Chapter 1

Sparse Representation for Time-Series Classification

Soheil Bahrampour*, Nasser M. Nasrabadi†, and Asok Ray*

* *Department of Electrical Engineering, Pennsylvania State University, University Park, PA 16802, USA; {soheil, axr2}@psu.edu.*

† *Army Research Laboratory, Adelphi, MD 20783, USA; nasser.m.nasrabadi.civ@mail.mil.*

This chapter studies the problem of time-series classification and presents an overview of recent developments in the area of feature extraction and information fusion. In particular, a recently proposed feature extraction algorithm, namely symbolic dynamic filtering (SDF), is reviewed. The SDF algorithm generates low-dimensional feature vectors using probabilistic finite state automata that are well-suited for discriminative tasks. The chapter also presents the recent developments in the area of sparse-representation-based algorithms for multimodal classification. This includes the joint sparse representation that enforces collaboration across all the modalities as well as the tree-structured sparsity that provides a flexible framework for fusion of modalities at multiple granularities. Furthermore, unsupervised and supervised dictionary learning algorithms are reviewed. The performance of the algorithms are evaluated on a set of field data that consist of passive infrared and seismic sensors.

1. Introduction

Unattended ground sensor (UGS) systems have been extensively used to monitor human activities for border security and target classification. Typical sensors used for this purpose are Seismic and Passive Infrared (PIR) sensors which are commonly used for target detection. Nevertheless discrimination of different types of targets from footstep signals is still a challenging problem due to environmental noise sources and locality of the sensors.[1] This study deals with the problem of target classification, and more generally time-series classification, in two main directions, feature extraction and information fusion.

Several feature extraction methods have been proposed to generate discriminative patterns from time-series data. This includes kurtosis algorithm,[2] Fourier and Wavelet analysis.[3–5] Recently, a symbolic dynamic filtering (SDF) algorithm has been proposed for feature extraction from time-series data and has shown promising results in several applications including robot motion classification and target classification.[6–8] In the SDF algorithm, the time series data are first converted into symbol sequences, and then probabilistic finite-state automata (PFSA) are constructed from these symbol sequences to compress the pertinent information into low-dimensional statistical patterns.[9] The advantage of the SDF algorithm is that it captures the local information of the signal and it is capable of mitigating noise. In this chapter, the SDF algorithm is briefly reviewed and is subsequently used for feature extraction from PIR and Seismic sensors.

This chapter also studies information fusion algorithms which is then utilized to integrate the information of the PIR and Seismic modalities. As it has been widely studied, information fusion often results in better situation awareness and decision making.[10,11] For this purpose, several sparsity models are discussed which provide a framework for feature-level fusion. Feature level fusion[12] is relatively less-studied topic compared to the decision level fusion,[13,14] mainly due to the difficulty in fusing heterogeneous feature vectors. However, as it will be shown, structured sparsity priors can be used to overcome this difficulty and result in state-of-the art performance. In particular, the joint sparse representation[15] is presented which enforces collaboration across all the modalities. The tree-structured sparse representation[16,17] is also presented that allows fusion of different modalities at multiple granularities. Moreover, unsupervised and supervised dictionary learning algorithms are discussed to train more compact dictionaries which are optimized for reconstructive or discriminative tasks, respectively.[18,19]

The rest of this chapter is organized as follows. Section 2 briefly describes the SDF-based feature extraction. Section 3 succinctly discusses the sparsity-based models for single-modal and multi-modal classification and Section 4 presents the dictionary learning algorithms. Section 5 provides a comparitive study on the performance of the discussed algorithm for the application of target classification, which is followed by conclusion and recommendations for future research in Section 6.

2. Symbolic dynamic filtering for feature extraction from time-series data

An important step for time-series classification is feature extraction from sensor signals. This step can be performed using different signal processing tools including principal component analysis,[20] Cepstrum,[21] wavelet analysis,[22] and SDF.[6,23] It has recently been shown that SDF can result in improved classification performance by compressing the information within a time-series window into a low-dimensional feature space while preserving the discriminative pattern of the data in several applications including target detection & classification[9,24] and prediction of lean blowout phenomena in confined combustion.[25] The detailed information and different versions of SDF can be found in earlier publications,[23,26] and this section briefly reviews a version of this algorithm which is used later for feature extraction.

The algorithm consists of a few steps. The signal space, which is approximated by the training samples, is first partitioned into a finite number of cells that are labeled as symbols. Then, PFSA are formed to represent different combinations of blocks of symbols on the symbol sequence. Finally, for a given time-series data, the state transition probability matrix is generated and the SDF feature is extracted. The pertinent steps of this procedure are described below.

Let $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$ be the set of N training time-series where $\boldsymbol{u}_j \in \mathbb{R}^{1\times M}, j = 1, \ldots, N$, consists of M consecutive data points. Let Σ be a set of finitely many symbols, also known as alphabet, and its cardinality denoted as $|\Sigma|$. In the partitioning step, the ensemble of training samples is divided into $|\Sigma|$ mutually exclusive and exhaustive cells. Each disjoint region forms a cell in the partitioning and is labeled with a symbol from the alphabet Σ. Consequently, each sample of a time-series is located in a particular cell and is coded with the corresponding symbol, which results in a string of symbols representing the (finite-length) time-series. There are at least two ways for performing the partitioning task: the maximum entropy partitioning and uniform partitioning.[26] The maximum entropy partitioning algorithm maximizes the entropy of the generated symbols and results in (approximately) equal number of data points in each region. Therefore, the information-rich cells of a data set are partitioned finer and those with sparse information are partitioned coarser. On the other hand, the uniform partitioning method results in equal-sized cells. Maximum entropy partitioning is adopted here. The choice of alphabet size $|\Sigma|$ largely depends on

the specific data set and can be set using cross-validation algorithms. A smaller alphabet size results in a more compressed feature vector which is more robust to the time-series noise at the cost of more information loss.

