. One notable representation of non-stationarity is that EEG feature distributions change from one session to another, which illustrates the non-stationary nature of the BCI signal and provides a rationale for the design of an adaptive BCI system [2].

Moreover, a good BCI system should be bi-directional in communication with the user. Besides providing visual/auditory feedback to a user, the system should be able to adapt to the user, possibly with an adaptive translation algorithm. Several studies have been conducted on adaptive BCI systems with positive results. Vidaurre et al. adopted an online updated classifier by adaptive estimation of the information matrix (ADIM) as well as an adaptive LDA with Kalman filtering [3, 4]. Blumberg et al. developed Adaptive Linear Discriminant Analysis, updating mean values and covariances continuously in time for different motor imaginary tasks [5]. However, most of the adaptive methods are based on supervised learning techniques (e.g., [1, 3, 4]), which need labeled test samples and are, thus, costly. Covariate shift adaptation is a method which can overcome this shortcoming, assuming that the input distributions of training and testing sessions are different while the conditional distribution of output given input remains unchanged [6]. Nevertheless, the plain covariate shift adaptation technique is rather unstable due to large variances.

To cope with this problem, we propose a novel method which combines covariate shift adaptation and bagging [7] [8]. Through applications on benchmark data, we demonstrate the effectiveness of the proposed approach.

2 Methods

In this section, we review baseline methods as well as our proposed approach.

2.1 Feature Extraction by CSP and the Baseline Classifier LDA

Common Spatial Patterns (CSP) is one of the most popular spatial filters of multi-channel EEG-based BCI in recent years. In contrast to other spatial filters, CSP generates features ready to be fed into the classifier. After band-pass filtering the EEG signals in the frequency range of interest, high or low signal variance reflects strong or attenuated rhythmic activity, respectively [9]. When classifying EEG into two tasks, CSP maximizes the variance of one class while minimizing the variance of the other and, thus, reflects the task specific activation patterns. Some example of CSP applications can be found in [9, 10, 11, 12].

Linear Discriminant Analysis (LDA) is a popular classification method in BCI application [13]. LDA can be realized by a linear least-squares method if the target labels $\{y_i\}_{i=1}^N$ corresponding to the feature vectors $\{x_i\}_{i=1}^N$ for class C_1 are set to be $-(N_1+N_2)/N_1$ and the target labels of class C_2 are set to $(N_1+N_2)/N_2$, where N_1 and N_2 are numbers of samples of classes C_1 and C_2, respectively. More specifically, for a linear model

$$\hat{f}(x;\theta) = \theta_0 + \sum_{i=1}^{d} \theta_i x^{(i)},$$

where $x^{(i)}$ is the ith element of an d-dimensional feature vector x, the parameters θ are learned by the least-squares method:

$$\min_{\theta} \sum_{i=1}^{N} \left(y_i - \hat{f}(x_i;\theta) \right)^2.$$

The least-squares solution is given as

$$\hat{\theta}_{LDA} = (X^T X)^{-1} X^T y,$$

where

$$X \equiv \begin{pmatrix} 1 & x_1^T \\ 1 & x_2^T \\ \vdots & \vdots \\ 1 & x_N^T \end{pmatrix},$$

$y = (y_1, y_2, ..., y_N)$, and X^T denotes the transpose of X.

2.2 Covariate Shift Adaptation by IWLDA

Covariate shift is defined as the situation where the training input points and test input points follow different distributions while the conditional distribution of output values given input points is unchanged [6]. A prime example of covariate shift in EEG-based BCIs occurs when, given different experimental sessions of the same imaginary tasks, event-related synchronization/desynchronization cortical distributions remain unchanged, but the means and variances shift in the feature distribution for each task.

Under covariate shift, ordinary Linear Discriminant Analysis (LDA) is not consistent [14, 6], i.e., even when infinitely many training samples are provided, one cannot obtain the optimal solution. To cope with this problem, Importance Weighted Linear Discriminant Analysis (IWLDA) was proposed [15, 6].

IWLDA is an extension of LDA based on the concept of *importance sampling*. The importance is defined as the ratio of test and training input densities:

$$w(x) = \frac{p_{te}(x)}{p_{tr}(x)}.$$

After the introduction of the importance and a regularizer, the parameters are learned as

$$\min_{\theta} \sum_{i=1}^{N} w(x_i) \left(y_i - \hat{f}(x_i; \theta) \right)^2 + \lambda \|\theta\|^2,$$

where λ (≥ 0) is the regularization parameter. The IWLDA solution is given by

$$\hat{\theta}_{IWLDA} = (X^T D X + \lambda I)^{-1} X^T D y,$$

where D is the diagonal matrix with the i-th diagonal element $D_{i,i} = w(x_i)$ and I is the identity matrix. IWLDA is proved to be consistent even in the presence of covariate shift.

2.3 Model Selection by IWCV

The IWLDA method contains a regularization parameter λ and this needs to be chosen appropriately for obtaining better performance. To this end, cross-validation is commonly used, which is known to be an unbiased estimator of the generalization error. However, ordinary cross-validation is no longer unbiased in the presence of covariate shift; importance-weighted cross validation (IWCV) is instead unbiased under covariate shift [6].

More specifically, we first divide the training samples $\{z_i \mid z_i = (x_i, y_i)\}_{i=1}^{N}$ into k disjoint subsets $\{\mathcal{Z}_r\}_{r=1}^{k}$ (we use $k = 5$ in the experiments). Then the parameter $\hat{\theta}_r$ is obtained using $\{\mathcal{Z}_j\}_{j \neq r}$ (i.e., without $\mathcal{Z}_r$) by IWLDA and its mean test error for the remaining samples $\mathcal{Z}_r$ is computed:

$$\frac{1}{|\mathcal{Z}_r|} \sum_{(x,y) \in \mathcal{Z}_r} w(x) \mathrm{loss} \left(\hat{f}(x; \hat{\theta}_r), y \right),$$

where

$$\mathrm{loss}\,(\widehat{y}, y) = \begin{cases} \frac{1}{2}(1 - \mathrm{sign}) & \text{Classification} \\ (\widehat{y} - y)^2 & \text{Regression} \end{cases}$$

We repeat this procedure for $r = 1, 2, \ldots, k$ and choose the regularization parameter λ so that the average of the above mean test error over all r is minimized.

2.4 Direct Importance Estimation by KLIEP or uLSIF

For computing the IWLDA solution and performing model selection by IWCV, the values of the importance are required, which are usually unknown. A naive approach to importance estimation would be to first estimate the training and testing densities separately from training input samples $\{x_i^{tr}\}_{i=1}^{n_{tr}}$ and testing input samples $\{x_j^{te}\}_{=1}^{n_{te}}$, then estimate the importance by taking the ratio of the estimated densities. However, density estimation is known to be a difficult problem, particularly in high-dimensional cases. Therefore, this naive approach may not be effective; directly estimating the importance without estimating the densities would be more promising [15].

2.4.1 KLIEP

KLIEP (Kullback-Leibler Importance Estimation Procedure) is a method to estimate the importance directly. First, the importance is modeled as

$$\hat{w}(x) = \sum_{l=1}^{b} \alpha_l \exp\left(-\frac{\|x - c_l\|^2}{2\sigma^2}\right),$$

where $\{\alpha_l\}_{l=1}^{b}$ are coefficients to be learned ($\alpha_l \geq 0$ for $l = 1, 2, \ldots, b$), $\{c_l\}_{l=1}^{b}$ are chosen randomly from $\{x_j^{te}\}_{=1}^{n_{te}}$, and the number of parameters is set to $b = \min(100, n_{te})$ in the experiments. The kernel width σ can be optimized by cross validation (see [15]).

Using the above importance model, we can obtain an estimate of the test input density as

$$\hat{p}_{te}(x) = \hat{w}(x) p_{tr}(x).$$

Based on this expression, $\{\alpha_l\}_{l=1}^{b}$ are determined so that the Kullback-Leibler divergence from $p_{te}(x)$ to $\hat{p}_{te}(x)$ is minimized.

$$\begin{aligned} \mathrm{KL}[p_{te}(x)||\hat{p}_{te}(x)] &= \int_D p_{te}(x) \log \frac{p_{te}(x)}{\hat{w}(x) p_{tr}(x)} dx \\ &= \int_D p_{te}(x) \log \frac{p_{te}(x)}{p_{tr}(x)} dx - \int_D p_{te}(x) \log \hat{w}(x) dx. \end{aligned}$$

Based on this, the optimization criterion of KLIEP is given as follows (see [15] for details):

$$\max_{\{\alpha_l\}_{l=1}^{b}} \sum_{j=1}^{n_{te}} \log \left[\sum_{l=1}^{b} \alpha_l \exp\left(-\frac{\|x_j^{te} - c_l\|^2}{2\sigma^2}\right) \right]$$

subject to

$$\sum_{i=1}^{n_{tr}} \sum_{l=1}^{b} \alpha_l \exp\left(-\frac{\|x_i^{tr} - c_l\|^2}{2\sigma^2}\right) = n_{tr} \text{ and } \alpha_1, \alpha_2, \ldots, \alpha_b \geq 0.$$

2.4.2 uLSIF

uLSIF (unconstrained Least-Squares Importance Fitting) [16] also estimates the importance directly. The modeling of the importance is the same as Equation (1), but the parameters are determined by minimizing the squared error:

$$\begin{aligned} J_0(\alpha) &= \frac{1}{2} \int \left(\hat{w}(x) - \frac{p_{te}(x)}{p_{tr}(x)} \right)^2 p_{tr}(x) dx \\ &= \frac{1}{2} \int \hat{w}(x)^2 p_{tr}(x) dx - \int \hat{w}(x) p_{te}(x) dx + C, \end{aligned}$$

where $C = \frac{1}{2} \int w(x) p_{te}(x) dx$ is a constant and thus can be ignored. As presented in detail in [16], the solution of uLSIF is given by

$$\hat{\alpha} = \max(0_b, (\hat{H} + \lambda I)^{-1} \hat{h}),$$

where

$$\hat{H}_{l,l'} = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \exp\left(-\frac{\|x_i^{tr} - c_l\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|x_i^{tr} - c_{l'}\|^2}{2\sigma^2}\right),$$

$$\hat{h}_l = \frac{1}{n_{te}} \sum_{j=1}^{n_{te}} \exp\left(-\frac{\|x_i^{te} - c_l\|^2}{2\sigma^2}\right).$$

2.5 Bagged IWLDA

IWLDA combined with KLIEP or uLSIF is shown to perform well under covariate shift. However, a weakness of this approach is that IWLDA can still produce a large-variance estimator, causing instability.

To ease this problem, we propose Bagged Importance Weighted Linear Discriminant Analysis (BIWLDA) which combines bagging (short for "Bootstrap aggregating") [7] [18] and IWLDA (with λ chosen by IWCV) to improve the stability of classifiers.

Bagging is the parallel approach to ensemble construction, which combines independently constructed accurate and diverse base learners [17]. The idea behind bagging is that averaging the predictions will lead to the improvement of classification accuracy, particularly variance reduction. Since plain covariate-shift adaptation methods tend to produce high-variance estimators, combining them with bagging would be promising.

More specifically, the proposed BIWLDA procedure is summarized as follows:

1. Randomly take M trials out of the whole N-sized training set, with $M = 0.8N$;
2. Train IWLDA (with λ chosen by IWCV) on the re-sampled training set;
3. Repeat 1) and 2) for 30 times;
4. Average the 30 predictors.

The classifiers realized with KLIEP with and without bagging are named BIWLDA1 and IWLDA1 respectively, while the classifiers realized with uLSIF with and without bagging are named BIWLDA2 and IWLDA2 respectively.

3 Experiments on BCI Competition III Dataset IVc

In this section, we show the experimental results on BCI Competition III Dataset IVc.

3.1 Dataset

Dataset IVc [19] in BCI competition III was recorded from one healthy subject. Visual cues of 3.5 seconds indicated which of the following 3 motor imageries the subject should perform: left hand, right foot and tongue. For training, 210 trials were provided with labels of left hand respectively right hand. 420 test trials were recorded 4 hours after

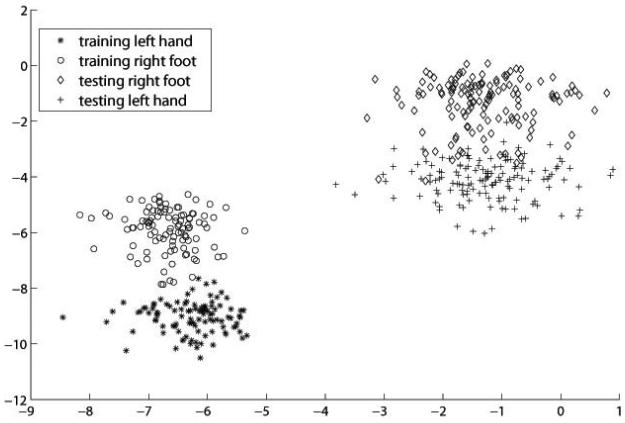


Figure 1: Different feature distributions between training and testing sessions in BCI Competition III Dataset IVc.

the training sessions. The testing sessions were similar to the training sessions, but the motor imagery had to be performed for 1 second only, compared to the 3.5 seconds in the training sessions. The other difference was that the class tongue was replaced by the class relax.

118 EEG channels were measured at positions of the extended international 10-20 system. Signals were band-pass filtered from 0.05 to 200 Hz and then digitized at 1000 Hz with 16 bit accuracy. The data downsampled to 100 Hz was used for analysis.

Since left hand and right foot imagery tasks were both included in the training and testing sessions, and these two sessions had a long time interval in between, checking these two classes would reveal whether there is a different feature distribution between two sessions.

3.2 Investigation of Feature Distributions and Improved Algorithms

It has been shown in many previous studies that filtering must precede CSP in order to make CSP optimal for the separation of two classes. After plotting and observing the power spectrum, we decided to apply only bandpass-filtering, from 12 to 14 Hz, though the competition winner considered a broader bandpass-filtering [18]. Also the competition winner claimed the optimal dimension of CSP was three, and this result was verified by us.

By plotting the features extracted by CSP for left hand and right foot imaginary movements (see Figure 1), it can be seen that a different feature distribution did occur and that there was a need to shift the classification boundary. Note that, for ease in visualization, only two dimensions of calculated features, which correspond to the two most important CSP filters from the training set, were drawn. Figure 2 shows the full

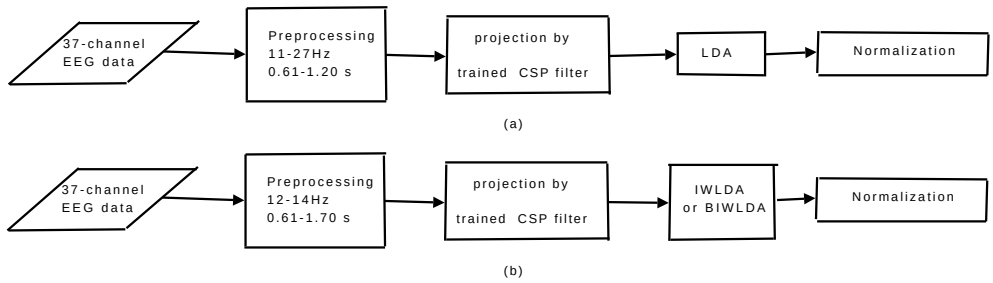


Figure 2: (a) Flowchart of the first winner; (b) Flowchart of the covariate shift methods.

Table 1: Testing results of LDA, IWLDA, BLDA and BIWLDA.

Method	LDA	IWLDA1	IWLDA2	BLDA	BIWLDA1	BIWLDA2
MSE (mean)	0.246	0.1726	0.1183	0.519	0.0994	0.1165
MSE(std)	0	0.0563	0.0014	0.6023	0.0142	0.0346

(a) MSEs by LDA (baseline), IWLDA, BLDA and BIWLDA.

