ARL-TR-7677 • MAY 2016

US Army Research Laboratory

Sparsity-Based Representation for Classification Algorithms and Comparison Results for Transient Acoustic Signals

by Minh Dao and Tung-Duong Tran-Luu

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-7677 ● MAY 2016

US Army Research Laboratory

Sparsity-Based Representation for Classification Algorithms and Comparison Results for Transient Acoustic Signals

by Minh Dao
MBO Partners, Herndon, VA

Tung-Duong Tran-Luu
Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
May 2016	Final	June 2015–June 2016

4. TITLE AND SUBTITLE
Sparsity-Based Representation for Classification Algorithms and Comparison Results for Transient Acoustic Signals

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Minh Dao and Tung-Duong Tran-Luu

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-SES-P
2800 Powder Mill Road
Adelphi, MD 20783-1138

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-7677

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

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

13. SUPPLEMENTARY NOTES
Authors' emails: <ducminh174@gmail.com> , <tung-duong.tran-luu.civ@mail.mil>.

14. ABSTRACT
In this report, we propose a general sparsity-based framework for the classification of transient acoustic signals; this framework enforces various sparsity structures like joint-sparse or group-and-joint-sparse within measurements of multiple acoustic sensors. We further robustify our models to deal with the presence of dense and large but correlated noise and signal interference (i.e., low-rank interference). Another contribution is the implementation of deep learning architectures to perform classification on the transient acoustic data set. Extensive experimental results are included in the report to compare the classification performance of sparsity-based and deep-network-based techniques with conventional classifiers such as Markov switching vector auto-regression, Gaussian mixture model, support vector machine (SVM), hidden Markov model (HMM), sparse logistic regression, and the combination of SVM and HMM methods (SVM-HMM) for 2 experimental sets of 4-class and 6-class classification problems.

15. SUBJECT TERMS
acoustic signal classification, multi-sensor, sparse representation, low rank, deep learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Tung-Duong Tran-Luu
Unclassified	Unclassified	Unclassified	UU	52	19b. TELEPHONE NUMBER (Include area code) 301-394-3082

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

1.1 Motivations

The US Army Research Laboratory (ARL) is developing an acoustic system to detect and localize launch points and impact points of different kinds of weapons in the field. Proper identification of the detected events would filter out the information, increase situational awareness, and allow the Soldier to take appropriate action. For example, it can provide a warning for incoming threats, or can cue a radar or a high-resolution camera.

Experimental work over the last decade on classification of acoustic signatures collected from ARL acoustic arrays suggests that the classifier can perform satisfactorily in environments on which it has been trained. However, it has difficulty generalizing to different environments. Yet, due to practical limitation and cost, the collected data cannot cover all possible atmospheric conditions, terrain geometries, and source-sensor ranges. This report examines this issue using a variety of classifiers and setups. It takes one attempt at quantifying the effect of propagation channels. It also points to other research directions that are believed to be able to get around this fundamental limitation.

1.2 Sparsity-Based Representation for Transient Acoustic Signals

For the last decade, sparse signal representations have proven to be extremely powerful tools in solving many inverse problems, where sparsity acts as a strong prior to alleviate the ill-posed nature of the problem. Recent research has pointed out that sparse representation is also useful for discriminative applications.[1–3] These applications rely on the crucial observation that test samples in the same class usually lie in a low-dimensional subspace of some proper bases or dictionaries. Thus, if the dictionary is constructed from all the training samples in all the classes, the test samples can be sparsely represented by only a few columns of this dictionary. Therefore, the sparse coefficient vector, which is recovered efficiently via ℓ_1-minimization techniques, can naturally be considered as the discriminative factor. In Wright et al.,[2] the authors successfully applied this idea to the face recognition problem. Since then, many more sophisticated techniques have been exploited and applied to various fields, such as hyperspectral target detection,[4] chemical plume detection and classification,[5] and visual classification.[1,6,7]

Nowadays, many real-world problems involve simultaneous representations of multiple correlated signals. These applications normally face the scenario where data sampling is performed simultaneously from multiple colocated sources (such as multiple channels or sensors) — yet within a small spatio-temporal neighborhood, recording the same physical event. This data collection scenario allows exploitation of correlated features within the signal sources to improve the resulting signal representation and guide successful decision making. Joint sparsity models, which assume the fact that multiple measurements belonging to the same class can be simultaneously sparsely represented by a few common training samples in the dictionaries, have been successfully applied in many applications. For instance, a joint sparse representation (JSR)-based method is proposed in Chen et al.[8] for target detection in hyperspectral imagery. The model exploits the fact that the sparse representations of hyperspectral pixels in a small neighborhood tend to share common sparsity patterns. Yuan and Yan[7] investigated a multitask model for visual classification, which also assumes that multiple observations from the same class could be simultaneously represented by a few columns of the training dictionary. Similarly, Nguyen et al.[9] investigated a multi-sensor classification framework via a multivariate sparse representation that forces different recording sensors to share the same sparse support distributions on their coefficient vectors.

In this report, we first develop a JSR model which imposes row-sparsity constraints across multiple acoustic sensors to collaboratively classify transient acoustic signals. Furthermore, we robustify our models to deal with the presence of large and dense but correlated signal-interference/noise (namely low-rank interference). This scenario is normally observed when the recorded data are the superimpositions of target signals with interferences, which can be signals from external sources, effects of propagation channel on multiple sensors, the underlying background that is inherently anchored in the data, or any pattern noise that remains stationary during signal collection. These interferences normally have correlated structure and appear as a low-rank signal interference since the sensors are spatially co-located and data samples are temporally recorded, thus any interference from external sources will affect similarly on all the multiple sensor measurements. Another extension of our sparsity-based representation models is the incorporation of group-structured-sparsity constraints among observations of multiple sensors. The group-sparse constraint is concurrently enforced with row-sparse constraints among the support coefficients of all sensors, thus yielding one more layer of classification robustness.

1.3 Contributions

The main contributions of this technical report are as follows:

- We develop a variety of novel sparsity-regularized regression methods that effectively incorporate simultaneous structured-sparsity constraints, demonstrated via a row-sparse and/or group-sparse coefficient matrix, across multiple acoustic sensors. We also robustify our models to deal with the presence of a large but low-rank interference term. While row-sparsity constraint in the JSR model has been extensively studied and applied in many application disciplines, the models with the incorporation of low-rank interference as well as the integration of both joint- and group-sparsity structure are our own developments. Preliminary classification results of these methods for a subset of ARL acoustic transients data set were presented as parts of our paper, which will soon be published in *IEEE Transactions on Signal Processing*.[10]

- We report comprehensive comparisons of the classification performance for 12 different classifiers, including 6 conventional classifiers previously developed and examined in ARL, 4 newly developed sparsity-based representation models, and 2 emerging deep learning architecture techniques.

- The classification performance of all competing methods is examined on a variety of experimental setups. Two broad sets of classification problems are conducted: a 4-class problem (launch and impact of 2 projectiles) and a 6-class problem (launch and impact of 3 projectiles). Furthermore, classification accuracy is evaluated for different partitioning of training and testing sets, effectively showing the performance degradation due to the propagation of the signals through the environment. In addition, classification rates are summarized and analyzed for both weighted and non-weighted classification results. Experimental setups and empirical results are discussed and analyzed in detail in Section 4 of the report.

1.4 Notations and Outline

The following notational conventions are used throughout this report. We denote vectors by boldface lowercase letters, such as $\boldsymbol{x}$, and denote matrices by boldface uppercase letters, such as $\boldsymbol{X}$. For a matrix $\boldsymbol{X}$, $\boldsymbol{X}_{i,j}$ represents the element at row i^{th} and column j^{th} of $\boldsymbol{X}$ while a bold lowercase letter with subscript, such as $\boldsymbol{x}_j$,

represents its j^{th} column. The ℓ_q-norm of a vector $\boldsymbol{x} \in \mathbb{R}^N$ is defined as $\|\boldsymbol{x}\|_q = (\sum_{i=1}^{N} |x_i|^q)^{1/q}$ where x_i is the i^{th} element of $\boldsymbol{x}$. Given a matrix $\boldsymbol{X} \in \mathbb{R}^{N \times M}$, $\|\boldsymbol{X}\|_F$, $\|\boldsymbol{X}\|_{1,q}$, and $\|\boldsymbol{X}\|_*$ are used to defined its Frobenious norm, mixed $\ell_{1,q}$-norm and nuclear-norm, respectively.

The remainder of this report is organized as follows. In Section 2, we give a brief overview of sparse representation for classification. Section 3 introduces various proposed sparsity models based on different assumptions on the structures of coefficient vectors and low-rank noise/interference. A fast and efficient algorithm based on the alternating direction method of multipliers (ADMM)[11] to solve the convex optimization problems that arise from these models and the guarantee of convergence to the optimal solutions are also outlined in this Section. Extensive experiments are conducted in Section 4, and conclusions and future works are drawn in Section 5.

2. Background Reviews

2.1 Sparse Representation for Classification

Recent years have witnessed the development of sparse representation techniques for both signal recovery and classification. In classification, in order to let the problem work, one often assumes that all of the samples that belong to the same class lie approximately in the same low-dimensional subspace.[2] Suppose we are given a dictionary representing C distinct classes $\boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_C] \in \mathbb{R}^{N \times P}$, where N is the feature dimension of each sample, and the c-th class subdictionary $\boldsymbol{D}_c$ has P_c training samples $\{\boldsymbol{d}_{c,p}\}_{p=1,...,P_c}$, resulting in a total of $P = \sum_{c=1}^{C} P_c$ samples in the dictionary $\boldsymbol{D}$. To label a test sample $\boldsymbol{y} \in \mathbb{R}^N$, it is often assumed that $\boldsymbol{y}$ can be represented by a subset of the training samples in $\boldsymbol{D}$. Mathematically, $\boldsymbol{y}$ is written as

$$\boldsymbol{y} = [\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_C] \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_C \end{pmatrix} + \boldsymbol{n} = \boldsymbol{D}\boldsymbol{a} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{a} \in \mathbb{R}^P$ is the unknown coefficient vector, and $\boldsymbol{n}$ is the low-energy noise due to the imperfection of the test sample and has little effect on the classification decision. For simplicity, the presence of $\boldsymbol{n}$ will be discarded from all model descriptions, though it is still taken into consideration by the fidelity constraint (a penalty

term with Frobenious norm) in the optimization process.

The sparsity assumption implies that only a few coefficients of $\boldsymbol{a}$ are nonzero and most of the others are insignificant. Particularly, only entries of $\boldsymbol{a}$ that are associated with the class of the test sample $\boldsymbol{y}$ are nonzero, and thus, $\boldsymbol{a}$ is a sparse vector. Taking this prior into account, many methods have been proposed to find the coefficient vector $\boldsymbol{a}$ efficiently, including ℓ_1-norm minimization,[12] greedy pursuit (e.g., CoSaMP[13] or subspace pursuit[14]), and iterative hard threshold,[15] to name a few. In this report, we favor the ℓ_1-minimization approach, which is described as follows:

$$\begin{aligned} \underset{\boldsymbol{a}}{min} \quad & \|\boldsymbol{a}\|_1 \\ s.t. \quad & \boldsymbol{y} = \boldsymbol{D}\boldsymbol{a}. \end{aligned} \tag{2}$$

Once the coefficient vector $\widehat{\boldsymbol{a}}$ is obtained from Eq. 2, the next step is to assign the test sample $\boldsymbol{y}$ to a class label. This can be determined by simply taking the minimal residual between $\boldsymbol{y}$ and its approximation from each class subdictionary:

$$\text{Class}(\mathbf{y}) = \underset{c=1,\ldots,C}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{D}_c\widehat{\boldsymbol{a}}_c\|_2 , \tag{3}$$

where $\widehat{\boldsymbol{a}}_c$ is the induced vector by keeping only the coefficients corresponding to the c-th class in $\widehat{\boldsymbol{a}}$. This step can be interpreted as assigning the class label of $\boldsymbol{y}$ to the class that can best represent $\boldsymbol{y}$. In the case a tested signal is the superposition of 2 or more classes, the class assignment step can be modified from Eq. 3 by considering if the residual between $\boldsymbol{y}$ and the closest representation with respect to 2 class subdictionaries is smaller than a certain threshold.

2.2 Joint Sparse Representation

Single-measurement sparse representation has been shown to be efficient for classification tasks because it provides an effective way to approximate the test sample from the training examples. However, in many practical applications, we are often given a set of test measurements collected from different observations of the same physical event. An obvious question is how to simultaneously exploit the information from various sources to come up with a more precise classification decision, rather than classifying each test sample independently and then assigning a class label via a simple fusion (e.g., a voting scheme). An active line of research recently

focuses on answering this question using JSR.[16–19] Mathematically, given an unlabeled set of M test samples $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_M] \in \mathbb{R}^{N \times M}$ from different nearby spatio-temporal observations, we again assume that each measurement $\boldsymbol{y}_m$ can be compactly represented by a few atoms in the training dictionary

$$\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_M] = [\boldsymbol{D}\boldsymbol{a}_1, \boldsymbol{D}\boldsymbol{a}_2, ..., \boldsymbol{D}\boldsymbol{a}_M] = \boldsymbol{D}\boldsymbol{A}, \tag{4}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_M] \in \mathbb{R}^{P \times M}$ is the unknown coefficient matrix. In the joint sparsity model, the sparse coefficient vectors $\{\boldsymbol{a}_m\}_{m=1}^{M}$ share the same support Γ, and thus the matrix $\boldsymbol{A}$ is a row-sparse matrix with only $|\Gamma|$ nonzero rows. This model is the extension of the aforementioned sparse representation for classification model to multiple observations and has been shown to enjoy better classification in various practical applications[7,16,17] as well as being able to reduce the sample size needed for signal reconstruction applications[18,20] when the row-sparsity assumption holds.