In the next step, a PFSA is used to capture the information of a given time-series. The PFSA states represent different combinations of blocks of symbols on the symbol sequence where the transition between a state to another state is governed by a transition probability matrix. Therefore, the "states" denote all possible symbol blocks within a window of certain length. For both algorithmic simplicity and computational efficiency, the D-Markov machine structure[6] has been adopted for construction of PFSA. It is noted that D-Markov machines form a proper subclass of hidden Markov models (HMM) and have been experimentally validated for applications in various fields of research (e.g., anomaly detection and robot motion classification[8]). A D-Markov chain is modeled as a statistically locally stationary stochastic process $S = \cdots s_{-1}s_0 \cdots s_1 \cdots$, where the probability of occurrence of a new symbol depends only on the last D symbols, i.e.,

$$P[s_n \mid \cdots s_{n-D} \cdots s_{n-1}] = P[s_n \mid s_{n-D} \cdots s_{n-1}].$$

Words of length D on a symbol string are treated as the states of the D-Markov machine before any state-merging is executed. The set of all possible states is denoted as $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$ and $|Q|$ is the number of (finitely many) states and $|Q| \leq |\Sigma|^D$. Here, a D-Markov Machine with the symbol block length of each state $D = 1$ is used, i.e., $|Q| = |\Sigma|$. In this case, the number of states are equal to the number of symbols, i.e., $|\mathcal{Q}| = |\Sigma|$, where the set of all possible states is denoted as $\mathcal{Q} = \{q_1, q_2, \ldots, q_{|\mathcal{Q}|}\}$ and $|\mathcal{Q}|$ is the number of (finitely many) states. The transition probabilities are defined as:

$$\mathcal{P}\left(q_k \mid q_l\right) = \frac{S\left(q_l, q_k\right)}{\sum_{i=1,2,\ldots,|\mathcal{Q}|} S\left(q_l, q_i\right)}, \forall q_k, q_l \in \mathcal{Q} \tag{1}$$

where $S\left(q_l, q_k\right)$ is the total count of events when q_k occurs adjacent to q_l in the direction of motion. Consequently, the state transition probability matrix of the PFSA is given as

$$\Pi = \begin{bmatrix} \mathcal{P}\left(q_1 \mid q_1\right) & \ldots \mathcal{P}\left(q_{|\mathcal{Q}|} \mid q_l\right) \\ \vdots & \ddots \vdots \\ \mathcal{P}\left(q_1 \mid q_{|\mathcal{Q}|}\right) & \cdots \mathcal{P}\left(q_{|\mathcal{Q}|} \mid q_{|\mathcal{Q}|}\right) \end{bmatrix}. \tag{2}$$

By appropriate choice of partitioning, the stochastic matrix Π is irreducible and the Markov chain is ergodic, i.e., the probability of every state being

reachable from any other state within finitely many transitions must be strictly positive under statistically stationary conditions.[27]

For a given time-series window, the SDF features is then constructed as the left eigenvector p corresponding to the unity eigenvalue of the stochastic matrix Π. It should be noted that Π is guaranteed to have unique unity eigenvalue. The extracted feature vector is indeed the stationary state probability vector.

3. Sparse representation for classification (SRC)

The SRC algorithm has been recently introduced[28] and has shown promising results in several classification applications such as robust face recognition,[28] visual tracking,[29] and transient acoustic signal classification.[30] Here the SRC algorithm is reviewed. Consider a C-class classification problem with N training samples from C different classes. Let $N_c, c \in \{1, \ldots, C\}$, be the number of training samples for the c^{th} class and n be the dimension of the feature vector. Here n is equal to the the SDF alphabet size $|\Sigma|$. Also let $\boldsymbol{x}_{c,j} \in \mathbb{R}^n$ denotes the j^{th} training sample of the c^{th} class where $j \in \{1, \ldots, N_c\}$. In the SRC algorithm, a *dictionary* $\boldsymbol{X} \triangleq [\boldsymbol{X}_1 \boldsymbol{X}_2 \ldots \boldsymbol{X}_C] \in \mathbb{R}^{n \times N}$ is constructed by stacking the training samples where sub-dictionary $\boldsymbol{X}_c = [\boldsymbol{x}_{c,1}, \boldsymbol{x}_{c,2}, \ldots, \boldsymbol{x}_{c,N_c}] \in \mathbb{R}^{n \times N_c}$ consists of the training samples for the c^{th} class, and $N = \sum_{c=1}^{C} N_c$ is the total number of train samples. Given a test sample $\boldsymbol{p} \in \mathbb{R}^n$, it is classified based on the minimum reconstruction error of it using the different classes. The underlying assumption of SRC is that a test sample from the c^{th} class lies (approximately) within the subspace formed by the training samples of the c^{th} class and can be represented using a linear combination of *a few* training samples in $\boldsymbol{X}_c$. In other words, the test sample $\boldsymbol{p}$ from the c^{th} class can be represented as

