AFRL-AFOSR-JP-TR-2018-0052

Bayesian Learning with Unbounded Capacity from Heterogeneous and Set-Valued Data

Dinh Phung
DEAKIN UNIVERSITY

06/13/2018
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ IOA
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-06-2018	Final	30 Sep 2016 to 29 Sep 2018

4. TITLE AND SUBTITLE
Bayesian Learning with Unbounded Capacity from Heterogeneous and Set-Valued Data

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA2386-16-1-4138

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Dinh Phung, Tu-Bao Ho

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEAKIN UNIVERSITY
221 BURWOOD HWY
BURWOOD, 3125 AU

8. PERFORMING ORGANIZATION REPORT NUMBER

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

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR IOA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-JP-TR-2018-0052

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The aim of this project was to advance machine learning methods grounded in Bayesian theory, optimal transport, point processes and random finite sets to deal with growing complexity and heterogeneity of large-scale data. The research team has focused on two main themes: i) developing necessary theory to perform parameter estimation with latent variables for set-valued data using point process theory, and ii) formulating and developing fast inference procedures for Bayesian models via Wasserstein and optimal transport theory. Both of these themes are related through their roles as the building blocks to construct more advanced and scalable models to deal with not only standard data types (such as vectors and matrices), but also set-valued data.

The most important results are twofold: a) new model-based method to reason and learn from set-valued (aka point pattern data). This includes new models for classification, clustering and novelty detection with point pattern data; and its extension to deal with sequential data. And, b) a new Wasserstein-based formulation for multi-level clustering in high-dimensional data; this formulation also provides a new scalable framework for Bayesian inference and as opposed to traditional information-theoretic approaches.

These results have been documented in 6 research papers where four have been accepted for publication (DSAA, ICPR and ICML), one is under review (Pattern Recognition) and another one is under preparation to be submitted to Journal of Machine Learning Research.

15. SUBJECT TERMS
Set-valued data, Bayesian inference, Bayesian nonparametric model, Heterogenous, Infinite model capacity

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			SINGLETON, BRIANA
Unclassified	Unclassified	Unclassified	SAR		19b. TELEPHONE NUMBER *(Include area code)* 315-227-7007

Final Report for AOARD Grant FA2386-16-1-4138

"Bayesian Learning with Unbounded Capacity from Heterogeneous and Set-Valued Data"

15 June 2018

Principal Investigator: Prof. **Dinh Phung**
Email: dinh.phung@deakin.edu.au
Deakin University of Technology, Australia
Address: 75 Pigdons Road, Waurn Ponds, VIC 3216, Australia
P: +61 3 5227 2082, F: +61 3 5227 2028

Co-PI: Prof. **Bao-Tu Ho**
Email: bao@jaist.ac.jp
Japan Advanced Institute of Science Technology (JAIST), Japan

Period of Performance: 09/30/2016 – 06/15/2018

Abstract: Large-scale and modern datasets have reshaped machine learning research and practices. They are not only bigger in size, but predominantly heterogeneous and growing in their complexity. The aim of this project is to advance machine learning methods grounded in Bayesian theory, optimal transport, point processes and random finite sets to deal with growing complexity and heterogeneity of large-scale data. Throughout this project, we have focused on two main themes: i) developed necessary theory to perform parameter estimation with latent variables for set-valued data using point process theory, and ii) formulated and developed fast inference procedures for Bayesian models via Wasserstein and optimal transport theory. Both of these themes are related through their roles as the building blocks to construct more advanced and scalable models to deal with not only standard data types (such as vectors and matrices), but also set-valued data.

The most important results came out from the support of this grant are twofold: a) new model-based method to reason and learn from set-valued (aka point pattern data). This includes new models for classification, clustering and novelty detection with point pattern data; and its extension to deal with sequential data. And, b) a new Wasserstein-based formulation for multi-level clustering in high-dimensional data; this formulation also provides a new scalable framework for Bayesian inference and as opposed to traditional information-theoretic approaches.

These results have been documented in 6 research papers where four have been accepted for publication (DSAA, ICPR and ICML), one is under review (Pattern Recognition) and another one is under preparation to be submitted to Journal of Machine Learning Research.

1 Model-based Learning with Point Pattern Data

The research outcomes reported in this section have been published in [1][2][3][5] (cf. Section 4).

1.1 What is point pattern data and why it is important?

Point patterns – which are also known as sets or multi-sets of unordered points – arise in many data analysis problems where they are commonly known as 'bags', e.g. in multiple instance learning [3, 15, 14], natural language processing and information retrieval ('bag-of-words') [40, 33, 41], image and scene categorization ('bag-of-visual-words') [17, 54, 69], and in sparse data ('bag-of-features') [16, 32]. A statistical data model, usually specified by the *likelihood* function, plays a fundamental role in model-based data analysis. However, statistical point pattern models have not received much attention in the development of machine learning algorithms for point pattern data.

To motivate the development of suitable likelihood functions for point patterns, let us consider an example in novelty detection. Suppose that apples from an apple tree land on the ground independently from each other, and that the daily point patterns of landing positions are also independent. Further, the probability density, p_f, of the landing position, learned from 'normal' training data, is shown in Fig. 1. Since the apple landing positions are independent, following common practice (see e.g., [40, 33, 41, 17, 12]) the likelihood that the apples land at positions $x_1, ..., x_m$ is given by the joint (probability) density $p(x_1, ..., x_m)$, which by the independence of the landing positions, is $\prod_{i=1}^{m} p_f(x_i)$.

Suppose we observe one apple landing at x_1 on day 1, and two apples landing at x_2 and x_3 on day 2 (see Fig. 1), which of these daily landing patterns is more likely to be a novelty? The common practice (see e.g., [39]) is to examine the 'normal' likelihoods $p(x_1) = p_f(x_1) = 0.2$ and $p(x_2, x_3) = p_f(x_2)\, p_f(x_3) = 0.36$, from which it is intuitive that the day 1 pattern is novel. However, had we measured distance in centimeters (p_f is scaled by 10^{-2}), then $p(x_1) = 0.002$ is greater than $p(x_2, x_3) = 0.000036$, which contradicts the previous conclusion! This phenomenon arises because $p(x_1)$ is measured in "m^{-1}" or "cm^{-1}" whereas $p(x_2, x_3)$ is measured in "m^{-2}" or "cm^{-2}".

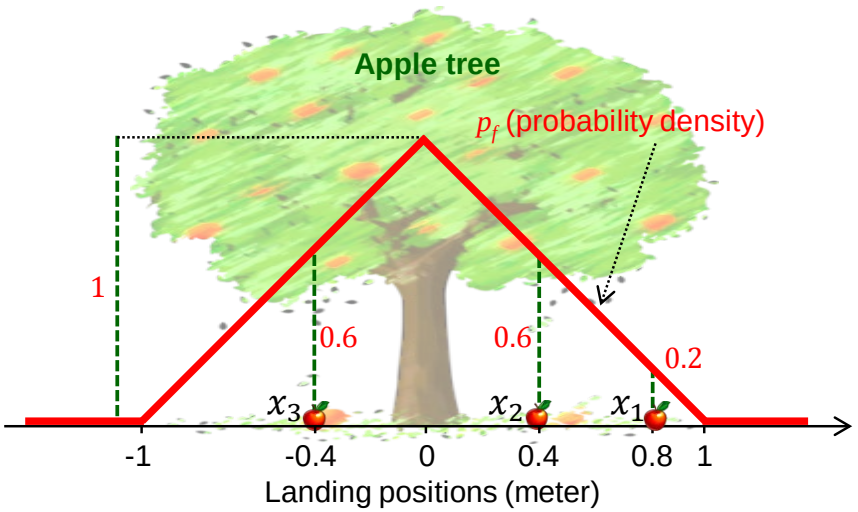
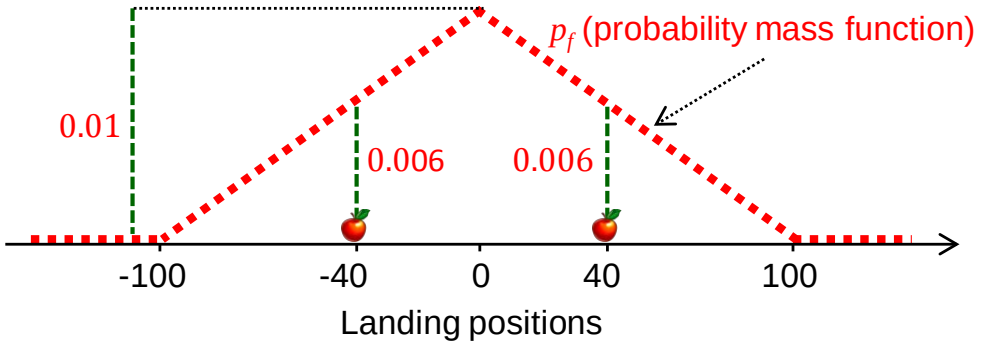


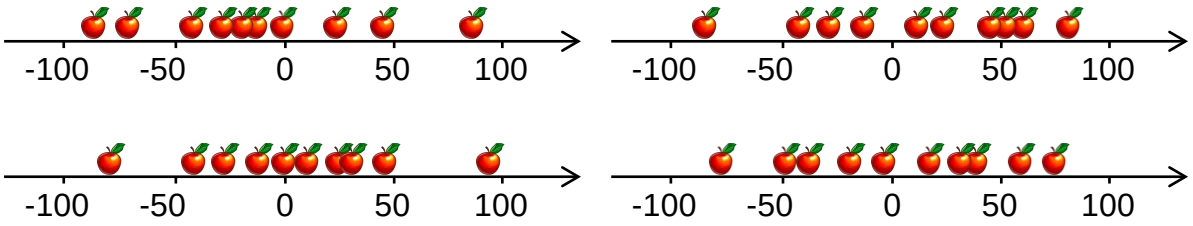
Figure 1: Distribution of landing positions, $x_1 = 0.8\,\mathrm{m}$ is 3 times less likely than $x_2 = 0.4\,\mathrm{m}$ and $x_3 = -0.4\,\mathrm{m}$ which are equally likely. Credit: clipartbest.com (apple tree clipart)

To rule-out the effect of unit incompatibility, we assume only 201 evenly spaced possible landing positions, and a (unit-less) probability mass function on the discrete set $\{-100, \ldots, 100\}$ shown in Fig. 2a (instead of a probability density). Fig. 2b, shows 4 point patterns from the 'normal' training data set, and Fig. 2c shows 2 new observations X_1 and X_2. Since X_2 has only 1 feature, whereas X_1 and the 'normal' observations each has around 10 features, it is intuitive that X_2 is novel. However, its likelihood is much higher than that of X_1. This counter intuitive phenomenon arises from the lack of cardinality information in the likelihood.

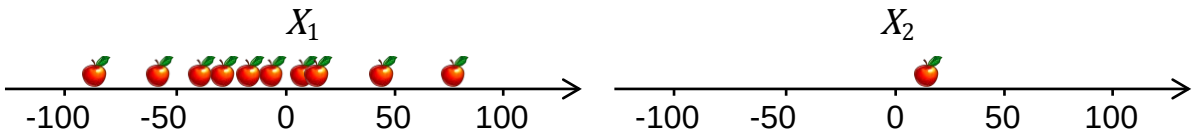
The above example demonstrates that the joint probability density of the constituent points is not the likelihood of a point pattern. Such likelihood functions could lead to erroneous results in point pattern learning tasks. Learning from point pattern data using point process theory [20, 60, 43] may alleviate the problem. Likelihood functions derived from point process theory are probability densities of random point patterns, which incorporate both cardinality and feature information, and avoid the unit of measurement inconsistency. Moreover, they enable the extension of model-based formulations for learning tasks such as classification, novelty detection,



(a) Distribution of discrete landing positions.



(b) Examples of 'Normal' observations.



(c) Input observations: $p(X_1) \approx 2 \times 10^{-23}$ and $p(X_2) = 0.009$.

Figure 2: An example with discrete landing positions.

and clustering to point pattern data in a conceptually transparent yet principled manner. Such a framework, facilitates the development of tractable point pattern models as well as solutions for learning and decision making.

We now summarize the elements of point process theory and presents some basic models for point pattern data. For further information, we refer the reader to textbooks such as [20, 60, 43].

Point Process. A point pattern is a set or multi-set of unordered points. While a multi-set may contain repeated elements, it can be equivalently represented by a set. Specifically, a multi-set with elements $x_1, \ldots, x_m$ of respective multiplicities $N_1, \ldots, N_m$, can be represented as the set $\{(x_1, N_1), \ldots, (x_m, N_m)\}$. A point pattern X can be characterized as a *counting measure* N on the space $\mathcal{X}$ of features, defined, for each (compact) set $A \subseteq \mathcal{X}$ by

$$N(A) = \text{number of points of } X \text{ falling in } A. \tag{1}$$

The values of the counting variables $N(A)$ for all subsets A provide sufficient information to reconstruct the point pattern X [20, 60]. The points of X are the set of x such that $N(\{x\}) > 0$. A point pattern is said to be: *finite* if it has a finite number of points, i.e., $N(\mathcal{X}) < \infty$; and *simple* if it contains no repeated points, i.e., $N(\{x\}) \leq 1$ for all $x \in \mathcal{X}$.

Formally a point process is defined as a *random counting measure*. A random counting measure N may be viewed as a collection of random variables $N(A)$ indexed by $A \subseteq \mathcal{X}$. A point process is *finite* if its realizations are finite almost surely, and *simple* if its realizations are simple almost surely. Likelihoods for point patterns in a countable space is conceptually straightforward, and hereon we only consider point processes on a compact subset $\mathcal{X}$ of $\mathbb{R}^d$.

Probability Density. In general the probability density of a point process may not exist [63, 5]. To ensure that probability densities are available, we restrict ourselves to simple finite point processes, which are equivalent to *random finite sets* (RFSs) [5], i.e., random variables taking values in $\mathcal{F}(\mathcal{X})$, the space of finite subsets of $\mathcal{X}$.

The probability density $f : \mathcal{F}(\mathcal{X}) \to [0, \infty)$ of a random finite set is usually taken with respect to the dominating measure μ, defined for each (measurable) $\mathcal{T} \subseteq \mathcal{F}(\mathcal{X})$, by (see e.g., [24, 43, 66]):

$$\mu(\mathcal{T}) = \sum_{i=0}^{\infty} \frac{1}{i!U^i} \int \mathbf{1}_{\mathcal{T}} (\{x_1, \ldots, x_i\}) \, \mathrm{d} (x_1, \ldots, x_i) , \tag{2}$$

where U is the unit of hyper-volume in $\mathcal{X}$, $\mathbf{1}_{\mathcal{T}}(\cdot)$ is the indicator function for $\mathcal{T}$, and by convention the integral for $i = 0$ is the integrand evaluated at $\emptyset$. It was shown in [66] that the integral of a function f with respect to μ, given by

$$\int f(X)\mu(\mathrm{d}X) = \sum_{i=0}^{\infty} \frac{1}{i!U^i} \int f (\{x_1, \ldots, x_i\}) \, \mathrm{d} (x_1, \ldots, x_i) , \tag{3}$$

is equivalent to Mahler's set integral [36, 37].

The probability density of a random finite set, with respect to μ, evaluated at $\{x_1, \ldots, x_i\}$ can be written as [63, p. 27] (Eqs. (1.5), (1.6), and (1.7)):

$$f (\{x_1, \ldots, x_i\}) = p_c(i) \, i! \, U^i f_i (x_1, \ldots, x_i) , \tag{4}$$

where $p_c(i)$ is the cardinality distribution, and $f_i (x_1, \ldots, x_i)$ is a symmetric function[1] denoting the joint probability density of $x_1, \ldots, x_i$ given cardinality i. Note that by convention $f_0 = 1$ and hence $f(\emptyset) = p_c(0)$. It can be seen from (4) that the probability density f captures the cardinality information as well as the dependence between the features. Also, U^i cancels out the unit of the probability density $f_i (x_1, \ldots, x_i)$ making f unit-less, thereby avoids the unit mismatch.

