Multiview Feature Selection for Single-view Classification

Majid Komeili, Narges Armanfard, and Dimitrios Hatzinakos

Abstract—In many real-world scenarios, data from multiple modalities (sources) are collected during a development phase. Such data are referred to as multiview data. While additional information from multiple views often improves the performance, collecting data from such additional views during the testing phase may not be desired due to the high costs associated with measuring such views or, unavailability of such additional views. Therefore, in many applications, despite having a multiview training data set, it is desired to do performance testing using data from only one view. In this paper, we present a multiview feature selection method that leverages the knowledge of all views and use it to guide the feature selection process in an individual view. We realize this via a multiview feature weighting scheme such that the local margins of samples in each view are maximized and similarities of samples to some reference points in different views are preserved. Also, the proposed formulation can be used for cross-view matching when the view-specific feature weights are pre-computed on an auxiliary data set. Promising results have been achieved on nine real-world data sets as well as three biometric recognition applications. On average, the proposed feature selection method has improved the classification error rate by 31% of the error rate of the state-of-the-art.

Index Terms—Feature Selection, Multiview, Feature weighting, Multiview training single view test, Classification.

1 Introduction

In many real-world scenarios, data from multiple modalities or multiple sources are collected during a development phase. Such data are known as multiview data. For example, in biomedical applications, data from MRI and PET images, genetics, cognitive tests and blood biomarkers collected from a set of subjects may be considered as different views. In general, incorporating data from multiple views (sources) has great potential to improve an algorithm's performance e.g. for detection/prediction. However, in many real-world scenarios, these multiple views are only available in the development phase and only a few of them are accessible in the test phase; because for example data collection from all the views is costly and/or time-consuming. In such cases, an algorithm is indeed appealing if it can incorporate relevant information from the multiple views available in the development phase to improve performance in the testing phase where only a few views are available. In the extreme case, only a single view is available in the testing phase i.e. “multiview learning single-view testing” which is the focus of this paper.

Nowadays, data are characterized by hundreds or even thousands of features. However, there is often an insufficient number of data samples to adequately represent the distribution of these high-dimensional feature spaces. Hence, dimensionality reduction is crucial in a wide range of scientific disciplines such as, e.g., the medical field, where oligonucleotide microarray data are used for the identification of cancer-associated gene expression profiles [1], [2], [3]. In this case, the number of available samples is less than a hundred, while the raw data are characterized by thousands of features.

Multiview dimensionality reduction approaches can be classified into two categories: dimension reduction and feature selection. The former often transforms samples of different views into a shared subspace [4], [5], [6], [7], [8] through combining original features to find a new set of features; hence extracted features lose their physical interpretation in terms of the original features. Feature selection approaches perform dimensionality reduction, with no transformation, by selecting a subset of the original features. Hence, feature selection approaches retain the physical interpretability property in terms of the selected features. In this paper, we consider the feature selection aspect of the dimensionality reduction for multiview data analysis.

Feature selection on multiview data has been mostly explored in an unsupervised setting with applications mainly in clustering [9], [10], [11], [12]. Such approaches cannot make use of data labels. There are few works on multiview feature selection in a supervised setting [13], [14]. However, they select a global subset across all views assuming that all views are available during both development and testing phases. To fit such a global feature subset in the “multiview learning, single-view testing” scenario, trivially, one may pick only features belonging to the single view. However such features may not be optimal for that single view because there might be some not-selected features which are effective for the single view but not effective when considering all views together when performing feature selection. Performing feature selection on individual views may alleviate this issue but it does not benefit from information in other views. Therefore the existing feature selection

- *M. Komeili is with the School of Computer Science, Carleton University, Ottawa, ON, Canada.*
E-mail: majid.komeili@carleton.ca
- *N. Armanfard is with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada.*
- *D. Hatzinakos is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada.*

methods are not suitable for the challenging problem of multiview learning, single-view testing.

The main contributions of the paper are as follows: 1) We propose a method for transferring knowledge from all available views to improve the feature selection process of a particular view. 2) The proposed approach is the first attempt to address the problem of multiview learning for single-view classification through feature selection. We refer to the proposed algorithm as the MVSV method. MVSV searches the view of interest just like conventional single-view feature selection methods with the addition of transferring some knowledge from other views. On the other hand, it reduces the number of variables involved in classification just like the multiview dimension reduction methods and yet preserves the interpretability of the input features and hence offers the potential for improved interpretability. Unlike the existing multiview feature selection methods, MVSV does not return a global feature subset. It considers all views just like the multiview feature selection methods but selects only from the view of interest. The proposed method makes no assumptions about the distribution of the data over the sample space. Therefore, it allows irregular and/or disjoint distributions of samples. The proposed MVSV algorithm is formulated as a convex optimization problem that can be readily solved and converge to its unique optimal solution.