$$\boldsymbol{p} = \boldsymbol{X}\boldsymbol{\alpha} + \boldsymbol{e}, \tag{3}$$

where $\boldsymbol{\alpha}$ is the coefficient vector whose entries have value 0's except for some of the entries associated with the c^{th} class, i.e. $\boldsymbol{\alpha} = \left[\boldsymbol{0}^T, \ldots, \boldsymbol{0}^T, \boldsymbol{\alpha}_c^T, \boldsymbol{0}^T, \ldots, \boldsymbol{0}^T\right]^T$, and $\boldsymbol{e}$ is a small error/noise term due to the imperfectness of the test and training samples. For this reason the algorithm seek to obtain the sparse coefficient vector $\boldsymbol{\alpha}$ through the following ℓ_1 optimization problem:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{p} - \boldsymbol{X}\boldsymbol{\alpha}\|_{\ell_2} + \lambda \|\boldsymbol{\alpha}\|_{\ell_1}, \tag{4}$$

with ℓ_1-norm defined as $\|\boldsymbol{\alpha}\|_{\ell_1} = \sum_{j=1}^{N}|\alpha_j|$ and λ is a regularization parameter. The solution of the above optimization problem is sparse provided that the regularization parameter λ is chosen sufficiently large. After the solution of the optimization problem is found, the test sample $\boldsymbol{p}$ is classified by comparing the reconstruction errors of different classes. For this purpose, let $\delta_c(\boldsymbol{\alpha}) \in \mathbb{R}^N$ be a vector whose only non-zero elements are those entries in $\boldsymbol{\alpha}$ that are associated with class c in $\boldsymbol{X}$, i.e. $\delta_c(\boldsymbol{\alpha}) = \left[\mathbf{0}^T, \ldots, \mathbf{0}^T, \boldsymbol{\alpha}_c^T, \mathbf{0}^T, \ldots, \mathbf{0}^T\right]^T$. The label of the test data is then predicted using

$$c^* = \underset{c}{\operatorname{argmin}} \|\boldsymbol{p} - \boldsymbol{X}\delta_c(\boldsymbol{\alpha}^*)\|_{\ell_2}. \tag{5}$$

3.1. *Joint sparse representation classification*

In many applications including target classification, there are several sources of information that should be fused to make an optimal classification decision. It is well-known that information fusion of sensors can generally result in better situation awareness and decision making.[10] Many algorithms have been proposed for sensor fusion in the classification problem. These methods can generally be categorized in two sets, feature fusion[12] and classifier fusion [14] algorithms. Classifier fusion algorithms aggregate the decisions from different classifiers which are individually built based on different sources. Different methods of decision fusion include majority vote,[31] fuzzy logic[32] and statistical inference.[11] For example, in the context of the sparse representation classification for target classification, the reconstruction error generated by using PIR and Seismic features individually can be combined, after proper normalization, for a fused decision. This is also known as holistic sparse representation classification (HSRC).[33] While classifier fusion is a well-studied topic, feature level fusion is a relatively less-studied, specifically for fusing heterogeneous source of information due to the incompatibility of feature sets.[34] Feature level fusion using sparse representation has also been recently introduced and has shown promising results.[35–38]

Among different sparsity based algorithms for information fusion, joint sparse representation classification (JSRC) is probably the most cited approach.[21,30,39] In JSRC, multiple observations of a pattern using different modalities are simultaneously represented by a few training samples. Consider the C-class classification problem and let $\mathcal{S} \triangleq \{1, \ldots, S\}$ be a finite set of available modalities. Similar to the previous section, let $N = \sum_{c=1}^{C} N_c$

be the number of the training samples from each modalities, where N_c is the number of training samples in the c^{th} class. Also let $n^s, s \in \mathcal{S}$, be the dimension of the feature vector for the s^{th} modality and $\boldsymbol{x}^s_{c,j} \in \mathbb{R}^{n^s}$ denote the j^{th} sample of the s^{th} modality that belongs to the c^{th} class, $j \in \{1, \ldots, N_c\}$. In JSRC, S dictionaries $\boldsymbol{X}^s \triangleq [\boldsymbol{X}^s_1 \boldsymbol{X}^s_2 \ldots \boldsymbol{X}^s_C] \in \mathbb{R}^{n^s \times N}, s \in \mathcal{S}$, are constructed from the (normalized) training samples, where the class-wise sub-dictionary $\boldsymbol{X}^s_c \triangleq \left[\boldsymbol{x}^s_{c,1}, \boldsymbol{x}^s_{c,2}, \ldots, \boldsymbol{x}^s_{c,N_c}\right] \in \mathbb{R}^{n^s \times N_c}$ consists of samples from the c^{th} class and s^{th} modality.

Given a multimodal test sample $\{\boldsymbol{p}^s\}$, where $\boldsymbol{p}^s \in \mathbb{R}^{n^s}, s \in \mathcal{S}$, the assumption is that the test sample $\boldsymbol{p}^s$ from the c^{th} class lies approximately within the subspace formed by the training samples of the c^{th} class and can be approximated (or reconstructed) from *a few* number of training samples in $\boldsymbol{X}^s_c$,[28] similar to the SRC algorithm. In other words, if the test sample $\boldsymbol{p}^s$ belongs to the c^{th} class, it is represented as:

$$\boldsymbol{p}^s = \boldsymbol{X^s \alpha^s} + \boldsymbol{e}, \tag{6}$$

where $\boldsymbol{\alpha}^s \in \mathbb{R}^N$ is a coefficient vector whose entries are mostly 0's except for some of the entries associated with the c^{th} class, i.e., $\boldsymbol{\alpha}^s = \left[\boldsymbol{0}^T, \ldots, \boldsymbol{0}^T, \boldsymbol{\alpha}^T_c, \boldsymbol{0}^T, \ldots, \boldsymbol{0}^T\right]^T$, and $\boldsymbol{e}$ is a small error term due to imperfectness of the samples. In addition, JSRC recognizes the relation between different modalities representing the same event and enforces collaboration among them to make a joint decision. This is achieved by constraining the coefficient vectors from different modalities to have the same sparsity pattern. Consequently, the same training samples from different modalities are used to reconstruct the test data. To illustrate the idea, consider the target classification application with PIR and Seismic sensors. Let $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$ be the feature vectors extracted from PIR and Seismic sensors, respectively. Also let the multimodal test sample belongs to the c^{th} class. Using the idea of sparse representation discussed in previous Section, test samples can be reconstructed using a linear combination of atoms in $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$, i.e., $\boldsymbol{p}^1 = \boldsymbol{X^1 \alpha^1} + \boldsymbol{e}^1, \boldsymbol{p}^2 = \boldsymbol{X^2 \alpha^2} + \boldsymbol{e}^2$, where $\boldsymbol{\alpha}^1$ and $\boldsymbol{\alpha}^2$ are the coefficient vectors whose entries have value 0's except for some of the entries associated with the c^{th} class, and $\boldsymbol{e}^1$ and $\boldsymbol{e}^2$ are small error terms. Let $\mathcal{I}^1$ and $\mathcal{I}^2$ be two index sets corresponding to non-zero rows of $\boldsymbol{\alpha}^1$ and $\boldsymbol{\alpha}^2$, respectively. In JSRC algorithm, it is further assumed that $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$ can be reconstructed from the same training samples corresponding to dictionaries $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$ with possibly different coefficients because they belong to the same event and therefore $\mathcal{I}^1 = \mathcal{I}^2$. In other words, $A = \left[\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2\right]$ is a row-sparse matrix with only a few non-zeros rows. In general, the coeffi-

cient matrix $\boldsymbol{A} = \left[\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^S\right] \in \mathbb{R}^{N \times S}$, where $\boldsymbol{\alpha}^s$ is the sparse coefficient vector for reconstructing $\boldsymbol{p}^s$, is recovered by solving the following ℓ_1/ℓ_2 joint optimization problem:

$$\underset{\boldsymbol{A}=[\boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^S]}{\operatorname{argmin}} f(\boldsymbol{A}) + \lambda \|\boldsymbol{A}\|_{\ell_1/\ell_q}, \tag{7}$$

where $f(\boldsymbol{A}) \triangleq \frac{1}{2}\sum_{s=1}^{S} \|\boldsymbol{p}^s - \boldsymbol{X}^{\boldsymbol{s}}\boldsymbol{\alpha}^{\boldsymbol{s}}\|_{\ell_2}^2$ is the reconstruction error, $\lambda > 0$ is a regularization parameter, and ℓ_1/ℓ_2 norm is defined as $\|\boldsymbol{A}\|_{\ell_1/\ell_2} = \sum_{j=1}^{N} \|\boldsymbol{a}_j\|_{\ell_2}$ in which $\boldsymbol{a}_j$'s are row vectors of $\boldsymbol{A}$. The above optimization problem encourages sharing of patterns across related observations which results in the solution $\boldsymbol{A}$ to have a common support at the column level.[21] The solution can be obtained by using the efficient alternating direction method of multipliers.[40]

Again let $\delta_c(\alpha) \in \mathbb{R}^N$ be a vector indication function in which the rows corresponding to c^{th} class are retained and the rest are set to zeros. Similar to the SRC algorithm, the test data is classified using the class-specific reconstruction errors as:

$$c^* = \underset{c}{\operatorname{argmin}} \sum_{s=1}^{S} \|\boldsymbol{p}^s - \boldsymbol{X}^s\delta_c(\boldsymbol{\alpha}^{s*})\|_{\ell_2}^2, \tag{8}$$

where $\boldsymbol{\alpha}^{s*}$'s are optimal solutions of (7).

3.2. *Tree-structured sparse representation classification*

The joint sparsity assumption of JSRC may be too stringent for applications in which not all the different modalities are equally important for classification. Recently tree-structured sparsity[16,17] is proposed to provide a flexible framework for information fusion. It uses the prior knowledge in grouping different modalities by encoding them in a tree and allows different modalities to be fused at multiple granularity. Th leaf nodes represent individual modalities in the tree and the internal nodes represent different grouping of the modalities. A tree-structured groups of modalities $\mathcal{G} \subseteq \left(2^{\mathcal{S}} \setminus \emptyset\right)$ is defined as a collection of subsets of the set of modalities $\mathcal{S}$ such that $\bigcup_{g\in\mathcal{G}} g = \mathcal{S}$ and $\forall g, \tilde{g} \in \mathcal{G}, (g \cap \tilde{g} \neq \emptyset) \Rightarrow ((g \subseteq \tilde{g}) \vee (\tilde{g} \subseteq g))$. It is assumed here that $\mathcal{G}$ is ordered according to relation $\preccurlyeq$ which is defined as $(g \preccurlyeq \tilde{g}) \Rightarrow ((g \subseteq \tilde{g}) \vee (g \cap \tilde{g} = \emptyset))$.