Method	LDA	IWLDA1	IWLDA2	BLDA	BIWLDA1	BIWLDA2
Accuracy (mean)	0.8429	0.9336	0.9539	0.8115	0.965	0.9546
Accuracy (std)	1.17E-16	0.035	0.0011	0.1322	0.0098	0.0343

(b) Accuracy by LDA (baseline), IWLDA, BLDA and BIWLDA.

two task classification process of the first winner as well as our algorithm. The main difference lies in the replacement of LDA by IWLDA or BIWLDA.

3.3 Results

Table 1 shows the testing results of all methods with the same data preprocessing. The means and standard deviations were based on ten iterations of testing. From Table 2(a), it can be seen that the covariate shift adaptation methods worked very well. Among them, BIWLDA1 proved to be much more stable than IWLDA1, while IWLDA2 and BIWLDA2 were comparable to each other. However, in real application, normalization of the outputs is impossible. Furthermore, as an additional evaluation criterion, classification accuracy was also calculated, as shown in Table 2(b).

It may be not appropriate for us to claim that our methods worked better than the method of competition winner, since this dataset contained three classes, and our methods only worked better in separating two of them. However, we think our method solved the non-stationarity problem caused by session-to-session transfer more efficiently, which can be revealed from the accuracy listed in table 2(b).

When estimating the importance, parameter b was established as $min(100, n_{te})$ (see section 2.4.1), where n_{te} is the number of testing trials. 100 trials were randomly chosen from the testing set in cases where n_{te} was greater than 100. To determine the effects

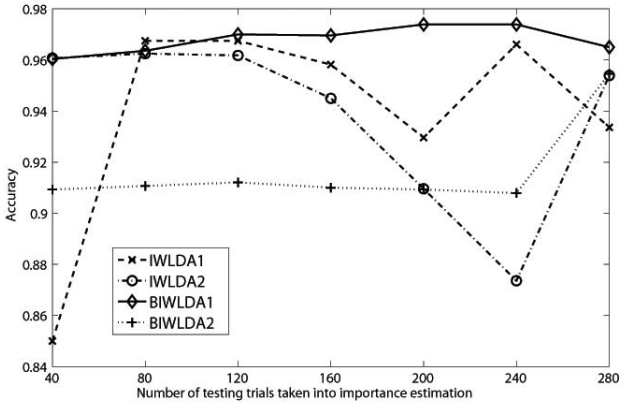


Figure 3: The first 40, 80,..., 280 trials were taken into importance estimation, with IWLDA1, IWLDA2, BIWLDA1 and BIWLDA2.

brought on by n_{te}, we tested with the first 40, 80,..., 240 and all 280 trials (with training set unchanged), which may be seen as a pseudo-online importance estimation scenario. The testing was repeated 10 times with four covariate shift methods, and the averaged accuracy was plotted in Figure 3. It can be concluded that BIWLDA1 is the most stable method for different numbers of testing trials taken for importance estimation.

3.4 Online application of bagged covariate shift method

From its application on the benchmark dataset, it is not difficult to see that BIWLDA1 performed well in terms of both accuracy and stability. Moreover, we wished to test its effectiveness in real online applications and, thus, performed an online experiment on three healthy female subjects (age 38, 23, 30). For the experiment, we used a G.tec USBamp system controlled with the software BCI2000 [20]. EEG was recorded using 10 or 15 electrodes positioned at locations FC_3, FC_1, FC_z, FC_2, FC_4, C_3, C_1, C_z, C_2, C_4, CP_3, CP_1, CP_z, CP_2, and CP_4 of the international 10-20 system. For subject 3 the former 10 channels were installed. Data was sampled at 256 Hz and the feedback was updated every 1 second. Pre-feedback of each trial was set as 2 seconds and the feedback time length was decided as 3 seconds.

For the online experiment, a ball was displayed traveling at a constant speed from the left to the right of a screen. Vertical position (distance from the midline of the screen) of the ball served as feedback, changing according to the classification output of the previous second. Subjects were asked to imagine moving their left hand or both hands and both feet to direct the ball downwards and upwards, respectively, and position it to hit a target bar at the right of the screen.

The experiment was carried out in two parts, separated by one or two days. In the first part, the subjects were trained to gain familiarity with both offline and online experiments, obtaining trial accuracies above 80%. The algorithm was two most discriminative features casted by CSP and classified with LDA. In the second part, only online experi-

Table 2: Mutual information estimated before and after BIWLDA1 updated

	Session before BIWLDA1		Sessions after BIWLDA1	
	MI(old)	MI(updated)	MI(old)	**MI(updated)**
subject 2	**0.3052**	0.3527	0.3452	**0.3660**
subject 3	**0.1005**	0.1679	0.1511	**0.1789**

ments were conducted, and the subjects were given a few minutes to practice before the experiment started. After the first session, BIWLDA1 was run to adjust the LDA coefficients according to the session transfer from the best performed online session recorded on the previous day. After running BIWLDA1, another two sessions of experiments were continued. Each online session consisted of 43 trials, and when running BIWLDA1, three 1 second (namely second 2, 3, 4 in one trial) non-overlapped windows were cut from each trial which means 129 one-second samples were obtained.

The trial accuracy was improved from 72.09% to 80.23% (subject 2) , and from 66% to 78% (subject 3) after adjustment of LDA coefficients. Results of only two subjects are presented here because subject 1 reached the same trial accuracy as the previous day at 83%, showing no sign of non-stationarities. In order to verify that these improvements were not due to the learning process itself, we applied the coefficients before and after BIWLDA1 adjustment to all the online sessions, analyzing the mutual information [21] between the targets and the outputs of online data (still 129 samples per session). Using mutual information as the evaluation criteria is natural because it takes not only the sign of the output into account but also the amplitude, which, in turn, is used to set the distance between the ball and vertical midline of the screen.

From Table 2, it can be concluded that these improvements cannot be attributed to the learning process because in sessions either before or after the BIWLDA1 adjustment, the updated coefficients generated higher mutual information and with values that are quite similar. Note in Table 2 that because the session before BIWLDA1 used old coefficients, the numbers of MI(old) are written in bold, as is MI(updated) in the session after BIWLDA1 adjustment.

Figure 4 gives a more direct description about the session-to-session transfer of feature distribution with subject 2. In it the training tasks meant the tasks from the best performed session on the previous day; the testing tasks referred to those performed in the first on-line session on the following day, which were classified more accurately after an adjustment. Although the session-to-session transfer phenomenon was not particularly obvious with subject 3 as figure 5(a) shows, an adjustment resulted in an increase of accuracy shown in figure 5(b). Adjusting the classifier may also help the subject get inspired with more controllable status, because the online experiment always involves intricate interaction between feedback and the subject.

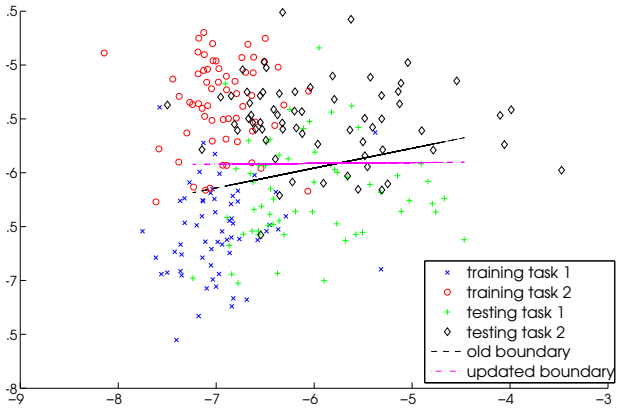


Figure 4: Session to session transfer phenomenon in subject 2 and classification boundary updating

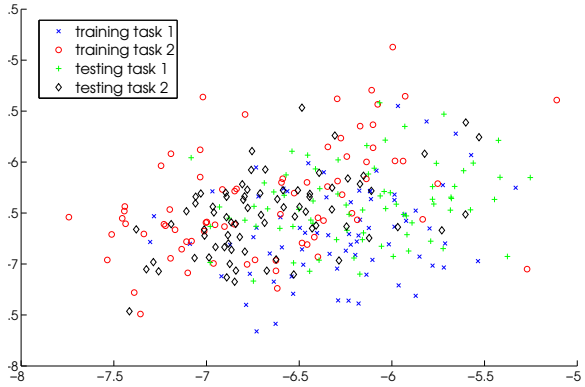
4 Discussions and Conclusions

In order to test the effectiveness of covariate shift adaptation schemes on a BCI, six classifiers, namely LDA, IWLDA1 and IWLDA2, BLDA, BIWLDA1 and BIWLDA2, were applied to Dataset IVc of the BCI Competition III. From the results, we arrive at the following conclusions.

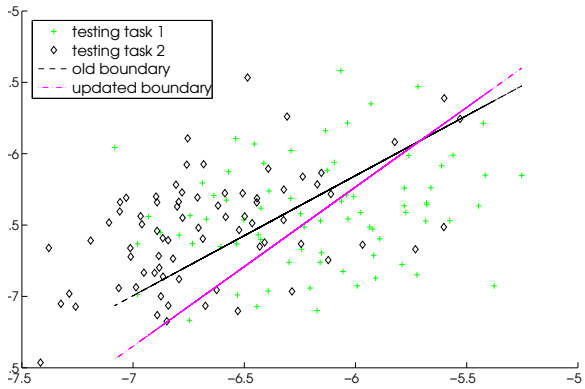
Lacking detailed descriptions regarding the experimental protocol of the two sessions in this dataset, we theorize that the non-stationarity originated from three aspects. First, if electrodes were reinstalled between sessions, slight differences in electrode placement may have caused shifts of the data in the feature space. If no reinstallation of electrodes was performed, it is also possible that the electrode gel dried after four hours, causing varying impedances. Second, the long breaks between runs may also have affected performance. Although in this dataset a good performance level was maintained, this is a normal occurrence in BCI experiments. An example was given in [22], where one of the breaks coincided with the end of a phase with good performance. Therefore, it is possible that, upon resuming the experiment, the subject was unable to regain the control acquired in the previous phase. Third, it may have been difficult for the subjects to maintain an adequate attention level due to fatigue or the learning process itself. Shenoy et al. [22] also pointed out in their study that the non-stationarity was due to different background EEG activities brought on by the introduction of visual feedback during the online feedback session. In our case, however, this cannot be considered to be a reason since the experiment setup remained unchanged between sessions.

In [22], two possible ways of adaptation were also discussed, namely shifting and rotating the boundary. The results of our study demonstrated that when there is a need for shifting or rotation, covariate shift methods are effective in adaptation.

Overall, covariate shift adaptation was shown to be effective for improving the classification accuracy when the feature distributions differ from one session to another. Especially when combined with bagging, even a small number of testing trials will result in



(a) feature distribution of subject 3, both sessions



(b) Second session feature distribution of subject 3 and classification boundary updating

Figure 5: (a) Session to session transfer phenomenon in subject 3, and (b) for a clearer view, the 2nd session (first session on the following day) was plotted separately

an accurate importance estimation.

It would be promising to integrate the proposed algorithm into a BCI system, where adaptation would be run at the beginning of every session. For this purpose, we designed an online experiment and proved the effectiveness of BIWLDA1.

LDA and quadratic discriminant analysis (QDA) are popular classification techniques, especially when adaptation is involved, due to their effectiveness and simplicity. Examples of adapted LDA/QDA applications can be found in [3, 4, 5].

Note that most of the existing adaptation studies focused on trial-to-trial adaptation [3, 4, 5], while we investigated session-to-session adaptation. For subjects, who have little experience with online experiments and may easily become frustrated with incorrect feedback results, the bagged-covariate shift method is helpful in reinforcing their confidence

by making slight adjustments to the settings of the previous day and, thus, avoiding the difficulties of offline training each time before an online experiment.

Acknowledgements

This work was supported with grants by the Japan Science and Technology Agency CREST program to Y. Sakurai, by Grant-in-Aid for Scientific Research on Priority Areas "Emergence of Adaptive Motor Function through Interaction between Body, Brain and Environment" and "Strategic Research Program for Brain Sciences" from MEXT, and by MEXT Grant-in-Aid for Young Scientists (A), 20680007.

References

[1] A.Buttfield, P. W. Ferrez and Jd R. Millan, *Towards a robust BCI: error potentials and online learning*, IEEE Trans. Neural Sys. Rehab. Eng, vol.14, pp.164-168, Jun. 2006.

[2] T. W. Berger et al.(2008, Aug 26), *WTEC Panel Report on International Assessment of Research and Development in Brain-Computer Interfaces [Online]*, Available:http://www.wtec.org/bci/BCI-finalreport-26Aug2008-lowres.pdf

[3] C. Vidaurre, A. Schlogl, R. Cabeza, R. Scherer and G. Pfurtscheller, *A fully online adaptive BCI*, IEEE Trans. Biomed. Eng. vol.53, pp. 1214-1219, Jun. 2006.

[4] C. Vidaurre, A. Schlogl, R. Cabeza, R. Scherer and G. Pfurtscheller, *Study of On-Line Adaptive Discriminant Analysis for EEG-Based Brain Computer Interfaces*, IEEE Trans. Biomed. Eng. vol.54, pp. 550-556, Mar. 2007.

[5] J. Blumberg, J. Rickert, S. Waldert, A. Schulze-Bonhage, A. Aertsen and C. Mehring, *Adaptive Classification for Brain Computer Interfaces*, Conf Proc IEEE Eng Med Biol Soc, 2007, pp. 2536-2539.

[6] M. Sugiyama, M. Krauledat and K.-R. Müller, *Covariate shift adaptation by importance weighted cross validation*, Journal of Machine Learning Research, vol.8, pp.985-1005, May 2007.

[7] S. Sun, C. Zhang and D. Zhang, *An experimental evaluation of ensemble methods for EEG signal classification*, Pattern Recognition Letters vol. 28, pp. 2157-2163, Nov.2007.

[8] L. Christoph et al, *Application and comparison of classification algorithms for recognition of Alzheimer's disease in electrical brain activity (EEG)*, Journal of Neuroscience Methods, vol 161, pp.342-350, Apr.2007.

[9] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller and G. Curio, *The non-invasive Berlin Brain–Computer Interface: Fast acquisition of effective performance in untrained subjects*, NeuroImage, vol.37, pp.539-550, Aug.2007.

[10] Y. Wang, B. Hong, X. Gao, and S. Gao, *Mu rhythm-based cursor control: an offline analysis*, Clinical Neurophysiology, vol.115, pp.745 -751, Apr.2004.

[11] Y. Wang, B. Hong, X. Gao and S. Gao, *Implementation of a Brain-Computer Interface Based on Three States of Motor Imagery*, Conf Proc IEEE Eng Med Biol Soc, 2007, pp.5059-5062.

[12] G. Pfurtscheller, C. Neuper, C. Guger, W. Harkam, H.Ramoser, A. Schlogl,B. Obermaier, and M. Pregenzer, *Current Trends in Graz Brain-Computer Interface (BCI) Research*, IEEE Trans. Rehab. Eng, vol. 8, pp.216-219, Jun.2000.

[13] F. Lotte1, M. Congedo, A. Lecuyer, F. Lamarche and B.Arnaldi, *A review of classification algorithms for EEG-based brain.computer interfaces*, J. Neural Eng. vol.4, pp.1-13, Jun. 2007.

[14] H. Shimodaira. *Improving predictive inference under covariate shift by weighting the log-likelihood function*, Journal of Statistical Planning and Inference, vol.90, pp.227-244, Nov.2000.

[15] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Buenau and M. Kawanabe, *Direct importance estimation for covariate shift adaptation*, Annals of the Institute of Statistical Mathematics, vol.60, pp.699-746, Dec.2008.