Intensity and Conditional Intensity. The *intensity function* λ of a point process is a function on $\mathcal{X}$ such that for any (compact) $A \subset \mathcal{X}$

$$\mathbb{E} [N(A)] = \int_A \lambda(x) \mathrm{d}x. \tag{5}$$

The intensity value $\lambda(x)$ is interpreted as the instantaneous expected number of points per unit hyper-volume at x.

For a *hereditary* probability density f, i.e., $f(X) > 0$ implies $f(Y) > 0$ for all $Y \subseteq X$, the *conditional intensity* at a point u is given by [5]

$$\lambda (u, X) = \frac{f (X \cup \{u\})}{f(X)}. \tag{6}$$

Loosely speaking, $\lambda(u, X)\mathrm{d}u$ can be interpreted as the conditional probability that the point process has a point in an infinitesimal neighborhood $\mathrm{d}u$ of u given all points of X outside this neighborhood. The intensity function is related to the conditional intensity by $\lambda(u) = \mathbb{E} [\lambda(u, X)]$.

The probability density of a finite point process is completely determined by its conditional intensity [60, 43]. Certain point process models are convenient to formulate in terms of the conditional intensity rather than probability density. Using the conditional intensity also eliminates the normalizing constant needed for the probability density. However, the functional form of the conditional intensity must satisfy certain consistency conditions.

IID-Cluster Model. Imposing the independence assumption among the features, the model in (4) reduces to the *IID-cluster* model [20, 60]:

$$f(X) = p_c(|X|) \, |X|! \, [Up_f]^X, \tag{7}$$

where $|X|$ denotes the cardinality (number of elements) of X, p_f is a probability density on $\mathcal{X}$, referred to as the *feature density,* and $h^X \triangleq \prod_{x \in X} h(x)$, with $h^{\emptyset} = 1$ by convention.

When the cardinality distribution p_c is Poisson with rate ρ we have the celebrated Poisson point process [20, 60].

$$f(X) = \rho^{|X|} \, e^{-\rho} \, [Up_f]^X. \tag{8}$$

[1]The notations $f_m (x_1, \ldots, x_m)$ and $f_m (\{x_1, \ldots, x_m\})$ can be used interchangeably, since f_m is symmetric.

The Poisson point process model is completely determined by the intensity function $\lambda = \rho p_f$, which also equals its conditional intensity. Note that the Poisson cardinality distribution is described by a single non-negative number ρ, hence there is only one degree of freedom in the choice of cardinality distribution for the Poisson point process model.

Finite Gibbs Model. A general model that accommodates dependence between its elements is a finite Gibbs process, which has probability density of the form [60, 43]

$$f(X) = \exp\left(V_0 + \sum_{i=1}^{|X|} \sum_{\{x_1,\ldots,x_i\}\subseteq X} V_i\left(x_1,\ldots,x_i\right)\right), \tag{9}$$

where V_i is called the i^{th} potential, given explicitly by

$$V_i\left(x_1,\ldots,x_i\right) = \sum_{Y\subseteq\{x_1,\ldots,x_i\}} (-1)^{|\{x_1,\ldots,x_i\}|-|Y|} \log f(Y).$$

Note that any hereditary probability density of a finite point process can be expressed in the Gibbs form [5]. The Poisson point process is indeed a first order Gibbs model. Gibbs models arise in statistical physics, where $\log f(X)$ may be interpreted as the potential energy of the point pattern. The term $-V_1(x)$ can be interpreted as the energy required to create a single point at a location x, and the term $-V_2(x_1, x_2)$ can be interpreted as the energy required to overcome the force between the points x_1 and x_2.

1.2 Model-based supervised learning with PPD

Classification is the supervised learning task that uses fully-observed training input-output pairs $\mathcal{D}_{\text{train}} = \{(X_n, y_n)\}_{n=1}^{N}$ to determine the output class label $y \in \{1,\ldots,K\}$ of each input observation [10, 44, 26, 42]. This fundamental machine learning task is the most widely used form of supervised machine learning, with applications spanning many fields of study.

Model-based classifiers for point pattern data have not been investigated. In multiple instance learning, existing classifiers in the Bag-Space paradigm are based on distances between point patterns, such as Hausdorff [29, 55], Chamfer [23], Earth Mover's [70, 57]. Such classifiers do not require any underlying data models and are simple to use. However, they may perform poorly with high dimensional inputs due to the curse of dimensionality, and are often computationally intractable for large datasets [44], not to mention that the decision procedure is unclear. On the other hand, knowledge of the underlying data model can be used to exploit statistical patterns in the training data, and to devise optimal decision procedures.

Using the notion of probability density for point process from subsection 1.1, the standard model-based classification formulation directly extends to point pattern classification:

- In the *training phase*, we seek likelihoods that 'best' fit the training data. Specifically, for each $k \in \{1,\ldots,K\}$, we seek a likelihood function $f(\cdot \mid y = k)$ that best fit the training input point patterns in $\mathcal{D}_{\text{train}}^{(k)} = \{X : (X,k) \in \mathcal{D}_{\text{train}}\}$, according to criteria such as *maximum likelihood* (ML) or Bayes optimal if suitable priors on the likelihoods are available.

- In the *classifying phase*, the likelihoods (learned from training data) are used to classify input observations. When a point pattern X is passed to query its label, the Bayes classifier returns the mode of the class label posterior $p\left(y = k \mid X\right)$ computed from the likelihood and the class prior p via Bayes' rule:

$$p\left(y = k \mid X\right) \propto p(y = k) f(X \mid y = k). \tag{10}$$

The simplest choices for the class priors are the uniform distribution, and the categorical distribution, usually estimated from the training data via

$$p(y = k) = \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} \delta_{y_n}[k],$$

where $\delta_i[j]$ is the Kronecker delta, which takes on the value 1 when $i = j$, and zero otherwise. Hence, the main computational effort in model-based classification lies in the training phase.

1.2.1 Learning Point Process Models

Learning the likelihood function for class k boils down to finding the value(s) of the parameter θ_k such that the (parameterized) probability density $f(\cdot \mid y = k, \theta_k)$ best explains the observations $X_1, \ldots, X_N$ in $\mathcal{D}_{\text{train}}^{(k)}$. In this subsection, we consider a fixed class label and its corresponding observations $X_1, \ldots, X_N$, and omit the dependence on k.

Methods for learning point process models have been available since the 1970's, see e.g., [43, 5]. We briefly summarize some recognized techniques and presents ML for IID-cluster models as a tractable point pattern classification solution.

Model fitting via summary statistics. The method of moments seeks the parameter θ such that the expectation of a given statistic of the model point process parameterized by θ is equal to the statistic of the observed point patterns [5]. However, this approach is only tractable when the solution is unique and the expectation is a closed form function of θ, which is usually not the case in practice, not to mention that moments are difficult to calculate.

The method of minimum contrast seeks the parameter θ that minimizes some dissimilarity between the expectation of a given summary statistic (e.g., the K-function) of the model point process and that of the observed point patterns [5]. Provided that the dissimilarity functional is convex in the parameter θ, this approach can avoid some of the problems in the method of moments. However, in general the statistical properties of the solution are not well understood, not to mention the numerical behaviour of the algorithm used to determine the minimum.

Maximum likelihood (ML). In the ML approach, we seek the ML estimate (MLE) of θ:

$$\text{MLE}\left(f(\cdot \mid \theta); X_{1:N}\right) \triangleq \underset{\theta}{\text{argmax}} \left(\prod_{n=1}^{N} f(X_n \mid \theta) \right). \tag{11}$$

The MLE has some desirable statistical properties such as asymptotic normality and optimality [5]. However, in general, there are problems with non-unique maxima. Moreover, analytic MLEs are not available because the likelihood (9) of many Gibbs models contains an intractable normalizing constant (which is a function of θ) [43].

To the best of our knowledge, currently there is no general ML technique for learning generic models such as Gibbs from real data. Numerical approximation methods in [50] and Markov Chain Monte Carlo (MCMC) methods in [24] are highly specific to the chosen model, computationally intensive, and require careful tuning to ensure good performance. Nonetheless, simple models such as the IID-cluster model (7) admits an analytic MLE (see subsection 1.2.1).

Remark: The method of estimating equation replaces the ML estimation equation

$$\nabla \left(\sum_{n=1}^{N} \log\left(f\left(X_n \mid \theta\right)\right) \right) = 0 \tag{12}$$

by an unbiased sample approximation $\sum_{n=1}^{N} \Psi(\theta, X_n) = 0$ of the general equation $\mathbb{E}_\theta\left[\Psi(\theta, X)\right] = 0$, For example, $\Psi(\theta, X_n) = \nabla \log(f(X_n \mid \theta))$ results in ML since it is well-known that (12) is an unbiased estimating equation. Setting $\Psi(\theta, X_n)$ to the difference between the empirical value and the expectation of the summary statistic results in the method of moments. Takacs-Fiksel is another well-known family of estimating equations [61, 21].

Maximum pseudo-likelihood. Maximum pseudo-likelihood (MPL) estimation is a powerful approach that avoids the intractable normalizing constant present in the likelihood while retaining desirable properties such as consistency and asymptotic normality in a large-sample limit [8, 9]. The key idea is to replace the likelihood of a point process (with parameterized conditional intensity $\lambda_\theta(u; X)$) by the pseudo-likelihood:

$$\text{PL}\left(\theta; X_{1:N}\right) = \prod_{n=1}^{N} e^{-\int \lambda_\theta(u; X_n) \mathrm{d}u} \left[\lambda_\theta\left(\cdot; X_n\right)\right]^{X_n}. \tag{13}$$

The rationale behind this strategy is discussed in [8]. Up to a constant factor, the pseudo-likelihood is indeed the likelihood if the model is Poisson, and approximately equal to the likelihood if the model is close to Poisson. The pseudo-likelihood may be regarded as an approximation to the likelihood which neglects the inter-point dependence.

An MPL algorithm has been developed by Baddeley and Turner in [6] for point processes with sufficient generality such as Gibbs whose conditional intensity has the form

$$\lambda(u, X) = \exp\left(\sum_{i=1}^{|X|+1} \sum_{\{x_1,\ldots,x_{i-1}\}\subseteq X} V_i(u, x_1, \ldots, x_{i-1})\right).$$

By turning the pseudo-likelihood of a general point process into a classical Poisson point process likelihood, MPL can be implemented with standard generalized linear regression software [6]. Due to its versatility, the Baddeley-Turner algorithm is the preferred model fitting tool for point processes.

The main hurdle in the application of the Baddeley-Turner algorithm to point pattern classification is the computational requirement. While this may not be an issue in spatial statistics applications, the computational cost is still prohibitive with large data sets often encountered in machine learning. On the other hand, disadvantages of MPL (relative to ML) such as small-sample bias and inefficiency [9, 31] become less significant with large data. Efficient algorithms for learning general point process models is an on going area of research.

ML learning for IID-clusters. Computationally efficient algorithms for learning point process models are important because machine learning usually involve large data sets (compared to applications in spatial statistics). Since learning a general point process is computationally prohibitive, the IID-cluster model (7) provides a good trade-off between tractability and versatility by neglecting interactions between the points.

Since an IID-cluster model is uniquely determined by its cardinality and feature distributions, we consider a parameterization of the form:

$$f(X \mid \xi, \varphi) = p_\xi(|X|)\, |X|!\, U^{|X|}\, p_\varphi^X, \tag{14}$$

where p_ξ and p_φ, are the cardinality and feature distributions parameterized by ξ and φ, respectively. Learning the underlying parameters of an IID-cluster model amounts to estimating the parameter $\theta = (\xi, \varphi)$ from training data.

The form of the IID-cluster likelihood function allows the MLE to separate into the MLE of the cardinality parameter ξ and MLE of the feature parameter φ. This is stated more concisely in Proposition 1 (the proof is straightforward, but included for completeness).

Proposition 1. *Let $X_1, \ldots, X_N$ be N i.i.d. realizations of an IID-cluster with parameterized cardinality distribution p_ξ and feature density p_φ. Then the MLE of (ξ, φ), is given by*

$$\hat{\xi} = \mathrm{MLE}\left(p_\xi; |X_1|, \ldots, |X_N|\right), \tag{15}$$

$$\hat{\varphi} = \mathrm{MLE}\left(p_\varphi; \uplus_{n=1}^N X_n\right), \tag{16}$$

where $\uplus$ denotes disjoint union.

Proof. Using (14), we have

$$\begin{aligned}\prod_{n=1}^N f(X_n \mid \xi, \varphi) &= \prod_{n=1}^N p_\xi(|X_n|)\ |X_n|!\, U^{|X_n|} p_\varphi^{X_n} \\ &= \prod_{n=1}^N |X_n|!\, U^{|X_n|} \prod_{n=1}^N p_\xi(|X_n|) \prod_{n=1}^N p_\varphi^{X_n}\end{aligned}$$

Hence, to maximize the likelihood we simply maximize the second and last products in the above separately. This is achieved with (15) and (16). □

Observe from Proposition 1 that the MLE of the feature density parameter is identical to that used in NB. For example: if the feature density is a Gaussian, then the MLEs of the mean and

covariance are

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N}\sum_{x\in X_n} x, \tag{17}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}\sum_{x\in X_n} (x-\hat{\mu})(x-\hat{\mu})^{\mathrm{T}}; \tag{18}$$

if the feature density is a Gaussian mixture, then the MLE of the Gaussian mixture parameters can be determined by the EM algorithm. Consequently, learning the IID-cluster model requires only one additional, but relatively inexpensive, task of computing the MLE of the cardinality parameters.

For a categorical cardinality distribution, i.e., $\xi = (\xi_1, \ldots, \xi_M)$ where $\xi_k = \Pr(|X| = k)$ and $\sum_{k=1}^{M}\xi_k = 1$, the MLE of the cardinality parameter is given by

$$\hat{\xi}_k = \frac{1}{N}\sum_{n=1}^{N}\delta_k[|X_n|]. \tag{19}$$

Note that to avoid over-fitting, the standard practice of placing a Laplace prior on the cardinality distribution can be applied, i.e. replacing the above equation by $\hat{\xi}_k \propto \epsilon + \sum_{n=1}^{N}\delta_k[|X_n|]$, where ϵ is a small number.

For a Poisson cardinality distribution parameterized by the rate $\xi = \rho$, the MLE is given by

$$\hat{\rho} = \frac{1}{N}\sum_{n=1}^{N}|X_n|. \tag{20}$$

It is also possible to derive MLEs for other families of cardinality distributions such as Panjer, multi-Bernoulli, etc.

Numerical results for point pattern classification, in which ML is used to learn a Poisson model, are given in subsection 1.4.1. The complexity of IID-cluster MLE is the same as NB, which is $O(NId)$ for training, and $O(KId)$ for classifying, where I is the average number of features per point pattern and d is the dimension of the features.

1.3 Novelty Detection with PPD

Novelty detection is the semi-supervised task of identifying observations that are significantly different from the rest of the data [39, 22]. In novelty detection, there is no novel training data, only 'normal' training data is available [22]. Hence it is not a special case of classification nor clustering [13, 28], and is a separate problem in its own right.