Given a tree-structured collection $\mathcal{G}$ of groups and a multimodal test sample $\{\boldsymbol{p}^s\}$, the tree-structured sparse representation classification

(TSRC) solves the following optimization problem:

$$\underset{\boldsymbol{A}=[\boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^S]}{\operatorname{argmin}} \quad f(\boldsymbol{A}) + \lambda\, \Omega\left(\boldsymbol{A}\right), \tag{9}$$

where $f(\boldsymbol{A})$ is defined the same as in Eq. (7), and the tree-structured sparsity prior $\Omega\left(\boldsymbol{A}\right)$ is defined as:

$$\Omega\left(\boldsymbol{A}\right) \triangleq \sum_{j=1}^{N} \sum_{g \in \mathcal{G}} \omega_g \|\boldsymbol{a}_{jg}\|_{\ell_2}. \tag{10}$$

In Eq. (10), ω_g is a positive weight for group g and $\boldsymbol{a}_{jg}$ is a $(1\times S)$ row vector whose coordinates are equal to the j^{th} row of $\boldsymbol{A}$ for indices in the group g, and 0 otherwise. An accelerated algorithm can be used to efficiently solve the optimization problem.[17]

The above optimization problem results in $\boldsymbol{A}^*$ that has a common support at the group level and the resulting sparsity is dependent on the relative weights ω_g of different groups.[16] In the special case where $\mathcal{G}$ consists of only one group, containing all modalities, then Eq. (9) reduces to that of JSRC in Eq. (7). On the other hand, if $\mathcal{G}$ consists of only singleton sets of individual modalities, no common sparsity pattern is sought across the modalities and the optimization problem of Eq. (9) reduces to S separate ℓ_1 optimization problems. The tree-structured sparsity prior provides flexibility in the expense of the need to apiori select the weights $\omega_g, g \in \mathcal{G}$, which will be discussed in more details in the next section.

4. Dictionary learning

In the discussed sparsity based classification algorithms, SRC, JSRC and TSRC, the dictionary is constructed by stacking the training samples. In contrast to the conventional classification algorithms, above algorithms do not have a"training" step and most of the computation is performed at the test time, i.e., an optimization problem needs to be solved for each test sample. These results in at least two difficulties. First, the computational cost of the optimization problem becomes more and more expensive as the number of training samples increases. Second, the dictionary constructed by stacking the training samples is not optimal neither for the reconstructive tasks[41] nor the discriminative tasks.[18] Recently it has been shown that *dictionary learning* can overcome the above limitations and significantly improve the performance in several applications including image restoration,[42] face recognition[43] and object recognition.[44,45] In contrast to principal component analysis and its variants, dictionary learning algorithms generally

do not impose orthogonality condition and are more flexible allowing to be well-tuned to the training data. Moreover, the size of the learned dictionaries are usually smaller than the number of training samples[46,47] and are more appropriated for real-time applications. Dictionary learning algorithms can generally be categorized into two groups: unsupervised and supervised which are discussed in the following sections.

4.1. *Unsupervised dictionary learning*

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{n\times N}$ be the collection of N (normalized) training samples. In an unsupervised setting, the dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times d}$ is obtained irrespective of the class labels of the training samples by minimizing of the following reconstructive cost:[43]

$$g_N(\boldsymbol{D}) \triangleq \frac{1}{N}\sum_{i=1}^{N} l_u(\boldsymbol{x}_i, \boldsymbol{D}), \tag{11}$$

over the regularizing convex set $\mathcal{D} \triangleq \{\boldsymbol{D} \in \mathbb{R}^{n\times d} | \|\boldsymbol{d}_k\|_{\ell_2} \leq 1, \forall k = 1, \ldots, d\}$, where $\boldsymbol{d}_k$ is the k^{th} column, or atom, of the dictionary. The loss l_u is defined as

$$l_u(\boldsymbol{x}, \boldsymbol{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathbb{R}^d} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_{\ell_2}^2 + \lambda\|\boldsymbol{\alpha}\|_{\ell_1}, \tag{12}$$

which is the optimal value of the sparse coding problem. It is usually preferred to find the dictionary by minimizing an expected risk, rather than the perfect minimization of the empirical cost for the generalization purpose.[48] A parameter-free online algorithm has also been proposedd[41] to find the dictionary $\boldsymbol{D}$ as the minimizer of the following stochastic cost over the convex set $\mathcal{D}$:

$$g(\boldsymbol{D}) \triangleq \mathrm{E}_{\boldsymbol{x}}\left[l_u(\boldsymbol{x}, \boldsymbol{D})\right], \tag{13}$$

where it is assumed that the data $\boldsymbol{x}$ is drawn from a finite probability distribution. The trained dictionary can then be integrated into the SRC algorithm.

The trained dictionary can also be used for feature extraction. In this setting, the sparse code $\boldsymbol{\alpha}^\star$, generated as a solution of (12), is used as a latent feature vector representing the input signal $\boldsymbol{x}$ in the classical expected risk optimization for training a classifier:[19]

$$\min_{\boldsymbol{w}\in\mathcal{W}} \mathrm{E}_{y,\boldsymbol{x}}\left[l(y, \boldsymbol{w}, \boldsymbol{\alpha}^\star)\right] + \frac{\nu}{2}\|\boldsymbol{w}\|_{\ell_2}^2, \tag{14}$$

where y is the ground truth class label associated with the input $\boldsymbol{x}$, $\boldsymbol{w}$ is the classifier parameters, ν is a regularizing parameter, and l is a convex loss function that measures how well one can predict y given $\boldsymbol{\alpha}^\star$ and $\boldsymbol{w}$. Note that in Eq. 14, the dictionary $\boldsymbol{D}$ is optimized independent of the classifier and class label.