[16] T. Kanamori, S. Hido, and M. Sugiyama. *A least-squares approach to direct importance estimation*, Journal of Machine Learning Research, vol.10, pp. 1391-1445, Jul.2009.

[17] Eugene Tuv. *Feature Extraction Foundations and Applications*, Springer,The Netherlands,2006, pp. 188-204.

[18] D. Zhang, Y. Wang, X. Gao, B. Hong and S. Gao (2007, Jul. 12), *An Algorithm for Idle-State Detection in Motor-Imagery-Based Brain-Computer Interface* Comput Intell Neurosci [Online], Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1994518/.

[19] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller, *Boosting bit rates in non-invasive EEG single-trial classifications by feature combination and multi-class paradigms*, IEEE Trans. Biomed. Eng., vol.51, pp. 993-1002, Jun.2004.

[20] G. Schalk, D. J. Mcfarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw *BCI2000: a general-purpose brain-computer interface (BCI) system* IEEE Trans. Biomed. Eng., Vol. 51, No. 6. (2004), pp. 1034-1043.

[21] H. Peng, F. Long, and C. Ding *Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 27, pp.1226-1238, Aug.2005.

[22] P. Shenoy, M. Krauledat, B. Blankertz, R. P N Rao and K-R Müller, *Towards adaptive classification for BCI* J. Neural Eng. vol.3, pp.13-23, Mar. 2006.

Efficient Exploration through Active Learning for Value Function Approximation in Reinforcement Learning

Takayuki Akiyama (akiyama@sg.cs.titech.ac.jp)
Tokyo Institute of Technology

Hirotaka Hachiya (hachiya@sg.cs.titech.ac.jp)
Tokyo Institute of Technology

Masashi Sugiyama (sugi@cs.titech.ac.jp)
Tokyo Institute of Technology
and
Japan Science and Technology Agency

Abstract

Appropriately designing sampling policies is highly important for obtaining better control policies in reinforcement learning. In this paper, we first show that the *least-squares policy iteration* (LSPI) framework allows us to employ statistical active learning methods for linear regression. Then we propose a design method of good sampling policies for efficient exploration, which is particularly useful when the sampling cost of immediate rewards is high. The effectiveness of the proposed method, which we call *active policy iteration* (API), is demonstrated through simulations with a batting robot.

Keywords

reinforcement learning, Markov decision process, least-squares policy iteration, active learning, batting robot

1 Introduction

Reinforcement learning (RL) is the problem of letting an agent learn intelligent behavior through trial-and-error interaction with unknown environment (Sutton & Barto, 1998). More specifically, the agent learns its control policy so that the amount of rewards it will receive in the future is maximized. Due to its potential possibilities, RL has attracted a great deal of attention recently in the machine learning community.

In practical RL tasks, it is often expensive to obtain immediate reward samples while state-action trajectory samples are readily available. For example, let us consider a robot-arm control task of hitting a ball by a bat and driving the ball as far away as possible (see Figure 9). Let us adopt the carry of the ball as the immediate reward. In this setting, obtaining state-action trajectory samples of the robot arm is easy and relatively cheap since we just need to control the robot arm and record its state-action trajectories over time. On the other hand, explicitly computing the carry of the ball from the state-action samples is hard due to friction and elasticity of links, air resistance, unpredictable disturbances such a current of air, and so on. Thus, in practice, we may have to put the robot in open space, let the robot really hit the ball, and measure the carry of the ball manually. Thus gathering immediate reward samples is much more expensive than the state-action trajectory samples.

When the sampling cost of immediate rewards is high, it is important to design the sampling policy appropriately so that a good control policy can be obtained from a small number of samples. So far, the problem of designing good sampling policies has been addressed in terms of the trade-off between *exploration* and *exploitation* (Kaelbling, Littman, & Moore, 1996). That is, an RL agent is required to determine either to explore new states for learning more about unknown environment or to exploit previously acquired knowledge for obtaining more rewards.

A simple framework of controlling the exploration-exploitation trade-off is the *ϵ-greedy policy* (Sutton & Barto, 1998)—with (small) probability ϵ, the agent chooses to explore unknown environment randomly; otherwise it follows the current control policy for exploitation. The choice of the parameter ϵ is critical in the ϵ-greedy policy. A standard and natural idea would be to decrease the probability ϵ as the learning process progresses, i.e., the environment is actively explored in the beginning and then the agent tends to be in the exploitation mode later. However, theoretically and practically sound methods for determining the value of ϵ seem to be still open research topics. Also, when the agent decides to explore unknown environment, merely choosing the next action randomly would be far from the best possible option.

An alternative strategy called *Explicit Explore or Exploit* (E^3) was proposed in Kearns & Singh (1998) and Kearns & Singh (2002). The basic idea of E^3 is to control the balance between exploration and exploitation so that the accuracy of environment model estimation is optimally improved. More specifically, when the number of known states is small, the agent actively explores unvisited (or less visited) states; as the number of known states increases, exploitation tends to be prioritized. The E^3 strategy is efficiently realized by an algorithm called *R-max* (Brafman & Tennenholtz, 2002; Strehl, Diuk, & Littman, 2007). R-max assigns the maximum 'value' to unknown states so that the unknown states are visited with high probability. An advantage of E^3 and R-max is that the polynomial-time convergence (with respect to the number of states) to a near-optimal policy is theoretically guaranteed. However, since the algorithms explicitly count the number of visits at every state, it is not straightforward to extend the idea to *continuous* state spaces (Li, Littman, & Mansley, 2008). This is a critical limitation in robotics applications since state spaces are usually spanned by continuous variables such as joint

angles and angular velocities.

In this paper, we address the problem of designing sampling policies from a different point of view—*active learning* (AL) for value function approximation. We adopt the framework of *least-squares policy iteration* (LSPI) (Lagoudakis & Parr, 2003) and show that statistical AL methods for linear regression (Fedorov, 1972; Cohn, Ghahramani, & Jordan, 1996; Wiens, 2000; Kanamori & Shimodaira, 2003; Sugiyama, 2006; Sugiyama & Nakajima, 2009) can be naturally employed. In the LSPI framework, the state-action value function is approximated by fitting a linear model with least-squares estimation. A traditional AL scheme (Fedorov, 1972; Cohn et al., 1996) is designed to find the input distribution such that the variance of the least-squares estimator is minimized. For justifying the use of the traditional AL scheme, the bias should be guaranteed not to increase when the variance is reduced, since the expectation of the squared approximation error of the value function is expressed as the sum of the squared bias and variance. To this end, we need to assume a strong condition that the linear model used for value function approximation is *correctly specified*, i.e., if the parameters are learned optimally, the true value function can be perfectly approximated.

However, such a correct model assumption may not be fulfilled in practical RL tasks since the profile of value functions may be highly complicated. To cope with this problem, a two-stage AL scheme has been proposed in Kanamori & Shimodaira (2003). The use of the two-stage AL scheme can be theoretically justified even when the model is *misspecified*, i.e., the true function is not included in the model. The key idea of this two-stage AL scheme is to use dummy samples gathered in the first stage for estimating the approximation error of the value function; then additional samples are chosen based on AL in the second stage. This two-stage scheme works well when a large number of dummy samples are used for estimating the approximation error in the first stage. However, due to high sampling costs in practical RL problems, the practical performance of the two-stage AL method in the RL scenarios would be limited.

To overcome the weakness of the two-stage AL method, single-shot AL methods have been developed (Wiens, 2000; Sugiyama, 2006). The use of the single-shot AL methods can be theoretically justified when the model is *approximately correct*. Since dummy samples are not necessary in the single-shot AL methods, good performance is expected even when the number of samples to be collected is not large. Moreover, the algorithms of the single-shot methods are very simple and computationally efficient. For this reason, we adopt the single-shot AL method proposed in Sugiyama (2006), and develop a new exploration scheme for the LSPI-based RL algorithm. The usefulness of the proposed approach, which we call *active policy iteration* (API), is demonstrated through batting-robot simulations.

The rest of this paper is organized as follows. In Section 2, we formulate the RL problem using Markov decision processes and review the LSPI framework. Then in Section 3, we show how a statistical AL method could be employed for optimizing the sampling policy in the context of value function approximation. In Section 4, we apply our AL strategy to the LSPI framework and show the entire procedure of the proposed API algorithm. In Section 5, we demonstrate the effectiveness of API through ball-batting simulations.

Finally, in Section 6, we conclude by summarizing our contributions and describing future work.

2 Formulation of Reinforcement Learning Problem

In this section, we formulate the RL problem as a Markov decision problem (MDP) following Sutton & Barto (1998), and review how it can be solved using a method of policy iteration following Lagoudakis & Parr (2003).

2.1 Markov Decision Problem

Let us consider an MDP specified by

$$(\mathcal{S}, \mathcal{A}, P_{\mathrm{T}}, R, \gamma), \tag{1}$$

where

- $\mathcal{S}$ is a set of states,
- $\mathcal{A}$ is a set of actions,
- $P_{\mathrm{T}}(s'|s,a)$ $(\in [0,1])$ is the conditional probability density of the agent's transition from state s to next state s' when action a is taken,
- $R(s,a,s')$ $(\in \mathbb{R})$ is a reward for transition from s to s' by taking action a,
- γ $(\in (0,1])$ is the discount factor for future rewards.

Let $\pi(a|s)$ $(\in [0,1])$ be a stochastic policy which is a conditional probability density of taking action a given state s. The state-action value function $Q^\pi(s,a)$ $(\in \mathbb{R})$ for policy π denotes the expectation of the discounted sum of rewards the agent will receive when taking action a in state s and following policy π thereafter, i.e.,

$$Q^\pi(s,a) \equiv \mathop{\mathbb{E}}_{\{s_n,a_n\}_{n=2}^\infty} \left[\sum_{n=1}^\infty \gamma^{n-1} R(s_n,a_n,s_{n+1}) \,\middle|\, s_1 = s, a_1 = a \right], \tag{2}$$

where $\mathbb{E}_{\{s_n,a_n\}_{n=2}^\infty}$ denotes the expectation over trajectory $\{s_n,a_n\}_{n=2}^\infty$ following $P_{\mathrm{T}}(s_{n+1}|s_n,a_n)$ and $\pi(a_n|s_n)$.

The goal of RL is to obtain the policy such that the expectation of the discounted sum of future rewards is maximized. The optimal policy can be expressed as

$$\pi^*(a|s) \equiv \delta(a - \mathop{\mathrm{argmax}}_{a'} Q^*(s,a')), \tag{3}$$

where $\delta(\cdot)$ is Dirac's delta function and

$$Q^*(s,a) \equiv \max_\pi Q^\pi(s,a) \tag{4}$$

is the *optimal* state-action value function.

$Q^\pi(s,a)$ can be expressed by the following recurrent form called the *Bellman equation*:

$$Q^\pi(s,a) = R(s,a) + \gamma \mathop{\mathbb{E}}_{P_{\mathrm{T}}(s'|s,a)} \mathop{\mathbb{E}}_{\pi(a'|s')} \left[Q^\pi(s',a')\right], \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}, \tag{5}$$

where

$$R(s,a) \equiv \mathop{\mathbb{E}}_{P_{\mathrm{T}}(s'|s,a)} \left[R(s,a,s')\right] \tag{6}$$

is the expected reward when the agent takes action a in state s, $\mathbb{E}_{P_{\mathrm{T}}(s'|s,a)}$ denotes the conditional expectation of s' over $P_{\mathrm{T}}(s'|s,a)$ given s and a, and $\mathbb{E}_{\pi(a'|s')}$ denotes the conditional expectation of a' over $\pi(a'|s')$ given s'.

2.2 Policy Iteration

Computing the value function $Q^\pi(s,a)$ is called *policy evaluation*. Using $Q^\pi(s,a)$, we may find a better policy $\pi'(a|s)$ by 'softmax' update:

$$\pi'(a|s) \propto \exp(Q^\pi(s,a)/\beta), \tag{7}$$

where β (>0) determines the randomness of the new policy π'; or by ϵ-greedy update:

$$\pi'(a|s) = \epsilon p_{\mathrm{u}}(a) + (1-\epsilon)\delta(a - \mathop{\mathrm{argmax}}_{a'} Q^\pi(s,a')), \tag{8}$$

where $p_{\mathrm{u}}(a)$ denotes the uniform probability density over actions and ϵ $(\in (0,1])$ determines how stochastic the new policy π' is. Updating π based on $Q^\pi(s,a)$ is called *policy improvement*. Repeating policy evaluation and policy improvement, we may find the optimal policy $\pi^*(a|s)$. This entire process is called *policy iteration* (Sutton & Barto, 1998).

2.3 Least-squares Framework for Value Function Approximation

Although policy iteration is a useful framework for solving an MDP problem, it is computationally expensive when the number of state-action pairs $|\mathcal{S}| \times |\mathcal{A}|$ is large. Furthermore, when the state space or action space is continuous, $|\mathcal{S}|$ or $|\mathcal{A}|$ becomes infinite and therefore it is no longer possible to directly implement policy iteration. To overcome this problem, we approximate the state-action value function $Q^\pi(s,a)$ using the following linear model:

$$\widehat{Q}^\pi(s,a;\boldsymbol{\theta}) \equiv \sum_{b=1}^{B} \theta_b \phi_b(s,a) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s,a), \tag{9}$$

where

$$\boldsymbol{\phi}(s,a) = (\phi_1(s,a), \phi_2(s,a), \ldots, \phi_B(s,a))^\top \tag{10}$$

are the fixed linearly independent basis functions, $\top$ denotes the transpose, B is the number of basis functions, and

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_B)^\top \tag{11}$$

are model parameters to be learned. Note that B is usually chosen to be much smaller than $|\mathcal{S}| \times |\mathcal{A}|$ for computational efficiency.

For N-step transitions, we ideally want to learn the parameters $\boldsymbol{\theta}$ so that the squared Bellman residual $G(\boldsymbol{\theta})$ is minimized (Lagoudakis & Parr, 2003):

$$\boldsymbol{\theta}^* \equiv \operatorname*{argmin}_{\boldsymbol{\theta}} G(\boldsymbol{\theta}), \tag{12}$$

$$G(\boldsymbol{\theta}) \equiv \mathop{\mathbb{E}}_{P_\pi}\left[\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\theta}^\top \boldsymbol{\psi}(s_n, a_n) - R(s_n, a_n))^2\right], \tag{13}$$

$$\boldsymbol{\psi}(s, a) \equiv \boldsymbol{\phi}(s, a) - \gamma \mathop{\mathbb{E}}_{P_{\mathrm{T}}(s'|s,a)} \mathop{\mathbb{E}}_{\pi(a'|s')} [\boldsymbol{\phi}(s', a')]. \tag{14}$$

$\mathbb{E}_{P_\pi}$ denotes the expectation over the joint probability density function of an entire trajectory:

$$P_\pi(s_1, a_1, s_2, a_2, \ldots, s_N, a_N, s_{N+1}) \equiv P_{\mathrm{I}}(s_1)\prod_{n=1}^{N} P_{\mathrm{T}}(s_{n+1}|s_n, a_n)\pi(a_n|s_n), \tag{15}$$

where $P_{\mathrm{I}}(s)$ denotes the initial-state probability density function.