The remaining portion of this paper is organized as follows: Section 2 briefly reviews the related work. Details of the proposed method for multiview feature selection are presented in Section 3. In Section 4, experimental results, which demonstrate the performance of the proposed method over a range of real-world applications, are presented. Conclusions are drawn in Section 5.

2 Related Work

The proposed method is related to single-view feature selection in the sense that it returns a feature subset from a particular view. On the other hand, it is related to multiview learning in the sense that it incorporates information from other views. In this section, we discuss the related work in both areas.

2.1 Single-view Feature Selection

Generally, feature selection approaches can be divided into two categories: supervised and unsupervised. Supervised approaches use label information to guide the selection process whereas unsupervised approaches aim to describe the structure of data in some feature space in the absence of label information [15], [16], [17], [18]. In this study, we focus on the supervised approaches.

From another perspective feature selection methods can be roughly categorized into filters, wrappers and embedded methods. Wrapper approaches such as sequential forward selection and sequential backward selection [19] evaluate a feature subset based on the accuracy of a specific classifier on a specific data set. A model for each candidate feature subset is trained and then tested and its performance is used to guide the feature selection process. wrappers are computationally very intensive especially if the chosen model is complex. The other drawback of such algorithms is the high risk of over-fitting because they are tuned to a specific model. Embedded methods, embeds feature selection in classification [20], [21]. Filter methods evaluate a feature subset based on its information content instead of optimizing the performance of any specific classifier. We focus on the filter approaches which are relatively faster as they do not require classifier training during feature selection.

Different criterion functions have been proposed for filter methods. Some feature selection approaches are based on mutual information and usually use some heuristics to handle the relevance-redundancy trade-off [22], [23], [24], [25], [26]. Some other feature selection algorithms are based on a maximum margin criteria [2], [27], [28], [29], [30], [31]. These methods are sample-based where the “margin” is defined as the difference between distance to the nearest same class sample (near-hit) and the nearest sample from opposite classes (near-miss). Relief [28] selects features that are statistically relevant to the target. The drawback of this method is that the nearest miss and nearest hit samples are computed in the original space. This was addressed in Simba algorithm [27] by reevaluation of the margins. However, its objective function is not convex and suffers from many local minima. Later, in [2] a local margin-based feature selection approach was presented in which uses a local learning approach to decompose a complex nonlinear problem into a set of locally linear problems within a large margin framework. However, the above methods are inherently single view and are not able to benefit from the information available in other views. Since only one view is available in the testing phase, the above methods can be applied to that view during the training phase and the resulting features can be used for model training and testing in the test view. The drawback of such an approach is that it has no mechanism to incorporate the other views that are available in the training phase. We argue that involving other views could help because they potentially contain additional information about the classes involved in the model training and testing.

2.2 Multiview Dimension Reduction

Multiview dimension reduction methods can be roughly divided in two categories: multiview subspace learning and Multiview feature selection.

Multiview subspace learning approaches often transform samples of different views into a common space where comparison is done in the common space. They can be divided into unsupervised and supervised approaches. Unsupervised approaches such as Canonical Correlation Analysis (CCA) [32] and its kernel version [32] project the views onto a common space such that the correlation between the projected views is maximized. The work in [33] is a deep extension of CCA based on deep neural networks and autoencoders. Supervised methods [4], [5], [6], [7], [8] use class labels to learn a discriminant common space. The methods in [5] and [6] consider inter-view discriminant information while maximizing the between-class and minimizing the within-class variations. In [7] a discriminative shared Gaussian process latent variable model is used to discover correlations between different views. The method in [8] extracts uncorrelated features in each view and computes transformations of each view such that the extracted features contain minimum redundancy.