4.2. *Supervised dictionary learning*

The dictionary trained in the unsupervised setting is not optimal for classification. In a supervised formulation, the class labels can be used to train a discriminative dictionary. In the most straightforward extension from the unsupervised setting, the dictionary $\boldsymbol{D}^*$ can instead be obtained by learning class-specific sub-dictionaries $\boldsymbol{D}_c^*, c = 1, \ldots, C$, in a C-class classification problem using the formulation of Eq. (13) by sampling the input from the corresponding c-th class population. The overall dictionary is then constructed as $\boldsymbol{D}^* = [\boldsymbol{D}_1^* \ldots \boldsymbol{D}_C^*]$.[43] The C different sub-dictionaries are trained independently and some of the sub-dictionaries can possibly share similar atoms that may adversely affect the discrimination performance. An *incoherence* term has been proposed to be added to the cost function to make the derived dictionaries independent.[49] In another formulation, a discriminative term is added to the reconstruction error and the overall cost is minimized.[18,50] More recently, it has been shown that better performance can be obtained by learning the dictionary in a task-driven formulation.[19] In the task-driven dictionary learning, the optimal dictionary and classifier parameters are obtained jointly by solving the following optimization problem:[19]

$$\min_{\boldsymbol{D}\in\mathcal{D},\boldsymbol{w}\in\mathcal{W}} \mathrm{E}_{y,\boldsymbol{x}}\left[l_{su}\left(y, \boldsymbol{w}, \boldsymbol{\alpha}^\star(\boldsymbol{x}, \boldsymbol{D})\right)\right] + \frac{\nu}{2}\|\boldsymbol{w}\|_{\ell_2}^2. \tag{15}$$

The learned task-driven dictionary has been shown to result in a superior performance compared to the unsupervised setting.[19] In this setting, the sparse codes are indeed the optimized latent features for the classifier. While the above dictionary learning algorithms are mostly developed for single-modal scenarios, it has been shown that learning multimodal dictionaries under structured sparsity priors can be beneficial.[51,52]

5. Results

This section presents the results of target classification on a real data set, which was generated from field data using one passive infrared (PIR) and

three seismic sensors. The time-series are classified to predict two targets: (i) human walking alone, and (ii) animal led by a walking human. These targets moved along an approximately 150 meters long trail and returned along the same trail to the starting point; all targets passed by the sensor sites at a distance of approximately 5 meters. Signals from both PIR and seismic sensors were acquired at a sampling frequency of 10 kHz and each test was conducted over a period of approximately 50 seconds. The subset of data used here consists of two days data. Day 1 includes 47 human targets and 35 animal-led-by-human targets while the corresponding numbers for Day 2 are 32 and 34, respectively. A two-way cross-validation is used to assess the performance of the classification algorithms, i.e., Day 1 data is used for training and Day 2 is used as test data and vice versa. The reported results are the average of the results on two different test data sets.

The alphabet size $|\Sigma|$ for the SDF feature extraction algorithm is chosen to be 30 as has been reported optimal for target classification in previous study.[9] The regularization parameters, and the dictionary size in cases where dictionary learning is used, are chosen based on minimizing the classifications cost on a small validation set.

For TSRC, the tree-structured set of groups is selected to be $\mathcal{G} = \{g_1, g_2, g_3, g_4, g_5, g_6\} = \{\{1\}, \{2\}, \{3\}, \{1, 2, 3\}, \{4\}, \{1, 2, 3, 4\}\}$ where 1, 2 and 3 refer to the seismic channels and 4 refers to the PIR channel. The relative weights for different groups are chosen to satisfy $\omega_{g_1} = \omega_{g_2} = \omega_{g_3} = \omega_{g_5} = 0.001\omega_{g_4}$ and $\omega_{g_6} = 10\omega_{g_4}$. This leaves the tuning to be done using only one parameter ω_{g_3} which is selected using validation set. The underlying assumption is that the three seismic channels are correlated and therefore, the weight for group g_4 is selected to be relatively big compared to the weights for individual seismic sources to encourage joint sparsity between the three seismic channels. However, the weight for g_6 is the biggest one which encourages collaboration among all sources. Although the value of correlation between different modalities are different, but the tree-structured sparsity allows collaboration at different granularities. After the relative relations of different weights are selected apriori, the value of ω_{g_3} is chosen using cross validation. It is noted that the results of the tree-structured sparsity is not sensitive to the exact values of the weights and similar results are obtained with different set of weights as long as the underlying prior information is correctly formulated.

A straightforward way of utilizing the unsupervised and supervised (single-modal) dictionary learning algorithms for multimodal classification is to train independent dictionaries and classifiers for each modality and

then combine the individual scores for a fused decision. The corresponding algorithms are referred as UDL and SDL. This way of fusion is equivalent to using the ℓ_{11} norm on $\boldsymbol{A}$, instead of ℓ_{12} norm, in Eq. (7) which does not enforce row sparsity in the sparse coefficients and is a decision-level fusion algorithm. The number of dictionary atoms for UDL and SDL is chosen to be 20, 10 per class, and the quadratic cost[19] is used. The dictionaries for JSRC and JDSRC are constructed using all available training samples.

The performances of the presented classification algorithms under different sparsity priors are also compared with those of the several state-of-the-art decision-level and feature-level fusion algorithms. For decision level fusion algorithms linear support vector machine (SVM)[20] and logistic regression (LR)[20] classifiers are trained on individual modalities and the fused decision is obtained by combining the score of individual modalities. These approaches are abbreviated as *SVM-Sum* and *LR-Sum*, respectively. The performance of the algorithms are also compared with feature-level fusion methods including the holistic sparse representation classifier (HSRC),[33] the joint dynamic sparse representation classifier (JDSRC),[33] relaxed collaborative representation (RCR),[38] and multiple kernel learning (MKL).[53] The HSRC is a simple modification of SRC for multimodal classification in which the feature vectors from different sources are concatenated into a longer feature vector and SRC is applied on the concatenated feature vectors. JDSRC and RCR are also recently applied to a number of applications with better performance than JSRC. For the MKL algorithm, linear, polynomial, and RBF kernels are used.