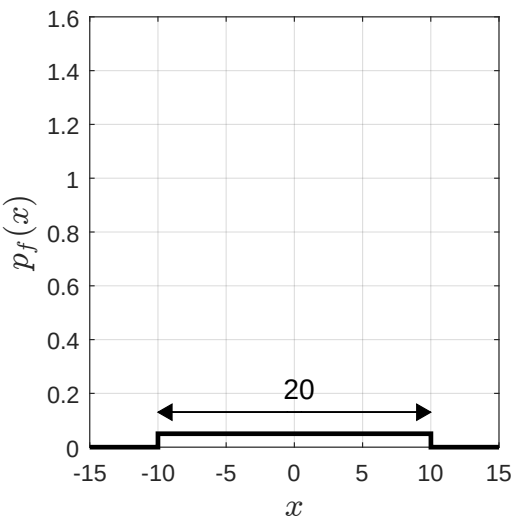
Similar to classification, novelty detection involves a training phase and a detection phase. Since novel training data is not available, input observations are ranked according to how well they fit the 'normal' training data and those not well-fitted are deemed novel or anomalous [13, 28]. The preferred measure of goodness of fit is the 'normal' likelihoods of the input data. To the best of our knowledge, there are no novelty detection solutions for point pattern data in the literature.

In this section we present a model-based solution to point pattern novelty detection. The training phase in novelty detection is the same as that for classification. However, in the detection phase the ranking of likelihoods is not applicable to point pattern data, even though point process probability density functions are unit-less and incorporates both feature and cardinality information. In subsection 1.3.1, we discuss why such probability densities are not suitable for ranking input point patterns, while in subsection 1.3.2 we propose a suitable ranking function for novelty detection.

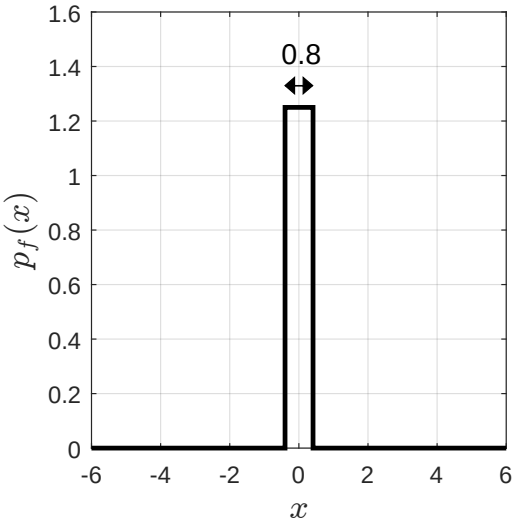
1.3.1 Probability Density and Likelihood

This subsection presents an example to illustrate that the probability density of a point pattern does not necessarily indicate how likely it is. For this example, we reserve the term *likelihood* for the measure of how likely or probable a realization is.

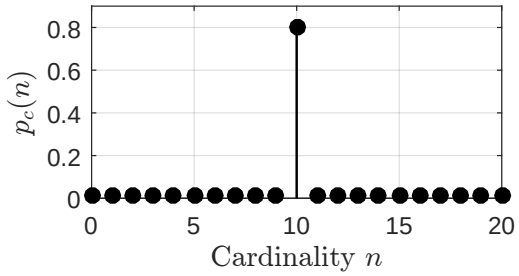
Consider two IID-cluster models with different uniform feature densities and a common cardinality distribution as shown in Fig. 3. Due to the uniformity of p_f, it follows from (7) that point patterns from each IID-cluster model with the same cardinality have the same probability density. Note from [20] that to sample from an IID-cluster model, we first sample the number of points from



(a) 'Short' uniform density



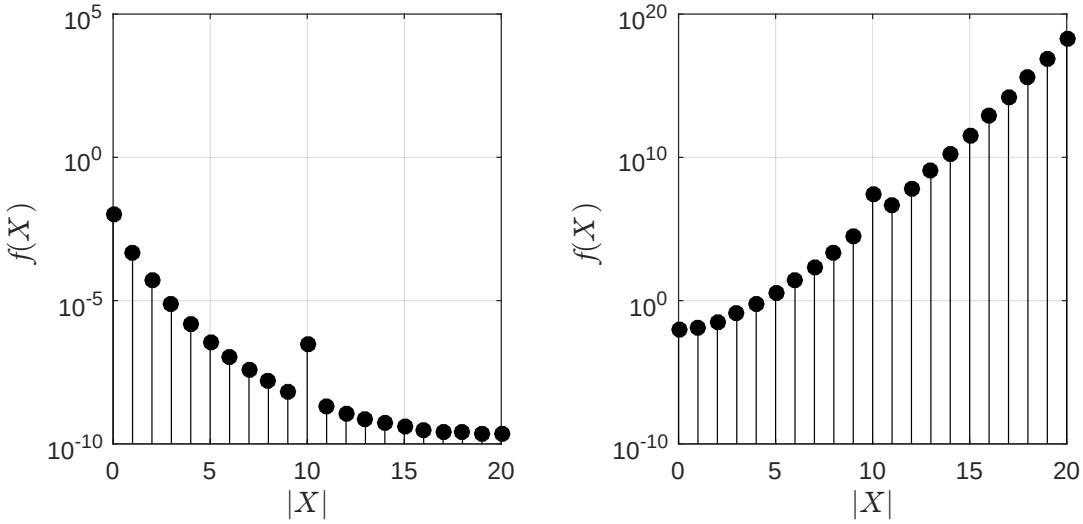
(b) 'Tall' uniform density



(c) Cardinality distribution on {0,...,20} with one mode at 10, the remaining cardinalities are equally likely with total mass 0.2.

Figure 3: Feature and cardinality distributions for 2 IID-clusters.

the cardinality distribution, and then sample the corresponding number of points independently from the feature distribution. For an IID-cluster model with uniform feature density, the joint distribution of the features is completely uninformative (total uncertainty) and so the likelihood of a point pattern should be proportional to the probability of its cardinality.



(a) 'Short' uniform feature density (b) 'Tall' uniform feature density

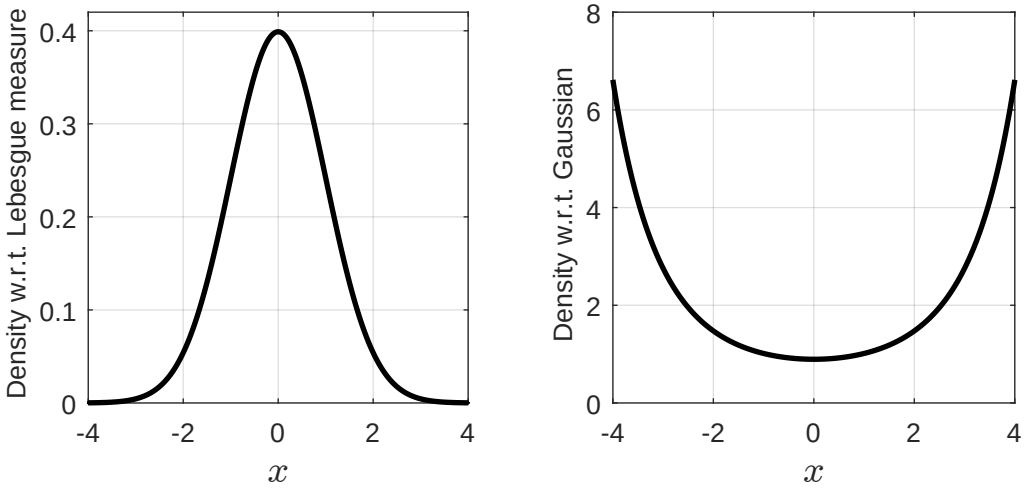
Figure 4: Probability density vs cardinality for 2 IID-clusters.

If the probability density were an indication of how likely a point pattern is, then the plot of probability density against cardinality should resemble the cardinality distribution. However, this is not the case. Fig. 4 indicates that for the IID-cluster with 'short' feature density, the probability density tends to decrease with increasing cardinality (Fig. 4a). This phenomenon arises because the feature density given cardinality n is $(1/20)^n$, which vanishes faster than the $n!$ growth (for $n \leq 20$). The converse is true for the IID-cluster with 'tall' feature density (Fig. 4b). Thus, point patterns with highest/least probability density are not necessarily the most/least probable.

Such problem arises from the non-uniformity of the reference measure. A measure μ is said to be uniform if for any measurable region A with $\mu(A) < \infty$, all points of A (except on set of measure zero) are equi-probable under the probability distribution $\mu/\mu(A)$. One example is the Lebesgue measure vol on $\mathbb{R}^n$: given any bounded measurable region A, all realizations in A are equally likely under the probability distribution $vol(\cdot)/vol(A)$. The probability density $f(X) = P(\mathrm{d}X)/\mu(\mathrm{d}X)$ (as a Radon-Nikodym derivative) at a point X is the ratio of probability measure to reference measure at an infinitesimal neighbourhood of X. Hence, unless the reference measure is uniform, $f(X)$ is not a measure of how likely X is. This is also true even for probability densities on the real line. For example, the probability density of a zero-mean Gaussian distribution with unit variance relative to the (uniform) Lebesgue measure is the usual Gaussian curve shown in Fig. 5a, while its density relative to a zero-mean Gaussian distribution with variance 0.8 is shown in Fig. 5b, where the most probable point has the least probability density value.

The reference measure μ defined by (2) is not uniform because for a bounded region $\mathcal{T} \subseteq \mathcal{F}(\mathcal{X})$, the probability distribution $\mu/\mu(\mathcal{T})$ is not necessarily uniform (unless all points of $\mathcal{T}$ have the same cardinality). Hence, probability densities of input point patterns relative to μ are not indicative of how well they fit the 'normal' data model.

Remark: In novelty detection we are interested in the likelihood of the input point pattern whereas in Bayesian classification we are interested in its likelihood ratio. Using standard properties of Radon-Nikodym derivative and relevant assumptions on absolute continuity, the posterior



(a) (b)

Figure 5: Density of a zero-mean unit-variance Gaussian w.r.t.: (a) Lebesgue measure; (b) zero-mean Gaussian with variance 0.8.

class probability

$$p\left(y \mid X\right) = \frac{p(y)f(X \mid y)}{\int p(y)f(X \mid y)\mathrm{d}y} = \frac{p(y)P(\mathrm{d}X \mid y)/\mu(\mathrm{d}X)}{\int p(y)(P(\mathrm{d}X \mid y)/\mu(\mathrm{d}X))\mathrm{d}y}$$
$$= \frac{p(y)P(\mathrm{d}X \mid y)}{\int p(y)P(\mathrm{d}X \mid y)\mathrm{d}y},$$

which is invariant to the choice of reference measure. In essence, the normalizing constant cancels out the influence of the reference measure, and hence, problems with the non-uniformity of the reference measure do not arise.

1.3.2 Ranking Functions

To the best of our knowledge, it is not known whether there exists a uniform reference measure on $\mathcal{F}(\mathcal{X})$ that dominates the probability distributions of interest (so that they admit densities). In this subsection, we propose a suitable point pattern ranking function for novelty detection by modifying the probability density.

The probability density (4) is the product of the cardinality distribution $p_c(|X|)$, the cardinality-conditioned feature (probability) density $f_{|X|}(X)$, and a trans-dimensional weight $|X|!\, U^{|X|}$. Note that the cardinality distribution and the conditional joint feature density completely describes the point process. The conditional density $f_{|X|}(X)$ enables the ranking of point patterns of the same cardinality, but cannot be used to rank across different cardinalities because it takes on different units of measurement. The weights $|X|!\, U^{|X|}$ reconcile for the differences in dimensionality and unit of measurement between $f_{|X|}(X)$ of different cardinalities. However, the example in subsection 1.3.1 demonstrates that weighting by $|X|!\, U^{|X|}$ leads to probability densities that are inconsistent with likelihoods.

In the generalization of the maximum aposteriori (MAP) estimator to point patterns [37], Mahler circumvented such inconsistency by replacing $|X|!\, U^{|X|}$ with $c^{|X|}$, where c is an arbitrary constant. Specifically, instead of maximizing the probability density $f(X)$, Mahler proposed to maximize $f(X)c^{|X|}/\,|X|!$. This generalized MAP estimate depends on the choice of the free parameter c.

Inspired by Mahler's generalized MAP estimator, we replace the weight $|X|!\, U^{|X|}$ in the probability density by a general function of the cardinality $C(|X|)$, resulting in a ranking function of the form

$$r(X) = p_c(|X|)\, C(|X|)\, f_{|X|}(X). \tag{21}$$

The example in subsection 1.3.1 demonstrated that, as a function of cardinality, the ranking

should be proportional to the cardinality distribution, otherwise unlikely samples can assume high ranking values. In general, the ranking function is not solely dependent on the cardinality, but also varies with the features. Nonetheless, the example suggests that the ranking function, on average, should be proportional to the cardinality distribution. Hence, we impose the following consistency requirement: for a given cardinality n, the expected ranking value is proportional to the probability of cardinality n, i.e.,

$$\mathbb{E}_{X||X|=n}\left[r(X)\right] \propto p_c(n). \tag{22}$$

Proposition 2. *For a point process with probability density (4), a ranking function consistent with the cardinality distribution, i.e., satisfies (22), is given by*

$$r(X) \propto \frac{p_c(|X|)}{\|f_{|X|}\|_2^2} f_{|X|}(X) \tag{23}$$

where $\|\cdot\|_2$ *denotes the* L_2*-norm.*

Proof. Noting $f(X \mid |X| = n) = n!\, U^n\, f_n(X)\, \delta_n[|X|]$ from (4), we have

$$\begin{aligned}
\mathbb{E}_{X||X|=n}\left[f_n(X)\right] &= \int f_n(X)\, f(X \mid |X| = n)\, \mu(\mathrm{d}X) \\
&= \int n!\, U^n \left(f_n\left(X\right)\right)^2 \delta_n[|X|]\, \mu(\mathrm{d}X) \\
&= \sum_{i=0}^{\infty} \frac{n!\, U^n}{i!\, U^i} \int \left(f_n\left(\{x_1, \ldots, x_i\}\right)\right)^2 \delta_n[i]\, \mathrm{d}\left(x_1, \ldots, x_i\right)
\end{aligned}$$

where the last step follows from definition (3) of the integral with respect to μ. Further due to $\delta_n[i]$, only the nth term in the sum remains, i.e.

$$\begin{aligned}
\mathbb{E}_{X||X|=n}\left[f_n(X)\right] &= \frac{n!\, U^n}{n!\, U^n} \int \left(f_n\left(\{x_1, \ldots, x_n\}\right)\right)^2 \mathrm{d}\left(x_1, \ldots, x_n\right) \\
&= \|f_n\|_2^2.
\end{aligned}$$

Hence taking the expectation of $r(X)$ in (23), we have

$$\begin{aligned}
\mathbb{E}_{X||X|=n}\left[r(X)\right] &\propto \mathbb{E}_{X||X|=n}\left[\frac{p_c(|X|)}{\|f_{|X|}\|_2^2} f_{|X|}(X)\right] \\
&= \frac{p_c(n)}{\|f_n\|_2^2} \mathbb{E}_{X||X|=n}\left[f_n(X)\right] \\
&= p_c(n).
\end{aligned}$$

□

Note that $\|f_{|X|}\|_2^2$ has units of $U^{-|X|}$, which is the same as the unit of $f(X)$, rendering the ranking function r unit-less, thereby avoids the unit of measurement inconsistency described in Section 1.1.