Multiview feature selection approaches aim to select a subset of features from multiview data and can be roughly divided into supervised and unsupervised. Most of the unsupervised feature selection methods learn features and common cluster structures across views and enforce a consensus on the cluster indicators from different views [34], [35], [36]. Generated pseudo labels are often combined with sparse learning. [11], [37], [38], [39]. On the other hand, supervised multiview feature selection methods can benefit from the class labels. The method in [13] uses Lasso and a low-rank weight matrix to measure the weights of samples. The method in [14] considers the importance of each view and guides the feature selection such that more important views contribute more to the final feature set. The importance of each view needs to be provided by an expert as an input to the algorithm. The above methods consider the multiview structure of data but their search space includes all the features. Therefore, they return a feature subset that is globally optimal. Such a feature subset, when split into views, produces view-specific subsets that are not necessarily optimal for individual views. This could happen for example when an informative feature of a view is not in the global feature subset because it has been considered redundant due to the presence of another feature from a different view. However, a globally redundant feature, may not be redundant in its corresponding view. The above approaches are unable to transfer knowledge from all available views to guide the feature selection process in an individual view. The proposed method is the first attempt to address this issue. To the best of our knowledge, research on supervised multiview feature selection has rarely been conducted [13], [14].

3 Multiview Feature Selection for Single-View Classification

3.1 Problem Definition

Let $\mathcal{D} = \left\{\mathbf{v}_1^{(i)}, \ldots, \mathbf{v}_V^{(i)}, z^{(i)}\right\}_{i=1}^{M} \subset \mathbb{R}^{J_1} \times \ldots \times \mathbb{R}^{J_V} \times \mathcal{Z}$ be a multiview dataset consisting of M samples for which V views are available. $\mathbf{v}_j^{(i)}$ is the i-th sample in the j-th view. $z^{(i)} \in \mathcal{Z}$ is class label of the i-th sample, where $\mathcal{Z} = \{Z_1, \ldots, Z_C\}$ is the set of all class labels. The multiview dataset $\mathcal{D}$ can be treated as a dataset with two overlapping views: a student view $\mathcal{X}$ and a teacher view $\mathcal{Y}$. The student view $\mathcal{X}$ consists of the view that is available during both training and testing. The teacher view $\mathcal{Y}$ includes all V views available during training. Without loss of generality, assume v_1 is the view that is available during testing. Therefore, we can reorganize $\mathcal{D}$ as $\mathcal{D} = \left\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, z^{(i)}\right\}_{i=1}^{M} \subset \mathbb{R}^{J_x} \times \mathbb{R}^{J_y} \times \mathcal{Z}$ where $\mathbf{x}^{(i)} = \mathbf{v}_1^{(i)}$ and $\mathbf{y}^{(i)}$ is defined as concatenation of all views –i.e. $\mathbf{y}^{(i)} = \left[\mathbf{v}_1^{(i)\intercal}, \ldots, \mathbf{v}_V^{(i)\intercal}\right]^{\intercal}$. Therefore $\mathcal{X} \subset \mathcal{Y}$, $J_x = J_1$ and $J_y = J_1 + \ldots + J_V$. The goal is to incorporate the information in the teacher view $\mathcal{Y}$ to guide the feature selection process in the student view $\mathcal{X}$. Classification is then performed on the student view feature subset selected with the guidance of the teacher view. The student view is included in the teacher view to ensure the teacher view is at least as good as the student view in terms of having informative features. All mathematical symbols are listed in Table 16 in Appendix.

3.2 Proposed Method

3.2.1 Overview

MVSV seeks to select a subset of features in the student view $\mathcal{X}$ by incorporating the information in the teacher view $\mathcal{Y}$. We realize this by introducing a feature weight vector for each of the student and teacher views such that the more informative features will receive higher weights. Real-valued feature weight vectors have some advantages over binary weight vectors: the number of features does not need to be specified in advance and optimization can be done using standard methods such as gradient descent. The proposed method jointly learns the feature weight vector of both views. It has two main components, one for encouraging cross-view matching and another one for maximizing class separability.

The cross-view matching term couples the views to each other. It encourages the feature selection process of student view to follow that of the teacher view, so that it may have a better generalization on test samples for which only view $\mathcal{X}$ is provided. Since $\mathcal{X}$ and $\mathcal{Y}$ views have different length, direct comparison does not help. To alleviate this, we represent samples by their distance to some reference points in the weighted space of each view.