Table 1 summarizes the average human detection rate (HDR), human false alarm rate (HFAR), and correct classification rates (CCR) obtained using different multimodal classification algorithms. As seen, the sparsity based feature-level fusion algorithm, JDSRC, JSRC, and TSRC have resulted in better performances than the competing algorithms with TSRC achieving the best classification result. Moreover, it is seen that the structured sparsity prior has indeed resulted in improved performance compared to the HSRC algorithm. While the SDL algorithm result in similar performance compared to the counterpart HSRC algorithm, it should be noted that SDL enjoys more compact dictionaries. Developing multimodal dictionaries using structured sparsity[51] can potentially improve the results which is a topic of future research.

	HDR	HFAR	CCR
SVM-Sum	0.94	0.13	91.22%
LR-Sum	0.97	0.13	92.57%
RCR	0.94	0.12	91.22%
MKL	0.94	**0.08**	93.24%
HSRC	0.96	0.10	93.24%
JSRC	1.00	0.12	94.59%
JDSRC	0.97	**0.08**	94.59%
TSRC	1.00	0.09	95.65%
UDL	0.92	0.13	89.86%
SDL	0.94	0.08	93.24%

6. Conclusions

This chapter has presented an overview of symbolic dynamic filtering as a feature extraction algorithm for time-series data. The recent developments in the area of sparse representation for multimodal classification have also been presented. For this purpose, structured sparsity priors, which enforce collaboration across different modalities, are studies. In particular, the tree-structured sparsity model allows extraction of cross-correlated information among multiple modalities at different granularities. Finally, unsupervised and supervised dictionary learning algorithms were reviewed. The performances of the discussed algorithms are evaluated for the application of target classification. The results show that the feature fusion algorithms using sparsity models achieve superior performance as compared to the counter-part decision-fusion algorithms. An interesting topic of future research includes development of multimodal dictionary learning algorithms under different structured sparsity priors.

References

1. R. A. Gramann, M. Bennett, and T. D. Obrien, Vehicle and personnel detection using seismic sensors, *Sensors, C3I, Information, and Training Technologies for Law Enforcement*. **3577**(1), 7485 (1999).
2. J. Altmann, Acoustic and seismic signals of heavy military vehicles for cooperative verification, *Proc. of the IEEE*. **273**(4-5), 713740 (2004).
3. Y. Tian and H. Qi. Target detection and classification using seismic signal processing in unattended ground sensor systems. In *ICASSP*, pp. 4172–4172 (2002).
4. D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, Detection, classification, and tracking of targets, *IEEE SPM*. **19**(2), 17–29 (2002).

5. G. Succi, D. Clapp, R. Gampert, and G. Prado, Footstep detection and tracking, *Unattended Ground Sensor Technologies and Applications III*. **4393** (1), 22–29 (2001).
6. A. Ray, Symbolic dynamic analysis of complex systems for anomaly detection, *Signal Processing*. **84**(7), 1115–1130 (July, 2004).
7. X. Jin, S. Sarkar, A. Ray, S. Gupta, and T. Damarla, Target detection and classification using seismic and PIR sensors, *IEEE SJ*. **12**(6), 1709–1718 (2012).
8. G. Mallapragada, A. Ray, and X. Jin, Symbolic dynamic filtering and language measure for behavior identification of mobile robots, *IEEE TSMC*. **42** (3), 647–659 (2012).
9. S. Bahrampour, A. Ray, S. Sarkar, T. Damarla, and N. Nasrabadi, Performance comparison of feature extraction algorithms for target detection and classification, *Pattern Recognition Letters*. pp. 2126–2134 (2013).
10. D. L. Hall and J. Llinas, An introduction to multisensor data fusion, *Proc. IEEE*. **85**(1), 6–23 (January, 1997).
11. H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang. Sensor fusion using Dempster-Shafer theory. In *Proc. 19th IEEE Instrum. and Meas. Technol. Conf. (IMTC)*, pp. 7–12 (2002).
12. A. Ross and R. Govindarajan. Feature level fusion using hand and face biometrics. In *SPIE proc. series*, pp. 196–204 (2005).
13. S. Pirooz Azad, S. Bahrampour, B. Moshiri, and K. Salahshoor. New fusion architectures for performance enhancement of a pca-based fault diagnosis and isolation system. In *SAFEPROCESS*, pp. 852–857 (2009).
14. D. Ruta and B. Gabrys, An overview of classifier fusion methods, *Comput. and Inf. syst.* **7**(1), 1–10 (2000).
15. Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *Proc. 26th IEEE Int. Conf. Mach. Learning (ICML)*, pp. 521–528 (2011).
16. S. Kim and E. Xing, Tree-guided group lasso for multi-task regression with structured sparsity, *arXiv:0909.1373* (2009).
17. S. Bahrampour, A. Ray, N. M. Nasrabadi, and W. K. Jenkins. Quality-based multimodal classification using tree-structured sparsity. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pp. 4114–4121 (2014).
18. J. Mairal, F. Bach, A. Zisserman, and G. Sapiro. Supervised dictionary learning. In *Advances Neural Inform. Process. Syst. (NIPS)*, pp. 1033–1040 (2008).
19. J. Mairal, F. Bach, and J. Ponce, Task-driven dictionary learning, *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(4), 791–804 (Apr., 2012).
20. C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer (2006).
21. N. H. Nguyen, N. M. Nasrabadi, and T. D. Tran. Robust multi-sensor classification via joint sparse representation. In *Proc. 14th Int. Conf. Information Fusion (FUSION)*, pp. 1–8 (2011).
22. S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way, 3rd ed.* Academic Press (2008).
23. K. Mukherjee and A. Ray, State splitting and merging in probabilistic finite state automata for signal representation and analysis, *Signal Processing*. **104**,