To recover the row-sparse matrix $\boldsymbol{A}$, the following joint sparse optimization is proposed:

$$\begin{aligned} \underset{\boldsymbol{A}}{min} \quad & \|\boldsymbol{A}\|_{1,q} \\ s.t. \quad & \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{A}, \end{aligned} \tag{5}$$

where the norm $\|\boldsymbol{A}\|_{1,q}$ with $q > 1$, defined as $\|\boldsymbol{A}\|_{1,q} = \sum_{i=1}^{P} \|\boldsymbol{a}_{i,:}\|_q$ with $\boldsymbol{a}_{i,:}$'s being rows of the matrix $\boldsymbol{A}$, encourages shared sparsity patterns across multiple sensors. This norm can be phrased as performing an ℓ_q-norm across the rows to enforce "joint" property and followed by an ℓ_1-norm along the columns to enforce "sparsity". It is clear that this $\ell_{1,q}$ regularization norm encourages shared sparsity patterns across related observations, and thus the solution of the optimization in Eq. 5 has common support at column level.

3. Robust Structural Sparsity-Based Representation for Classification of Transient Acoustic Signals

3.1 Joint Sparsity Model for Classification of Acoustic Transients

In this section, we discuss a general JSR framework for the classification of transient acoustic signals. In this application, the acoustic data are collected during the launch and impact of different types of munitions using a tetrahedral acoustic sensor array, hence there are 4 measurements for each data sample (i.e., $M = 4$). A feature vector (e.g., cepstral features that have been proved effective in speech

recognition and acoustic signal classification) is then extracted from each measurement and concatenated into a matrix $\boldsymbol{Y}$, as shown in Fig. 1. Since the 4 sensors simultaneously measure the same event, the coefficient matrix $\boldsymbol{A}$ in the linear representation of training samples in the dictionary $\boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_C]$ should have common sparse supports. In other words, $\boldsymbol{A}$ should be row-sparse. After a joint sparsity optimization is engaged in $\boldsymbol{Y}$, its class label is then determined by the following minimal residual rule:

$$\text{Class}(\mathbf{Y}) = \underset{c=1,...,C}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \boldsymbol{D}_c \widehat{\boldsymbol{A}}_c \right\|_F, \tag{6}$$

where $\|\cdot\|_F$ is the Frobenious norm of a matrix; $\boldsymbol{D}_c$ and $\widehat{\boldsymbol{A}}_c$ are the submatrices from class c-th of the training dictionary $\boldsymbol{D}$ and the coefficient matrix $\widehat{\boldsymbol{A}}$ solved by Eq. 5, respectively. This step can be interpreted as collaboratively assigning the class label of $\boldsymbol{Y}$ to the class that can best represent all samples in $\boldsymbol{Y}$.

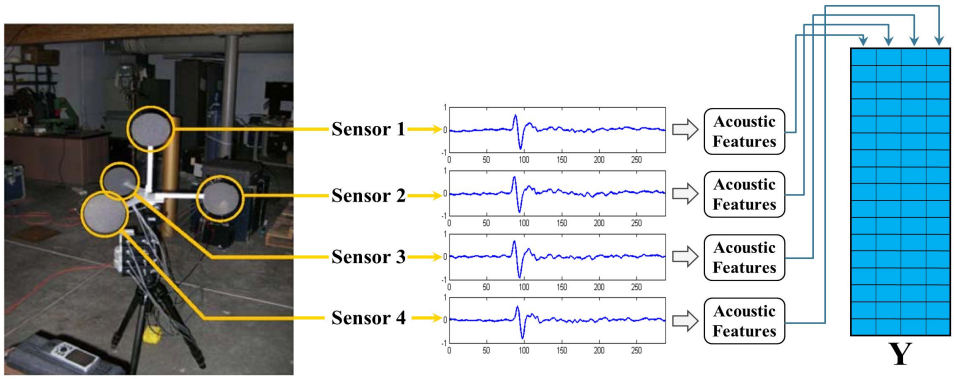


Fig. 1 Joint sparsity framework for classification of transient acoustic signals

3.2 Joint Sparse Representation with Low-Rank Interference

In this section, we study a robust joint-sparsity model that is capable of coping with the dense and large but correlated noise, so-termed low-rank interference. This scenario often happens when there are external sources interfering with the recording process of all sensors. When the sensors are closely spaced, interference sources look similar across the sensors, resulting in a large but low-rank corruption. Figure 2 illustrates a typical setup with multiple colocated sensors simultaneously recording the same physical events. In a multi-model setting, sensor colocation normally ensures that interference/noise patterns are very similar, hence justifying the low-rank assumption. These interference sources may include sound and vibration from a

car passing by, a helicopter hovering nearby, interference from any radio-frequency source, or the effect of propagation on the signal (in the cepstral space). Furthermore, in many situations the recorded data may contain not only the signal of interest but also the intrinsic background that normally stays stationary, hence promoting a low-rank background interference. Our proposed joint sparse representation with low-rank interference (JSR+L) model is expected to tackle this problem by extracting the low-rank component while collaboratively taking advantage of correlated information from different sensors.

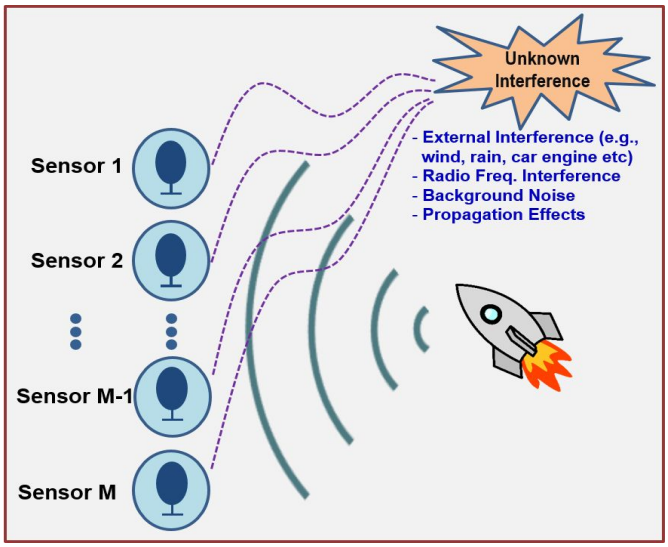


Fig. 2 A general multisensor problem with unknown low-rank interference

Mathematically, let $\boldsymbol{Y}$ be the measurement matrix. Under some circumstances, we are not able to observe the joint sparse representation $\boldsymbol{DA}$ in $\boldsymbol{Y}$ directly; instead, we observe its corrupted version $\boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{DA}$. The matrix $\boldsymbol{L}$ captures interference with the prior knowledge that $\boldsymbol{L}$ is low-rank. To separate $\boldsymbol{L}$ and $\boldsymbol{DA}$, a strategy model that simultaneously fits the low-rank approximation on $\boldsymbol{L}$ and a joint sparse ℓ_{1q}-regularization on the coefficient matrix $\boldsymbol{A}$ is proposed as

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{L}}{min} \quad & \|\boldsymbol{A}\|_{1,q} + \lambda_L \|\boldsymbol{L}\|_* \\ s.t. \quad & \boldsymbol{Y} = \boldsymbol{DA} + \boldsymbol{L}, \end{aligned} \tag{7}$$

where the nuclear matrix norm $\|\boldsymbol{L}\|_*$ is a convex-relaxation version of the rank defined as the sum of all singular values of the matrix $\boldsymbol{L}$[21,22]; and $\lambda_L > 0$ is a weighting parameter balancing the 2 regularization terms.

Once the solution $\{\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{L}}\}$ of Eq. 7 is computed, the class label of $\boldsymbol{Y}$ is decided by

the minimal residual rule

$$\text{Class}(\mathbf{Y}) = \underset{c=1,\ldots,C}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \boldsymbol{D}_c \widehat{\boldsymbol{A}}_c - \widehat{\boldsymbol{L}}_c \right\|_F^2, \tag{8}$$

where $\boldsymbol{D}_c$, $\widehat{\boldsymbol{A}}_c$, and $\widehat{\boldsymbol{L}}_c$ are the corresponding induced matrices associated with the c-th class.

3.3 Simultaneous Group-and-Joint Sparse Representation with Low-Rank Interference

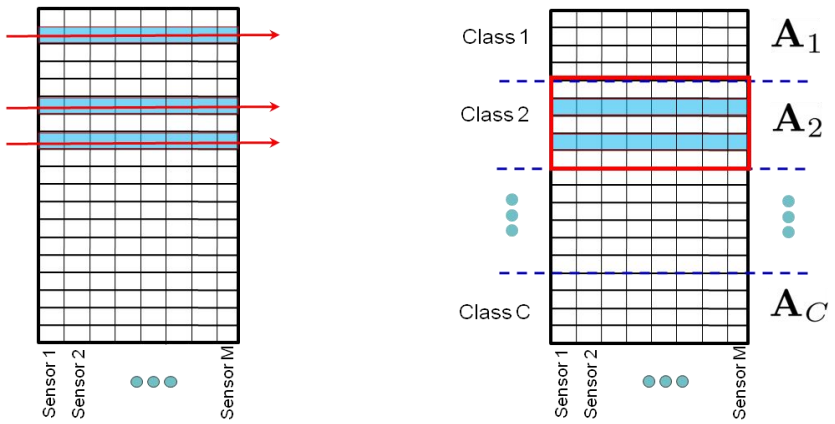
JSR+L has the capability to extract correlated noise/interference while simultaneously performing intercorrelation of multiple sensors in the coefficient matrix at row-level, hence boosting the overall classification result. Moreover, this model can even be strengthened by further incorporating the group sparsity constraint into the coefficient matrix $\boldsymbol{A}$. The idea of adding group structure has been intensively studied[23,24] and theoretically as well as empirically proven to better represent signals in discriminative purposes. Especially, this concept is critically beneficial for classification tasks where multiple measurements do not necessarily represent the same signals but rather come from the same set of classes. This leads to group sparse representation where the dictionary atoms are grouped and the sparse coefficients are enforced to have only a few active groups at a time, resulting a 2-level sparsity model: group-sparse and row-sparse in a combined cost function.

We tentatively apply this concept to the JSR+L model. The new model simultaneously searches for the group-and-row sparse structure representation among all sensors and low-rank interference and is termed group-and-joint sparse representation with low-rank interference (GJSR+L).

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{L}}{min} \quad & \|\boldsymbol{A}\|_{1,q} + \lambda_G \sum_{c=1}^{C} \|\boldsymbol{A}_c\|_F + \lambda_L \|\boldsymbol{L}\|_* \\ s.t. \quad & \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{A} + \boldsymbol{L}, \end{aligned} \tag{9}$$

where $\boldsymbol{A}_c$ is the subcoefficient matrix of $\boldsymbol{A}$ induced by the labeled indexes corresponding to class c; and $\lambda_G \geq 0$ is the weighting parameter of the group constraint. The optimization in Eq. 9 can be interpreted as follows: the first term $\|\boldsymbol{A}\|_{1,q}$ encourages row-wise sparsity across all sensors, whereas the group regularizer defined by the second term tends to minimize the number of active groups in the same coefficient matrix $\boldsymbol{A}$. The third term penalizes the nuclear norm of the interference as

discussed in the previous section. In succession, the model enforces $\boldsymbol{A}$ to be both group-sparse and row-sparse, in parallel with extracting the low-rank interference appearing in all measurements altogether. Figure 3 shows the visual comparison of coefficient matrices of joint-sparsity structure and the combined group-and-joint sparsity model. Once the solutions of the coefficient matrix and low-rank term are recovered, the class label of $\boldsymbol{Y}$ is decided by the same Eq. 8 as in the JSR+L method.



(a) (b)

Fig. 3 a) Coefficient matrix of joint sparsity model and b) coefficient matrix of group-and-joint sparsity model

3.4 Algorithm

In this section, we provide a fast and efficient algorithm based on the ADMM[11] to solve for the proposed multisensor sparsity-based representation models. GJSR+L is the most general method; hence, here we discuss the algorithm to solve Eq. 9 and then simplify the algorithm to generate solutions for the other methods. The augmented Lagrangian function of Eq. 9 is defined as

$$\begin{aligned}\mathcal{L}(\boldsymbol{A},\boldsymbol{L},\boldsymbol{Z}) = \|\boldsymbol{A}\|_{1,q} \quad &+\lambda_G \sum_{c=1}^{C} \|\boldsymbol{A}_c\|_F + \lambda_L \|\boldsymbol{L}\|_* \\ &+ \langle \boldsymbol{Y} - \boldsymbol{D}\boldsymbol{A} - \boldsymbol{L}, \boldsymbol{Z}\rangle + \frac{\mu}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{A} - \boldsymbol{L}\|_F^2 ,\end{aligned} \tag{10}$$

where $\boldsymbol{Z}$ is the Lagrangian multiplier, and μ is a positive penalty parameter. The algorithm, formally presented in Algorithm 1, then minimizes $\mathcal{L}(\boldsymbol{A},\boldsymbol{L},\boldsymbol{Z})$ with respect to one variable at a time by keeping others fixed and then updating the variables sequentially.