For an IID-cluster with feature density p_f the ranking function reduces to

$$r(X) \propto p_c(|X|) \left(\frac{p_f}{\||p_f\|_2^2}\right)^X. \tag{24}$$

The feature density p_f, in the example of subsection 1.3.1, is uniform and so $p_f/\|p_f\|_2^2 = 1$ on its support. Hence the ranking is equal to the cardinality distribution, as expected. Fig. 6 illustrates the effect of dividing a non-uniform feature density p_f, by its energy $\|p_f\|_2^2$: 'tall' densities become shorter and 'short' densities become taller, providing adjustments for multiplying together many large/small numbers.

Numerical results for point pattern novelty detection are given in subsection 1.4.2, where ML is used to learn a 'normal' Poisson model and input data are ranked via the proposed ranking function. The complexity is the same as NB, which is $O\left(NId\right)$ for training, and $O\left(Id\right)$ for detection, where I is the average number of features per point pattern.

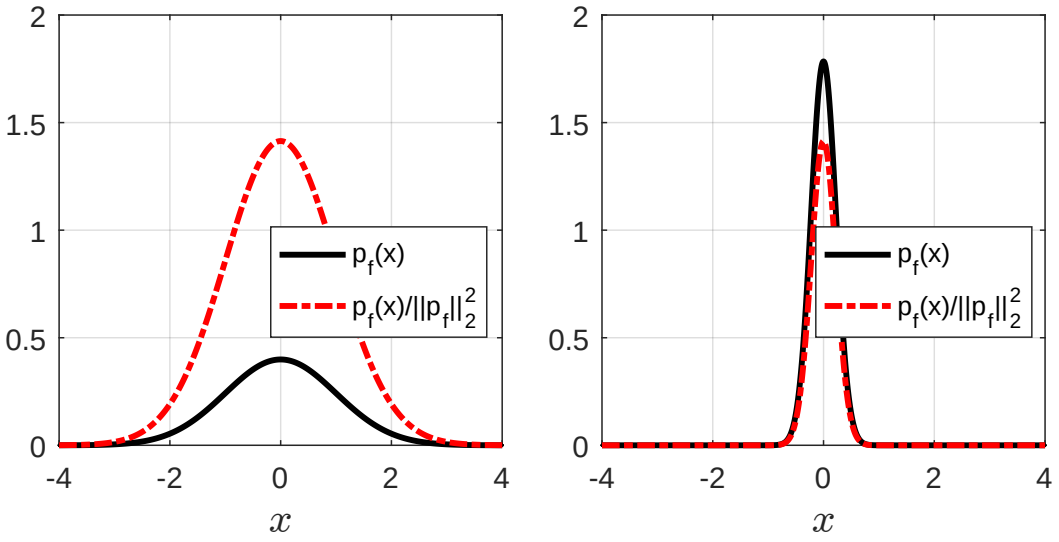


Figure 6: Probability density divided by energy: (a) 'short' Gaussian (mean = 0, variance = 1); (b) 'tall' Gaussian (mean = 0, variance = 0.05).

1.4 Results and Discussions

This section demonstrates the viability of the proposed framework using the IID-cluster model with Poisson and Categorical cardinality distributions. A Poisson point process with Gaussian intensity is specified by the triple (ρ, μ, Σ) where ρ is the rate and μ, Σ are the mean and covariance of the feature density. The NB model is used as a performance benchmark since it has been used for this type of problems (see e.g. [40, 33, 41, 17, 12]) and assumes i.i.d. features.

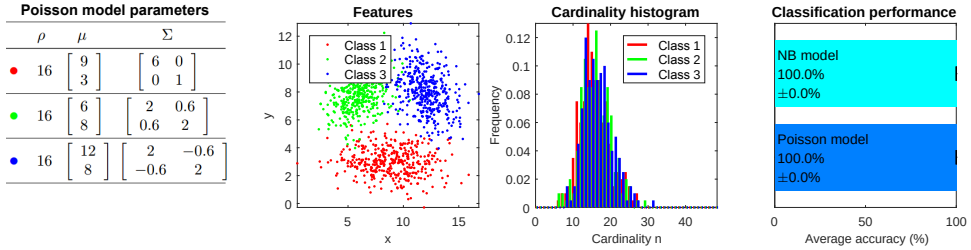
1.4.1 Classification Experiments

This subsection presents three classification experiments on simulated data, the Texture images dataset [35], and the StudentLife dataset [67]. In the training phase, ML is used to learn the parameters of the NB model and the Poisson model (as per subsection 1.2.1) from fully observed training data. Both trained models agree on the feature distribution. For simplicity we use a uniform class prior in the test phase. For the last dataset, we use the IID-cluster model instead of the Poisson model.

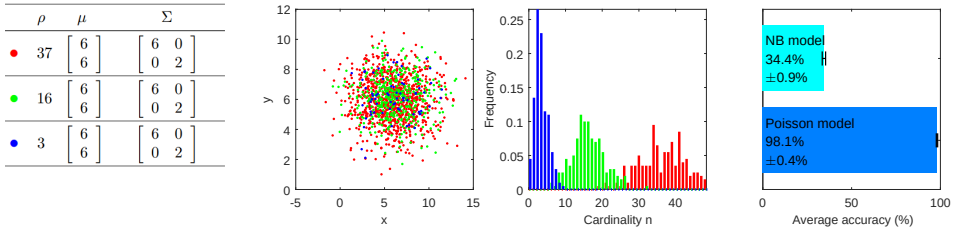
Classification on simulated data We consider three diverse scenarios, each comprising three classes simulated from Poisson point processes with Gaussian intensities shown in Fig. 7. In scenario (a), point patterns from each class are well-separated from other classes in feature, but significantly overlapping in cardinality (see Fig. 7a). In scenario (b), point patterns from each class are well-separated from other classes in cardinality, but significantly overlapping in feature (see Fig. 7b). Scenario (c) is a mix of (a) and (b), where: point patterns from Class 1 are well-separated from other classes in features, but significantly overlapping with Class 2 in cardinality; and the point patterns from Classes 2 and 3 significantly overlap in feature, but well-separated in cardinality (see Fig. 7c).

The fully observed training dataset comprises 600 point patterns (200 per class) is used to train the NB/Poisson model in which each class is modeled by a Gaussian density/intensity. In the test phase, 10 different test sets each comprises 300 point patterns (100 per class) are used. The average classification performance is reported in Fig. 7.

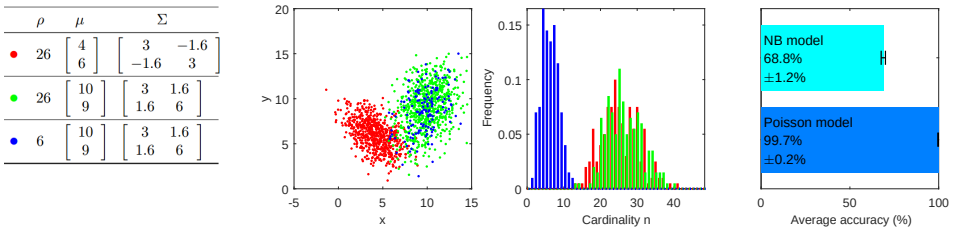
In scenario (a) both models achieve perfect classification using only the features of the test point patterns because the classes are so well-separated in the feature space. In scenario (b) on the other hand, neither models are able to differentiate the classes using the features in the test



(a) All 3 classes are well-separated from each other in feature, but overlapping in cardinality.



(b) All 3 classes are overlapping in feature, but well-separated from each other in cardinality.

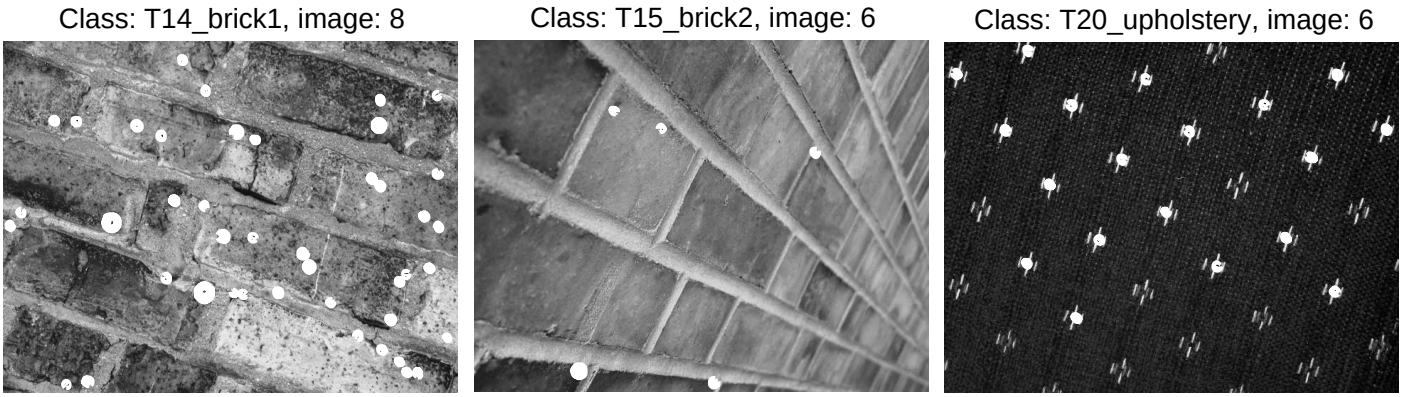


(c) Classes 2 and 3: overlap in feature, well-separated in cardinality. Classes 1 and 2: overlap in cardinality, well-separated in feature.

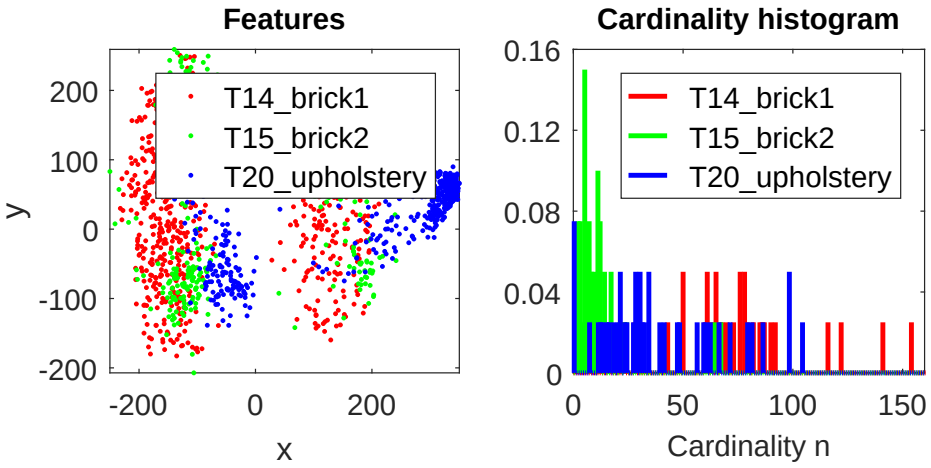
Figure 7: Model, data and classification accuracy (No. correct classifications / No. of observations in the test set [38]) for three scenarios.

data. Nonetheless, the Poisson model achieved excellent performance by exploiting the separation in cardinality of the classes from the test data. In contrast, the NB model's inability to exploit cardinality information results in very poor performance.

In scenario (c) both models can differentiate Class 1 from 2 and 3 by exploiting the separation in features. However, the Poisson model achieved near perfect performance by further exploiting the well-separated cardinality to differentiate Class 3 from 1 and 2, whereas the NB model could not do so.



(a) Example images (circles represent detected SIFT keypoints).



(b) Extracted 2-D point patterns.

Figure 8: Three classes of the Texture dataset.

Classification on the Texture dataset. Three classes "T14 brick1", "T15 brick2", and "T20 upholstery" of the Texture images dataset [35] are considered. Each class comprises 40 images, with some examples shown in Fig. 8a. Each image is processed by the SIFT algorithm (using the VLFeat library [64]) to produce a point pattern of 128-D SIFT features, which is then compressed into a 2-D point pattern by Principal Component Analysis (PCA). Fig. 8b shows the superposition of the 2-D point patterns from the three classes along with their cardinality histograms.

A 4-fold cross validation scheme is used for performance evaluation. In each fold, the fully observed training dataset comprising 30 images per class is used to learn the NB/Poisson model in which each class is parameterized by a 3-component Gaussian mixture density/intensity. The test set comprises the remaining images (10 per class).

Fig. 8b shows that the classes are neither well-separated in feature nor cardinality. Note also that there are possible dependencies between the features of the point patterns not captured by the simple Poisson model. However, the Poisson model still shows good performance (even on a small training set), and outperforms NB, as shown in Fig 9, by exploiting cardinality information.

Classification on the StudentLife dataset. To demonstrate the scalability of the proposed solutions, we choose the StudentLife dataset [67] that is widely-used in pervasive computing research. This dataset contains various data types (e.g. Wi-Fi signals, Bluetooth scan) collected from the smartphones of 49 voluntary students at Dartmouth College over a 10-week term in 2013. Pre-processing of the data is described in [46]. For the purpose of our experiments, we only use the Wi-Fi signal strength readings, which are grouped into 10-minute time frames, based on their time-stamps. If there are multiple readings of a Wi-Fi ID within a 10-minute frame, we use the mean signal strength as its (sole) observation. Only10-minute frames with at least 1 observation of any Wi-Fi ID after aggregation are retained. The resulting dataset is a collection of records,

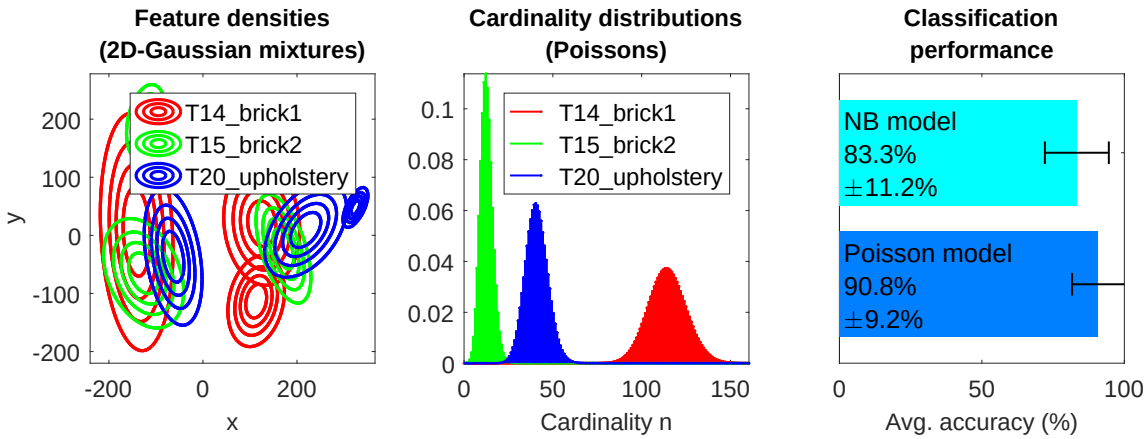


Figure 9: MLE of model parameters and classification performance on the Texture dataset. The feature densities are the same for both Poisson and NB models. The error-bars represent standard deviations.