2.4 Value Function Approximation from Samples

Suppose that roll-out data samples consisting of M episodes with N steps are available for training purposes. The agent initially starts from randomly selected state s_1 following the initial-state probability density $P_{\mathrm{I}}(s)$ and chooses an action based on *sampling policy* $\widetilde{\pi}(a_n|s_n)$. Then the agent makes a transition following the transition probability $P_{\mathrm{T}}(s_{n+1}|s_n, a_n)$ and receives a reward $r_n(= R(s_n, a_n, s_{n+1}))$. This is repeated for N steps—thus the training dataset $\mathcal{D}^{\widetilde{\pi}}$ is expressed as

$$\mathcal{D}^{\widetilde{\pi}} \equiv \{d_m^{\widetilde{\pi}}\}_{m=1}^{M}, \tag{16}$$

where each episodic sample $d_m^{\widetilde{\pi}}$ consists of a set of 4-tuple elements as

$$d_m^{\widetilde{\pi}} \equiv \{(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}, r_{m,n}^{\widetilde{\pi}}, s_{m,n+1}^{\widetilde{\pi}})\}_{n=1}^{N}. \tag{17}$$

We use two types of policies for different purposes: the *sampling policy* $\widetilde{\pi}(a|s)$ for collecting data samples and the *evaluation policy* $\pi(a|s)$ for computing the value function $\widehat{Q}^\pi$. Minimizing the *importance-weighted* empirical generalization error $\widehat{G}(\boldsymbol{\theta})$, we can

obtain a *consistent* estimator of $\boldsymbol{\theta}^*$ as follows:

$$\widehat{\boldsymbol{\theta}} \equiv \operatorname*{argmin}_{\boldsymbol{\theta}} \widehat{G}(\boldsymbol{\theta}), \tag{18}$$

$$\widehat{G}(\boldsymbol{\theta}) \equiv \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} (\boldsymbol{\theta}^\top \widehat{\boldsymbol{\psi}}(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}; \mathcal{D}^{\widetilde{\pi}}) - r_{m,n}^{\widetilde{\pi}})^2 w_{m,N}^{\widetilde{\pi}}, \tag{19}$$

$$\widehat{\boldsymbol{\psi}}(s, a; \mathcal{D}) \equiv \boldsymbol{\phi}(s, a) - \frac{\gamma}{|\mathcal{D}_{(s,a)}|} \sum_{s' \in \mathcal{D}_{(s,a)}} \mathop{\mathbb{E}}_{\pi(a'|s')} [\boldsymbol{\phi}(s', a')], \tag{20}$$

where $\mathcal{D}_{(s,a)}$ is a set of 4-tuple elements[1] containing state s and action a in the training data $\mathcal{D}$, $\sum_{s' \in \mathcal{D}_{(s,a)}}$ denotes the summation over s' in the set $\mathcal{D}_{(s,a)}$, and

$$w_{m,N}^{\widetilde{\pi}} \equiv \frac{\prod_{n'=1}^{N} \pi(a_{m,n'}^{\widetilde{\pi}} | s_{m,n'}^{\widetilde{\pi}})}{\prod_{n'=1}^{N} \widetilde{\pi}(a_{m,n'}^{\widetilde{\pi}} | s_{m,n'}^{\widetilde{\pi}})} \tag{21}$$

is called the *importance weight* (Sutton & Barto, 1998).

It is important to note that consistency of $\widehat{\boldsymbol{\theta}}$ can be maintained even if $w_{m,N}^{\widetilde{\pi}}$ is replaced by the *per-decision importance weight* $w_{m,n}^{\widetilde{\pi}}$ (Precup, Sutton, & Singh, 2000), which is computationally more efficient and stable. $\widehat{\boldsymbol{\theta}}$ can be analytically expressed with the matrices $\widehat{\boldsymbol{L}}$ ($\in \mathbb{R}^{B \times MN}$), $\widehat{\boldsymbol{X}}$ ($\in \mathbb{R}^{MN \times B}$), $\boldsymbol{W}$ ($\in \mathbb{R}^{MN \times MN}$), and the vector $\boldsymbol{r}^{\widetilde{\pi}}$ ($\in \mathbb{R}^{MN}$) as

$$\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{L}} \boldsymbol{r}^{\widetilde{\pi}}, \tag{22}$$

$$\widehat{\boldsymbol{L}} \equiv (\widehat{\boldsymbol{X}}^\top \boldsymbol{W} \widehat{\boldsymbol{X}})^{-1} \widehat{\boldsymbol{X}}^\top \boldsymbol{W}, \tag{23}$$

$$\boldsymbol{r}_{N(m-1)+n}^{\widetilde{\pi}} \equiv r_{m,n}^{\widetilde{\pi}}, \tag{24}$$

$$\widehat{\boldsymbol{X}}_{N(m-1)+n,b} \equiv \widehat{\psi}_b(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}; \mathcal{D}^{\widetilde{\pi}}), \tag{25}$$

$$\boldsymbol{W}_{N(m-1)+n,N(m'-1)+n'} \equiv w_{m,n}^{\widetilde{\pi}} I(m = m') I(n = n'), \tag{26}$$

where $I(c)$ denotes the indicator function:

$$I(c) = \begin{cases} 1 & \text{if the condition } c \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

When the matrix $\widehat{\boldsymbol{X}}^\top \boldsymbol{W} \widehat{\boldsymbol{X}}$ is ill-conditioned, it is hard to compute its inverse accurately. To cope with this problem, we may practically employ a *regularization* scheme (Tikhonov & Arsenin, 1977; Hoerl & Kennard, 1970; Poggio & Girosi, 1990):

$$(\widehat{\boldsymbol{X}}^\top \boldsymbol{W} \widehat{\boldsymbol{X}} + \lambda \boldsymbol{I})^{-1}, \tag{28}$$

where $\boldsymbol{I}$ ($\in \mathbb{R}^{B \times B}$) is the identity matrix and λ is a small positive scalar.

[1] When the state-action space is continuous, the set $\mathcal{D}_{(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}})}$ contains only a single sample $(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}, r_{m,n}^{\widetilde{\pi}}, s_{m,n+1}^{\widetilde{\pi}})$ and then consistency of $\widehat{\boldsymbol{\theta}}$ may not be guaranteed. A possible measure for this would be to use several neighbor samples around $(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}})$. However, in our experiments, we decided to use the single-sample approximation since it performed reasonably well.

3 Efficient Exploration with Active Learning

The accuracy of the estimated value function depends on the training samples collected following sampling policy $\widetilde{\pi}(a|s)$. In this section, we propose a new method for designing a good sampling policy based on a statistical AL method proposed in Sugiyama (2006).

3.1 Preliminaries

Let us consider a situation where collecting state-action trajectory samples is easy and cheap, but gathering immediate reward samples is hard and expensive (for example, the batting robot explained in the introduction). In such a case, immediate reward samples are too expensive to be used for designing the sampling policy; only state-action trajectory samples may be used for sampling policy design.

The goal of AL in the current setup is to determine the sampling policy so that the expected generalization error is minimized. The generalization error is not accessible in practice since the expected reward function $R(s,a)$ and the transition probability $P_{\mathrm{T}}(s'|s,a)$ are unknown. Thus, for performing AL, the generalization error needs to be estimated from samples. A difficulty of estimating the generalization error in the context of AL is that its estimation needs to be carried out only from state-action trajectory samples *without* using immediate reward samples. This means that standard generalization error estimation techniques such as *cross-validation* (Hachiya, Akiyama, Sugiyama, & Peters, 2009) cannot be employed since they require both state-action and immediate reward samples. Below, we explain how the generalization error could be estimated under the AL setup (i.e., without the reward samples).

3.2 Decomposition of Generalization Error

The information we are allowed to use for estimating the generalization error is a set of roll-out samples *without* immediate rewards:

$$\overline{\mathcal{D}}^{\widetilde{\pi}} \equiv \{\overline{d}_m^{\widetilde{\pi}}\}_{m=1}^{M}, \tag{29}$$

$$\overline{d}_m^{\widetilde{\pi}} \equiv \{(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}, s_{m,n+1}^{\widetilde{\pi}})\}_{n=1}^{N}. \tag{30}$$

Let us define the deviation of immediate rewards from the mean as

$$\epsilon_{m,n}^{\widetilde{\pi}} \equiv r_{m,n}^{\widetilde{\pi}} - R(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}). \tag{31}$$

Note that $\epsilon_{m,n}^{\widetilde{\pi}}$ could be regarded as additive noise in the context of least-squares function fitting. By definition, $\epsilon_{m,n}^{\widetilde{\pi}}$ has mean zero and its variance generally depends on $s_{m,n}^{\widetilde{\pi}}$ and $a_{m,n}^{\widetilde{\pi}}$, i.e., *heteroscedastic* noise (Bishop, 2006). However, since estimating the variance of $\epsilon_{m,n}^{\widetilde{\pi}}$ without using reward samples is not generally possible, we ignore the dependence of the variance on $s_{m,n}^{\widetilde{\pi}}$ and $a_{m,n}^{\widetilde{\pi}}$. Let us denote the input-independent common variance by σ^2.

Now we would like to estimate the generalization error

$$\overline{G}(\widehat{\boldsymbol{\theta}}) \equiv \mathop{\mathbb{E}}_{P_\pi}\left[\frac{1}{N}\sum_{n=1}^{N}(\widehat{\boldsymbol{\theta}}^\top\widehat{\boldsymbol{\psi}}(s_n,a_n;\overline{\mathcal{D}}^{\widetilde{\pi}}) - R(s_n,a_n))^2\right] \tag{32}$$

from $\overline{\mathcal{D}}^{\widetilde{\pi}}$. Its expectation over 'noise' can be decomposed as follows (Sugiyama, 2006).

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}^{\widetilde{\pi}}}\left[\overline{G}(\widehat{\boldsymbol{\theta}})\right] = \text{Bias} + \text{Variance} + \text{ModelError}, \tag{33}$$

where $\mathbb{E}_{\boldsymbol{\epsilon}^{\widetilde{\pi}}}$ denotes the expectation over 'noise' $\{\epsilon^{\widetilde{\pi}}_{m,n}\}_{m=1,n=1}^{M,N}$. Bias, Variance, and ModelError are the *bias* term, the *variance* term, and the *model error* term defined by

$$\text{Bias} \equiv \mathop{\mathbb{E}}_{P_\pi}\left[\frac{1}{N}\sum_{n=1}^{N}\left\{(\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}^{\widetilde{\pi}}}\left[\widehat{\boldsymbol{\theta}}\right] - \boldsymbol{\theta}^*)^\top\widehat{\boldsymbol{\psi}}(s_n,a_n;\overline{\mathcal{D}}^{\widetilde{\pi}})\right\}^2\right], \tag{34}$$

$$\text{Variance} \equiv \mathop{\mathbb{E}}_{P_\pi}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}^{\widetilde{\pi}}}\left[\frac{1}{N}\sum_{n=1}^{N}\left\{(\widehat{\boldsymbol{\theta}} - \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}^{\widetilde{\pi}}}\left[\widehat{\boldsymbol{\theta}}\right])^\top\widehat{\boldsymbol{\psi}}(s_n,a_n;\overline{\mathcal{D}}^{\widetilde{\pi}})\right\}^2\right], \tag{35}$$

$$\text{ModelError} \equiv \mathop{\mathbb{E}}_{P_\pi}\left[\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\theta}^{*\top}\widehat{\boldsymbol{\psi}}(s_n,a_n;\overline{\mathcal{D}}^{\widetilde{\pi}}) - R(s_n,a_n))^2\right]. \tag{36}$$

$\boldsymbol{\theta}^*$ is the optimal parameter in the model, defined by Eq.(12). Note that the variance term can be expressed in a compact form as

$$\text{Variance} = \sigma^2\text{tr}(\boldsymbol{U}\widehat{\boldsymbol{L}}\widehat{\boldsymbol{L}}^\top), \tag{37}$$

where the matrix $\boldsymbol{U}$ $(\in \mathbb{R}^{B\times B})$ is defined as

$$\boldsymbol{U}_{b,b'} \equiv \mathop{\mathbb{E}}_{P_\pi}\left[\frac{1}{N}\sum_{n=1}^{N}\widehat{\psi}_b(s_n,a_n;\overline{\mathcal{D}}^{\widetilde{\pi}})\widehat{\psi}_{b'}(s_n,a_n;\overline{\mathcal{D}}^{\widetilde{\pi}})\right]. \tag{38}$$

3.3 Estimation of Generalization Error for AL

The model error is constant and thus can be safely ignored in generalization error estimation since we are interested in finding a minimizer of the generalization error with respect to $\widetilde{\pi}$. So we focus on the bias term and the variance term. However, the bias term includes the unknown optimal parameter $\boldsymbol{\theta}^*$, and thus it may not be possible to estimate the bias term without using reward samples; similarly, it may not be possible to estimate the 'noise' variance σ^2 included in the variance term without using reward samples.

It is known that the bias term is small enough to be neglected when the model is *approximately correct* (Sugiyama, 2006), i.e., $\boldsymbol{\theta}^{*\top}\widehat{\boldsymbol{\psi}}(s,a)$ approximately agrees with the true function $R(s,a)$. Then we have

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}^{\widetilde{\pi}}}\left[\overline{G}(\widehat{\boldsymbol{\theta}})\right] - \text{ModelError} - \text{Bias} \;\propto\; \text{tr}(\boldsymbol{U}\widehat{\boldsymbol{L}}\widehat{\boldsymbol{L}}^\top), \tag{39}$$

which does not require immediate reward samples for its computation. Since $\mathbb{E}_{P_\pi}$ included in $\boldsymbol{U}$ is not accessible (see Eq.(38)), we replace $\boldsymbol{U}$ by its consistent estimator $\widehat{\boldsymbol{U}}$:

$$\widehat{\boldsymbol{U}} \equiv \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\widehat{\boldsymbol{\psi}}(s^{\widetilde{\pi}}_{m,n}, a^{\widetilde{\pi}}_{m,n}; \overline{\mathcal{D}}^{\widetilde{\pi}})\widehat{\boldsymbol{\psi}}(s^{\widetilde{\pi}}_{m,n}, a^{\widetilde{\pi}}_{m,n}; \overline{\mathcal{D}}^{\widetilde{\pi}})^\top w^{\widetilde{\pi}}_{m,n}. \tag{40}$$

Consequently, we have the following generalization error estimator:

$$J \equiv \mathrm{tr}(\widehat{\boldsymbol{U}}\widehat{\boldsymbol{L}}\widehat{\boldsymbol{L}}^\top), \tag{41}$$

which can be computed only from $\overline{\mathcal{D}}^{\widetilde{\pi}}$ and thus can be employed in the AL scenarios. If it is possible to gather $\overline{\mathcal{D}}^{\widetilde{\pi}}$ multiple times, the above J may be computed multiple times and its average J' may be used as a generalization error estimator.

Note that the values of the generalization error estimator J and the true generalization error $\overline{G}$ are not directly comparable since irrelevant additive and multiplicative constants are ignored (see Eq.(39)). We expect that the estimator J has a similar *profile* to the true error $\overline{G}$ as a function of sampling policy $\widetilde{\pi}$ since the purpose of deriving a generalization error estimator in AL is not to approximate the true generalization error itself, but to approximate the *minimizer* of the true generalization error with respect to sampling policy $\widetilde{\pi}$. We will experimentally investigate this issue in Section 3.5.

3.4 Designing Sampling Policies

Based on the generalization error estimator derived above, we give an algorithm for designing a good sampling policy, which fully makes use of the roll-out samples without immediate rewards.