Inputs: Matrices $\boldsymbol{Y}$ and $\boldsymbol{D}$, weighting parameters λ_G and λ_L.
Initializations: $\boldsymbol{A}_0 = \boldsymbol{0}$, $\boldsymbol{L}_0 = \boldsymbol{0}$, $j = 0$.
While not converged **do**
1. Solve for $\boldsymbol{L}_{j+1}$: $\boldsymbol{L}_{j+1} = \text{argmin}_{\boldsymbol{L}} \mathcal{L}(\boldsymbol{A}_j, \boldsymbol{L}, \boldsymbol{Z}_j)$
2. Solve for $\boldsymbol{A}_{j+1}$: $\boldsymbol{A}_{j+1} = \text{argmin}_{\boldsymbol{A}} \mathcal{L}(\boldsymbol{A}, \boldsymbol{L}_{j+1}, \boldsymbol{Z}_j)$
3. Update the multiplier: $\boldsymbol{Z}_{j+1} = \boldsymbol{Z}_j + \mu(\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{A}_{j+1} - \boldsymbol{L}_{j+1})$
4. $j = j + 1$.
end while
Outputs: $(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{L}}) = (\boldsymbol{A}_j, \boldsymbol{L}_j)$.

Algorithm 1. ADMM for GJSR+L

Algorithm 1 involves 2 main subproblems to solve for the intermediate minimizations with respect to variables $\boldsymbol{L}$ and $\boldsymbol{A}$ at each iteration j, respectively. The first optimization subproblem that updates variable $\boldsymbol{L}$ can be recast as

$$\begin{aligned} \boldsymbol{L}_{j+1} &= \underset{\boldsymbol{L}}{\text{argmin}}\, \lambda_L \|\boldsymbol{L}\|_* + \frac{\mu}{2} \left\| \boldsymbol{L} - (\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{A}_j - \frac{\boldsymbol{Z}_j}{\mu}) \right\|_F^2 \\ &= \underset{\boldsymbol{L}}{\text{argmin}}\, \lambda_L \|\boldsymbol{L}\|_* + \frac{\mu}{2} \|\boldsymbol{L} - \boldsymbol{P}_j\|_F^2 , \end{aligned} \tag{11}$$

where we define $\boldsymbol{P}_j = (\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{A}_j - \frac{1}{\mu}\boldsymbol{Z}_j)$. The proximal minimization in Eq. 11 can be solved via the singular value thresholding operator[21] in which we first define a singular value decomposition $(\boldsymbol{U}, \boldsymbol{\Delta}, \boldsymbol{V}) = svd(\boldsymbol{Y} - \boldsymbol{P}_j + \frac{1}{\mu}\boldsymbol{Z}_j)$.The intermediate solution of $\boldsymbol{L}_{j+1}$ is then determined by applying the soft-thresholding operator to the singular values: $\boldsymbol{L}_{j+1} = \boldsymbol{U} S_{\frac{\lambda_L}{\mu}}(\boldsymbol{\Delta})\boldsymbol{V}$, where the soft-thresholding operator of $\boldsymbol{\Delta}$ over $\frac{\lambda_L}{\mu}$ is element-wise defined for each δ in the diagonal of $\boldsymbol{\Delta}$ as $S_{\frac{\lambda_L}{\mu}}(\delta) = max(|\delta| - \frac{\lambda_L}{\mu}, 0)\, sgn(\delta)$.

The second subproblem to update $\boldsymbol{A}$ can be rewritten as

$$\boldsymbol{A}_{j+1} = \underset{\boldsymbol{A}}{\text{argmin}} \|\boldsymbol{A}\|_{1,q} + \lambda_G \sum_{c=1}^{C} \|\boldsymbol{A}_c\|_F + \frac{\mu}{2} \left\| \boldsymbol{D}\boldsymbol{A} - (\boldsymbol{Y} - \boldsymbol{L}_{j+1} + \frac{1}{\mu}\boldsymbol{Z}_j) \right\|_F^2 . \tag{12}$$

This subproblem is a convex utility function. Unfortunately, its closed-form solution is not easily determined. The difficulties come from not only the joint regularization of row-sparse and group-sparse on the variable $\boldsymbol{A}$, but also the operation over dictionary transformation $\boldsymbol{D}\boldsymbol{A}$ as well as the engagement of multiple modalities. In order to tackle these difficulties, we do not solve for an exact solution of Eq. 12.

Instead, the third term in the objective function is approximated by its Taylor series expansion at $\boldsymbol{A}_j$ (achieved from iteration j) up to the second derivative order

$$\begin{aligned}\left\|\boldsymbol{D}\boldsymbol{A}-(\boldsymbol{Y}-\boldsymbol{L}_{j+1}+\tfrac{1}{\mu}\boldsymbol{Z}_j)\right\|_F^2 \;&\approx \left\|\boldsymbol{D}\boldsymbol{A}_j-(\boldsymbol{Y}-\boldsymbol{L}_{j+1}+\frac{1}{\mu}\boldsymbol{Z}_j)\right\|_F^2 \\ &\quad +2\left\langle \boldsymbol{A}-\boldsymbol{A}_j,\boldsymbol{T}_j\right\rangle+\frac{1}{\theta}\left\|\boldsymbol{A}-\boldsymbol{A}_j\right\|_F^2,\end{aligned} \tag{13}$$

where θ is a positive proximal parameter and $\boldsymbol{T}_j=\boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{A}_j-(\boldsymbol{Y}-\boldsymbol{L}_{j+1}+\frac{1}{\mu}\boldsymbol{Z}_j))$ is the gradient at $\boldsymbol{A}_j$ of the expansion. The first component in the right-hand side of Eq. 13 is constant with $\boldsymbol{A}$. Consequently, by replacing Eq. 13 into the subproblem Eq. 12 and manipulating the last 2 terms of Eq. 13 into one component, the optimization to update $\boldsymbol{A}$ can be simplified to

$$\boldsymbol{A}_{j+1}=\underset{\boldsymbol{A}}{\operatorname{argmin}}\left\|\boldsymbol{A}\right\|_{1,q}+\lambda_G\sum_{c=1}^{C}\left\|\boldsymbol{A}_c\right\|_F+\frac{\mu}{2\theta}\left\|\boldsymbol{A}-(\boldsymbol{A}_j-\theta\boldsymbol{T}_j)\right\|_F^2. \tag{14}$$

The derivation in the second line of Eq. 14 is again based on the separable structure of $\|\cdot\|_F^2$ with $\boldsymbol{T}_j=[\boldsymbol{T}_j^1,\boldsymbol{T}_j^2,...,\boldsymbol{T}_j]$. Note that while $\|\cdot\|_F^2$ has element-wise separable structure promoting both row- and column-separable properties, the norm $\|\cdot\|_F$ does not perform any separable structure and $\|\cdot\|_{1,q}$ has separable structure with respect to rows (i.e., $\|\boldsymbol{A}\|_{1,q}=\sum_{c=1}^{C}\|\boldsymbol{A}_c\|_{1,q}$ with $\boldsymbol{A}$ being the row-concatenation matrix of all $\boldsymbol{A}_c$'s). Applying this row-separable property into the first and third terms of Eq. 14, we can further simplify it to solve for the subcoefficient matrix of each class separately:

$$\begin{aligned}(\boldsymbol{A}_{j+1})_c \;&=\underset{\boldsymbol{A}_c}{\operatorname{argmin}}\left\|\boldsymbol{A}_c\right\|_{1,q}+\lambda_G\left\|\boldsymbol{A}_c\right\|_F+\frac{\mu}{2\theta}\left\|\boldsymbol{A}_c-((\boldsymbol{A}_j)_c-\theta(\boldsymbol{T}_j)_c)\right\|_F^2 \\ &\qquad (\forall c=1,2,...,C).\end{aligned} \tag{15}$$

The explicit solution of Eq. 15 can then be solved via the following lemma.

Lemma 1*: Given a matrix* $\boldsymbol{R}$*, the optimal solution to*

$$\underset{\boldsymbol{X}}{min}\;\alpha_1\left\|\boldsymbol{X}\right\|_{1,q}+\alpha_2\left\|\boldsymbol{X}\right\|_F+\frac{1}{2}\left\|\boldsymbol{X}-\boldsymbol{R}\right\|_F^2 \tag{16}$$

is the matrix $\hat{\boldsymbol{X}}$

$$\hat{\boldsymbol{X}}=\begin{cases}\frac{\|\boldsymbol{S}\|_F-\alpha_2}{\|\boldsymbol{S}\|_F}\boldsymbol{S} & \textit{if } \|\boldsymbol{S}\|_F>\alpha_2\\ \boldsymbol{0} & \textit{otherwise,}\end{cases} \tag{17}$$

where the i-th row of $\boldsymbol{S}$ is given by

$$\boldsymbol{S}_{i,:} = \begin{cases} \frac{\|\boldsymbol{R}_{i,:}\|_q - \alpha_1}{\|\boldsymbol{R}_{i,:}\|_q} \boldsymbol{R}_{i,:} & \textit{if } \|\boldsymbol{R}_{i,:}\|_q > \alpha_1 \\ \boldsymbol{0} & \textit{otherwise.} \end{cases} \tag{18}$$

Furthermore, Algorithm 1 is guaranteed to provide the global optimum of the convex program in Eq. 9 as stated in the following theorem:

Theorem 1*: If the proximal parameter θ satisfies the condition $\sigma_{max}((\boldsymbol{D})^T\boldsymbol{D}) < \frac{1}{\theta}$, where $\sigma_{max}(\cdot)$ is the largest eigenvalue of a matrix, then $\{\boldsymbol{A}_j, \boldsymbol{L}_j\}$ generated by algorithm 1 for any value of the penalty coefficient μ converges to the optimal solution $\{\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{L}}\}$ of Eq. 9 as $j \to \infty$.*

4. Experimental Results

4.1 Data Set Description

In this section, we perform experimental results for a challenging classification of transient acoustic signals produced by various impulsive sources such as detonations (e.g., launch and impact) of different kinds of weapons collected by ground-based tetrahedral microphone arrays (hence there are 4 measurements for each data sample) at separate locations and on different dates and times at the sampling rate of 1001.6 Hz. This classification is difficult since many factors affect the propagation of signals from source to sensors. Significant noises and external interfering signals may also be present in the data jungle. In particular, we note several challenges: 1) the long distances between receivers and sources; 2) the presence of obstacles on the terrain and nature of the ground; 3) the various amplitudes of the sources; 4) the different times of day; and 5) the meteorological conditions (e.g., cloud cover, wind, and humidity). Several preliminary classification results have been reported using well-known conventional classifiers, such as Markov switching vector auto-regression (MSVAR), Gaussian mixture model (GMM), support vector machine (SVM), hidden Markov model (HMM), or the combination of SVM and HMM (so-termed SVM-HMM or SHMM).

The given original data set contains a total of 7,420 acoustic signatures and their associated metadata. After a clean-up round that exhaustively investigates and compares the true signal contents, it is revealed that there are 2,088 duplicate samples. Moreover, some of the signatures in the data set are unusable for various reasons,

such as 1) the recording equipment on some signals only has outputs from one sensors; 2) some data files contain incomplete events or inaccurate information (e.g., the starting time of segmentation is even after its ending time); and 3) the amplitudes of certain recorded signals are too small. In total, another 1,188 files will not be used in our experiments; these files may be fixed and reused in our future works.

The 4,144 cleaned and unique files are reported as acoustic signatures of 9 different weapons recorded at 19 experiment fields in which the set of fields {02, 03, 06, 11–15} (so termed V1 fields) are categorized as desert vegetation, and fields {01, 04, 05, 07–10, 16–19} (known as V2 fields) are classified as grassland vegetation. In this report, we classify launches and impacts for 3 weapon types (namely, W1, W2, and W3), which account for a total of 2,950 samples. The availability of these data samples allows us to examine the effect of recording environments on the classification accuracy. The data set used in our experiments is summarized in Table 1, which shows the distribution of the number of signatures for each location/weapon type combination.

Table 1 Data set summary

Acoustic Data	W1	W2	W3	Total
Launch	1,719	106	107	**1,932**
Impact	918	36	64	**1,018**
Total	**2,637**	**142**	**171**	**2,950**

4.2 Comparison Methods

In this report, we compare the classification results of transient acoustic signals using 12 different methods:

- MSVAR: Markov switching vector autoregressive.[25]
- GMM: Gaussian mixture model.
- SLR: sparse logistic regression.
- HMM: hidden Markov model.
- SVM: support vector machine classifier (reporting results on the concatenated feature vector of the 4 sensors).

- SVM-HMM (or SHMM): combined SVM and HMM model where HMM is first developed to capture the temporal dynamics of time series of transient acoustic signals, and SVM is then embedded to model the emission probabilities for each HMM.[26]
- JSR: joint sparse representation model.
- GJSR: group-and-joint sparse representation. This model is simplified from GJSR+L model in which the low-rank interference term is set to be zero along the optimization procedure.
- JSR+L: joint sparse representation with low-rank interference.
- GJSR+L: group-and-joint sparse representation with low-rank interference model. This is the most general model among sparsity-based technique.
- DNN: deep neural network.
- DBN: deep belief network.

As mentioned previously, among the classification methods above, MSVAR, GMM, HMM, SVM, and SVM-HMM have been previously studied for the acoustic transients data set. Furthermore, besides sparsity-based representation techniques, we also develop and compare the classification results of SLR as well as 2 deep learning models, arguably as among the most advanced discrimination techniques in recent years.