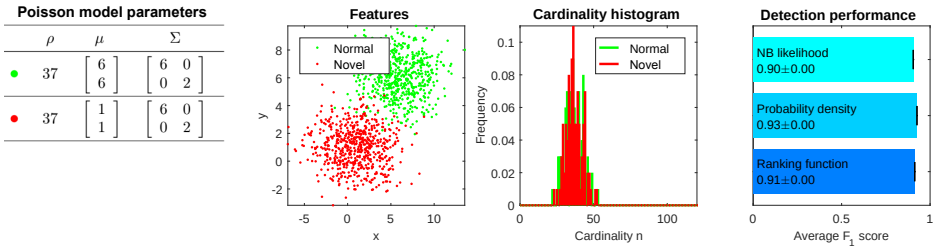
	No. PPs	**No. feat.**	**Min**	**Mode**	**Max**	**NB** (%)	**IC** (%)
1 day	2,132	10,002	1	2	39	$\mathbf{77.0 \pm 3.5}$	76.4 ± 2.4
2 days	4,449	24,141	1	2	56	73.3 ± 2.0	$\mathbf{78.4 \pm 2.4}$
3 days	6,796	36,297	1	2	49	66.3 ± 1.9	$\mathbf{77.5 \pm 1.7}$
4 days	8,833	46,514	1	2	56	67.4 ± 1.0	$\mathbf{79.4 \pm 2.0}$
5 days	10,396	53,804	1	2	56	65.7 ± 1.0	$\mathbf{79.7 \pm 1.4}$
6 days	12,718	64,334	1	2	56	66.2 ± 1.0	$\mathbf{79.3 \pm 1.0}$
7 days	15,034	76,060	1	2	56	67.9 ± 1.1	$\mathbf{81.0 \pm 0.7}$
2 weeks	30,231	152,203	1	2	56	63.3 ± 0.9	$\mathbf{80.8 \pm 0.3}$
3 weeks	46,219	221,995	1	2	57	63.0 ± 0.9	$\mathbf{79.8 \pm 0.7}$
4 weeks	61,945	295,863	1	2	57	61.8 ± 0.9	$\mathbf{78.7 \pm 0.6}$

Table 1: Statistics for the10 subdatasets constructed from StudentLife dataset, and classification accuracies for NB and IID-cluster (IC) models.

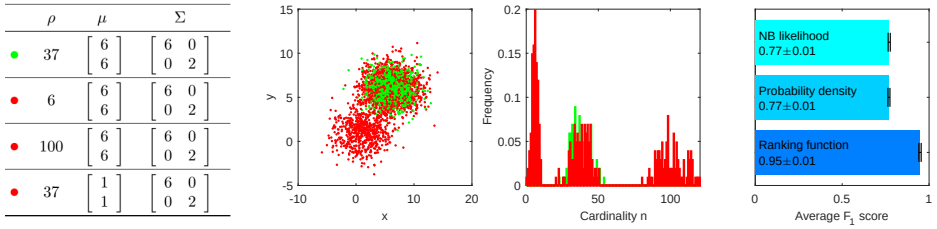
each of which is a 1271-D vector corresponding to readings of the 1271 Wi-Fi IDs in 10-minute intervals, and is compatible with the benchmark NB-based classifier and K-means clustering algorithm. Each point pattern observation is obtained by retaining only the non-zero entries of each 1271-D vector (hence, the cardinality of this point pattern is the number of non-zero entries of the vector). An element of the converted point pattern is an ordered pair of Wi-Fi ID and its signal strength. For the experiments with the StudentLife dataset, we use an IID-cluster model with Categorical cardinality distribution and feature density consisting of a Categorical distribution for the Wi-Fi ID and a 3-component 1-D Gaussian mixture for the corresponding signal strength. The described dataset and model will be used in both classification and clustering experiments.

In our classification experiment, we construct (from the full StudentLife dataset) 10 subdatasets, with respective total observation periods of 1 day, 2 days, ... , 7 days, 2 weeks, 3 weeks, and 4 weeks. Further, for each subdataset, we select only the top 20 users with the most number of non-empty observations. The total numbers of point patterns and features, the minimum, mode, and maximum of the point pattern cardinalities for each subdataset are shown in Table 1. The user IDs are used as ground-truth classification labels, hence we have 20 classes in each classification task. In each task, we employ a 10-fold cross validation scheme.

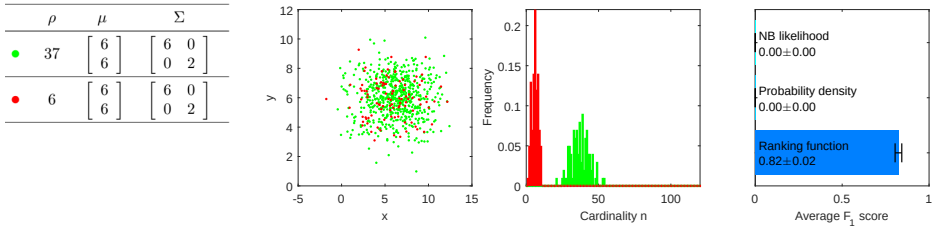
The average accuracies of the NB model and the IID-cluster model are reported respectively in the last 2 columns of Table 1. Except for the first subdataset, the proposed classifier outperforms the benchmark classifier by a large margin. Observe the overall trend that as we have more observations, the accuracy of IID-cluster model tends to increase whilst the accuracy of NB model tends to decrease.



(a) Novelty well-separated from 'normal' data in feature, but overlap in cardinality.



(b) Novelty and 'normal' data partially overlap in feature and cardinality.



(c) Low-cardinality novelty that overlap with 'normal' data in feature.

Figure 10: Model, data and novelty detection performance for three scenarios.

1.4.2 Novelty Detection Experiments

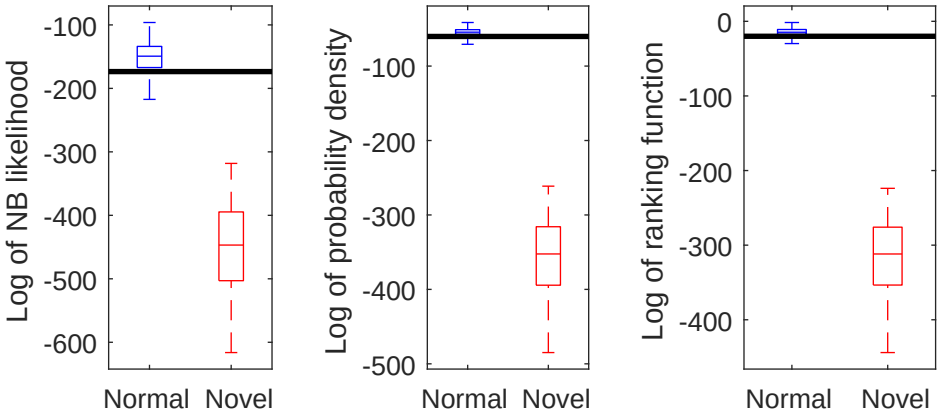
This subsection presents two novelty detection experiments on simulated and real data using the Poisson model to illustrate the effectiveness of the proposed ranking function against the NB likelihood and standard probability density (for the purpose of demonstrating scalability, classification on the StudentLife dataset is sufficient since the proposed novelty detection and classification use the same ML algorithm). Like the classification experiments, ML is used to learn the parameters of the 'normal' NB and Poisson models in the training phase. The novelty threshold is set at the 2nd 10-quantile of the ranking values of the 'normal' training data. The detection performance measure is the F_1 *score* [38]:

$$F_1(\text{precision, recall}) \triangleq 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

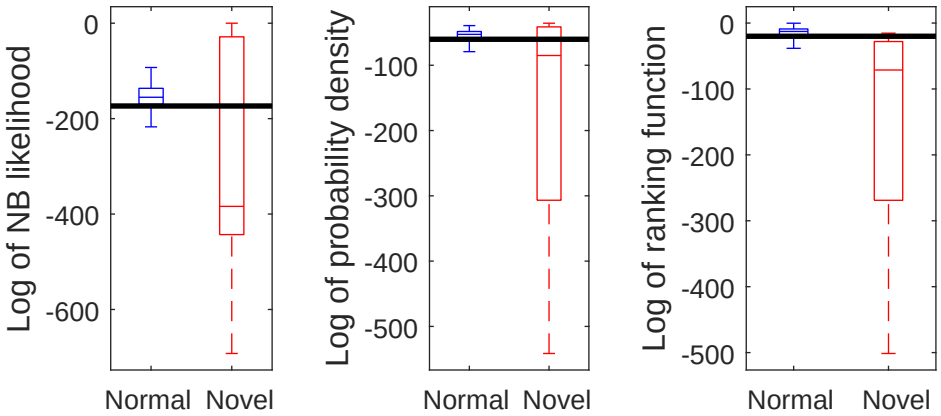
where *precision* is the proportion of correct decisions in the output of the detector, and *recall* is the proportion of correctly identified novelties in the test set. To ensure functional continuity of F_1, we define $F_1(0,0) \triangleq 0$.

Novelty detection on simulated data. We consider three simulated scenarios comprising 'normal' and novel point patterns generated from Poisson point processes with 2-D Gaussian intensities as shown in Fig. 10. All scenarios have the same 'normal' point patterns, with cardinalities between 20 and 60. In scenario (a) novelties are well-separated from 'normal' data in feature, but overlapping in cardinality (see Fig. 10a). In scenario (b) novelties are overlapping with 'normal' data in feature, but only partially overlapping in cardinality (see Fig. 10b). In scenario (c) we remove the high cardinality novelties from (b) (see Fig. 10c).

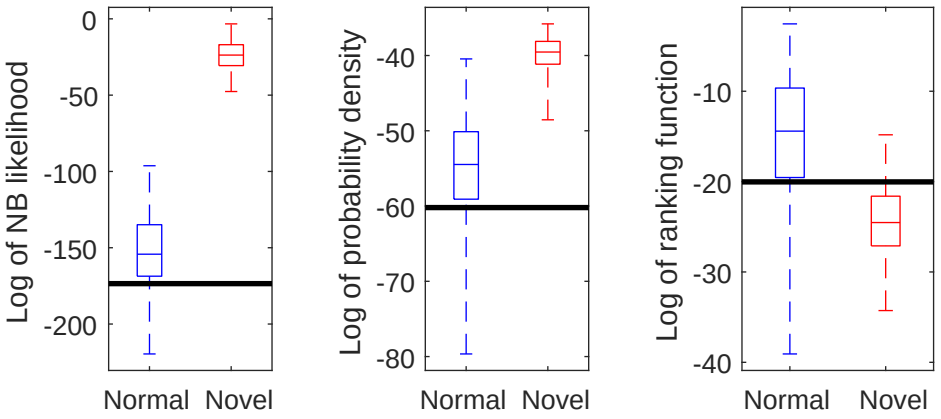
In the training phase, the same 300 'normal' point patterns for each scenario are used to learn the 'normal' NB/Poisson model that consists of a Gaussian density/intensity. In the testing phase,



(a)



(b)



(c)

Figure 11: Boxplots of: NB likelihood, probability density, and ranking function for the test data in the three simulated scenarios in Fig. 10 (solid line through each graph indicates the novelty threshold chosen from training data).

10 tests are ran per scenario with each test set comprising 100 'normal' point patterns and 100 novelties generated according to their respective models.

Observe from Fig. 10a that in scenario (a) the NB likelihood, probability density, and ranking function all perform well. Even though the NB likelihood and probability density are not consistent in ranking, the good separation in features of 'novel' from 'normal' test data is sufficient to differentiate them. The box plots in Fig. 11a shows that the range of ranking values for 'normal' data (for all three functions) are well-separated from 'novel' data, and hence the good detection performance.

Fig. 10b shows that the proposed ranking function out performs the others in scenario (b). The performance of the NB likelihood and probability density are actually inflated by erroneously ranking all high cardinality point patterns lower than they should be due to the multiplication of many small numbers, which inadvertently include some novelties. The box plots in Fig. 11b show that the ranges of NB likelihood and probability density values for 'normal' data fall within those for 'novel' data, making them difficult to differentiate. On the other hand the range of ranking function values for 'normal' data sits above that for 'novel' data, which allows them to be differentiated.

In scenario (c), where the high cardinality novelties are removed from the training and test sets, Fig. 10c shows that only the ranking function performed well while the others completely failed. The reason for such failure (apart from failing to detect low cardinality novelties) is that there are no high cardinality novelties for NB likelihood and probability density to inadvertently detect this time. The boxplots in Fig. 11c shows that the NB likelihood and the probability density even rank novelties much higher than 'normal' data. Only the proposed ranking function is consistent in all three scenarios.

Novelty detection on the Texture dataset. For this experiment, data from class "T14 brick1" of the Texture dataset from subsection 1.4.1, are considered 'normal' while novel data are taken from class "T20 upholstery".

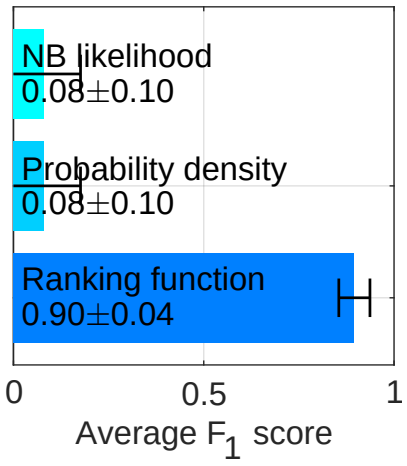


Figure 12: Averaged novelty detection performance on the Texture dataset for: NB likelihood, probability density, and proposed ranking function. The error-bars are standard deviations of the F1-scores.

A 4-fold cross validation scheme is used for performance evaluation. In each fold, training data comprising 30 'normal' images is used to learn the 'normal' NB/Poisson model that consists of a 3-component Gaussian mixture density. The test set comprises the remaining 10 'normal' and 10 novel images. The learned models are similar to those of class "T14 brick1" in Fig. 9.

Fig. 12 showed that ranking the data using the NB likelihood or the probability density failed to detect most novelties, whereas the proposed ranking function achieved a high F_1 score. The poor performance can be attributed to the fact (established in subsection 1.3.1) that the NB likelihood and probability density do not indicate how probable or likely a point pattern is. This

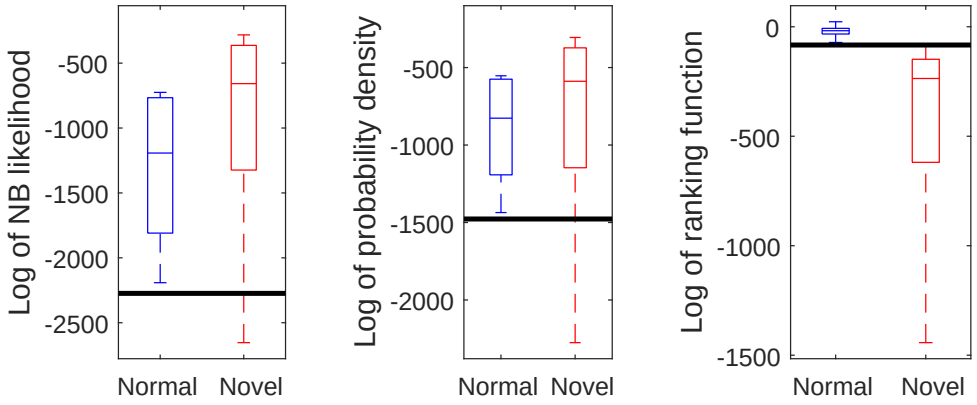


Figure 13: Boxplots of: NB likelihood; probability density, and ranking function; for 'normal' and novel test data in one fold of the Texture dataset.

inconsistency is illustrated by the box plots in Fig. 13. Note that even with hindsight it is not possible to separate 'novel' from 'normal' data using their NB likelihood and probability values. On the other hand, the box plots verified that the proposed ranking function provides a consistent ranking.

Conclusion and discussion. We have demonstrated the use of point process models for various learning tasks. Our main aim is to introduce a tool-set that facilitates research in machine learning with point pattern data. While the utility of the framework was only demonstrated on representative learning tasks such as classification, novelty detection, the framework is flexible enough to accommodate other learning tasks. For tractability, the proposed algorithms are based on very simple models. Improved performance can be achieved with more sophisticated models, albeit at higher computational costs. More complex datasets, where the i.i.d. assumption is no longer adequate, require sophisticated point process models such as Gibbs to capture interactions between the elements of the point patterns. Developing efficient techniques for learning such models is an active research area in statistics.