1. Prepare K candidates of sampling policy: $\{\widetilde{\pi}_k\}_{k=1}^K$.
2. Collect episodic samples without immediate rewards for each sampling-policy candidate: $\{\overline{\mathcal{D}}^{\widetilde{\pi}_k}\}_{k=1}^K$.
3. Estimate $\boldsymbol{U}$ using all samples $\{\overline{\mathcal{D}}^{\widetilde{\pi}_k}\}_{k=1}^K$:

$$\widetilde{\boldsymbol{U}} = \frac{1}{KMN}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{n=1}^{N}\widehat{\boldsymbol{\psi}}(s^{\widetilde{\pi}_k}_{m,n}, a^{\widetilde{\pi}_k}_{m,n}; \{\overline{\mathcal{D}}^{\widetilde{\pi}_k}\}_{k=1}^K)\widehat{\boldsymbol{\psi}}(s^{\widetilde{\pi}_k}_{m,n}, a^{\widetilde{\pi}_k}_{m,n}; \{\overline{\mathcal{D}}^{\widetilde{\pi}_k}\}_{k=1}^K)^\top w^{\widetilde{\pi}_k}_{m,n}. \tag{42}$$

4. Estimate the generalization error for each k:

$$J_k \equiv \mathrm{tr}(\widetilde{\boldsymbol{U}}\widehat{\boldsymbol{L}}^{\widetilde{\pi}_k}\widehat{\boldsymbol{L}}^{\widetilde{\pi}_k\top}), \tag{43}$$

$$\widehat{\boldsymbol{L}}^{\widetilde{\pi}_k} \equiv (\widehat{\boldsymbol{X}}^{\widetilde{\pi}_k\top}\boldsymbol{W}^{\widetilde{\pi}_k}\widehat{\boldsymbol{X}}^{\widetilde{\pi}_k})^{-1}\widehat{\boldsymbol{X}}^{\widetilde{\pi}_k\top}\boldsymbol{W}^{\widetilde{\pi}_k}, \tag{44}$$

$$\widehat{\boldsymbol{X}}^{\widetilde{\pi}_k}_{N(m-1)+n,b} \equiv \widehat{\psi}_b(s^{\widetilde{\pi}_k}_{m,n}, a^{\widetilde{\pi}_k}_{m,n}; \{\overline{\mathcal{D}}^{\widetilde{\pi}_k}\}_{k=1}^K), \tag{45}$$

$$\boldsymbol{W}^{\widetilde{\pi}_k}_{N(m-1)+n,N(m'-1)+n'} \equiv w^{\widetilde{\pi}_k}_{m,n} I(m=m')I(n=n'). \tag{46}$$

5. (If possible) repeat 2. to 4. several times and calculate the average for each k: $\{J'_k\}_{k=1}^K$.
6. Determine the sampling policy: $\widetilde{\pi}_{\rm AL} \equiv {\rm argmin}_k \, J'_k$.
7. Collect training samples with immediate rewards following $\widetilde{\pi}_{\rm AL}$: $\mathcal{D}^{\widetilde{\pi}_{\rm AL}}$.
8. Learn the value function by LSPI using $\mathcal{D}^{\widetilde{\pi}_{\rm AL}}$.

3.5 Numerical Examples

Here we illustrate how the above AL method behaves in the 10-state chain-walk environment shown in Figure 1. The MDP consists of 10 states

$$\mathcal{S} = \{s^{(i)}\}_{i=1}^{10} = \{1, 2, \ldots, 10\} \tag{47}$$

and 2 actions

$$\mathcal{A} = \{a^{(i)}\}_{i=1}^{2} = \{\text{'L'}, \text{'R'}\}. \tag{48}$$

The immediate reward function is defined as

$$R(s, a, s') \equiv f(s'), \tag{49}$$

where the profile of the function $f(s')$ is illustrated in Figure 2.

The transition probability $P_{\rm T}(s'|s, a)$ is indicated by the numbers attached to the arrows in Figure 1; for example, $P_{\rm T}(s^{(2)}|s^{(1)}, \text{'R'}) = 0.8$ and $P_{\rm T}(s^{(1)}|s^{(1)}, \text{'R'}) = 0.2$. Thus the agent can successfully move to the intended direction with probability 0.8 (indicated by solid-filled arrows in the figure) and the action fails with probability 0.2 (indicated by dashed-filled arrows in the figure). The discount factor γ is set to 0.9. We use the 12 basis functions $\boldsymbol{\phi}(s, a)$ defined as

$$\phi_{2(i-1)+j}(s, a) = \begin{cases} I(a = a^{(j)}) \exp\left(-\dfrac{(s - c_i)^2}{2\tau^2}\right) & \text{for } i = 1, 2, \ldots, 5 \text{ and } j = 1, 2 \\ I(a = a^{(j)}) & \text{for } i = 6 \text{ and } j = 1, 2, \end{cases} \tag{50}$$

where $c_1 = 1$, $c_2 = 3$, $c_3 = 5$, $c_4 = 7$, $c_5 = 9$, and $\tau = 1.5$.

For illustration purposes, we evaluate the selection of sampling policies only in one-step policy evaluation; evaluation through iterations will be addressed in the next section. Sampling policies and evaluation policies are constructed as follows. First, we prepare a deterministic 'base' policy $\overline{\pi}$, e.g., 'LLLLLRRRRR', where the i-th letter denotes the action taken at $s^{(i)}$. Let $\overline{\pi}^\epsilon$ be the 'ϵ-greedy' version of the base policy $\overline{\pi}$, i.e., the intended action can be successfully chosen with probability $1 - \epsilon/2$ and the other action is chosen with probability $\epsilon/2$. We perform experiments for three different evaluation policies:

$$\overline{\pi}_1 : \text{'RRRRRRRRRR'}, \tag{51}$$

$$\overline{\pi}_2 : \text{'RRLLLLLRRR'}, \tag{52}$$

$$\overline{\pi}_3 : \text{'LLLLLRRRRR'} \tag{53}$$

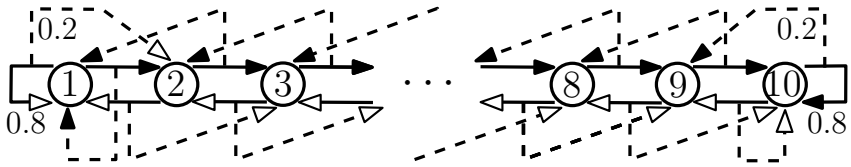


Figure 1: 10-state chain walk. Filled/unfilled arrows indicate the transitions when taking action 'R'/'L' and solid/dashed lines indicate the successful/failed transitions.

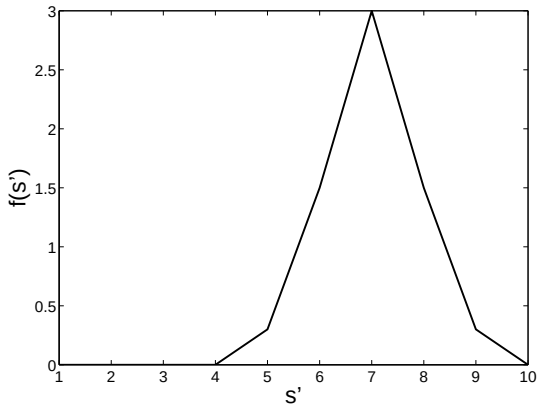


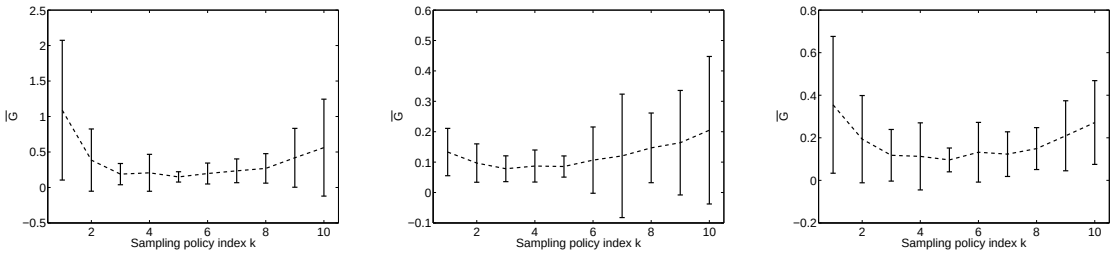
Figure 2: Profile of the function $f(s')$.

with $\epsilon = 0.1$. For each evaluation policy $\overline{\pi}_i^{0.1}$ $(i = 1, 2, 3)$, we prepare 10 candidates of the sampling policy $\{\widetilde{\pi}_i^{(k)}\}_{k=1}^{10}$, where the k-th sampling policy $\widetilde{\pi}_i^{(k)}$ is defined as $\overline{\pi}_i^{k/10}$. Note that $\widetilde{\pi}_i^{(1)}$ is equivalent to the evaluation policy $\overline{\pi}_i^{0.1}$.

For each sampling policy, we calculate the J-value 5 times and take the average. The numbers of episodes and steps are set to $M = 10$ and $N = 10$, respectively. The initial-state probability $P_{\mathrm{I}}(s)$ is set to be uniform. The regularization parameter is set to $\lambda = 10^{-3}$ for avoiding matrix singularity. This experiment is repeated 100 times with different random seeds and the mean and standard deviation of the true generalization error and its estimate are evaluated.

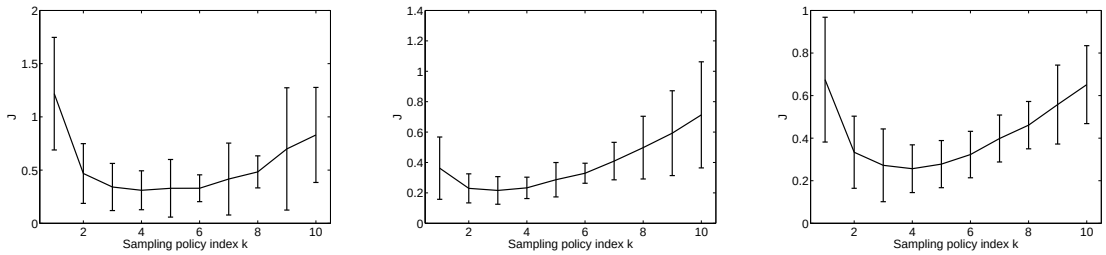
The results are depicted in Figure 3 (the true generalization error) and Figure 4 (its estimate) as functions of the index k of the sampling policies. Note that in these figures, we ignored irrelevant additive and multiplicative constants when deriving the generalization error estimator (see Eq.(39)). Thus, directly comparing the values of the true generalization error and its estimate is meaningless. The graphs show that the proposed generalization error estimator overall captures the trend of the true generalization error well for all three cases.

For active learning purposes, we are interested in choosing the value of k so that the true generalization error is minimized. Next, we investigate the values of the obtained generalization error $\overline{G}$ when k is chosen so that J is minimized (active learning; AL), the evaluation policy $(k = 1)$ is used for sampling (passive learning; PL), and k is chosen



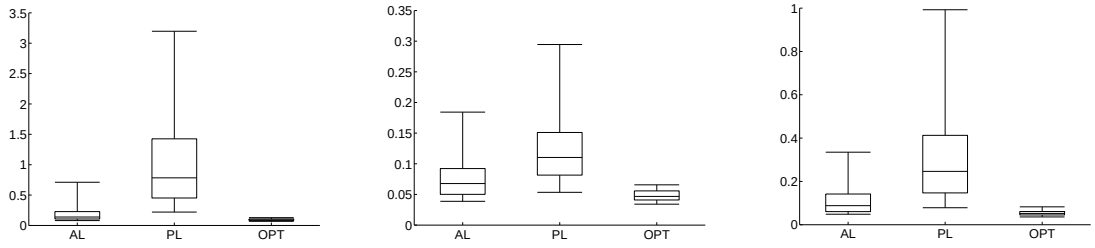
(a) $\overline{\pi}_1^{0.1}$ (b) $\overline{\pi}_2^{0.1}$ (c) $\overline{\pi}_3^{0.1}$

Figure 3: The mean and standard deviation of the true generalization error over 100 trials.



(a) $\overline{\pi}_1^{0.1}$ (b) $\overline{\pi}_2^{0.1}$ (c) $\overline{\pi}_3^{0.1}$

Figure 4: The mean and standard deviation of the estimated generalization error J over 100 trials.



(a) $\overline{\pi}_1^{0.1}$ (b) $\overline{\pi}_2^{0.1}$ (c) $\overline{\pi}_3^{0.1}$

Figure 5: The box-plots of the values of the obtained generalization error $\overline{G}$ over 100 trials when k is chosen so that J is minimized (active learning; AL), the evaluation policy ($k = 1$) is used for sampling (passive learning; PL), and k is chosen optimally so that the true generalization error is minimized (optimal; OPT). The box-plot notation indicates the 5%-quantile, 25%-quantile, 50%-quantile (i.e., median), 75%-quantile, and 95%-quantile from bottom to top.

optimally so that the true generalization error is minimized (optimal; OPT). Figure 5 depicts the box-plots of the generalization error values for AL, PL, and OPT over 100 trials, where the box-plot notation indicates the 5%-quantile, 25%-quantile, 50%-quantile (i.e., median), 75%-quantile, and 95%-quantile from bottom to top. The graphs show that the proposed AL method compares favorably with PL and performs well for reducing the generalization error.

We will continue the performance evaluation of the proposed AL method through iterations in Section 4.2.

4 Active Learning in Policy Iteration

In Section 3, we have shown that the unknown generalization error could be accurately estimated without using immediate reward samples in one-step policy evaluation. In this section, we extend the idea to the full policy-iteration setup.

4.1 Sample Reuse Policy Iteration with Active Learning

Sample reuse policy iteration (SRPI) (Hachiya et al., 2009) is a recently-proposed framework of *off-policy RL* (Sutton & Barto, 1998; Precup et al., 2000), which allows us to reuse previously-collected samples effectively. Let us denote the evaluation policy at the l-th iteration by π_l and the maximum number of iterations by L.

In the policy iteration framework, new data samples $\mathcal{D}^{\pi_l}$ are collected following the new policy π_l for the next policy evaluation step. In ordinary policy-iteration methods, only the new samples $\mathcal{D}^{\pi_l}$ are used for policy evaluation. Thus the previously-collected data samples $\{\mathcal{D}^{\pi_1}, \mathcal{D}^{\pi_2}, \ldots, \mathcal{D}^{\pi_{l-1}}\}$ are not utilized:

$$\pi_1 \overset{\mathrm{E}:\{\mathcal{D}^{\pi_1}\}}{\to} \widehat{Q}^{\pi_1} \overset{\mathrm{I}}{\to} \pi_2 \overset{\mathrm{E}:\{\mathcal{D}^{\pi_2}\}}{\to} \widehat{Q}^{\pi_2} \overset{\mathrm{I}}{\to} \pi_3 \overset{\mathrm{E}:\{\mathcal{D}^{\pi_3}\}}{\to} \cdots \overset{\mathrm{I}}{\to} \pi_{L+1}, \tag{54}$$

where 'E : $\{\mathcal{D}\}$' indicates policy evaluation using the data sample $\mathcal{D}$ and 'I' denotes policy improvement. On the other hand, in SRPI, all previously-collected data samples are reused for policy evaluation as

$$\pi_1 \overset{\mathrm{E}:\{\mathcal{D}^{\pi_1}\}}{\to} \widehat{Q}^{\pi_1} \overset{\mathrm{I}}{\to} \pi_2 \overset{\mathrm{E}:\{\mathcal{D}^{\pi_1},\mathcal{D}^{\pi_2}\}}{\to} \widehat{Q}^{\pi_2} \overset{\mathrm{I}}{\to} \pi_3 \overset{\mathrm{E}:\{\mathcal{D}^{\pi_1},\mathcal{D}^{\pi_2},\mathcal{D}^{\pi_3}\}}{\to} \cdots \overset{\mathrm{I}}{\to} \pi_{L+1}, \tag{55}$$

where appropriate importance weights are applied to each set of previously-collected samples in the policy evaluation step.