Sparse logistic regression. The main idea of this model is to associate to all the training data in the training dictionary $\boldsymbol{D} \in \mathbb{R}^{N\times P}$ an appropriate coefficient vector $\boldsymbol{a} \in R^N$, and the logistic loss is taken over the sum of training samples. Sparsity regularization can be incorporated into the optimization to retrieve the coefficient vectors $\boldsymbol{a}$. This regularization implies that some features are assumed to be irrelevant for classification, thus they should be minimized via the ℓ_1-norm sparsity constraint. In particular, the probabilistic model of the logistic regression[27] is formulated as follows:

$$\mathbb{P}(y_j|\boldsymbol{d}_j) = \frac{\exp(\boldsymbol{d}_j^T\boldsymbol{a}y_j)}{1+\exp(\boldsymbol{d}_j^T\boldsymbol{a})}, \tag{19}$$

where the feature vector $\boldsymbol{d}_j$ is the j-th column vector of the training dictionary $\boldsymbol{D}$, and the index y_j receiving values 0 or 1 indicates the class label.

To recover $\boldsymbol{a}$'s, a maximum likelihood over all the training data is applied together with imposing sparsity on $\boldsymbol{a}$. In particular, the following function is minimized:

$$\min_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}) + \lambda \left\|\boldsymbol{a}\right\|_1, \tag{20}$$

where the loss function $\mathcal{L}(\boldsymbol{a})$ is defined as

$$\mathcal{L}(\boldsymbol{a}) = \sum_{j=1}^{P} y_j (\boldsymbol{d}_j)^T \boldsymbol{a} - \sum_{j=1}^{P} \log(1 + \exp(\boldsymbol{d}_j^T \boldsymbol{a})). \tag{21}$$

Once the coefficient vector $\boldsymbol{a}$ for each training sample is obtained, each measurement of the test sample is then assigned to a class, and the final decision is made by selecting the label that occurs most frequently.

Deep learning models. Recently, deep learning or deep networks have increasingly attracted the interest of researchers in various diverse disciplines and have been applied to many fields like computer vision, natural language processing, automatic speech recognition, and bioinformatics. Remarkably, deep-learning-based techniques have been shown to produce state-of-the-art results on various discrimination tasks.[28,29] However, it has been argued that in order to perform well, these techniques normally require a very large and diversified training data set. Experiments with deep learning architectures are critical in helping us comprehensively compare the classification performance with state-of-the-art techniques.

In this report, we develop 2 deep learning architectures, namely, deep neural network (DNN) and deep belief network (DBN), to perform classification on the transient acoustic data set. A general deep network architecture is an artificial neural network with a number of hidden layers between its inputs and its outputs (Fig. 4). DNNs are feed-forward networks whose hidden layers can be hierarchically trained by backpropagating derivatives of a cost function that measures the discrepancy between the target outputs and the actual outputs produced for each training case.[30] On the other hand, DBNs can be formed by stacking multiple layers of restricted Boltzmann machines (RBMs), which are trained in a greedy fashion,[31,32] and the results are fine-tuned using backpropagation. An RBM is a generative undirected graphical model that learns connections between a layer of visible units (represent-

ing stochastic binary input data) and a layer of stochastic binary hidden units with the restriction that no visible-visible or hidden-hidden units are connected.

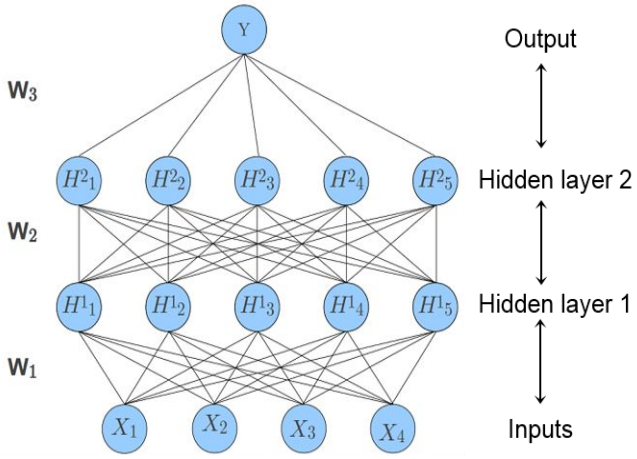


Fig. 4 A general deep network architecture.

4.3 Experimental Setups

4.3.1 Segmentation and Feature Extraction

To accurately perform classification, it is necessary to extract the actual events from collected signals. A recorded signal can last several seconds. However, the event occurs within a small interval, typically of half a second duration or less. After segmentation, the cepstral features[33] are extracted in each segment. Cepstral features have been proven to be very effective in speech recognition and acoustic signal classification. The power cepstrum of a signal $y(t)$, formulated as $\left|F^{-1}(\log_{10}(|F(y(t))|^2))\right|^2$ where F is a Fourier transform,[34] captures the rate of signal change with time with respect to a different frequency band. We discard the first cepstral coefficient (corresponding to the zero-frequency component) and use the next 50 coefficients for classification.

4.3.2 Training and Testing Splits

We evaluated the performance of different training and testing splitting frameworks:

- Randomly splitting the whole data set into training and testing sets with an equal amount of samples (i.e., half for training and half for testing). This is the optimal setup when training samples of diversified battlefields are available.

This means that the results of this setting should deliver the upper-bound results in real-world practical applications. Most of the published experiments so far have been evaluated on this half-half splitting setup.

- Splitting the training and testing sets by opposite vegetation environments. This experimental setup consists of 2 different scenarios that perform classification from samples of V1 vegetation (i.e., desert vegetation) as training and V2 vegetation (i.e., grassland vegetation) as testing and vice versa. These are viewed as pessimistic setups (corresponding to lower-bound results) since training and testing samples are completely nonoverlapped in date, time, location, experiment field, and vegetation environment.

4.3.3 Data Imbalance

As can be seen from Table 1, there is a severe class imbalance among the number of samples in different projectiles. More specifically, the number of W1 samples equals 89.4% of the total amount of available signatures while W2 and W3 samples account for only 4.8% and 5.8% of all signatures, respectively. This means that there are at least 15 times more signatures available for W1 than for the other 2 weapons. Moreover, while the imbalance between data samples of the 2 actions, launch and impact, is less serious compared with that of weapon types, the collection of launch actions is still almost twice that of the impact signatures. In total, the number of signatures coming from the largest-sized class (i.e., W1-launch class) is approximately 48 times more than the smallest counterpart (i.e., W2-impact class).

This imbalance property of the data set is a critical problem that influences the classification results of all methods since a test sample is more likely to be categorized into majority classes which have more volume with high density and contain more diversified information. Particularly, this problem seriously affects the results of sparsity-based techniques because the class assignment steps of these methods are decided by the coefficients vectors, which lack balance in sizes among data samples in the training classes, hence their outputs are greatly asymmetric within different classes. Therefore, for the 4 sparsity-based methods, we process a balancing step to balance data samples in the training set to the median number of all class sizes. This step includes shrinking the majority classes by randomly under-sampling to the median size and expanding the minority classes by randomly over-sampling to the same class size. This balancing strategy, despite generating over-fitting due to

multiple tied instances in minority classes and loss of features in majority classes, provides more equally distributed classification accuracy rates among classes on sparsity-based methods. On the other hand, the data balancing may slightly vary the performance of the other classification techniques but would not show systematic changes on the overall results. Therefore, in this report, all results generated by sparsity-based models are performed with balancing training samples while those of the other methods are reported with the full data set.

4.4 Comparison of Classification Performance

In this section, we perform extensive experimental results on the classification of the transient acoustic data set for 2 main classification problems:

- Four-class problem to classify W1-launch, W1-impact, W2-launch, and W2-impact
- Six-class problem to classify W1-launch, W1-impact, W2-launch, W2-impact, W3-launch, and W3-impact

Note that a W3 munition is a special type of W2. Therefore, in the 4-class problem, we combine the signatures of W2 and W3 weapon types and just consider them as W2 while the 6-class problem further discriminates between W2 and W3 weapons.

4.4.1 Classification Results for 4-class Problem

Now we demonstrate the classification results for the 4-class classification problem, particularly in discriminating between the launches and impacts of W1 and W2 projectiles. The detail confusion matrices of all competing methods for the 3 splitting setups—1) half-half, 2) V1 as training and V2 as testing, and 3) V2 as training and V1 as testing—are visualized in Figs. 5–7, respectively. The heights of all bars are normalized with respect to the actual number of testing samples in that class (i.e., each row is normalized to have sum to unity). Furthermore, the black number on the top of each bar shows the true number of samples that are classified into the associated class while the red number displays the percentage of the number of predicted samples over the total number of testing samples in that class. The results on the diagonal of each matrix (surrounded by rectangles drawn with solid red lines) demonstrate accurate classification rates of the 4 classes. Also, the results from the 4 sparsity-based techniques are surrounded by rectangles drawn with solid blue lines.

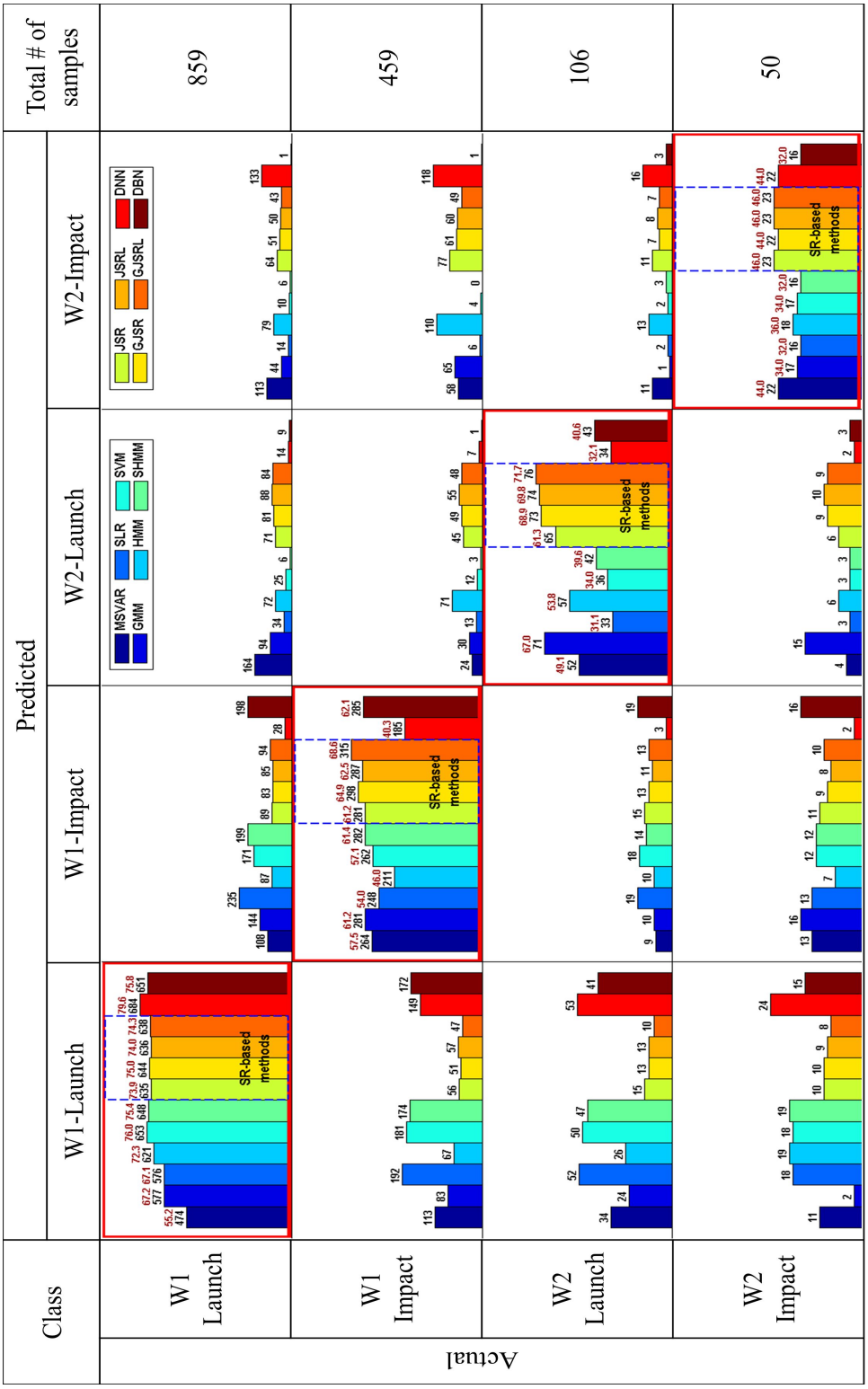


Fig. 5 Comparison of 4-class confusion matrices with random half-half separation of training and testing sets