2 Wasserstein distance-based Multilevel Clustering

The research outcomes reported in this section have been published in [4][6] (cf. Section 4).

2.1 Why optimal transport for multilevel clustering?

In numerous applications in engineering and sciences, data are often organized in a multilevel structure. For instance, a typical structural view of text data in machine learning is to have words grouped into documents, documents are grouped into corpora. A prominent strand of modeling and algorithmic works in the past couple decades has been to discover latent multilevel structures from these hierarchically structured data. For specific clustering tasks, one may be interested in simultaneously partitioning the data in each group (to obtain local clusters) and partitioning a collection of data groups (to obtain global clusters). Another concrete example is the problem of clustering images (i.e., global clusters) where each image contains partions of multiple annotated regions (i.e., local clusters) [51]. While hierachical clustering techniques may be employed to find a tree-structed clustering given a collection of data points, they are not applicable to discovering the nested structure of multilevel data. Bayesian hierarchical models provide a powerful approach, exemplified by influential works such as [11, 53, 62]. More specific to the simultaneous and multilevel clustering problem, we mention the paper of [56]. In this interesting work, a Bayesian nonparametric model, namely the nested Dirichlet process (NDP) model, was introduced that enables the inference of clustering of a collection of probability distributions from

which different groups of data are drawn. With suitable extensions, this modeling framework has been further developed for simultaneous multilevel clustering, see for instance, [68, 47, 30].

The optimal transport distances, also known as Wasserstein distances [65], have been shown to be the natural distance metric for the convergence theory of latent mixing measures arising in both mixture models [48] and hierarchical models [49]. They are also intimately connected to the problem of clustering — this relationship goes back at least to the work of [52], where it is pointed out that the well-known K-means clustering algorithm can be directly linked to the quantization problem — the problem of determining an optimal finite discrete probability measure that minimizes its second-order Wasserstein distance from the empirical distribution of given data [25].

If one is to perform simultaneous K-means clustering for hierarchically grouped data, both at the global level (among groups), and local level (within each group), then this can be achieved by a joint optimization problem defined with suitable notions of Wasserstein distances inserted into the objective function. In particular, multilevel clustering requires the optimization in the space of probability measures defined in *different* levels of abstraction, including the space of measures of measures on the space of grouped data. Our goal, therefore, is to formulate this optimization precisely, to develop algorithms for solving the optimization problem efficiently, and to make sense of the obtained solutions in terms of statistical consistency.

We now summarize the preliminary background on Wasserstein distance, Wasserstein barycenter, and the connection between K-means clustering and the quantization problem.

For any given subset $\Theta \subset \mathbb{R}^d$, let $\mathcal{P}(\Theta)$ denote the space of Borel probability measures on Θ. The Wasserstein space of order $r \in [1, \infty)$ of probability measures on Θ is defined as $\mathcal{P}_r(\Theta) = \left\{ G \in \mathcal{P}(\Theta) : \int \|x\|^r dG(x) < \infty \right\}$, where $\|.\|$ denotes Euclidean metric in $\mathbb{R}^d$. Additionally, for any $k \geq 1$ the probability simplex is denoted by $\Delta_k = \left\{ u \in \mathbb{R}^k : u_i \geq 0, \sum_{i=1}^{k} u_i = 1 \right\}$. Finally, let $\mathcal{O}_k(\Theta)$ (resp., $\mathcal{E}_k(\Theta)$) be the set of probability measures with at most (resp., exactly) k support points in Θ.

Wasserstein distances For any elements G and G' in $\mathcal{P}_r(\Theta)$ where $r \geq 1$, the Wasserstein distance of order r between G and G' is defined as (cf. [65]):

$$W_r(G, G') = \left(\inf_{\pi \in \Pi(G, G')} \int_{\Theta^2} \|x - y\|^r d\pi(x, y) \right)^{1/r}$$

where $\Pi(G, G')$ is the set of all probability measures on $\Theta \times \Theta$ that have marginals G and G'. In words, $W_r^r(G, G')$ is the optimal cost of moving mass from G to G', where the cost of moving unit mass is proportional to r-power of Euclidean distance in Θ. When G and G' are two discrete measures with finite number of atoms, fast computation of $W_r(G, G')$ can be achieved (see, e.g., [18]). By a recursion of concepts, we can speak of measures of measures, and define a suitable distance metric on this abstract space: the space of Borel measures on $\mathcal{P}_r(\Theta)$, to be denoted by $\mathcal{P}_r(\mathcal{P}_r(\Theta))$. This is also a Polish space (that is, complete and separable metric space) as $\mathcal{P}_r(\Theta)$ is a Polish space. It will be endowed with a Wasserstein metric of order r that is induced by a metric W_r on $\mathcal{P}_r(\Theta)$ as follows (cf. Section 3 of [49]): for any $\mathcal{D}, \mathcal{D}' \in \mathcal{P}_r(\mathcal{P}_r(\Theta))$

$$W_r(\mathcal{D}, \mathcal{D}') \triangleq \left(\inf \int_{\mathcal{P}_r(\Theta)^2} W_r^r(G, G') d\pi(G, G') \right)^{1/r}$$

where the infimum in the above ranges over all $\pi \in \Pi(\mathcal{D}, \mathcal{D}')$ such that $\Pi(\mathcal{D}, \mathcal{D}')$ is the set of all probability measures on $\mathcal{P}_r(\Theta) \times \mathcal{P}_r(\Theta)$ that has marginals $\mathcal{D}$ and $\mathcal{D}'$. In words, $W_r(\mathcal{D}, \mathcal{D}')$ corresponds to the optimal cost of moving mass from $\mathcal{D}$ to $\mathcal{D}'$, where the cost of moving unit mass in its space of support $\mathcal{P}_r(\Theta)$ is proportional to the r-power of the W_r distance in $\mathcal{P}_r(\Theta)$. Note a slight notational abuse - W_r is used for both $\mathcal{P}_r(\Theta)$ and $\mathcal{P}_r(\mathcal{P}_r(\Theta))$, but it should be clear which one is being used from context.

Wasserstein barycenter Next, we present a brief overview of Wasserstein barycenter problem, first studied by [1] and subsequentially many others (e.g., [7, 58, 2]). Given probability measures $P_1, P_2, \ldots, P_N \in \mathcal{P}_2(\Theta)$ for $N \geq 1$, their Wasserstein barycenter $\overline{P}_{N,\lambda}$ is such that

$$\overline{P}_{N,\lambda} = \arg \min_{P \in \mathcal{P}_2(\Theta)} \sum_{i=1}^{N} \lambda_i W_2^2(P, P_i) \tag{25}$$

where $\lambda \in \Delta_N$ denote weights associated with $P_1, \ldots, P_N$. When $P_1, \ldots, P_N$ are discrete measures with finite number of atoms and the weights λ are uniform, it was shown by [4] that the problem of finding Wasserstein barycenter $\overline{P}_{N,\lambda}$ over the space $\mathcal{P}_2(\Theta)$ in (25) is reduced to search only over a much simpler space $\mathcal{O}_l(\Theta)$ where $l = \sum_{i=1}^{N} s_i - N + 1$ and s_i is the number of components of P_i for all $1 \leq i \leq N$. Efficient algorithms for finding local solutions of the Wasserstein barycenter problem over $\mathcal{O}_k(\Theta)$ for some $k \geq 1$ have been studied recently in [19]. These algorithms will prove to be a useful building block for our method as we shall describe in the sequel. The notion of Wasserstein barycenter has been utilized for approximate Bayesian inference [59].

K-means as quantization problem The well-known K-means clustering algorithm can be viewed as solving an optimization problem that arises in the problem of quantization, a simple but very useful connection [52, 25]. The connection is the following. Given n unlabelled samples $Y_1, \ldots, Y_n \in \Theta$. Assume that these data are associated with at most k clusters where $k \geq 1$ is some given number. The K-means problem finds the set S containing at most k elements $\theta_1, \ldots, \theta_k \in \Theta$ that minimizes the following objective

$$\inf_{S:|S|\leq k} \frac{1}{n} \sum_{i=1}^{n} d^2(Y_i, S). \tag{26}$$

Let $P_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{Y_i}$ be the empirical measure of data $Y_1, \ldots, Y_n$. Then, problem in Eq. (26) is equivalent to finding a discrete probability measure G which has finite number of support points and solves:

$$\inf_{G \in \mathcal{O}_k(\Theta)} W_2^2(G, P_n). \tag{27}$$

Due to the inclusion of Wasserstein metric in its formulation, we call this a *Wasserstein means problem*. This problem can be further thought of as a Wasserstein barycenter problem where $N = 1$. In light of this observation, as noted by [19], the algorithm for finding the Wasserstein barycenter offers an alternative for the popular Loyd's algorithm for determing local minimum of the K-means objective.

2.2 Optimal transport for clustering with multilevel structure data

The focus of this work is on the multilevel clustering problem motivated in the aforementioned modeling works [68, 47, 30], but we shall take a purely optimization approach. We aim to formulate optimization problems that enable the discovery of multilevel clustering structures hidden in grouped data. Our technical approach is inspired by the role of optimal transport distances in hierarchical modeling and clustering problems. We now presents several optimization formulations of the multilevel clusering problem, and the algorithms for solving them.

Problem statement Given m groups of n_j exchangeable data points $X_{j,i}$ where $1 \leq j \leq m, 1 \leq i \leq n_j$, i.e., data are presented in a two-level grouping structure, our goal is to learn about the two-level clustering structure of the data. We want to obtain simultaneously local clusters for each data group, and global clusters among all groups. For any $j = 1, \ldots, m$, we denote the empirical measure for group j by $P_{n_j}^j := \frac{1}{n_j} \sum_{i=1}^{n_j} \delta_{X_{j,i}}$. Throughout this section, for simplicity of exposition we assume that the number of both local and global clusters are either known or bounded above by a given number. In particular, for local clustering we allow group j to have at most k_j clusters

for $j = 1, \ldots, m$. For global clustering, we assume to have M group (Wasserstein) means among the m given groups.

High level idea For local clustering, for each $j = 1, \ldots, m$, performing a K-means clustering for group j, as expressed by (27), can be viewed as finding a finite discrete measure $G_j \in \mathcal{O}_{k_j}(\Theta)$ that minimizes squared Wasserstein distance $W_2^2(G_j, P_{n_j}^j)$. For global clustering, we are interested in obtaining clusters out of m groups, each of which is now represented by the discrete measure G_j, for $j = 1, \ldots, m$. Adopting again the viewpoint of Eq. (27), provided that all of G_js are given, we can apply K-means quantization method to find their distributional clusters. The global clustering in the space of measures of measures on Θ can be succintly expressed by

$$\inf_{\mathcal{H} \in \mathcal{E}_M(\mathcal{P}_2(\Theta))} W_2^2\left(\mathcal{H}, \frac{1}{m}\sum_{j=1}^m \delta_{G_j}\right).$$

However, G_j are not known - they have to be optimized through local clustering in each data group.

MWM problem formulation We have arrived at an objective function for jointly optimizing over both local and global clusters

$$\inf_{\substack{G_j \in \mathcal{O}_{k_j}(\Theta), \\ \mathcal{H} \in \mathcal{E}_M(\mathcal{P}_2(\Theta))}} \sum_{j=1}^m W_2^2(G_j, P_{n_j}^j) + W_2^2(\mathcal{H}, \frac{1}{m}\sum_{j=1}^m \delta_{G_j}). \tag{28}$$

We call the above optimization the problem of *Multilevel Wasserstein Means (MWM)*. The notable feature of MWM is that its loss function consists of two types of distances associated with the hierarchical data structure: one is distance in the space of measures, e.g., $W_2^2(G_j, P_{n_j}^j)$, and the other in space of measures of measures, e.g., $W_2^2(\mathcal{H}, \frac{1}{m}\sum_{j=1}^m \delta_{G_j})$. By adopting K-means optimization to both local and global clustering, the multilevel Wasserstein means problem might look formidable at the first sight. Fortunately, it is possible to simplify this original formulation substantially, by exploiting the structure of $\mathcal{H}$.

Indeed, we can show that formulation (28) is equivalent to the following optimization problem, which looks much simpler as it involves only measures on Θ:

$$\inf_{G_j \in \mathcal{O}_{k_j}(\Theta), \boldsymbol{H}} \sum_{j=1}^m W_2^2(G_j, P_{n_j}^j) + \frac{d_{W_2}^2(G_j, \boldsymbol{H})}{m}, \tag{29}$$

where $d_{W_2}^2(G, \boldsymbol{H}) := \min_{1 \le i \le M} W_2^2(G, H_i)$ and $\boldsymbol{H} = (H_1, \ldots, H_M)$, with each $H_i \in \mathcal{P}_2(\Theta)$. The proof of this equivalence is deferred to [27, Proposition B.4].

Algorithm Description Now we are now to describe our algorithm in the general case. This is a procedure for finding a local minimum of Problem (29) and is summarized in Algorithm 1.

We prepare the following details regarding the initialization and updating steps required by the algorithm:

- The initialization of local measures $G_j^{(0)}$ (i.e., the initialization of their atoms and weights) can be obtained by performing K-means clustering on local data $X_{j,i}$ for $1 \le j \le m$. The initialization of elements $H_i^{(0)}$ of $H^{(0)}$ is based on a simple extension of the K-means algorithm.

- The updates $G_j^{(t+1)}$ can be computed efficiently by simply using algorithms from [19] to search for local solutions of these barycenter problems within the space $\mathcal{O}_{k_j}(\Theta)$ from the atoms and weights of $G_j^{(t)}$;

- Since all $G_j^{(t+1)}$ are finite discrete measures, finding the updates for $H_i^{(t+1)}$ over the whole space $\mathcal{P}_2(\Theta)$ can be reduced to searching for a local solution within space $\mathcal{O}_{l^{(t)}}$ where $l^{(t)} =$

Algorithm 1 Multilevel Wasserstein Means (MWM)

Input: Data $X_{j,i}$, Parameters k_j, M.
Output: Probability measures G_j and elements H_i of $\boldsymbol{H}$.
Initialize measures $G_j^{(0)}$, elements $H_i^{(0)}$ of $\boldsymbol{H}^{(0)}$, $t = 0$.
while $Y_j^{(t)}, b_j^{(t)}, H_i^{(t)}$ have not converged **do**
1. Update $Y_j^{(t)}$ and $b_j^{(t)}$ for $1 \leq j \leq m$:
for $j = 1$ **to** m **do**
$i_j \leftarrow \arg\min_{1 \leq u \leq M} W_2^2(G_j^{(t)}, H_u^{(t)})$; $\quad G_j^{(t+1)} \leftarrow \arg\min_{G_j \in \mathcal{O}_{k_j}(\Theta)} W_2^2(G_j, P_{n_j}^j) + W_2^2(G_j, H_{i_j}^{(t)})/m$.
end for
2. Update $H_i^{(t)}$ for $1 \leq i \leq M$:
for $j = 1$ **to** m **do**
$i_j \leftarrow \arg\min_{1 \leq u \leq M} W_2^2(G_j^{(t+1)}, H_u^{(t)})$.
end for
for $i = 1$ **to** M **do**
$C_i \leftarrow \{l : i_l = i\}$ for $1 \leq i \leq M$; $\quad H_i^{(t+1)} \leftarrow \arg\min_{H_i \in \mathcal{P}_2(\Theta)} \sum_{l \in C_i} W_2^2(H_i, G_l^{(t+1)})$.
end for
$t \leftarrow t + 1$.
end while

$\sum_{j \in C_i} |\text{supp}(G_j^{(t+1)})| - |C_i|$ from the global atoms $H_i^{(t)}$ of $\boldsymbol{H}^{(t)}$. This again can be done by utilizing algorithms from [19]. Note that, as $l^{(t)}$ becomes very large when m is large, to speed up the computation of Algorithm 1 we impose a threshold L, e.g., $L = 10$, for $l^{(t)}$ in its implementation.