Here, we apply the AL technique proposed in the previous section to the SRPI framework. More specifically, we optimize the sampling policy at each iteration. Then the iteration process becomes

$$\pi_1 \overset{\mathrm{E}:\{\mathcal{D}^{\widetilde{\pi}_1}\}}{\to} \widehat{Q}^{\pi_1} \overset{\mathrm{I}}{\to} \pi_2 \overset{\mathrm{E}:\{\mathcal{D}^{\widetilde{\pi}_1},\mathcal{D}^{\widetilde{\pi}_2}\}}{\to} \widehat{Q}^{\pi_2} \overset{\mathrm{I}}{\to} \pi_3 \overset{\mathrm{E}:\{\mathcal{D}^{\widetilde{\pi}_1},\mathcal{D}^{\widetilde{\pi}_2},\mathcal{D}^{\widetilde{\pi}_3}\}}{\to} \cdots \overset{\mathrm{I}}{\to} \pi_{L+1}. \tag{56}$$

Thus, we do not gather samples following the current evaluation policy π_l, but following the sampling policy $\widetilde{\pi}_l$ optimized based on the AL method given in the previous section.

We call this framework *active policy iteration* (API). Figure 6 and Figure 7 show the pseudo code of the API algorithm. Note that the previously-collected samples are used not only for value function approximation, but also for sampling-policy selection. Thus API fully makes use of the samples.

4.2 Numerical Examples

Here we illustrate how the API method behaves using the same 10-state chain-walk problem as Section 3.5 (see Figure 1).

The initial evaluation policy π_1 is set as

$$\pi_1(a|s) \equiv 0.15p_{\mathrm{u}}(a) + 0.85I(a = \operatorname*{argmax}_{a'} \widehat{Q}_0(s,a')), \tag{57}$$

where

$$\widehat{Q}_0(s,a) \equiv \sum_{b=1}^{12} \phi_b(s,a). \tag{58}$$

Policies are updated in the l-th iteration using the ϵ-greedy rule with $\epsilon = 0.15/l$. In the sampling-policy selection step of the l-th iteration, we prepare the four sampling-policy candidates

$$\{\widetilde{\pi}_l^{(1)}, \widetilde{\pi}_l^{(2)}, \widetilde{\pi}_l^{(3)}, \widetilde{\pi}_l^{(4)}\} \equiv \{\overline{\pi}_l^{0.15/l}, \overline{\pi}_l^{0.15/l+0.15}, \overline{\pi}_l^{0.15/l+0.5}, \overline{\pi}_l^{0.15/l+0.85}\}, \tag{59}$$

where $\overline{\pi}_l$ denotes the policy obtained by greedy update using $\widehat{Q}^{\pi_{l-1}}$. The number of iterations to learn the policy is set to $L = 7$, the number of steps is set to $N = 10$, and the number M of episodes is different in each iteration and defined as

$$\{M_1, M_2, M_3, M_4, M_5, M_6, M_7\}, \tag{60}$$

where M_l $(l = 1, 2, \ldots, 7)$ denotes the number of episodes collected in the l-th iteration. In this experiment, we compare two types of scheduling: $\{5, 5, 3, 3, 3, 1, 1\}$ and $\{3, 3, 3, 3, 3, 3, 3\}$, which we refer to as the 'decreasing M' strategy and the 'fixed M' strategy, respectively. The J-value calculation is repeated 5 times for AL. In order to avoid matrix singularity, the regularization parameter is set to $\lambda = 10^{-3}$. The performance of learned policy π_{L+1} is measured by the discounted sum of immediate rewards for test samples $\{r_{m,n}^{\pi_{L+1}}\}_{m,n=1}^{50}$ (50 episodes with 50 steps collected following π_{L+1}):

$$\text{Performance} = \frac{1}{50} \sum_{m=1}^{50} \sum_{n=1}^{50} \gamma^{n-1} r_{m,n}^{\pi_{L+1}}, \tag{61}$$

where the discount factor γ is set to 0.9.

We compare the performance of passive learning (PL; the current policy is used as the sampling policy in each iteration) and the proposed AL method (the best sampling policy is chosen from the policy candidates prepared in each iteration). We repeat the same

Algorithm 1: *ActivePolicyIteration*$(\boldsymbol{\phi}, \pi_1, \lambda, Z)$

$//\boldsymbol{\phi}$ Basis functions, $\boldsymbol{\phi}(s,a) = (\phi_1(s,a), \phi_2(s,a), \ldots, \phi_B(s,a))^\top$
$//\pi_1$ Initial policy, $\pi_1(a|s) \in [0,1]$
$//\lambda$ Regularization parameter, $\lambda > 0$
$//Z$ The number of J-value calculations to take the average J', $Z \in \mathbb{N}$

$l \leftarrow 1$

for $l \leftarrow 1, 2, \ldots, L$

// Determine sampling policy $\widetilde{\pi}_l$ by the active learning method
$\widetilde{\pi}_l \leftarrow$ *SamplingPolicySelection*$(\{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l-1}, \boldsymbol{\phi}, \pi_l, \lambda, Z)$

// Collect episodic samples using policy $\widetilde{\pi}_l$
$\mathcal{D}^{\widetilde{\pi}_l} \leftarrow$ *DataSampling*$(\widetilde{\pi}_l)$

do

// Learn the value function Q^{π_l} from the samples $\{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l}$
$$\boldsymbol{A} \leftarrow \frac{1}{lMN} \sum_{l'=1}^{l} \sum_{m=1}^{M} \sum_{n=1}^{N} \widehat{\boldsymbol{\psi}}(s_{m,n}^{\widetilde{\pi}_{l'}}, a_{m,n}^{\widetilde{\pi}_{l'}}; \{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l}) \widehat{\boldsymbol{\psi}}(s_{m,n}^{\widetilde{\pi}_{l'}}, a_{m,n}^{\widetilde{\pi}_{l'}}; \{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l})^\top w_{m,n}^{\widetilde{\pi}_{l'}}$$
$$\boldsymbol{B} \leftarrow \frac{1}{lMN} \sum_{l'=1}^{l} \sum_{m=1}^{M} \sum_{n=1}^{N} \widehat{\boldsymbol{\psi}}(s_{m,n}^{\widetilde{\pi}_{l'}}, a_{m,n}^{\widetilde{\pi}_{l'}}; \{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l}) r_{m,n}^{\widetilde{\pi}_{l'}}$$
$$\widehat{\boldsymbol{\theta}}_l \leftarrow (\boldsymbol{A} + \lambda \boldsymbol{I})^{-1} \boldsymbol{B}$$

// Update π_l using $\widehat{Q}^{\pi_l}$
$\pi_{l+1} \leftarrow$ *PolicyImprovement*$(\widehat{\boldsymbol{\theta}}_l, \boldsymbol{\phi})$

return (π_{L+1})

Figure 6: The pseudo code of *ActivePolicyIteration*. By the *DataSampling* function, episodic samples (M episodes and N steps) are collected using the input policy. By the *PolicyImprovement* function, the current policy is updated with policy improvement such as ϵ-greedy update or softmax update. The pseudo code of *SamplingPolicySelection* is shown in Algorithm 2 in Figure 7.

Algorithm 2: *SamplingPolicySelection*$(\{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l-1}, \boldsymbol{\phi}, \pi_l, \lambda, Z)$

$//\{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l-1}$ The previously-collected training samples up to $(l-1)$-th iteration
$//\boldsymbol{\phi}$ Basis functions, $\boldsymbol{\phi}(s,a) = (\phi_1(s,a), \phi_2(s,a), \ldots, \phi_B(s,a))^\top$
$//\pi_l$ The evaluation policy in the l-th iteration, $\pi_l(a|s)$ $(\in [0,1])$
$//\lambda$ Regularization parameter, λ (> 0)
$//Z$ The number of J-value calculations to compute the average J', Z $(\in \mathbb{N})$

for $z \leftarrow 1, 2, \ldots, Z$

do

- **for** $k \leftarrow 1, 2, \ldots, K$
  - **do**
    - //Generate sampling policy candidate $\widetilde{\pi}_k^{(l)}$ and collect episodic samples
    - //without immediate rewards using $\widetilde{\pi}_k^{(l)}$
    - $\overline{\mathcal{D}}^{\widetilde{\pi}_k^{(l)}} \leftarrow$ *RewardlessDataSampling*$(\widetilde{\pi}_k^{(l)})$
- //Estimate matrix $\widetilde{\boldsymbol{U}}$
- $//\overline{\mathcal{D}}_0 \equiv \{\overline{\mathcal{D}}^{\widetilde{\pi}_k^{(l)}}\}_{k=1}^K \cup \{\overline{\mathcal{D}}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l-1}$, $\Pi_0 \equiv \{\widetilde{\pi}_k^{(l)}\}_{k=1}^K \cup \{\widetilde{\pi}_{l'}\}_{l'=1}^{l-1}$
- $\widetilde{\boldsymbol{U}} \leftarrow \frac{1}{(K+l-1)MN} \sum_{\pi \in \Pi_0} \sum_{m=1}^{M} \sum_{n=1}^{N} \widehat{\boldsymbol{\psi}}(s_{m,n}^\pi, a_{m,n}^\pi; \overline{\mathcal{D}}_0) \widehat{\boldsymbol{\psi}}(s_{m,n}^\pi, a_{m,n}^\pi; \overline{\mathcal{D}}_0)^\top w_{m,n}^\pi$

do

- **for** $k \leftarrow 1, 2, \ldots, K$
  - **do**
    - //Calculate J_k^z
    - $//\Pi_k \equiv \{\widetilde{\pi}_k^{(l)}\} \cup \{\widetilde{\pi}_{l'}\}_{l'=1}^{l-1}$, $\boldsymbol{h}_{m,n}^\pi \equiv w_{m,n}^\pi \boldsymbol{e}^{(N(m-1)+n)}$ $(\in \mathbb{R}^{MN})$,
    - $//\boldsymbol{e}^{(i)}$ $(\in \mathbb{R}^{MN})$ is the standard basis vector: $\boldsymbol{e}_j^{(i)} \equiv I(i=j)$
    - $\boldsymbol{A} \leftarrow \frac{1}{lMN} \sum_{\pi \in \Pi_k} \sum_{m=1}^{M} \sum_{n=1}^{N} \widehat{\boldsymbol{\psi}}(s_{m,n}^\pi, a_{m,n}^\pi; \overline{\mathcal{D}}_0) \widehat{\boldsymbol{\psi}}(s_{m,n}^\pi, a_{m,n}^\pi; \overline{\mathcal{D}}_0)^\top w_{m,n}^\pi$
    - $\boldsymbol{B} \leftarrow \frac{1}{lMN} \sum_{\pi \in \Pi_k} \sum_{m=1}^{M} \sum_{n=1}^{N} \widehat{\boldsymbol{\psi}}(s_{m,n}^\pi, a_{m,n}^\pi; \overline{\mathcal{D}}_0) \boldsymbol{h}_{m,n}^{\pi\top}$
    - $\widehat{\boldsymbol{L}_k} \leftarrow (\boldsymbol{A} + \lambda \boldsymbol{I})^{-1} \boldsymbol{B}$
    - $J_k^z \leftarrow \mathrm{tr}(\widetilde{\boldsymbol{U}} \widehat{\boldsymbol{L}_k} \widehat{\boldsymbol{L}_k}^\top)$

//Choose the policy $\widetilde{\pi}_{\mathrm{AL}}$ which minimizes $J_k' = \frac{1}{Z} \sum_{z=1}^{Z} J_k^z$ $(k = 1, 2, \ldots, K)$
$\widetilde{\pi}_{\mathrm{AL}} \leftarrow \mathrm{argmin}_{\pi_k'} J_k'$
return $(\widetilde{\pi}_{\mathrm{AL}})$

Figure 7: The pseudo code of *SamplingPolicySelection*. In the function *RewardlessDataSampling*, episodic samples without immediate rewards (M episodes and N steps) are collected. Previously-collected training samples $\{\mathcal{D}^{\widetilde{\pi}_{l'}}\}_{l'=1}^{l}$ are used for the calculation of matrices $\widetilde{\boldsymbol{U}}$, $\boldsymbol{A}$, and $\boldsymbol{B}$ in J-value calculation.

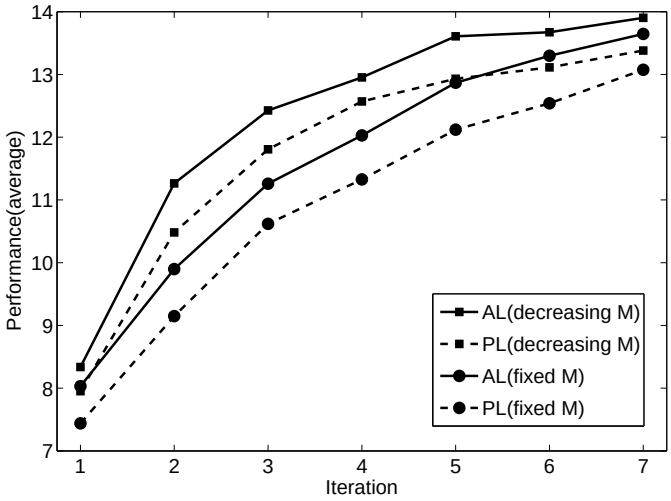


Figure 8: The mean performance over 1000 trials in the 10-state chain-walk experiment. The dotted lines denote the performance of passive learning (PL) and the solid lines denote the performance of the proposed active learning (AL) method. The error bars are omitted for clear visibility. For both the 'decreasing M' and 'fixed M' strategies, the performance of AL after the 7-th iteration is significantly better than that of PL according to the two-tailed paired Student t-test at the significance level 1% applied to the error values at the 7-th iteration.

experiment 1000 times with different random seeds and evaluate the average performance of each learning method. The results are depicted in Figure 8, showing that the proposed AL method works better than PL in both types of episode scheduling with statistical significance by the two-tailed paired *Student t-test* at the significance level 1% (Henkel, 1979) for the error values obtained at the 7-th iteration. Furthermore, the 'decreasing M' strategy outperforms the 'fixed M' strategy for both PL and AL, showing the usefulness of the 'decreasing M' strategy.

5 Experiments

Finally, we evaluate the performance of the proposed API method using a ball-batting robot illustrated in Figure 9, which consists of two links and two joints. The goal of the ball-batting task is to control the robot arm so that it drives the ball as far away as possible. The state space $\mathcal{S}$ is continuous and consists of angles φ_1[rad] ($\in [0, \pi/4]$) and φ_2[rad] ($\in [-\pi/4, \pi/4]$) and angular velocities $\dot{\varphi}_1$[rad/s] and $\dot{\varphi}_2$[rad/s]. Thus a state s ($\in \mathcal{S}$) is described by a four-dimensional vector:

$$s = (\varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2)^\top. \tag{62}$$

The action space $\mathcal{A}$ is discrete and contains two elements:

$$\mathcal{A} = \{a^{(i)}\}_{i=1}^{2} = \{(50, -35)^\top, (-50, 10)^\top\}, \tag{63}$$

where the i-th element ($i = 1, 2$) of each vector corresponds to the torque [N $\cdot$ m] added to joint i.

We use the *Open Dynamics Engine* ('http://ode.org/') for physical calculations including the update of the angles and angular velocities, and collision detection between the robot arm, ball, and pin. The simulation time-step is set to 7.5 [ms] and the next state is observed after 10 time-steps. The action chosen in the current state is kept taken for 10 time-steps. To make the experiments realistic, we add noise to actions: if action $(f_1, f_2)^\top$ is taken, the actual torques applied to the joints are $f_1 + \varepsilon_1$ and $f_2 + \varepsilon_2$, where ε_1 and ε_2 are drawn independently from the Gaussian distribution with mean 0 and variance 3.