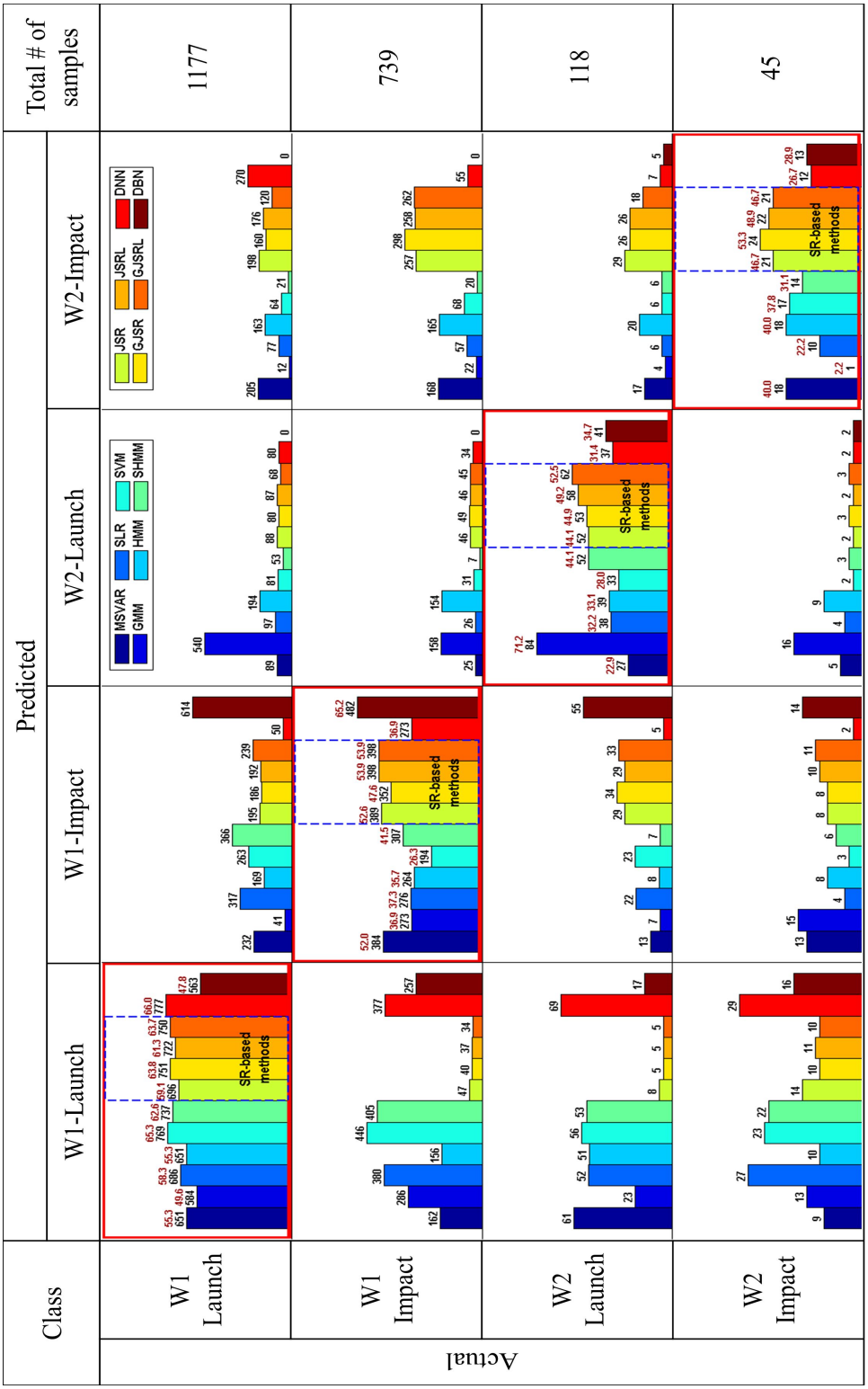


Fig. 6 Comparison of 4-class confusion matrices, V1 as training and V2 as testing

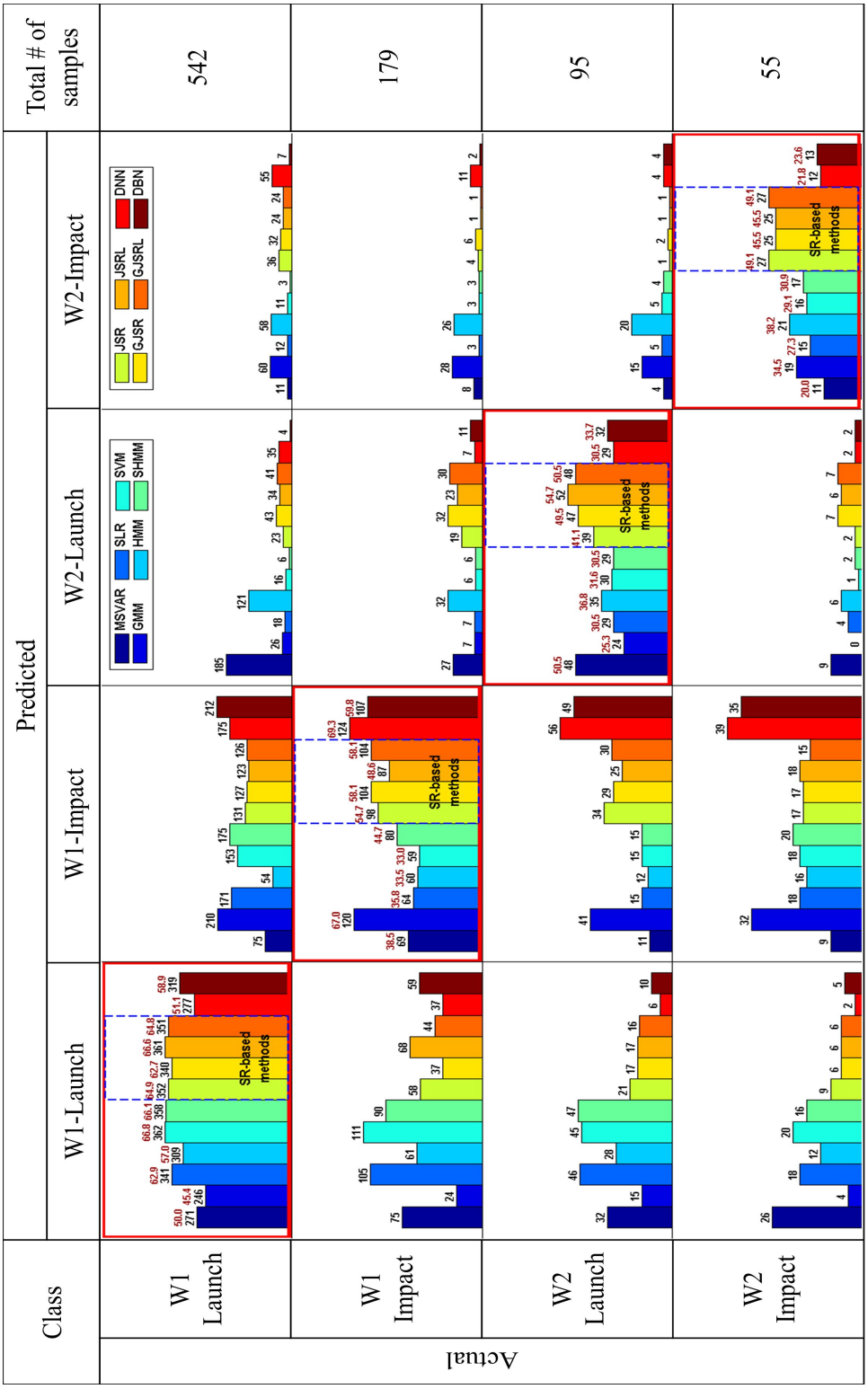
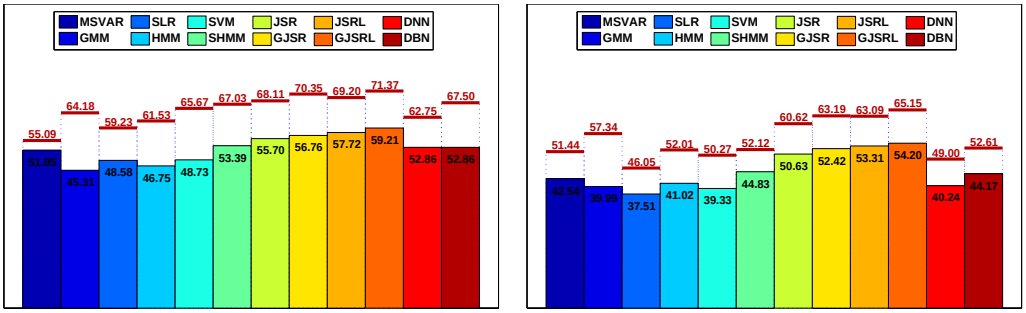


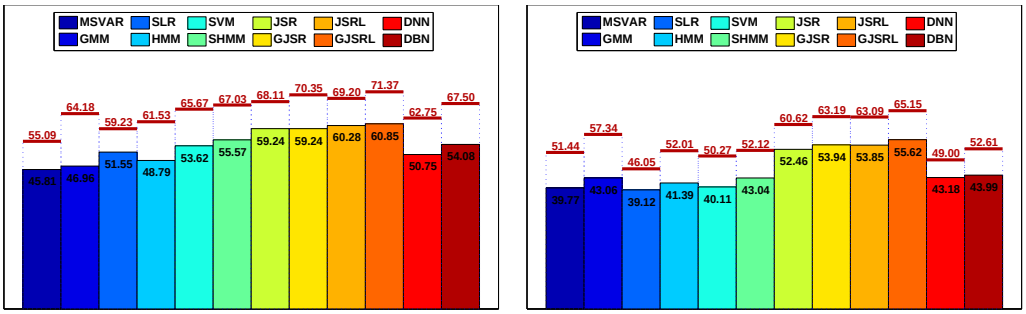
Fig. 7 Comparison of 4-class confusion matrices, V2 as training and V1 as testing

Figures 8 and 9 summarize the classification performance of all comparison methods for the 2 splitting setups: V1 as training and V2 as testing (Fig. 8) and V2 as training and V1 as testing (Fig. 9). We report both weighted results in relation to the actual class sizes and non-weighted results, which simply average the classification rates of the 4 classes without taking into account the class sizes. In addition, the results of each opposite-vegetation splitting setup, displayed in black percentage numbers, are compared to those of the optimistic half for training and half for testing setup, exhibited by red numbers. The classification rates of the half-half setting consistently exceed those of the opposite-vegetation splits by around 10%.



(a) (b)

Fig. 8 Comparison of classification performance of 4-class problem with V1 as training and V2 as testing vs. half-half split of training and testing sets: a) weighted results and b) non-weighted results



(a) (b)

Fig. 9 Comparison of classification performance of 4-class problem with V2 as training and V1 as testing vs. half-half split of training and testing sets: a) weighted results and b) non-weighted results

Both Figs. 8 and 9 show that sparsity-based techniques generally perform among the best, in which the GJSR+L model always yields the best classification rates for both weighted average and non-weighted average in all 3 splitting setups, il-

lustrating the efficacy of its group-and-joint sparsity structure as well as low-rank interference. On the other hand, deep network architectures provide rates similar to traditional classification techniques and are surpassed by GJSR+L by more than 6% in weighted results and 10% in non-weighted results. Among the 2 deep learning methods, DBN is slightly superior to DNN in all cases. The main reasons for the moderate results of these techniques are the limitation and nondiversity of the data set as well as the harmful effects of noise and/or external interferences, such as the presence of propagation effects in recorded signals.

4.4.2 Classification Results for 6-class Problem

In this section, we further compare the performance of the 12 competing classification techniques on the 6-class problem which discriminates between 2 detonations (launches and impacts) of 3 projectiles. The detail confusion matrices for the 3 training-testing splitting scenarios are exhibited in detail in Figs. 10–12, and their weighted and non-weighted classification results are encapsulated in Figs. 13 and 14. We can again observe similar performance orders among classification techniques in competition, in which GJSR+L and other sparsity-based models are superior to conventional classifiers as well as the 2 deep network architectures. The training data samples are even more limited than the 4-class problem case in some classes. Specifically, no more than 25 training samples are available in the 4 classes W2-launch, W2-impact, W3-launch, and W3-impact for the V2 as training and V1 as testing split, in which the W3-impact class only contains 9 training signatures.

In addition, we compare the classification rates of 4-class and 6-class problems in Fig. 15, where the results are averaged over the 3 splitting setups. The black numbers display the classification rates in percentage resulted by 6-class problem set while the red numbers display outputs of the 4-class case. The performance discrepancies between the 2 problems are varied by methods but are typically small and stay in the range of 2%–4%. This means that W3 signatures contain significantly discriminative features with signals of other W2-weapon types, hence should be categorized in a separated class.