105–119 (2014).
24. B. Smith, P. Chattopadhyay, A. Ray, S. Phoha, and T. Damarla. Performance robustness of feature extraction for target detection and classification. In *Proc. American Control Conf.*, pp. 3814–3819 (June, 2014).
25. S. Sarkar, A. Ray, A. Mukhopadhyay, and S. Sen, Dynamic data-driven prediction of lean blowout in a swirl-stabilized combustor (2015).
26. V. Rajagopalan and A. Ray, Symbolic time series analysis via wavelet-based partitioning, *Signal Processing*. **86**(11), 3309–3320 (2006).
27. A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. SIAM (1994).
28. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, Robust face recognition via sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.* **31** (2), 210–227 (Feb., 2009).
29. X. Mei and H. Ling, Robust visual tracking and vehicle classification via sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(11), 2259–2272 (Nov., 2011).
30. H. Zhang, Y. Zhang, N. M. Nasrabadi, and T. S. Huang, Joint-structured-sparsity-based classification for multiple-measurement transient acoustic signals, *IEEE Trans. Syst., Man, Cybern.* **42**(6), 1586–98 (Dec., 2012).
31. Y. A. Zuev and S. Ivanov, The voting as a way to increase the decision reliability, *Journal of the Franklin Institute*. **336**(2), 361–378 (1999).
32. T. M. Chen and R. C. Luo, Multilevel multiagent based team decision fusion for autonomous tracking system, *Mach. Intell. Robot. Control*. **1**(2), 63–69 (1999).
33. H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang. Multi-observation visual recognition via joint dynamic sparse representation. In *Proc. IEEE Conf. Comput. Vision (ICCV)*, pp. 595–602 (2011).
34. A. Rattani, D. R. Kisku, M. Bicego, and M. Tistarelli. Feature level fusion of face and fingerprint biometrics. In *Proc. 1st IEEE Int. Conf. Biometrics: Theory, Applicat., and Syst.*, pp. 1–6 (2007).
35. S. Shekhar, V. Patel, N. M. Nasrabadi, and R. Chellappa, Joint sparse representation for robust multimodal biometrics recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(1), 113–126 (Jan., 2013).
36. U. Srinivas, H. Mousavi, C. Jeon, V. Monga, A. Hattel, and B. Jayarao, Simultaneous sparsity model for histopathological image representation and classification, *IEEE Trans. Med. Imag.* **33**(5), 1163 – 1179 (May, 2014).
37. H. S. Mousavi, V. Srinivas, U.and Monga, Y. Suo, M. Dao, and T. D. Tran. Multi-task image classification via collaborative, hierarchical spike-and-slab priors. In *IEEE Intl. Conf. Image Processing (ICIP)*, pp. 4236–4240 (2014).
38. M. Yang, L. Zhang, D. Zhang, and S. Wang. Relaxed collaborative representation for pattern classification. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pp. 2224–2231 (2012).
39. S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Joint sparsity-based robust multimodal biometrics recognition. In *European Conf. Comput. Vision*, pp. 365–374 (2012).
40. J. Yang and Y. Zhang, Alternating direction algorithms for ℓ_1-problems in

compressive sensing, *SISC.* **33**(1), 250–278 (2011).
41. J. Mairal, F. Bach, J. Ponce, and G. Sapiro, Online dictionary learning for sparse coding, *Proc. 26th Annu. Int. Conf. Mach. Learning (ICML).* pp. 689–696 (2009).
42. J. Mairal, M. Elad, and G. Sapiro, Sparse representation for color image restoration, *IEEE Trans. Image Process.* **17**(1), 53–69 (Jan., 2008).
43. M. Yang, L. Zhang, J. Yang, and D. Zhang. Metaface learning for sparse representation based face recognition. In *Proc. IEEE Conf. Image Process. (ICIP)*, pp. 1601–1604 (2010).
44. Y. L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pp. 2559–2566 (2010).
45. Z. Jiang, Z. Lin, and L. S. Davis, Label consistent K-SVD: Learning a discriminative dictionary for recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(11), 2651–2664 (Nov., 2013).
46. M. Aharon, M. Elad, and A. Bruckstein, K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (Nov., 2006).
47. J. Mairal, F. Bach, J. Ponce, and G. Sapiro, Online learning for matrix factorization and sparse coding, *The J. of Mach. Learning Research.* **11**, 19–60 (2010).
48. L. Bottou and O. Bousquet. The trade-offs of large scale learning. In *Advances Neural Inform. Process. Syst. (NIPS)*, pp. 161–168 (2007).
49. I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pp. 3501–3508 (2010).
50. Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, pp. 2691–2698 (2010).
51. S. Bahrampour, N. M. Nasrabadi, A. Ray, and W. K. Jenkins, Multimodal task-driven dictionary learning for image classification, *arXiv:1502.01094* (2015).
52. S. Bahrampour, N. M. Nasrabadi, A. Ray, and W. K. Jenkins. Kernel task-driven dictionary learning for hyperspectral image classification. In *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Process. (ICASSP)* (2015).
53. A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, Simplemkl., *J. of Mach. Learning Research.* **9**(11) (2008).