The class separability term maximizes the sample margins in both views. For every sample, we consider its two nearest samples, one with the same label (nearest hit) and the other one with a different label (nearest miss). The margin for a sample is defined as the difference between the distance to its nearest miss and distance to its nearest hit. The margin can be seen as how much a sample can wander in the weighted space while still is correctly classified using a 1-nearest neighbour classifier. This term implicitly minimizes the upper bound of the leave-one-out classification error rate of 1-nearest neighbour classifier in the weighted space of each view. In the original space, finding the nearest miss and nearest hit is known to be challenging because of the curse of dimensionality [40]. MVSV iteratively finds the nearest samples in the weighted space and updates feature weights. Since finding the nearest sample at the problem outset may not be reliable, we adopt a probabilistic approach and estimate the margin as expectation over all possible candidates for being the nearest sample. In the remainder of this section, we will describe details of the cross-view matching and class separability components, followed by problem reformulation and final objective function.

3.2.2 Cross-view Matching

Considering the i-th sample, we represent $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ by their distances to N reference points in each view as follows:

$$\phi(\mathbf{x}^{(i)}) = \left(d_x^{(i,1)}, \ldots, d_x^{(i,n)}, \ldots, d_x^{(i,N)}\right) \tag{1}$$

$$\phi(\mathbf{y}^{(i)}) = \left(d_y^{(i,1)}, \ldots, d_y^{(i,n)}, \ldots, d_y^{(i,N)}\right) \tag{2}$$

where $\phi(\mathbf{x}^{(i)})$ and $\phi(\mathbf{y}^{(i)})$ are N dimensional vectors. Considering l_1 distance, $d_x^{(i,n)}$ and $d_y^{(i,n)}$, $n \in \{1, \ldots, N\}$, can be computed as follows:

$$d_x^{(i,n)} = \mathbf{1}^\top \mathbf{d}_x^{(i,n)} \qquad d_y^{(i,n)} = \mathbf{1}^\top \mathbf{d}_y^{(i,n)} \tag{3}$$

where $(\cdot)^\top$ is transpose operator and

$$\mathbf{d}_x^{(i,n)} = |\mathbf{x}^{(i)} - \mathbf{r}_x^{(n)}| \qquad \mathbf{d}_y^{(i,n)} = |\mathbf{y}^{(i)} - \mathbf{r}_y^{(n)}|. \tag{4}$$

The reference points $\mathbf{r}_x^{(n)}$ and $\mathbf{r}_y^{(n)}$, $n = 1, \ldots, N$ can be defined as the class centres or simply all samples. For example in the $\mathcal{X}$ view, in the former, $\phi(\mathbf{x}^{(i)})$ is a $C-1$ dimensional vector corresponding to the distance from $\mathbf{x}^{(i)}$ to the centre of $C-1$ other classes with a class label different than $z^{(i)}$. In the latter, $\phi(\mathbf{x}^{(i)})$ is a $M-1$ dimensional vector corresponding to the distance from $\mathbf{x}^{(i)}$ to $M-1$ all other samples. However, in both cases, $\phi(\mathbf{x}^{(i)})$ and $\phi(\mathbf{y}^{(i)})$ have the same dimension. $|.|$ is element-wise absolute operator. While other options are possible (e.g. Euclidean distance by substituting $|.|$ with $(.)^2$ in (4)), throughout this study we use l_1 distance.

Since $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ different views of the same sample, we want their representations $\phi(\mathbf{x}^{(i)})$ and $\phi(\mathbf{y}^{(i)})$ be similar. We realize this goal by "weighting" the feature space such that the cross-view matching error $||\phi(\mathbf{x}^{(i)}) - \phi(\mathbf{y}^{(i)})||_2$ be minimized. Equation (3) defines the distances in the original space. Similarly, we can compute distances in the weighted space as follows:

$$d_x^{(i,n)} = \mathbf{f}_x^\top \mathbf{d}_x^{(i,n)} \qquad d_y^{(i,n)} = \mathbf{f}_y^\top \mathbf{d}_y^{(i,n)} \tag{5}$$

where $\mathbf{f}_x$ and $\mathbf{f}_y$ are nonnegative weight vectors. Hence, the problem of minimizing the cross-view error in the weighted spaces can be stated as follows:

$$\min_{\mathbf{f}_x, \mathbf{f}_y} \sum_{i=1}^{M} ||\mathbf{f}_x^\top \mathbf{D}_x^{(i)} - \mathbf{f}_y^\top \mathbf{D}_y^{(i)}||_2^2 \quad \text{s.t.} \quad \mathbf{f}_x, \mathbf{f}_y \geq 0 \tag{6}$$