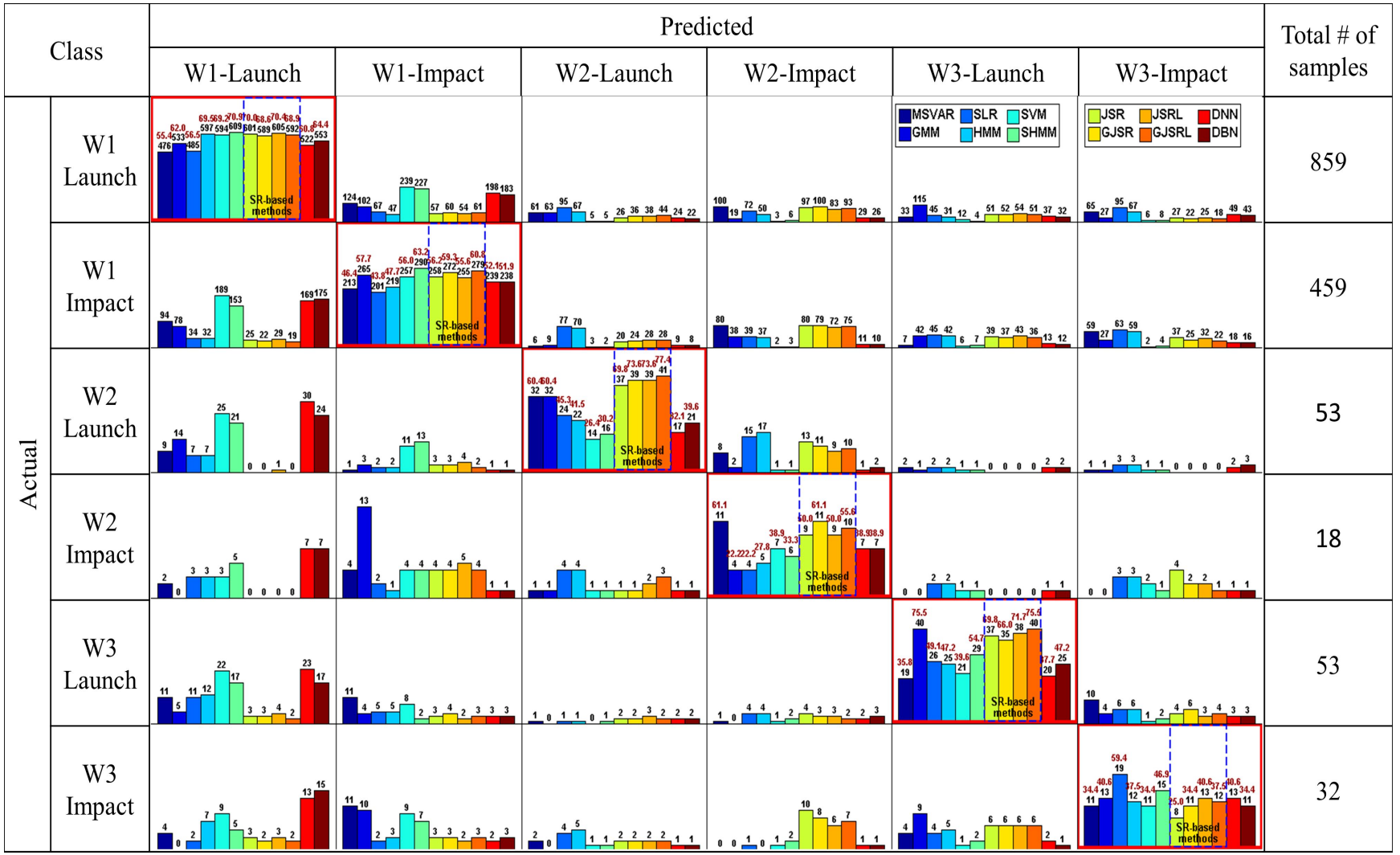


Fig. 10 Comparison of 6-class confusion matrices with random half-half separation of training and testing sets

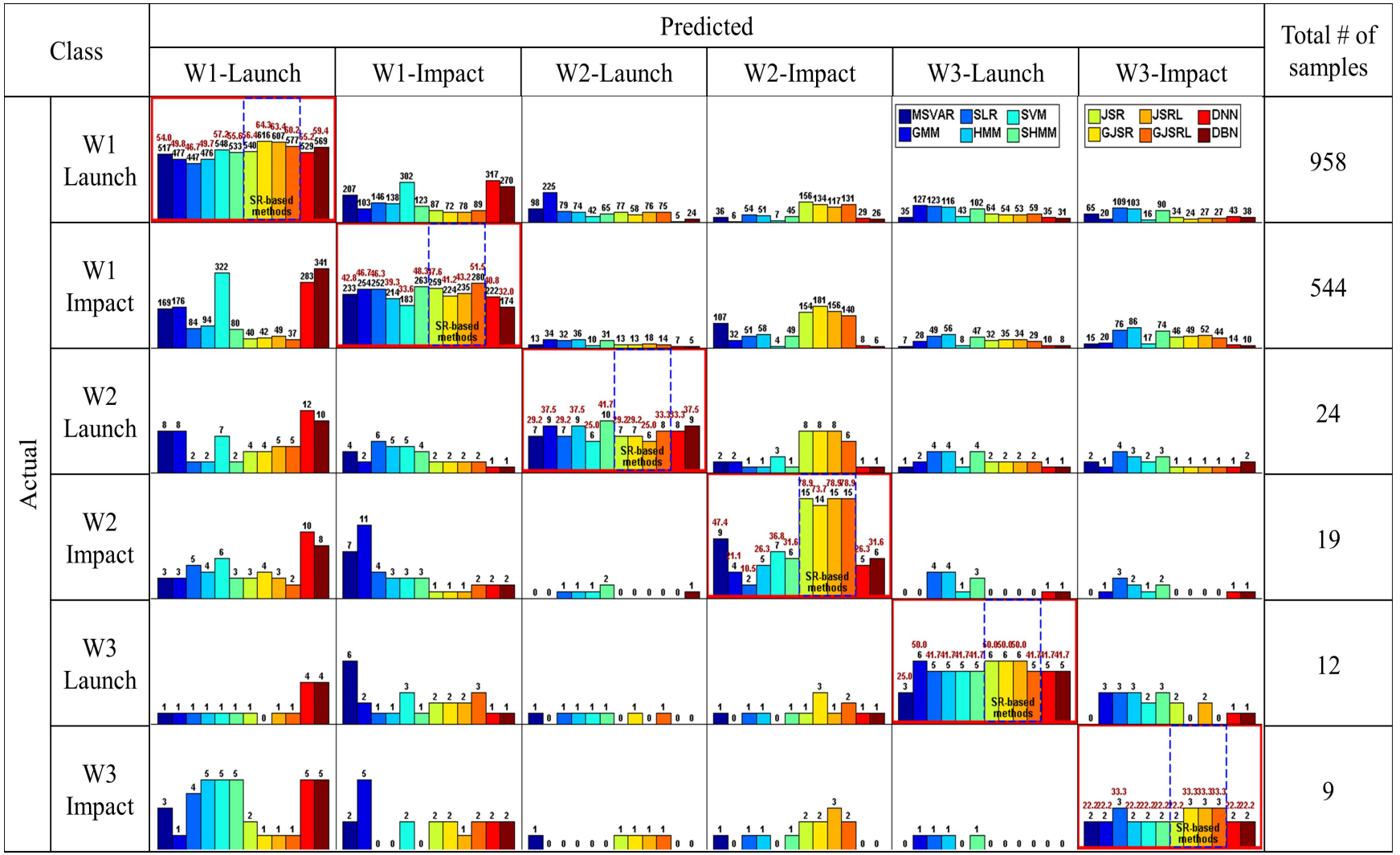


Fig. 11 Comparison of 6-class confusion matrices, V1 as training and V2 as testing

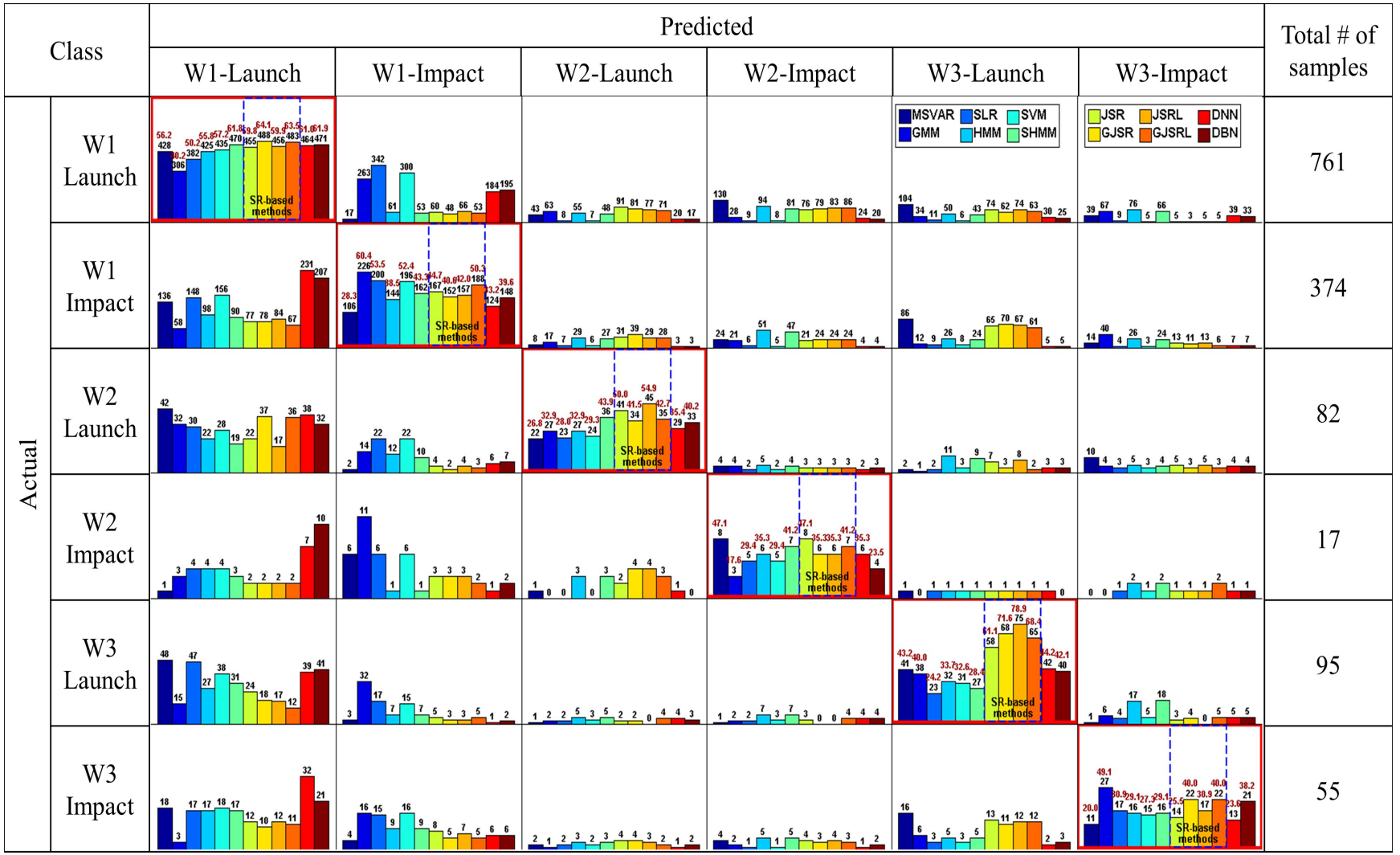
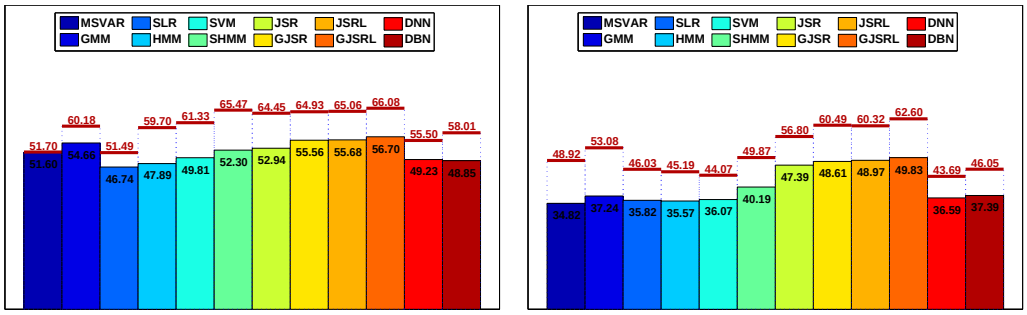
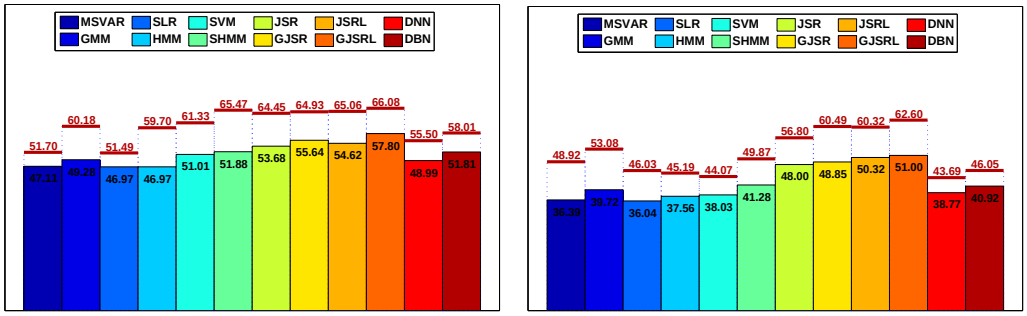


Fig. 12 Comparison of 6-class confusion matrices, V2 as training and V1 as testing



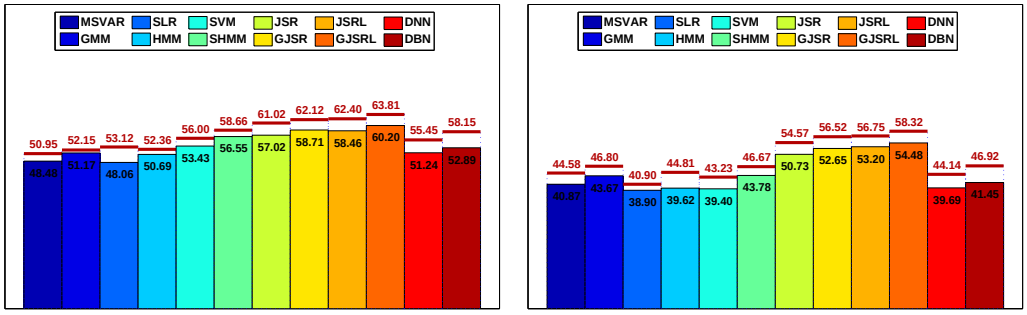
(a) (b)

Fig. 13 Comparison of classification performance of 6-class problem with V1 as training and V2 as testing vs. half-half split of training and testing sets: a) weighted results and b) non-weighted results



(a) (b)

Fig. 14 Comparison of classification performance of 6-class problem with V2 as training and V1 as testing vs. half-half split of training and testing sets: a) weighted results and b) non-weighted results



(a) (b)

Fig. 15 Comparison of classification results of 4-class problem vs. 6-class problem: a) non-weighted results and b) weighted results

4.5 Discussions on Classification Strengths and Weaknesses.

Section 4.3.3 has provided a comprehensive comparison on the performance of the 12 classification methods. In this section, we will discuss general strengths and weaknesses of the classifiers under examination, focusing on both performance and complexity. From the performance perspective, the results from the previous section demonstrate the superior performance of sparsity-based models. On the negative

side, however, these methods are expensive in both training and testing procedures. Furthermore, the results are highly dependent on the weighting parameters, such as λ_L and λ_G, which encode low-rank and group-structure information.