Multilevel Wasserstein Means with Sharing The *multilevel Waserstein means* formulation may not encourage the sharing components locally among m groups in its solution. However, enforced sharing has been demonstrated to be a very useful technique, which leads to the "borrowing of strength" among different parts of the model, consequentially improving the inferential efficiency [49, 62]. In this section, we seek to encourage the borrowing of strength among groups by imposing additional constraints on the atoms of $G_1, \ldots, G_m$ in the original MWM formulation (28). Denote $\mathcal{A}_{M,\mathcal{S}_K} = \left\{ G_j \in \mathcal{O}_K(\Theta),\ \mathcal{H} \in \mathcal{E}_M(\mathcal{P}(\Theta)) : \text{supp}(G_j) \subseteq \mathcal{S}_K\ \forall 1 \leq j \leq m \right\}$ for any given $K, M \geq 1$ where the constraint set $\mathcal{S}_K$ has exactly K elements. To simplify the exposition, let us assume that $k_j = K$ for all $1 \leq j \leq m$. Consider the following locally constrained version of the multilevel Wasserstein means problem

$$\inf \sum_{j=1}^{m} W_2^2(G_j, P_{n_j}^j) + W_2^2(\mathcal{H}, \frac{1}{m} \sum_{j=1}^{m} \delta_{G_j}). \tag{30}$$

where $\mathcal{S}_K,\ G_j, \mathcal{H} \in \mathcal{A}_{M,\mathcal{S}_K}$ in the above infimum. We call the above optimization the problem of *Multilevel Wasserstein Means with Sharing (MWMS)*. The local constraint assumption $\text{supp}(G_j) \subseteq \mathcal{S}_K$ had been utilized previously in the literature — see for example the work of [34], who developed an optimization-based approach to the inference of the HDP [62], which also encourages explicitly the sharing of local group means among local clusters. Now, we can rewrite objective function (30) as follows

$$\inf_{\mathcal{S}_K, G_j, \boldsymbol{H} \in \mathcal{B}_{M,\mathcal{S}_K}} \sum_{j=1}^{m} W_2^2(G_j, P_{n_j}^j) + \frac{d_{W_2}^2(G_j, \boldsymbol{H})}{m} \tag{31}$$

Algorithm 2 Multilevel Wasserstein Means with Sharing (MWMS)

Input: Data $X_{j,i}$, K, M.
Output: global set S_K, local measures G_j, and elements H_i of $\boldsymbol{H}$.
Initialize $S_K^{(0)} = \left\{a_1^{(0)}, \ldots, a_K^{(0)}\right\}$, elements $H_i^{(0)}$ of $\boldsymbol{H}$, and $t = 0$.
while $S_K^{(t)}, G_j^{(t)}, H_i^{(t)}$ have not converged **do**
1. Update global set $S_K^{(t)}$:
for $j = 1$ **to** m **do**
$i_j \leftarrow \arg\min_{1 \le u \le M} W_2^2(G_j^{(t)}, H_u^{(t)})$; $T^j \leftarrow$ optimal coupling of $G_j^{(t)}$, P_n^j.
$U^j \leftarrow$ optimal coupling of $G_j^{(t)}$, $H_{i_j}^{(t)}$.
end for
for $i = 1$ **to** M **do**
$h_i^{(t)} \leftarrow$ atoms of $H_i^{(t)}$ with $h_{i,v}^{(t)}$ as v-th column.
end for
for $i = 1$ **to** K **do**
$$mD \leftarrow m \sum_{u=1}^{m} \sum_{v=1}^{n_i} T_{i,v}^u + \sum_{u=1}^{m} \sum_{v \neq i} U_{i,v}^u.$$
$$a_i^{(t+1)} \leftarrow \left(m \sum_{u=1}^{m} \sum_{v=1}^{n_i} T_{i,v}^u X_{u,v} + \sum_{u=1}^{m} \sum_{v} U_{i,v}^u h_{j_u,v}^{(t)} \right) / mD.$$
end for
2. Update $G_j^{(t)}$ for $1 \le j \le m$:
for $j = 1$ **to** m **do**
$$G_j^{(t+1)} \leftarrow \arg \min_{G_j : \mathrm{supp}(G_j) \equiv \mathcal{S}_K^{(t+1)}} W_2^2(G_j, P_{n_j}^j) + W_2^2(G_j, H_{i_j}^{(t)})/m.$$
end for
Update $H_i^{(t)}$ for $1 \le i \le M$ as Algorithm *1.*
$t \leftarrow t + 1$.
end while

where $\mathcal{B}_{M,\mathcal{S}_K} = \Big\{ G_j \in \mathcal{O}_K(\Theta),\ \boldsymbol{H} = (H_1, \ldots, H_M) : \text{supp}(G_j) \subseteq \mathcal{S}_K\ \forall 1 \leq j \leq m \Big\}$. The high level idea of finding local minimums of objective function (31) is to first, update the elements of constraint set $\mathcal{S}_K$ to provide the supports for local measures G_j and then, obtain the weights of these measures as well as the elements of global set H by computing appropriate Wasserstein barycenters. The details of these steps are summarized in the MWMS Algorithm (Algorithm 2).

2.3 Results and Discussion

Synthetic data We are interested in evaluating the effectiveness of both MWM and MWMS clustering algorithms by considering different synthetic data generating processes. Unless otherwise specified, we set the number of groups $m = 50$, number of observations per group $n_j = 50$ in $d = 10$ dimensions, number of global clusters $M = 5$ with 6 atoms. For Algorithm 1 (MWM) local measures G_j have 5 atoms each; for Algorithm 2 (MWMS) number of atoms in constraint set S_K is 50. As a benchmark for the comparison we will use a basic 3-stage K-means approach (cf. [27, Alg. 3]). The Wasserstein distance between the estimated distributions (i.e. $\hat{G}_1, \ldots, \hat{G}_m$; $\hat{H}_1, \ldots, \hat{H}_M$) and the data generating ones will be used as the comparison metric.

Recall that the MWM formulation does not impose constraints on the atoms of G_i, while the MWMS formulation explicitly enforces the sharing of atoms across these measures. We used multiple layers of mixtures while adding Gaussian noise at each layer to generate global and local clusters and the no-constraint (NC) data. We varied number of groups m from 500 to 10000. We notice that the 3-stage K-means algorithm performs the best when there is no constraint structure *and* variance is constant across clusters (Fig. 14(a) and 15(a)) - this is, not surprisingly, a favorable setting for the basic K-means method. As soon as we depart from the (unrealistic) constant-variance, no-sharing assumption, both of our algorithms start to outperform the basic three-stage K-means. The superior performance is most pronounced with local-constraint (LC) data (with or without constant variance conditions). See Fig. 14(c,d). It is worth noting that even when group variances are constant, the 3-stage K-means is no longer longer effective because now fails to account for the shared structure. When $m = 50$ and group sizes are larger, we set $S_K = 15$. Results are reported in Fig. 15 (c), (d). These results demonstrate the effectiveness and flexibility of our both algorithms.

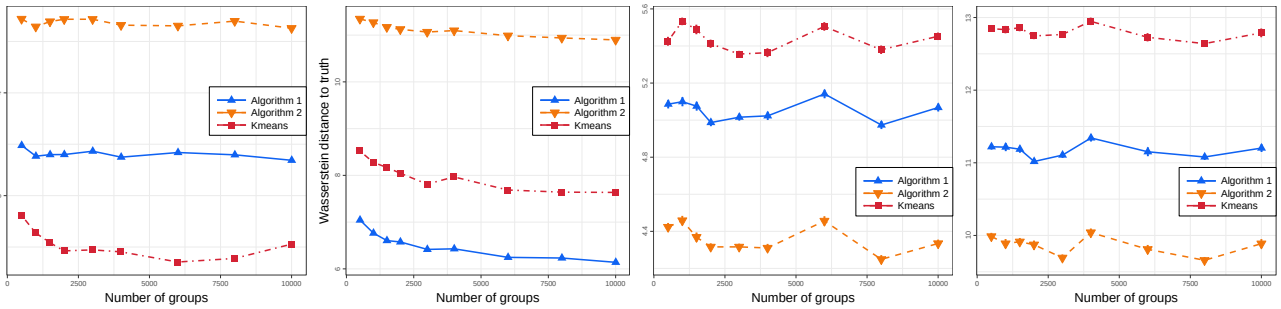


Figure 14: Data with a lot of small groups: (a) NC data with constant variance; (b) NC data with non-constant variance; (c) LC data with constant variance; (d) LC data with non-constant variance

Real-world data We applied our multilevel clustering algorithms to two real-world datasets: *LabelMe* and *StudentLife*.

LabelMe dataset consists of 2,688 annotated images which are classified into 8 scene categories including *tall buildings, inside city, street, highway, coast, open country, mountain,* and *forest* [51] . Each image contains multiple annotated regions. Each region, which is annotated by users, represents an object in the image. As shown in Figure 16, the left image is an image from *open country* category and contains 4 regions while the right panel denotes an image of *tall buildings*

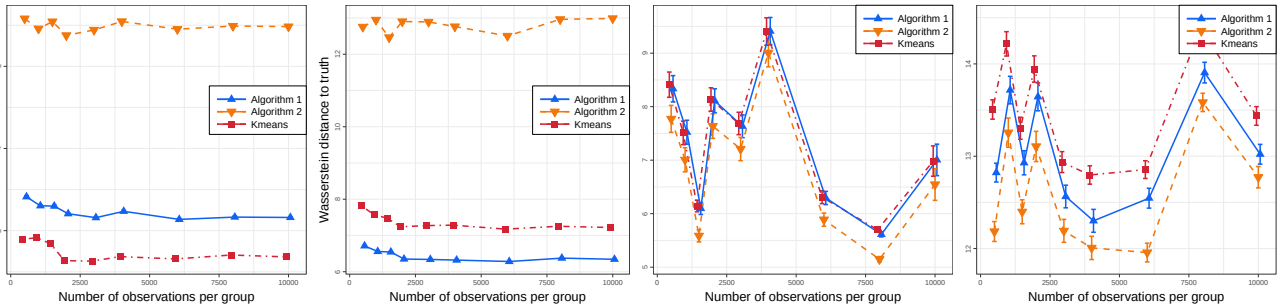


Figure 15: Data with few big groups: (a) NC data with constant variance; (b) NC data with non-constant variance; (c) LC data with constant variance; (d) LC data with non-constant variance

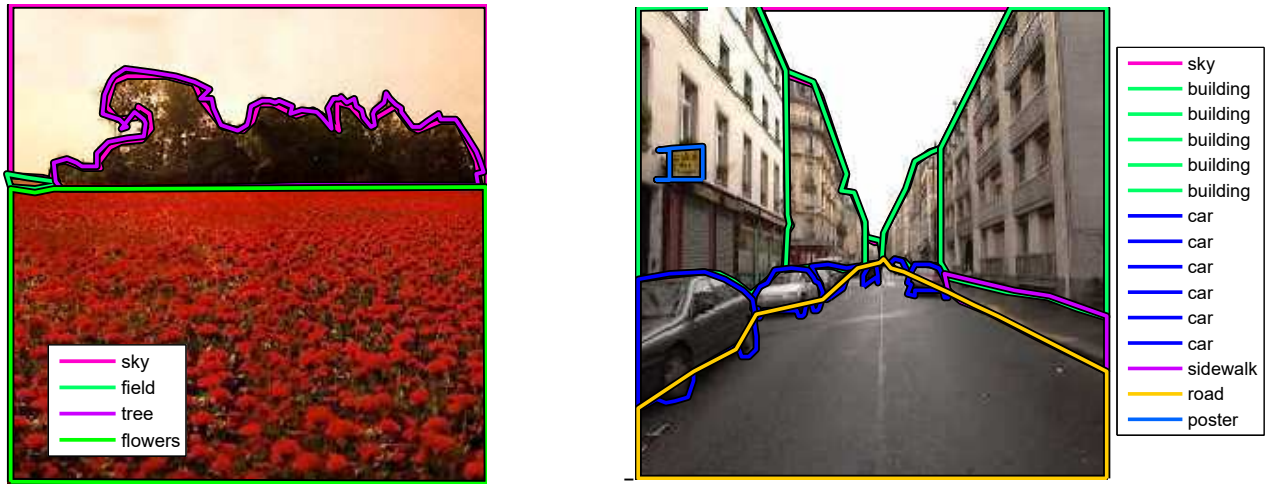


Figure 16: Examples of images used in LabelMe dataset. Each image consists of different annotated regions.

category including 16 regions. Note that the regions in each image can be overlapped. We remove the images containing less then 4 regions and obtain $1,800$ images.

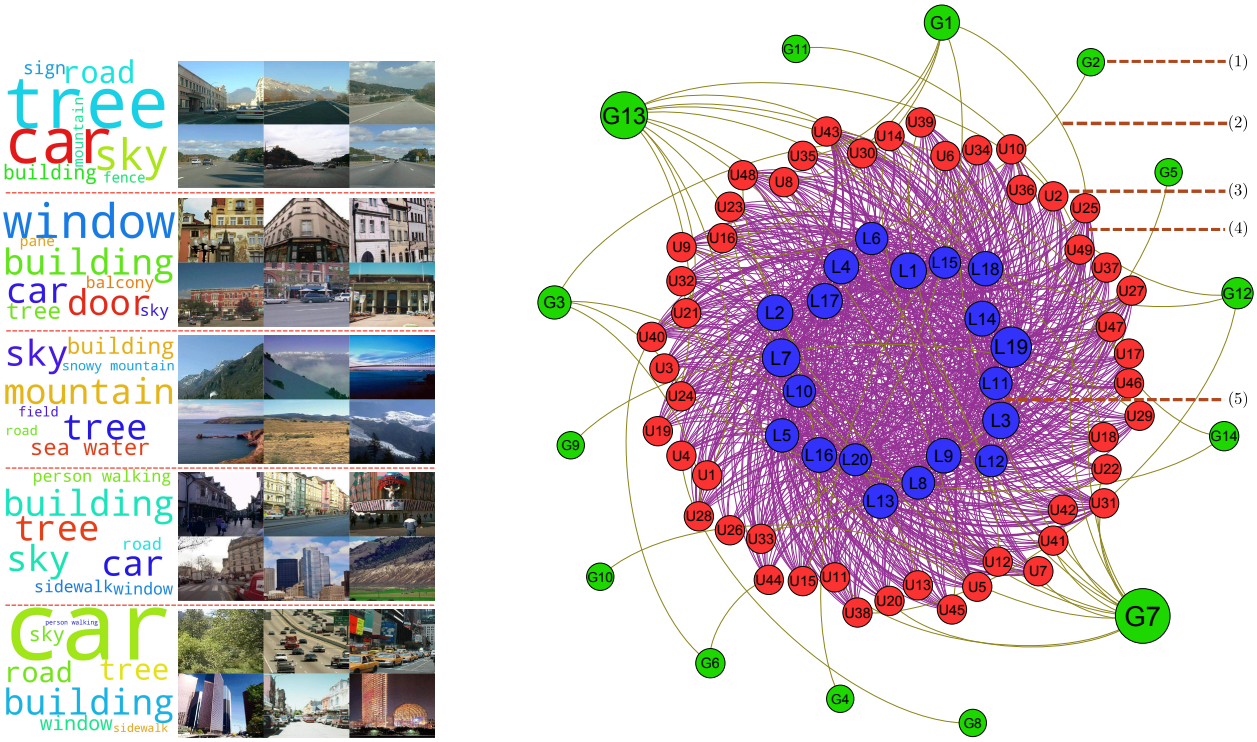
We then extract GIST feature [51] for each region in a image. GIST is a visual descriptor to represent perceptual dimensions and oriented spatial structures of a scene. Each GIST descriptor is a 512-dimensional vector. We further use PCA to project GIST features into 30 dimensions. Finally, we obtain $1,800$ "documents", each of which contains regions as observations. Each region now is represented by a 30-dimensional vector. We now can perform clustering regions in every image since they are visually correlated. In the next level of clustering, we can cluster images into scene categories.