The immediate reward is defined as the carry of the ball. This reward is given only when the robot arm collides with the ball for the first time at state s' after taking action a at current state s. For value function approximation, we use the 110 basis functions defined as

$$\phi_{2(i-1)+j} = \begin{cases} I(a = a^{(j)}) \exp\left(-\dfrac{\| s - c_i \|^2}{2\tau^2}\right) & \text{for } i = 1, 2, \ldots, 54 \text{ and } j = 1, 2, \\ I(a = a^{(j)}) & \text{for } i = 55 \text{ and } j = 1, 2, \end{cases} \tag{64}$$

where τ is set to $3\pi/2$ and the Gaussian centers c_i ($i = 1, 2, \ldots, 54$) are located on the regular grid

$$\{0, \pi/4\} \times \{-\pi, 0, \pi\} \times \{-\pi/4, 0, \pi/4\} \times \{-\pi, 0, \pi\}. \tag{65}$$

We set $L = 7$ and $N = 10$. We again compare the 'decreasing M' strategy and the 'fixed M' strategy. The 'decreasing M' strategy is defined by $\{10, 10, 7, 7, 7, 4, 4\}$ and the 'fixed M' strategy is defined by $\{7, 7, 7, 7, 7, 7, 7\}$. The initial state is always set to $s = (\pi/4, 0, 0, 0)^\top$. The regularization parameter is set to $\lambda = 10^{-3}$ and the number of J-calculations in the AL method is set to 5. The initial evaluation policy π_1 is set to the ϵ-greedy policy defined as

$$\pi_1(a|s) \equiv 0.15 p_{\mathrm{u}}(a) + 0.85 I(a = \operatorname*{argmax}_{a'} \widehat{Q}_0(s, a')), \tag{66}$$

$$\widehat{Q}_0(s, a) \equiv \sum_{b=1}^{110} \phi_b(s, a). \tag{67}$$

Policies are updated in the l-th iteration using the ϵ-greedy rule with $\epsilon = 0.15/l$. The way we prepare sampling-policy candidates is the same as the chain-walk experiment in Section 4.2.

The discount factor γ is set to 1 and the performance of learned policy π_{L+1} is measured by the discounted sum of immediate rewards for test samples $\{r_{m,n}^{\pi_{L+1}}\}_{m=1,n=1}^{20,10}$ (20 episodes

with 10 steps collected following π_{L+1}):

$$\text{Performance} = \sum_{m=1}^{M} \sum_{n=1}^{N} r_{m,n}^{\pi_{L+1}}. \tag{68}$$

The experiment is repeated 500 times with different random seeds and the average performance of each learning method is evaluated. The results are depicted in Figure 10, showing that the proposed API method outperforms the PL strategy; for the 'decreasing M' strategy, the performance difference is statistically significant by the two-tailed paired Student t-test at the significance level 1% for the error values at the 7-th iteration.

Based on the experimental evaluation, we conclude that the proposed sampling-policy design method, API, is useful for improving the RL performance. Moreover, the 'decreasing M' strategy is shown to be a useful heuristic to further enhance the performance of API.

6 Conclusions and Future Work

When we cannot afford to collect many training samples due to high sampling costs, it is crucial to choose the most 'informative' samples for efficiently learning the value function. In this paper, we proposed a new data sampling strategy for reinforcement learning based on a statistical active learning method proposed by Sugiyama (2006). The proposed procedure called *active policy iteration* (API)—which effectively combines the framework of sample-reuse policy iteration (Hachiya et al., 2009) with active sampling-policy selection—was shown to perform well in simulations with chain-walk and batting robot control.

Our active learning strategy is a batch method and does not require previously collected reward samples. However, in the proposed API framework, reward samples are available from the previous iterations. A naive extension would be to include those previous samples in the generalization error estimator, for example, following the two-stage active learning scheme proposed by Kanamori & Shimodaira (2003), in which both the bias and variance terms are estimated using the labeled samples. However, such a bias-and-variance approach was shown to perform poorly compared with the variance-only approach (which we used in the current paper) (Sugiyama, 2006). Thus, developing an active learning strategy which can effectively make use of previously collected samples is an important future work.

For the case where the number of episodic samples to be gathered is fixed, we gathered many samples in earlier iterations, rather than gathering samples evenly in each iteration. Although this strategy was shown to perform well in the experiments, so far we do not have strong justification for this heuristic yet. Thus theoretical analysis would be necessary for understanding the mechanism of this approach and further improving the performance.

In the proposed method, the basis function $\boldsymbol{\psi}(s,a)$ defined by Eq.(14) was approximated by $\widehat{\boldsymbol{\psi}}(s,a,\mathcal{D})$ defined by Eq.(20) using samples. When the state-action space is continuous, this is theoretically problematic since only a single sample is available for

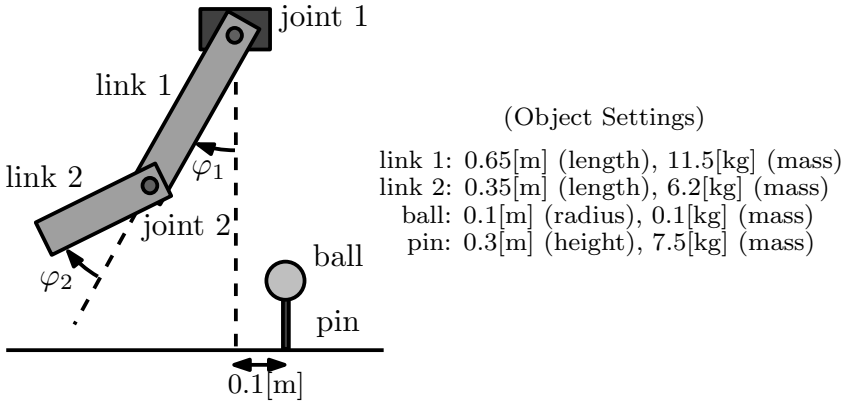


Figure 9: A ball-batting robot.

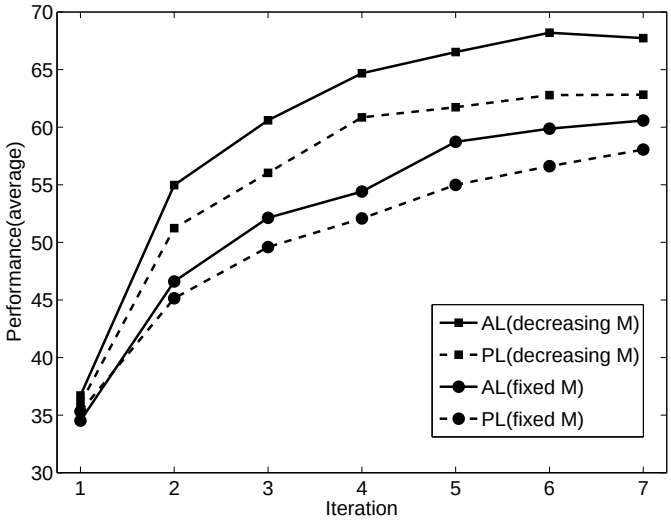


Figure 10: The mean performance over 500 trials in the ball-batting experiment. The dotted lines denote the performance of passive learning (PL) and the solid lines denote the performance of the proposed active learning (AL) method. The error bars are omitted for clear visibility. For the 'decreasing M' strategy, the performance of AL after the 7-th iteration is significantly better than that of PL according to the two-tailed paired Student t-test at the significance level 1% for the error values at the 7-th iteration.

approximation and thus consistency may not be guaranteed. Although we experimentally confirmed that the single-sample approximation gave reasonably good performance, it is important to theoretically investigate the convergence issue in the future work.

The R-max strategy (Brafman & Tennenholtz, 2002) is an approach to controlling the trade-off between exploration and exploitation in the model-based RL framework. The *LSPI R-max* method (Strehl et al., 2007; Li, Littman, & Mansley, 2009) is an application

of the R-max idea to the LSPI framework. It is therefore interesting to investigate the relation between the LSPI R-max method and the proposed method. Moreover, exploring alternative active learning strategies in the model-based RL formulation would be a promising research direction in the future.

Acknowledgments

We thank fruitful comments from anonymous reviewers. MS was supported by Grant-in-Aid for Young Scientists (A), 20680007, AOARD, and SCAT.

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Brafman, R. I., & Tennenholtz, M. (2002). R-max—a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, *3*, 213–231.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Fedorov, V. V. (1972). *Theory of optimal experiments*. New York: Academic Press.

Hachiya, H., Akiyama, T., Sugiyama, M., & Peters, J. (2009). Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Networks*, *22*(10), 1399–1410.

Henkel, R. E. (1979). *Tests of significance*. Beverly Hills: SAGE Publication.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(3), 55–67.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kanamori, T., & Shimodaira, H. (2003). Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference*, *116*(1), 149–162.

Kearns, M., & Singh, S. (1998). Near-optimal reinforcement learning in polynomial time. In *Proceedings of international conference on machine learning* (pp. 260–268).

Kearns, M., & Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, *49*, 209–232.

Lagoudakis, M. G., & Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*, *4*(Dec), 1107–1149.

Li, L., Littman, M. L., & Mansley, C. R. (2008). *Exploration in least-squares policy iteration* (Tech. Rep. No. DCS-TR-641). Rutgers, The State University of New Jersey.

Li, L., Littman, M. L., & Mansley, C. R. (2009). Online exploration in least-squares policy iteration. In *Proceedings of autonomous agents and multiagent systems.*

Poggio, T., & Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, *78*(9), 1481–1497.

Precup, D., Sutton, R. S., & Singh, S. (2000). Eligibility traces for off-policy policy evaluation. In *Proceedings of the seventeenth international conference on machine learning* (pp. 759–766). Morgan Kaufmann.

Strehl, A. L., Diuk, C., & Littman, M. L. (2007). Efficient structure learning in factored-state mdps. In *Proceedings of the twenty-second national conference on artificial intelligence* (pp. 645–650).

Sugiyama, M. (2006). Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, *7*, 141–166.

Sugiyama, M., & Nakajima, S. (2009). Pool-based active learning in approximate linear regression. *Machine Learning*, *75*(3), 249-274.

Sutton, R. S., & Barto, G. A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. Washington DC: V. H. Winston.

Wiens, D. P. (2000). Robust weights and designs for biased regression models: Least squares and generalized M-estimation. *Journal of Statistical Planning and Inference*, *83*(2), 395–412.

Lighting Condition Adaptation for Perceived Age Estimation

Kazuya Ueki
NEC Soft, Ltd., Japan

Masashi Sugiyama
Tokyo Institute of Technology, Japan

Yasuyuki Ihara
NEC Soft, Ltd., Japan

Abstract

Over the recent years, a great deal of effort has been made to age estimation from face images. It has been reported that age can be accurately estimated under controlled environment such as frontal faces, no expression, and static lighting conditions. However, it is not straightforward to achieve the same accuracy level in real-world environment because of considerable variations in camera settings, facial poses, and illumination conditions. In this paper, we apply a recently-proposed machine learning technique called *covariate shift adaptation* to alleviating lighting condition change between laboratory and practical environment. Through real-world age estimation experiments, we demonstrate the usefulness of our proposed method.

Keywords

face recognition, age estimation, covariate shift adaptation, lighting condition change, Kullback-Leibler importance estimation procedure, importance-weighted regularized least-squares

1 Introduction

In recent years, demographic analysis in public places such as shopping malls and stations is attracting a great deal of attention. Such demographic information is useful for various purposes including designing effective marketing strategies and targeted advertisement based on customers' gender and age. For this reason, a number of approaches have been explored for age estimation from face images [2, 3], and several databases became publicly available recently [1, 6].

The recognition performance of age prediction systems is significantly influenced, e.g., by the type of camera, camera calibration, and lighting variations, and the publicly available databases were mainly collected in semi-controlled environment. For this reason, existing age prediction systems built upon such databases tend to perform poorly in real-world environment.

The situation where training and test data are drawn from different distributions is called *covariate shift* [8, 11, 12]. In this paper, we formulate the problem of age estimation in real-world environment as a supervised learning problem under covariate shift. Within the covariate shift framework, a method called *importance-weighted least-squares* allows us to alleviate the influence of environmental changes, by assigning higher weights to data samples having high test input densities and low training input densities. We demonstrate through real-world experiments that age estimation based on covariate shift adaptation achieves higher accuracy than baseline approaches.

2 Proposed Method

In this section, we formulate the problem of age estimation as a supervised learning problem under covariate shift, and then describe our proposed method.

2.1 Formulation

Throughout this paper, we perform age estimation based not on subjects' real age, but on their *perceived* age. Thus, the 'true' age of the subject y is defined as the average perceived age evaluated by those who observed the subject's face images (the value is rounded-off to the nearest integer).

Let us consider a regression problem of estimating the age y^* of subject $\boldsymbol{x}$ (face features). We use the following model for regression.

$$f(\boldsymbol{x};\boldsymbol{\alpha}) = \sum_{i=1}^{n_{tr}} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i^{tr}), \tag{1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n_{tr}})^\top$ is a model parameter, $^\top$ denotes the transpose, and $K(\boldsymbol{x}, \boldsymbol{x}')$ is a *positive definite kernel* [7].

Suppose we are given labeled training data $\{(\boldsymbol{x}_i^{tr}, y_i^{tr})\}_{i=1}^{n_{tr}}$. A standard approach to learning the model parameter $\boldsymbol{\alpha}$ would be *regularized least-squares* [4].

$$\min_{\boldsymbol{\alpha}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} (y_i^{tr} - f(\boldsymbol{x}_i^{tr};\boldsymbol{\alpha}))^2 + \lambda \|\boldsymbol{\alpha}\|^2 \right], \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm, and $\lambda(>0)$ is the regularization parameter to avoid overfitting.

Below, we explain that merely using regularized least-squares is not appropriate in real-world perceived age prediction, and show how to cope with this problem.

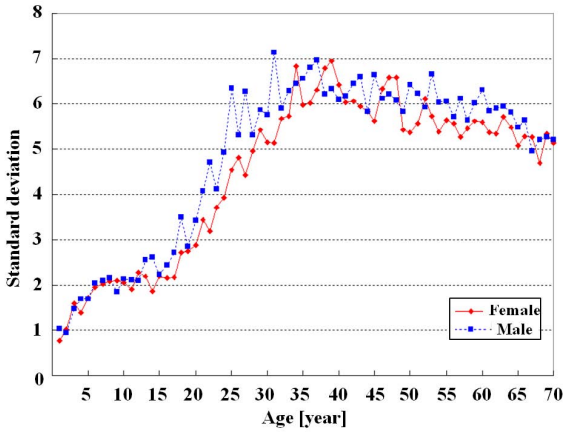


Figure 1: The relation between subjects' perceived age y^* (horizontal axis) and its standard deviation (vertical axis)

.

2.2 Incorporating Age Perception Characteristics

Human age perception is known to have heterogeneous characteristics, e.g., it is rare to misjudge the age of a 5-year-old child as 15 years old, but the age of a 35-year-old person is often misjudged as 45 years old. In order to quantify this phenomenon, a large-scale questionnaire survey was carried out in [15]: Each of 72 volunteers were asked to give age labels y to approximately 1000 face images. Figure 1 depicts the relation between subjects' perceived age y^* and its standard deviation. This shows that the perceived age deviation tends to be small in younger age brackets and large in older age brackets.

In order to match characteristics of our age prediction system to those of human age perception, we weight the goodness-of-fit term in Eq.(2) according to the inverse variance of the perceived age:

$$\min_{\boldsymbol{\alpha}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{(y_i^{tr} - f(\boldsymbol{x}_i^{tr}; \boldsymbol{\alpha}))^2}{w_{age}(y_i^{tr})^2} + \lambda \|\boldsymbol{\alpha}\|^2 \right], \tag{3}$$

where $w_{age}(y)$ is the standard deviation of the perceived age (see Figure 1 again).

2.3 Coping with Lighting Condition Change

When designing age estimation systems, the environment of recording training face images is often different from the test environment in terms of lighting conditions. Typically, training data are recorded indoors such as a studio with appropriate illumination. On the other hand, in real-world environment, lighting conditions have considerable varieties, e.g., strong sunlight might be cast from a side of faces or there is no enough light. In such situations, age estimation accuracy is significantly degraded.