Table 2 summarizes the advantages and disadvantages in terms of complexity and classification accuracy of the 12 classification methods. This does not compare methodology in general; instead, the classifiers are only evaluated based on their specific performance on the transient acoustic data set.

Table 2 Advantages and disadvantages of the competing methods

Methods	Strengths	Weaknesses	Other comments
MSVAR	Quite low computational time for testing.	Requires multiple predefined parameters of transition stages.	MSVAR was previously studied and is a good reference for performance comparison.
GMM	(i) Simple and easy to implement; (ii) low computation and small memory allocation.	Requires predefining the number of mixtures of the model.	GMM is a simple model that can yield generally good results.
SLR	Training time is generally fast.	Classification accuracy is quite low.	This method is moderate in both performance and computation.
SVM	Supporting kernels, hence can model nonlinear relations.	Expensive in both training/testing time and memory usage.	This is a typical deterministic, discriminative model.
HMM	Low complexity in both running time and memory usage.	Requires predefining the number of stages and number of mixtures in each state.	This is a typical generative, probabilistic model.
SVM-HMM	(i) Good classification performance; (ii) testing time is generally fast.	Long training time.	Performance is second best after sparsity-based methods while requiring much less computational time.
Sparsity-based methods	(i) Highest classification performance among competing methods; (ii) robust with noise/interference; (iii) low memory allocation for testing.	(i) Very long testing time; (ii) results are dependent on parameter selection; (iii) require extensive cross-validation computation for parameter learning.	The 4 sparsity-based methods provide slight variance on classification results and computational complexity, in which JSR has lowest classification rates but is least dependent on parameter selections while GJSR+L performs the best but is most dependent on parameter selections.
Deep Network Architectures	Fast testing time.	(i) Long training procedure; (ii) require large training set.	Deep learning methods do not perform very well on the available data set. However, the performance may significantly improve if many more training samples are available.

Furthermore, we demonstrate the computational complexity of the 12 methods, testing for the 4-class problem and half-half split in Table 3. In particular, Table 3 exhibits the running time in milliseconds and the maximum memory allocation in kilobytes calculated specifically for the classification functions of one testing sample. All the calculations are conducted and timed on the same desktop with an Intel quad-core 3.60-GHz CPU that has 16-GB memory, running Windows 7 and Matlab version 8.3.0. Note that these running times would depend on the parameters chosen (e.g., number of states, correlation order). A detailed study of speed for each algorithm is outside the scope of this report. In addition, it might not provide the final word in the context of continuing hardware progress. It is also worth mentioning that these numbers are for the testing side after data training and parameter learning have already been processed. Moreover, while the training complexity is difficult to quantify, its general analysis has been provided in Table 2.

Table 3 Computational complexity and memory usage comparison

Methods	Running time per testing sample (ms)	Maximum memory allocation (KBs)
MSVAR	18.2	29,742
GMM	9.4	5,566
SLR	55.4	128,262
HMM	5.2	11,506
SVM	181.2	220,386
SVM-HMM	20.7	116,714
JSR	938	2,096
GJSR	1,062	2,122
JSR+L	1,286	2,138
GJSR+L	1,308	2,264
DNN	0.7	4,838
DBN	0.6	6,972

5. Conclusions and Future Work

In this technical report, we have proposed a general sparsity-based framework for the classification of transient acoustic signals; this framework enforces various sparsity structures like joint-sparse or group-and-joint-sparse within measurements of multiple acoustic sensors. We further robustify our models to deal with the presence of dense and large but correlated noise and signal interference (i.e., low-rank interference). Another contribution is the implementation of deep learning archi-

tectures to classify the transient acoustic data set. Extensive experimental results are included in the report to compare the classification performance of sparsity-based and deep-network-based techniques with conventional classifiers, such as MSVAR, GMM, SVM, HMM, SLR, and the combined HMM-SVM methods for 2 experimental sets of 4-class and 6-class classification problems. Based on relative performance and overall computational requirements, we would pick the JSR as a stop-gap solution. Its performance is among the leading pack while its training and testing time is moderate. However, the reality is that all classifiers drop around 10% in accuracy when tested on unseen environments. This is clearly a fundamental problem that is still open. Much work remains to be done, as is discussed next.

In the future, we plan to work on 1) searching for invariant features in the z-domain; 2) developing a collaborative multi-array multi-sensor classification framework, which takes into consideration the correlations as well as complementary information among data samples of a single event recorded by multiple sensor arrays at different locations; 3) learning dictionary instead of using an off-the-shelf dictionary to improve discrimination characteristics; 4) using online/on-the-fly dictionary update; and 5) studying unsupervised transfer learning methods to exploit available information from unlabeled data samples and thus further improve classification results. The detailed approaches are presented in Subsections 5.1–5.5. Furthermore, some other tasks under our future investigations include 6) re-running experiments on the full and clean data set; 7) developing more robust deep network methods; 8) learning about other invariant feature spaces, such as the symbolic dynamic filtering features[35] that can effectively encapsulate time-series information; and 9) publishing results in journals and/or present at conferences in acoustics fields.

5.1 Invariant Features Search in the Z-Domain

A fundamental obstacle to long-range classification is the effect of the propagation channel on the acoustic signal. Under the linear regime, it is reasonable to assume that the channel is linear time-invariant. We can further assume that the signals themselves, after propagating far enough from the point of explosion, are small enough to be in the linear regime and can be modeled by a linear time-invariant system.

Under these assumptions, the measurements can be modeled in the z-domain as

$$X(z) = H(z)M(z) + N(z) = k\frac{\prod_{j=1}^{p} z - \chi_j}{\prod_{j=1}^{q} z - \pi_j}\frac{\prod_{j=1}^{m} z - z_j}{\prod_{j=1}^{n} z - p_j} + N(z), \qquad (22)$$

where $H(z)$ is the channel, $M(z)$ is the signal of interest, and $N(z)$ is the additive noise. χ_j and π_j are the zeros and poles of $H(z)$, and z_j and p_j are the zeros and poles of $M(z)$.

Clearly, the "natural" invariant features are the poles and zeros of $M(z)$, as any $H(z)$ will leave them unchanged, as long as there is no pole zero cancellation. To find them, one will need to remove the poles/zeros of $H(z)$ and $N(z)$, effectively, to factor out channel and signal. The answer is not obvious if one only measures $X(z)$. However, it is conceivable to use past noise measurements and multiple measurements of the same signal to estimate $M(z)$.

A simple approach is to estimate all the poles and zeros of $X(z)$ and compare the pole-zero pattern to known ones, using the known patterns as templates. There are many robust approaches to solving for the poles and zeros of $X(z)$. One can use the state-space realization from noisy impulse response.[36] Another option is to use the system identification approach. We have already implemented and tested the first approach. However, to keep the focus, we will report our preliminary results in another paper.

If accurate, a practical by-product is a pole-zero pattern that can be displayed to human operators, who could then use their own judgment to do the classification. The challenge with this approach is the order selection and unstable realization.

5.2 Collaborative Multi-array Multi-sensor Classification

In this report, we have theoretically and empirically demonstrated the effectiveness of incorporating correlation across different sensors attached to the same sensor array. In practice, multiple sensor arrays are stationed at different locations, concurrently listening to detonation events; hence, this information can be accommodated to further improve classification performance. In other words, we exploit not only correlation among sensors of the same array, but also complementary information across different sensor arrays. Mathematically, suppose there are K sensor arrays in the system, $\boldsymbol{Y}^k$ is the corresponding measurement matrix collected by the k-th

array ($k = 1, 2, ..., K$), and $\boldsymbol{Y} = [\boldsymbol{Y}^1, \boldsymbol{Y}^2, ..., \boldsymbol{Y}^K]$ is the concatenated matrix of all sensors and arrays. The GJSR+L model can then be extended to fuse multiple arrays for a collaborative classification as follow:

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{L}}{min} \quad & \sum_{k=1}^{K} \left\|\boldsymbol{A}^k\right\|_{1,q} + \lambda_G \sum_{k=1}^{K}\sum_{c=1}^{C} \left\|\boldsymbol{A}_c^k\right\|_F + \lambda_L \left\|\boldsymbol{L}\right\|_* \\ s.t. \quad & \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{A} + \boldsymbol{L}, \end{aligned} \tag{23}$$

or equivalently

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{L}}{min} \quad & \sum_{k=1}^{K} \left\|\boldsymbol{A}^k\right\|_{1,q} + \lambda_G \sum_{c=1}^{C} \left\|\boldsymbol{A}_c\right\|_F + \lambda_L \left\|\boldsymbol{L}\right\|_* \\ s.t. \quad & \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{A} + \boldsymbol{L}, \end{aligned} \tag{24}$$

where $\boldsymbol{A} = [\boldsymbol{A}_1; \boldsymbol{A}_2; ...; \boldsymbol{A}_C]$ is the row-concatenated coefficient matrix of different classes, $\boldsymbol{A} = [\boldsymbol{A}^1, \boldsymbol{A}^2, ..., \boldsymbol{A}^K]$ is the column-concatenated coefficient matrices of K sensor arrays, and $\boldsymbol{A}_c^k$ is the coefficient submatrix associated with class c and array k. The minimization of the first component in Eq. 24 can be phrased as enforcing the row-sparsity property within each sensor array while different arrays do not necessarily share common sparsity patterns. On the other hand, the group-sparsity function (i.e., the second term in Eq. 24) promotes a group structure within each array as well as across multiple arrays. Finally, a nuclear norm minimization is devoted to $\boldsymbol{L}$ to encourage low-rank property on the interference among all measurements.

5.3 Dictionary Learning for Sparse Coding on Acoustic Signals

The sparsity-based techniques proposed in this report have been based on the assumption that acoustic signatures usually lie in low-dimensional subspaces of a deterministic dictionary, which is constructed by directly concatenating the acquired training samples. However, it is probable that learning the dictionary instead of using off-the-shelf training samples will improve classification performance. It means that a learning procedure will be added to the training side where not only a sparsity constraint is enforced on the coefficient matrix, but also a dictionary is learned in parallel to increase the sparsity characteristic and better capture the discrimination property. This learned dictionary will then be used for the classification of testing samples instead of a deterministic dictionary. Given the training data $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_M]$, a general dictionary learning (DL) method is designed to si-

multaneously learn a dictionary $\boldsymbol{D}$ and the corresponding sparse coefficients $\boldsymbol{A}$ as follows[37]:

$$\min_{\boldsymbol{D},\boldsymbol{A}} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{A}\|_F^2 + \lambda \|\boldsymbol{A}\|_1 . \tag{25}$$

The nonconvex optimization problem in Eq. 25 is usually solved by iterating between sparse coding and dictionary updating. In the sparse coding stage, the sparse coefficient $\boldsymbol{A}$ is found with respect to a fixed dictionary $\boldsymbol{D}$. In the dictionary updating stage, each dictionary atom $\boldsymbol{d}_j$ in $\boldsymbol{D}$ is updated using only data with nonzero sparse coefficients on index j. This subproblem can be solved by either block coordinate descent[37] or singular value decomposition.[38] Furthermore, we propose to design the dictionary and sparse code with more discriminating properties by enforcing extra structural constraints $f_{\boldsymbol{A}}(\cdot)$ and $f_{\boldsymbol{D}}(\cdot)$, leading to

$$\min_{\boldsymbol{D},\boldsymbol{A}} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{A}\|_F^2 + \lambda_A f_{\boldsymbol{A}}(\boldsymbol{A}) + \lambda_D f_{\boldsymbol{D}}(\boldsymbol{D}), \tag{26}$$

where the structural-sparsity promoting function $f_{\boldsymbol{A}}(\cdot)$ enforces the correlation along multiple measurements, which can be element-wise-sparse, joint-sparse, or group-sparse functions as previously discussed in this report; and $f_{\boldsymbol{D}}(\cdot)$ forces the subdictionaries of different classes to be as incoherent as possible.[39] For simplicity, the presence of low-rank interference as well as the incorporation of multiple sensor arrays are omitted from the model description in Eq. 26.