StudentLife dataset is a large dataset frequently used in pervasive and ubiquitous computing research. Data signals consist of multiple channels (e.g., WiFi signals, Bluetooth scan, etc.), which are collected from smartphones of 49 students at Dartmouth College over a 10-week spring term in 2013. However, in our experiments, we use only WiFi signal strengths. We applied a similar procedure described in [45] to pre-process the data. We aggregate the number of scans by each Wifi access point and select 500 Wifi Ids with the highest frequencies. Eventually, we obtain 49 "documents" with totally approximately 4.6 million 500-dimensional data points.

Experimental results. To quantitatively evaluate our proposed methods, we compare our algorithms with several base-line methods: K-means, three-stage K-means (TSK-means) as described in the Supplement, MC2-SVI without context [30]. Clustering performance in Table 2 is evaluated with the image clustering problem for *LabelMe dataset*. With *K-means*, we average all data points to obtain a single vector for each images. K-means needs much less time to run since the number of data points is now reduced to $1,800$. For MC2-SVI, we used stochastic varitational and a parallelized Spark-based implementation in [30] to carry out experiments. This implementation has the advantage of making use of all of 16 cores on the test machine. The running time

Table 2: Clustering performance for LabelMe dataset.

Methods	NMI	ARI	AMI	Time (s)
K-means	0.349	0.237	0.324	**0.3**
TSK-means	0.236	0.112	0.22	218
MC2	0.315	0.206	0.273	4.2
MWM	0.373	0.263	0.352	332
MWMS	**0.391**	**0.284**	**0.368**	544



(a) (b)

Figure 17: Clustering representation for two datasets: (a) Five image clusters from *Labelme* data discovered by MWMS algorithm: tag-clouds on the left are accumulated from all images in the clusters while six images on the right are randomly chosen images in that cluster; (b) StudentLife discovered network with three node groups: (1) discovered student clusters, (3) student nodes, (5) discovered activity location (from Wifi data); and two edge groups: (2) Student to cluster assignment, (4) Student involved to activity location. Node sizes (of discovered nodes) depict the number of element in clusters while edge sizes between *Student* and *activity location* represent the popularity of student's activities.

for MC2-SVI is reported after scanning one epoch. In terms of clustering accuracy, MWM and MWMS algorithms perform the best.

Fig. 17a demonstrates five representative image clusters with six randomly chosen images in each (on the right) which are discovered by our MWMS algorithm. We also accumulate labeled tags from all images in each cluster to produce the tag-cloud on the left. These tag-clouds can be considered as visual ground truth of clusters. Our algorithm can group images into clusters which are consistent with their tag-clouds.

We use StudentLife dataset to demonstrate the capability of multilevel clustering with large-scale datasets. This dataset not only contains a large number of data points but presents in high dimension. Our algorithms need approximately 1 hour to perform multilevel clustering on this dataset. Fig. 17b presents two levels of clusters discovered by our algorithms. The innermost (blue) and outermost (green) rings depict local and global clusters respectively. Global clusters represent groups of students while local clusters shared between students ("documents") may be used to infer locations of students' activities. From these clusteing we can dissect students' shared location (activities), e.g. Student 49 ($U49$) mainly takes part in activity location 4 ($L4$).

3 Collaborations and Partnerships

With the partial support of this grant, several fruitful collaboration has been established. We have worked closely with Prof Ba-Ngu Vo and his team at Curtin University on the theme of point pattern data. Prof Vo's group is a well-known for his pioneering role in multitarget tracking using finite-random sets, which are more accessible to be known as point pattern data (PPD) in the data analysis field. This theory is largely unknown to the machine learning research community. This grant has made it possible to partially support the long-term collaboration between us and Prof Vo's research where we have initiated the idea of applying point pattern data to machine learning problem, and developing, to our knowledge, for the first time, learning and parameter estimation framework for statistical models dealing with PPD relevant to common tasks in machine learning.

The scientific inquiry into optimal transport theory for scaling up Bayesian inference has also brought us to work closely with Prof. Long Nguyen and his research group at Michigan University at Ann Arbor, Dr Hung Bui from Adobe Research and now at Google DeepMind, and Dr Nhat Ho now at Berkeley UC. This has been an interesting journey for all of us, and our results are one of the very first to address the Bayesian inference through this len of Wasserstein geometry, published in the prestigious ICML conference in 2017. This work has enabled us to continue to use this theory for research into deep generative models, which are one of most active research areas in deep learning research.

4 Publication Outcomes

1. Ba-Ngu Vo, Nhat-Quang Tran, Dinh Phung, and Ba-Tuong Vo. Model-based classification and novelty detection for point pattern data. In *23rd Intl. Conf. on Pattern Recognition (ICPR)*, Dec. 2016.

2. Nhat-Quang Tran, Ba-Ngu Vo, Dinh Phung, and Ba-Tuong Vo. Clustering for point pattern data. In *23rd Intl. Conf. on Pattern Recognition (ICPR)*, Dec. 2016.

3. Nhan Dam, Dinh Phung, Ba-Ngu Vo, and Viet Huynh. Forward-Backward Smoothing for Hidden Markov Models of Point Pattern Data. In *Proc. of International Conference on Data Science and Advanced Analytics (DSAA)*, Tokyo, Japan, 2017.

4. Nhat Ho, XuanLong Nguyen, Mikhail Yurochkin, Hung Bui, Viet Huynh, and Dinh Phung. Multilevel clustering via Wasserstein means. In *Proc. of International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017.

5. Ba-Ngu Vo, Nhan Dam, Dinh Phung, Quang Tran, and Ba-Tuong Vo. Model-Based Learning for Point Pattern Data. In *Pattern Recognition*, (under revision), 2018.

6. Nhat Ho, XuanLong Nguyen, Mikhail Yurochkin, Hung Bui, Viet Huynh, Nhan Dam and Dinh Phung. Multilevel clustering with Contexts using Wasserstein distance. To be submitted to *Journal of Machine Learning Research (JMLR)*, (under preparation), 2018.

5 Attachments

Paper numbers 1, 2, 3, 4, 5 are attached.

References

[1] M. Agueh and G. Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43:904–924, 2011.

[2] P. C. Alvarez-Estebana, E. del Barrioa, J.A. Cuesta-Albertosb, and C. Matrán. A fixed-point approach to barycenters in wasserstein space. *Journal of Mathematical Analysis and Applications*, 441:744–762, 2016.

[3] Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

[4] E. Anderes, S. Borgwardt, and J. Miller. Discrete wasserstein barycenters: optimal transport for discrete data. *http://arxiv.org/abs/1507.07218*, 2015.

[5] Adrian Baddeley, Imre Bárány, and Rolf Schneider. Spatial point processes and their applications. *Stochastic Geometry: Lectures given at the CIME Summer School held in Martina Franca, Italy, September 13–18, 2004*, pages 1–75, 2007.

[6] Adrian Baddeley and Rolf Turner. Practical maximum pseudolikelihood for spatial point patterns. *Australian & New Zealand J. Statistics*, 42(3):283–322, 2000.

[7] J. D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Payré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 2:1111–1138, 2015.

[8] Julian Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

[9] JULIAN Besag. Some methods of statistical analysis for spatial data. *Bulletin of the Int. Statistical Institute*, 47(2):77–92, 1977.

[10] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[11] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res*, 3:993–1022, 2003.

[12] Igor V Cadez, Scott Gaffney, and Padhraic Smyth. A general probabilistic framework for clustering individuals and objects. In *Proc. 6th ACM SIGKDD Int. Conf. knowledge discovery and data mining*, pages 140–149, 2000.

[13] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surveys (CSUR)*, 41(3):15, 2009.

[14] Veronika Cheplygina, David MJ Tax, and Marco Loog. Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264–275, 2015.

[15] Veronika Cheplygina, David MJ Tax, and Marco Loog. On classification with bags, groups and sets. *Pattern Recognition Letters*, 59:11–17, 2015.

[16] David Maxwell Chickering and David Heckerman. Fast learning from sparse data. In *Proc. 15th Conf. Uncertainty in artificial intelligence*, pages 109–115. Morgan Kaufmann Publishers Inc., 1999.

[17] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop statistical learning in computer vision, ECCV*, 2004.

[18] M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems 26*, 2013.

[19] M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[20] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes*, volume 2. Springer, 1988.

[21] Thomas Fiksel. Estimation of interaction potentials of gibbsian point processes. *Statistics*, 19(1):77–86, 1988.

[22] Maurizio Filippone and Guido Sanguinetti. Information theoretic novelty detection. *Pattern Recognition*, 43(3):805–814, 2010.

[23] Dariu M Gavrila and Vasanth Philomin. Real-time object detection for "smart" vehicles. In *Proc. 7th Int. Conf. Comput. Vision, 1999*, volume 1, pages 87–93, 1999.

[24] Charles J Geyer et al. Likelihood inference for spatial point processes. *Stochastic geometry: likelihood and computation*, 80:79–140, 1999.

[25] S. Graf and H. Luschgy. *Foundations of quantization for probability distributions*. Springer-Verlag, New York, 2000.

[26] Javad Hamidzadeh, Reza Monsefi, and Hadi Sadoghi Yazdi. Irahc: instance reduction algorithm using hyperrectangle clustering. *Pattern Recognition*, 48(5):1878–1889, 2015.

[27] Nhat Ho, XuanLong Nguyen, Mikhail Yurochkin, Hung Hai Bui, Viet Huynh, and Dinh Phung. Multilevel clustering via wasserstein means. *arXiv preprint arXiv:1706.03883*, 2017.

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

[29] Daniel P Huttenlocher, Jae J Noh, and William J Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. Comput. Vision, 1993*, pages 93–101. IEEE, 1993.

[30] V. Huynh, D. Phung, S. Venkatesh, X. Nguyen, M. Hoffman, and H. Bui. Scalable nonparametric bayesian multilevel clustering. *Proceedings of Uncertainty in Artificial Intelligence*, 2016.

[31] Jens Ledet Jensen and Jesper Møller. Pseudolikelihood for exponential family models of spatial point processes. *Annals of Applied Probability*, pages 445–461, 1991.

[32] Liping Jing, Michael K Ng, and Joshua Zhexue Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowl. Data Eng.*, 19(8):1026–1041, 2007.

[33] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, DTIC Document, 1996.

[34] B. Kulis and M. I. Jordan. Revisiting k-means: new algorithms via bayesian nonparametrics. *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[35] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1265–1278, 2005.

[36] R. Mahler. Multi-target Bayes filtering via first-order multi-target moments. *IEEE Trans. Aerospace & Electronic Systems*, 39(4):1152–1178, 2003.

[37] Ronald PS Mahler. *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[38] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge univ. press Cambridge, 2008.

[39] Markos Markou and Sameer Singh. Novelty detection: a review – part 1: statistical approaches. *Signal Process.*, 83(12):2481–2497, 2003.

[40] Melvin Earl Maron. Automatic indexing: an experimental inquiry. *JACM*, 8(3):404–417, 1961.

[41] Andrew McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop learning for text categorization*, volume 752, pages 41–48, 1998.

[42] Vahid Hooshmand Moghaddam and Javad Hamidzadeh. New hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier. *Pattern Recognition*, 60:921–935, 2016.

[43] Jesper Moller and Rasmus Plenge Waagepetersen. *Statistical inference and simulation for spatial point processes*. CRC Press, 2003.

[44] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[45] Thanh-Binh Nguyen, Vu Nguyen, Svetha Venkatesh, and Dinh Phung. Mcnc: Multi-channel nonparametric clustering from heterogeneous data. In *Proceedings of ICPR*, 2016.

[46] Thanh-Binh Nguyen, Vu Nguyen, Svetha Venkatesh, and Dinh Q. Phung. MCNC: multi-channel nonparametric clustering from heterogeneous data. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pages 3633–3638, 2016.

[47] V. Nguyen, D. Phung, X. Nguyen, S. Venkatesh, and H. Bui. Bayesian nonparametric multilevel clustering with group-level contexts. *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[48] X. Nguyen. Convergence of latent mixing measures in finite and infinite mixture models. *Annals of Statistics*, 4(1):370–400, 2013.

[49] X. Nguyen. Borrowing strengh in hierarchical bayes: Posterior concentration of the dirichlet base measure. *Bernoulli*, 22:1535–1571, 2016.

[50] Yosihiko Ogata and Masaharu Tanemura. Likelihood analysis of spatial point patterns. *J. Royal Statistical Society. Series B (Methodological)*, pages 496–518, 1984.

[51] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[52] D. Pollard. Quantization and the method of k-means. *IEEE Transactions on Information Theory*, 28:199–205, 1982.

[53] J. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

[54] Bharath Ramesh, Cheng Xiang, and Tong Heng Lee. Shape classification using invariant features and contextual information in the bag-of-words model. *Pattern Recognition*, 48(3):894–906, 2015.

[55] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*. Springer Sci. & Business Media, 2009.

[56] A. Rodriguez, D. Dunson, and A.E. Gelfand. The nested Dirichlet process. *J. Amer. Statist. Assoc.*, 103(483):1131–1154, 2008.

[57] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *6th IEEE Int. Conf. Comput. Vision*, pages 59–66, 1998.

[58] J. Solomon, G. Fernando, G. Payré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. In *The International Conference and Exhibition on Computer Graphics and Interactive Techniques*, 2015.

[59] S. Srivastava, V. Cevher, Q. Dinh, and D. Dunson. Wasp: Scalable bayes via barycenters of subset posteriors. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015.

[60] Dietrich Stoyan, Wilfrid S Kendall, and Joseph Mecke. *Stochastic geometry and its applications*. John Wiley & Sons, 1995.

[61] R Takacs. Estimator for the pair–potential of a gibbsian point process. *Statistics: A J. Theoretical and Applied Statistics*, 17(3):429–433, 1986.

[62] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 101:1566–1581, 2006.

[63] M. van Lieshout. *Markov Point Processes and their Applications*. Imperial College Press, 2000.

[64] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of comput. vision algorithms. http://www.vlfeat.org/, 2008.

[65] Cédric Villani. *Topics in Optimal Transportation*. American Mathematical Society, 2003.

[66] Ba-Ngu Vo, Sumeetpal Singh, and Arnaud Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *Aerosp. Electron. Syst., IEEE Trans.*, 41(4):1224–1245, 2005.

[67] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 3–14. ACM, 2014.

[68] D. F. Wulsin, S. T. Jensen, and B. Litt. Nonparametric multi-level clustering of human epilepsy seizures. *Annals of Applied Statistics*, 10:667–689, 2016.

[69] Konstantinos Zagoris, Ioannis Pratikakis, Apostolos Antonacopoulos, Basilis Gatos, and Nikos Papamarkos. Distinction between handwritten and machine-printed text based on the bag of visual words model. *Pattern Recognition*, 47(3):1051–1062, 2014.

[70] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vision*, 73(2):213–238, 2007.