Let $p_{tr}(\boldsymbol{x})$ be the training input density and $p_{te}(\boldsymbol{x})$ be the test input density. When these two densities are different, it would be natural to emphasize the influence of train-

ing samples $(\boldsymbol{x}_i^{tr}, y_i^{tr})$ which have high similarity to data in the test environment. Such adjustment can be systematically carried out as follows [8, 11, 12]:

$$\min_{\boldsymbol{\alpha}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} w_{imp}(\boldsymbol{x}_i^{tr}) \frac{(y_i^{tr} - f(\boldsymbol{x}_i^{tr}; \boldsymbol{\alpha}))^2}{w_{age}(y_i^{tr})^2} + \lambda \|\boldsymbol{\alpha}\|^2 \right], \tag{4}$$

i.e., the goodness-of-fit term in Eq.(3) is weighted according to the *importance function*:

$$w_{imp}(\boldsymbol{x}) = \frac{p_{te}(\boldsymbol{x})}{p_{tr}(\boldsymbol{x})}.$$

The solution of Eq.(4) can be obtained analytically by

$$\widehat{\boldsymbol{\alpha}} = (\boldsymbol{K}^{tr} \boldsymbol{W}^{tr} \boldsymbol{K}^{tr} + n_{tr} \lambda \boldsymbol{I}_{n_{tr}})^{-1} \boldsymbol{K}^{tr} \boldsymbol{W}^{tr} \boldsymbol{y}^{tr}, \tag{5}$$

where $\boldsymbol{K}^{tr}$ is the kernel matrix whose (i, i')-th element is defined by

$$K_{i,i'}^{tr} = K(\boldsymbol{x}_i^{tr}, \boldsymbol{x}_{i'}^{tr}),$$

$\boldsymbol{W}^{tr}$ is the n_{tr}-dimensional diagonal matrix with (i, i)-th diagonal element defined by

$$W_{i,i}^{tr} = \frac{w_{imp}(\boldsymbol{x}_i^{tr})}{w_{age}(y_i^{tr})^2},$$

$\boldsymbol{I}_{n_{tr}}$ is the n_{tr}-dimensional identity matrix, and $\boldsymbol{y}^{tr}$ is the n_{tr}-dimensional vector with i-th element defined by y_i^{tr}.

When the number of training data n_{tr} is large, we may reduce the number of kernels in Eq.(1) so that the inverse matrix in Eq.(5) can be computed with limited memory; or we may compute the solution numerically by a *stochastic gradient-decent method*.

2.4 Importance-Weighted Cross-Validation (IWCV)

In supervised learning, the choice of models (for example, the basis functions and the regularization parameter) is crucial for obtaining better performance. *Cross-validation* (CV) would be one of the most popular techniques for model selection [9]. CV has been shown to give an *almost* unbiased estimate of the generalization error with finite samples [7], but such almost unbiasedness is no longer fulfilled under covariate shift.

To cope with this problem, a variant of CV called *importance-weighted CV* (IWCV) has been proposed [11]. Let us randomly divide the training set

$$\mathcal{Z} = \{(\boldsymbol{x}_i^{tr}, y_i^{tr})\}_{i=1}^{n_{tr}}$$

into T disjoint non-empty subsets $\{\mathcal{Z}_t\}_{t=1}^{T}$ of (approximately) the same size. Let $f_{\mathcal{Z}_t}(\boldsymbol{x})$ be a function learned from $\mathcal{Z} \backslash \mathcal{Z}_t$ (i.e., without $\mathcal{Z}_t$). Then the *T-fold IWCV* (IWCV) estimate of the generalization error is given by

$$\frac{1}{T} \sum_{t=1}^{T} \frac{1}{|\mathcal{Z}_t|} \sum_{(\boldsymbol{x}, y) \in \mathcal{Z}_t} \frac{w_{imp}(\boldsymbol{x})}{w_{age}(y)^2} (f_{\mathcal{Z}_t}(\boldsymbol{x}) - y)^2,$$

Table 1: Pseudo code of KLIEP. './' indicates the element-wise division. Inequalities and the 'max' operation for vectors are applied in an element-wise manner.

Input: $\{\boldsymbol{x}_i^{tr}\}_{i=1}^{n_{tr}}, \{\boldsymbol{x}_j^{te}\}_{j=1}^{n_{te}}$
Output: $\widehat{w}(\boldsymbol{x})$

Choose $\{\boldsymbol{c}_k\}_{k=1}^{b}$ as a subset of $\{\boldsymbol{x}_j^{te}\}_{j=1}^{n_{te}}$;
$A_{j,k} \leftarrow \exp\left(-\|\boldsymbol{x}_j^{te} - \boldsymbol{c}_k\|^2/(2\gamma^2)\right)$;
$b_k \leftarrow \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \exp\left(-\|\boldsymbol{x}_i^{tr} - \boldsymbol{c}_k\|^2/(2\gamma^2)\right)$;
Initialize $\boldsymbol{\beta}(> \mathbf{0})$ and ε $(0 < \varepsilon \ll 1)$;
Repeat until convergence
    $\boldsymbol{\beta} \leftarrow \varepsilon A^\top(\mathbf{1}./A\boldsymbol{\beta})$;
    $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + (1 - \boldsymbol{b}^\top\boldsymbol{\beta})\boldsymbol{b}/(\boldsymbol{b}^\top\boldsymbol{b})$;
    $\boldsymbol{\beta} \leftarrow \max(\mathbf{0}, \boldsymbol{\beta})$;
    $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}/(\boldsymbol{b}^\top\boldsymbol{\beta})$;
end

where $|\mathcal{Z}_t|$ denotes the number of samples in the subset $\mathcal{Z}_t$.

It was proved that IWCV gives an *almost* unbiased estimate of the generalization error even under covariate shift [11].

2.5 Kullback-Leibler Importance Estimation Procedure (KLIEP)

In order to compute the solution (5) or performing IWCV, we need the importance weights $w_{imp}(\boldsymbol{x}_i^{tr}) = p_{te}(\boldsymbol{x}_i^{tr})/p_{tr}(\boldsymbol{x}_i^{tr})$, which include two probability densities $p_{tr}(\boldsymbol{x})$ and $p_{te}(\boldsymbol{x})$. However, since density estimation is a hard problem, a two-stage approach of first estimating $p_{tr}(\boldsymbol{x})$ and $p_{te}(\boldsymbol{x})$ and then taking their ratio may not be reliable. Here we describe a method called *Kullback-Leibler Importance Estimation Procedure* (KLIEP) [12], which allows us to directly estimate the importance function $w_{imp}(\boldsymbol{x})$ without going through density estimation of $p_{tr}(\boldsymbol{x})$ and $p_{te}(\boldsymbol{x})$.

Let us model $w_{imp}(\boldsymbol{x})$ using the following model:

$$\widehat{w}_{imp}(\boldsymbol{x}) = \sum_{k=1}^{b} \beta_k \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_k\|^2}{2\gamma^2}\right), \tag{6}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_b)^\top$ is a parameter, and $\{\boldsymbol{c}_k\}_{k=1}^{b}$ is a subset of test input samples $\{\boldsymbol{x}_j^{te}\}_{j=1}^{n_{te}}$. Using the model $\widehat{w}_{imp}(\boldsymbol{x})$, we can estimate the test input density $p_{te}(\boldsymbol{x})$ by

$$\widehat{p}_{te}(\boldsymbol{x}) = \widehat{w}_{imp}(\boldsymbol{x}) p_{tr}(\boldsymbol{x}). \tag{7}$$

We determine the parameter $\boldsymbol{\beta}$ in the model (7) so that the Kullback-Leibler divergence

Figure 2: Examples of face images under different lighting conditions (left: standard lighting, middle: dark, right: strong light from a side)

from p_{te} to $\widehat{p}_{te}$ is minimized:

$$\begin{aligned} KL(p_{te}\|\widehat{p}_{te}) &= \int p_{te}(\boldsymbol{x}) \log \frac{p_{te}(\boldsymbol{x})}{\widehat{p}_{te}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\ &= \int p_{te}(\boldsymbol{x}) \log \frac{p_{te}(\boldsymbol{x})}{p_{tr}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} - \int p_{te}(\boldsymbol{x}) \log \widehat{w}_{imp}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \end{aligned}$$

We ignore the first term (which is a constant) and impose $\widehat{w}_{imp}(\boldsymbol{x})$ to be non-negative and normalized. Then we obtain the following convex optimization problem:

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \left[\sum_{j=1}^{n_{te}} \log \left(\sum_{k=1}^{b} \beta_k \exp\left(-\frac{\|\boldsymbol{x}_j^{te} - \boldsymbol{c}_k\|^2}{2\gamma^2} \right) \right) \right], \\ &\text{s.t.} \begin{cases} \beta_k \geq 0 \quad \text{for } k = 1, \ldots, b, \\ \dfrac{1}{n_{tr}} \displaystyle\sum_{i=1}^{n_{tr}} \left(\sum_{k=1}^{b} \beta_k \exp\left(-\frac{\|\boldsymbol{x}_i^{tr} - \boldsymbol{c}_k\|^2}{2\gamma^2} \right) \right) = 1. \end{cases} \end{aligned}$$

A pseudo code of KLIEP is described in Table 1. The tuning parameter γ can be optimized based on *likelihood cross-validation* (LCV) [12].

3 Empirical Evaluation

In this section, we experimentally evaluate the performance of the proposed method using in-house face-age datasets.

We use the face images recorded under 17 different lighting conditions: for instance, average illuminance from above is approximately 1000 lux and 500 lux from the front in the standard lighting condition, 250 lux from above and 125 lux from the front in the dark setting, and 190 lux from left and 750 lux from right in another setting (see Figure 2). Note that these 17 lighting conditions are diverse enough to cover real-world lighting conditions. Images were recorded as movies with camera at depression angle 15 degrees. The number of subjects is approximately 500 (250 for each gender). We used a

face detector for localizing the two eye-centers, and then rescaled the image to 64 × 64 pixels. The number of face images in each environment is about 2500 (5 face images × 500 subjects).

As pre-processing, a neural network feature extractor [14] was used to extract 100-dimensional features from 64 × 64 face images. The kernel regression model (1) with the following Gaussian kernel was employed for the extracted 100-dimensional data:

$$K_\sigma(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right).$$

We constructed the male/female age prediction models only using male/female data, assuming that gender classification had been correctly carried out.

We split the 250 subjects into the *training set* (200 subjects) and the *test set* (50 subjects). The training set was used for training the kernel regression model (1), and the test set was used for evaluating its generalization performance. For the test samples $\{(\boldsymbol{x}_i^{te}, y_i^{te})\}_{i=1}^{n_{te}}$ taken from the test set in the environment with strong light from a side, age-weighted mean square error (WMSE)

$$\mathrm{WMSE} = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \frac{(y_i^{te} - f(\boldsymbol{x}_i^{te}; \widehat{\boldsymbol{\alpha}}))^2}{w_{age}(y_i^{te})^2}$$

was calculated as a performance measure. The training test sets were shuffled 5 times in such a way that each subject was selected as a test sample once. The final performance was evaluated based on the average WMSE over the 5 trials.

We compared the performance of the proposed method with the two baseline methods:

Baseline method 1: Training samples were taken only from the standard lighting condition and age-weighted regularized least-squares (3) was used for training.

Baseline method 2: Training samples were taken from all 17 different lighting conditions and age-weighted regularized least-squares (3) was used for training.

The importance weights were not used in these baseline methods. The Gaussian width σ and the regularization parameter λ were determined based on 4-fold CV over WMSE, i.e., the training set was further divided into a training part (150 subjects) and a validation part (50 subjects).

In the proposed method, training samples were taken from all 17 different lighting conditions (which is the same as the baseline method 2). The importance weights were estimated by KLIEP using the training samples and additional *unlabeled* test samples; the hyper-parameter γ in KLIEP was determined based on 2-fold LCV [12]. We then computed the average importance score over different samples for each lighting condition and used the average importance score for training the regression model. The Gaussian width σ and the regularization parameter λ in the regression model were determined based on 4-fold IWCV [11].

Table 2: The test performance measured by WMSE.

	Male	Female
Baseline method 1	2.83	6.51
Baseline method 2	2.64	4.40
Proposed method	**2.54**	**3.90**

Table 2 summarizes the experimental results, showing that, for both male and female data, the baseline method 2 is better than the baseline method 1 and the proposed method is better than the baseline method 2. This illustrates the effectiveness of the proposed method. Note that WMSE for female subjects is substantially larger than that for male subjects. The reason for this would be that female subjects tend to have more divergence such as short/long hair and with/without makeup, which makes prediction harder [16].

4 Summary and Future Works

Lighting condition change is one of the critical causes of performance degradation in age prediction from face images. In this paper, we proposed to employ a machine learning technique called *covariate shift adaptation* for alleviating the influence of lighting condition change. We demonstrated the effectiveness of our proposed method through real-world perceived age prediction experiments.

In the experiments in Section 3, test samples were collected from a particular lighting condition, and samples from the same lighting condition were also included in the training set. Although we believe this setup to be practical, it would be interesting to evaluate the performance of the proposed method when no overlap in the lighting conditions exists between training and test data.

In principle, the covariate shift framework allows us to incorporate not only lighting condition change, but also various types of environment change such as face pose variation and camera setting change. In our future work, we will investigate whether the proposed approach is still useful in such challenging scenarios.

Recently, novel approaches to density ratio estimation for high-dimensional problems have been explored [5, 10, 17, 13]. In our future work, we would like to incorporating these new ideas into our framework of perceived age estimation, and see how the prediction performance can be further improved.

References

[1] *The FG-NET Aging Database*. http://www.fgnet.rsunit.com/.

[2] Y. Fu, Y. Xu, and T. S. Huang. Estimating human age by manifold analysis of face pictures and regression on aging features. *Proceedings of the IEEE Multimedia and Expo*, pages 1383–1386, 2007.

[3] G. Guo, G. Mu, Y. Fu, C. Dyer, and T. Huang. A study on automatic age estimation using a large database. *International Conference on Computer Vision in Kyoto (ICCV 2009)*, pages 1986–1991, 2009.

[4] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

[5] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

[6] K. J. Ricanek and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. *Proceedings of the IEEE 7th International Conference on Automatic Face and Gesture Recognition (FGR 2006)*, pages 341–345, 2006.

[7] B. Schölkopf and A. J. Smola. *Learning with Kernels*, MIT Press, Cambridge, MA, USA, 2002.

[8] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[9] M. Stone. Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society, Series B*, 36:111-147, 1974.

[10] M. Sugiyama, M. Kawanabe, P. L. Chui. Dimensionality reduction for density ratio estimation in high-dimensional spaces. *Neural Networks*, 23(1):44-59, 2010.

[11] M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, May 2007.

[12] M. Sugiyama, T. Suzuki, T., S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[13] M. Sugiyama, M. Yamada, P. von Bünau, T. Suzuki, T. Kanamori, and M. Kawanabe. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search, *Neural Networks*, to appear.

[14] F. H. C. Tivive and A. Bouzerdoum. A gender recognition system using shunting inhibitory convolutional neural networks. *Proceedings of the International Joint Conference on Neural Networks (IJCNN '06)*, pages 5336–5341, 2006.

[15] K. Ueki, M. Sugiyama, Y. Ihara. A semi-supervised approach to perceived age prediction from face images. *IEICE Transactions on Information and Systems*, to appear.

[16] K. Ueki, M. Miya, T. Ogawa, T. Kobayashi. Class distance weighted locality preserving projection for automatic age estimation. *Proceedings of IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2008)*, pages 1–5, 2008.

[17] M. Yamada, M. Sugiyama, G. Wichern, and J. Simm. Direct importance estimation with a mixture of probabilistic principal component analyzers. *IEICE Transactions on Information and Systems*, E93-D(10), 2846–2849, 2010.