5.4 Online Dictionary Learning Update

In the long run, it is desired to develop a system that can automatically update the dictionary with more and more training samples continuously collected on the battlefields (both labeled and unlabeled). For the sparsity models with deterministic dictionaries, the dictionary update procedure typically includes 2 steps: the first step selectively adds dictionary atoms collected on the fields and labels them to the corresponding classes, and the second step involves relearning the models' parameters via cross-validation technique. This normally requires very high computations, especially when the training set is large and thus is impractical to process in real time. Therefore, we propose a dynamic dictionary updating framework based on a DL approach that can capture the representation and the label of the signal on-the-fly.

Researchers have proposed to update online dictionaries by block-coordinate descent methods with warm restarts,[40] recursive least squares,[41] or an efficient feed-forward architecture.[42] Another related line of research is learning algorithms on

manifolds for independent component analysis. Researchers have developed efficient conjugate gradient or steepest descent algorithms leveraging differential geometry techniques.[43] Our proposed online dictionary update may include several approaches to remove the outdated dictionary atoms in a chosen subdictionary or add new dictionary atoms to capture more on-the-fly features. First, a linear dynamic system (LDS) model describing the signal evolution can be used to capture the transformation of the dictionary. This quadratic problem can be formulated as follows:

$$\min_{\boldsymbol{D}_t} \frac{1}{2}||\boldsymbol{Y}_t - \boldsymbol{D}_t\boldsymbol{A}_t||_F^2 + \gamma||\boldsymbol{D}_t - \boldsymbol{B}_t\boldsymbol{D}_{t-1}||_F^2 + \lambda_D f_{\boldsymbol{D}}(\boldsymbol{D}), \tag{27}$$

where $\boldsymbol{D}_t$ is the updated dictionary at time t, $\boldsymbol{B}_t$ is used to capture the subspace deformation, and γ is a trade-off weighting parameter to balance the 2 terms. Here we only change the dictionary updating stage in the DL model in Eq. 26 while the sparse coding stage remains unchanged as described in Section 5.

Another approach is to update the dictionary by descent on the oblique manifold.[44] The oblique manifold is the set of vectors with unit Frobenious norm, which is exactly the constraint typically enforced on DL. We propose to employ the gradient on the oblique manifold to dynamically update the dictionary, where the gradient can be derived as

$$\nabla f(\boldsymbol{D}_{t+1}) = P(f^{'}(\boldsymbol{D}_t)) = f^{'}(\boldsymbol{D}_t) - \boldsymbol{D}_t \text{ ddiag}(\boldsymbol{D}_t^T f^{'}(\boldsymbol{D}_t)). \tag{28}$$

Here, $P(\cdot)$ captures the orthogonal projection onto the tangent space of the dictionary manifold, $f(\cdot)$ represents the loss function, and ddiag$(\cdot)$ sets all off-diagonal entries of a matrix to zero. Notice that in our problem setup, the structure usually indicates the boundary of subspaces with different labels. Therefore, we will further incorporate this information into the dictionary update procedure to improve performance.

5.5 Unsupervised Transfer Learning

One more approach that we plan to investigate in the future is the unsupervised transfer learning that deals with the problem of automatically labeling newly collected signals, then using them as new training samples for the classification of acoustic signals.[45,46] This is a critical problem since, in practice, many more acoustic samples can be collected in the field but only a small subset of those can be

manually labeled. The question is how to make use of these unlabeled samples to further improve classification performance. This motivates us to study the unsupervised transfer learning that takes into account the prior knowledge of labeled training examples (source tasks) in a transformed domain to develop a hypothesis for the set of unlabeled samples (target tasks)—that is, probabilistically assigning every unlabeled sample to a specific class. This can be done by using a transformation matrix to transfer both the source and target data onto a common subspace in which each target datum can be linearly reconstructed by the data from the source domain. This problem can be formulated as

$$\min_{\boldsymbol{T},\boldsymbol{A}} \|\boldsymbol{T}\boldsymbol{Y}_U - \boldsymbol{T}\boldsymbol{Y}_L\boldsymbol{A}\|_F^2 , \tag{29}$$

where $\boldsymbol{Y}_L$ and $\boldsymbol{Y}_U$ are the corresponding labeled (source) and unlabeled (target) measurement matrices; $\boldsymbol{T}$ is the linear transformation, and $\boldsymbol{A}$ is the coefficient matrix that linearly represents target tasks by the source tasks in the common subspace. Furthermore, the reconstruction coefficient matrix $\boldsymbol{A}$ should have a block-wise structure that promotes both low-rank and sparsity properties on $\boldsymbol{A}$. Therefore, the unsupervised transfer learning for classification can be recast as the following optimization:

$$\min_{\boldsymbol{T},\boldsymbol{A}} \frac{1}{2} \|\boldsymbol{T}\boldsymbol{Y}_U - \boldsymbol{T}\boldsymbol{Y}_L\boldsymbol{A}\|_F^2 + \lambda_1 \|\boldsymbol{A}\|_1 + \lambda_L \|\boldsymbol{A}\|_* . \tag{30}$$

The low-rank constraint penalized by the nuclear norm $\|\boldsymbol{A}\|_*$ encourages the data correlation among samples of the same classes while the sparsity constraint $\|\boldsymbol{A}\|_1$ is helpful to preserve the data local structure such that each target sample can be well reconstructed by only a few samples from the source domain. The benefits of solving the transfer learning problem (Eq. 30) are 2-fold. First, it efficiently exploits information on the unlabeled signatures and can thus automatically update the dictionary on the battlefield. Second, by transfer learning, we can enforce the consistency of source and target samples in the transferred domain even when there are uncommon features (such as propagation effects, signal interferences, or vegetation) between the 2 sets. This may eventually lead to a possible answer to the invariant feature search problem for transient acoustic signals, which still remains unsolved.

6. References

1. Obozinski G, Taskar B, Jordan MI. Joint covariate selection and joint subspace selection for multiple classification problems. Journal of Statistics and Computing. 2010;20(2):231–252.

2. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y. Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2009;31(2):210–227.

3. Wright J, Ma Y, Mairal J, Sapiro G, Huang T, Yan S. Sparse representation for computer vision and pattern recognition. Proceedings of the IEEE. 2010;98(6):1031–1044.

4. Chen Y, Nasrabadi NM, Tran TD. Sparse representation for target detection in hyperspectral imagery. IEEE Journal of Selected Topics in Signal Processing. 2011;5(3):629–640.

5. Dao M, Nguyen D, Tran T, Chin S. Chemical plume detection in hyperspectral imagery via joint sparse representation. Military Communications Conference (MILCOM); 2012. p. 1–5.

6. Mairal J, Bach F, Ponce J, Sapiro G, Zisserman A. Discriminative learned dictionaries for local image analysis. Conference on Computer Vision and Pattern Recognition (CVPR); 2008. p. 1–8.

7. Yuan XT, Yan S. Visual classification with multi-task joint sparse representation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR); 2010. p. 3493–3500.

8. Chen Y, Nasrabadi NM, Tran TD. Hyperspectral image classification using dictionary-based sparse representation. IEEE Transactions on Geoscience and Remote Sensing. 2011;49(10):3973–3985.

9. Nguyen NH, Nasrabadi NM, Tran TD. Robust multi-sensor classification via joint sparse representation. IEEE International Conference on Information Fusion (FUSION); 2011 Jul. p. 1–8.

10. Dao M, Nguyen NH, Nasrabadi NM, Tran TD. Collaborative multi-sensor classification via sparsity-based representation. Accepted to IEEE Transactions on Signal Processing. 2016.

11. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning. 2011;3(1):1–122.

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

13. Needell D, Tropp JA. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis. 2008;26:301–321.

14. Dai W, Milenkovic O. Subspace pursuit for compressive sensing signal reconstruction. IEEE Transactions on Information Theory. 2009;55(5):2230–2249.

15. Blumensath T, Davies ME. Iterative hard thresholding for compressed sensing. Applied and Computational Harmonic Analysis. 2009;27(3):265–274.

16. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society - Series B. 2006;68(1):49–67.

17. Zhao P, Rocha G, Yu B. The composite absolute penalties family for grouped and hierarchical variable selection. Annals of Statistics. 2009;37:3469–3497.

18. Obozinski G, Wainwright MJ, Jordan MI. Support union recovery in high-dimensional multivariate regression. Annals of Statistics. 2001;39(1):1–47.

19. Zhang H, Nasrabadi NM, Zhang Y, Huang TS. Joint dynamic sparse representation for multi-view face recognition. Pattern Recognition. 2012;45(4):1290–1298.

20. Lounici K, Pontil M, Tsybakov AB, Van SDG. Taking advantage of sparsity in multi-task learning. 2009. ArXiv preprint:0903.1468.

21. Cai J, Candès EJ, Shen Z. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization. 2010;20(4):1956–1982.

22. Candès EJ, Li X, Ma Y, Wright J. Robust principal component analysis. Journal of ACM. 2011;58(3):1–37.

23. Sprechmann P, Ramírez I, Sapiro G, Eldar YC. C-hilasso: A collaborative hierarchical sparse modeling framework. IEEE Transactions on Signal Processing. 2011;59(9):4183–4198.

24. Suo Y, Dao M, Srinivas U, Monga V, Tran TD. Structured dictionary learning for classification. 2014. ArXiv preprint:1406.1943.

25. Roberts WJ, Sabrin HW, Tenney S. Transient signal classification for force protection systems. Proceedings of the Military Sensing Symposia (MSS) on Battlefield Acoustic and Seismic Sensing. Magnetic and Electric Field Sensors (BAMS); 2010.

26. Wu H, Gurram P, Kwon H, Prasad S. A hybrid csvm-hmm model for acoustic signal classification using a tetrahedral sensor array. IEEE Sensors. 2014. p. 1352–1355.

27. Meier L, Geer SVD, Bahlmann P. The group lasso for logistic regression. Journal of the Royal Statistical Society: Series B. 2008;70(1):53–71.

28. Lee H, Pham P, Largman Y, Ng AY. Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Advances in Neural Information Processing Systems; 2009. p. 1096–1104.

29. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems; 2012. p. 1097–1105.

30. Hinton G, Deng L, Yu D, Dahl GE, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine. 2012;29(6):82–97.

31. Mohamed A, Dahl GE, Hinton G. Acoustic modeling using deep belief networks. IEEE Transactions on Audio, Speech, and Language Processing. 2012;20(1):14–22.

32. Bengio Y. Learning deep architectures for ai. Foundations and trends® in Machine Learning. 2009;2(1):1–127.

33. Childers DG, Skinner DP, Kemerait RC. The cepstrum: A guide to processing. Proceedings of the IEEE. 1977;65(10):1428–1443.

34. Norton MP, Karczub DG. Fundamentals of noise and vibration analysis for engineers. Cambridge (UK): Cambridge University Press; 2003.

35. Gupta S, Ray A. Pattern Recognition: Theory and Application. New York (NY): Nova Science Publishers; 2007. Chapter 2, Symbolic dynamic filtering for data-driven pattern recognition; p. 17–71.

36. HO B, Kálmán RE. Editorial: Effective construction of linear state-variable models from input/output functions. Automatisierungstechnik. 1966;14(12):545–548.

37. Mairal J, Bach F, Ponce J, Sapiro G. Online dictionary learning for sparse coding. Proceedings of 26th Annual International Conference on Machine Learning. 2009 Jun. p. 689–696.

38. Aharon M, Elad M, Bruckstein A. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing. 2006;54(11):4311–4322.

39. Ramírez I, Sprechmann P, Sapiro G. Classification and clustering via dictionary learning with structured incoherence and shared features. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2010. p. 3501–3508.

40. Mairal J, Bach F, Ponce J, Sapiro G. Online learning for matrix factorization and sparse coding. The Journal of Machine Learning Research. 2010;11:19–60.

41. Mateos G, Giannakis GB. Distributed recursive least-squares: stability and performance analysis. IEEE Transactions on Signal Processing. 2012;60(7):3740–3754.

42. Sprechmann P, Bronstein AM, Sapiro G. Learning robust low-rank representations. 2012. ArXiv preprint:1209.6393.

43. Edelman A, Arias TA, Smith ST. The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications. 1998;20(2):303–353.

44. Absil PA, Mahony R, Sepulchre R. Optimization algorithms on matrix manifolds. Princeton (NJ): Princeton University Press; 2009.

45. Pan SJ, Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering. 2010;22(10):1345–1359.

46. Raina R, Battle A, Lee H, Packer B, Ng AY. Self-taught learning: transfer learning from unlabeled data. Proceedings of the 24th International Conference on Machine Learning; 2007. p. 759–766.

List of Symbols, Abbreviations, and Acronyms

ADMM	alternating direction method of multipliers
ARL	US Army Research Laboratory
DBN	deep belief network
DL	dictionary learning
DNN	deep neural network
GJSR	group-and-joint sparse representation
GJSR+L	group-and-joint sparse representation with low-rank interference
GMM	Gaussian mixture model
HMM	hidden Markov model
JSR	joint sparse representation
MSVAR	Markov switching vector auto-regression
RBM	restricted Boltzmann machine
SLR	sparse logistic regression
SVM	support vector machine

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (PDF)	GOVT PRINTG OFC A MALHOTRA

ABERDEEN PROVING GROUND

11 (PDF) DIR USARL
RDRL SES P
M SCANLON
S TENNEY
L SIM
TD TRAN-LUU
C REIFF
H VU
D GONSKI
J GOLDMAN
W ALBERTS
RDRL SES A
T PHAM
RDRL CI
L SOLOMON

1 (PDF) MBO PARTNERS
M DAO