$\mathbf{D}_x^{(i)}$ and $\mathbf{D}_y^{(i)}$ are difference matrices where their columns are the difference vectors with respect to the i-th sample and are computed as follows:

$$\mathbf{D}_x^{(i)} = \left(\mathbf{d}_x^{(i,1)}, \ldots, \mathbf{d}_x^{(i,n)}, \ldots, \mathbf{d}_x^{(i,N)}\right) \tag{7}$$

$$\mathbf{D}_y^{(i)} = \left(\mathbf{d}_y^{(i,1)}, \ldots, \mathbf{d}_y^{(i,n)}, \ldots, \mathbf{d}_y^{(i,N)}\right). \tag{8}$$

Our formulation builds on the second-order distances [40] and extend it to multiview settings. Second-order distances are characterized in [40], in the context of distribution clustering where they proved that the distance between samples almost always depends only on the mean and variances of the underlying distributions and not on samples values. The second-order features are defined as columns of the affinity matrix and they showed that if i^{th} and j^{th} samples are from the same distribution-cluster, then i^{th} and j^{th} rows/columns of the affinity matrix will be near identical. In the proposed method, (1) and (2) may be seen as the second-order distances in the induced spaces parametrized by feature weights.

3.2.3 *Class Separability*

In addition to minimizing the cross-view match error, we want to maximize the class separability. To this end, we maximize the margin around each sample. For the sake of brevity, in the following, only equations for $\mathcal{X}$ view are described. Driving the equations for $\mathcal{Y}$ view is straightforward. Let $\ell_x^{(i)}$ be the margin of the i-th sample:

$$\ell_x^{(i)} = \mathbf{f}_x^\top \mathbf{d}_x^{(i)} \tag{9}$$

where $\mathbf{d}_x^{(i)} = \mathbf{d}_{NM_x}^{(i)} - \mathbf{d}_{NH_x}^{(i)}$ and $\mathbf{d}_{NM_x}^{(i)}$ and $\mathbf{d}_{NH_x}^{(i)}$ are absolute difference vectors determined as follows:

$$\mathbf{d}_{NM_x}^{(i)} = \left|\mathbf{x}^{(i)} - \mathrm{NM}(\mathbf{x}^{(i)})\right|, \ \mathbf{d}_{NH_x}^{(i)} = \left|\mathbf{x}^{(i)} - \mathrm{NH}(\mathbf{x}^{(i)})\right| \tag{10}$$

where $\mathrm{NM}(\mathbf{x}^{(i)})$ is the nearest neighbor of $\mathbf{x}^{(i)}$ with a different class label (nearest miss) and $\mathrm{NH}(\mathbf{x}^{(i)})$ is the nearest neighbor of $\mathbf{x}^{(i)}$ with the same class label as $\mathbf{x}^{(i)}$ (nearest hit). Intuitively, a positive margin allows the sample to wander in the sample space and still be correctly classified by a nearest neighbor classifier in a leave-one-out fashion, i.e. a better generalization on unseen data.

At the problem outset, $\mathbf{f}_x$ and $\mathbf{f}_y$ are unknown. Therefore, determining the nearest miss and nearest hit in the weighted space defined by $\mathbf{f}_x$ and $\mathbf{f}_y$ is a challenging issue. To overcome this issue, we use an iterative approach for computing $\mathbf{f}_x$ and $\mathbf{f}_y$, where at each iteration $\mathbf{f}_x$ and $\mathbf{f}_y$ are determined based on the distances in the weighted space defined at the previous iteration. However, due to presence of lots of noisy features, determining $\mathrm{NM}_x(\mathbf{x}^{(i)})$ and $\mathrm{NH}_x(\mathbf{x}^{(i)})$ in the original space, may not be accurate. To address this issue, margin is estimated as the expectation of $\ell_x^{(i)}(\mathbf{f}_x)$ over all possible candidates for $\mathrm{NM}_x(\mathbf{x}^{(i)})$ and $\mathrm{NH}_x(\mathbf{x}^{(i)})$ as follows:

$$\bar{\ell}_x^{(i)}(\mathbf{f}_x) = \mathbf{f}_x^\top \bar{\mathbf{d}}_x^{(i)} \tag{11}$$

where

$$\bar{\mathbf{d}}_x^{(i)} = \bar{\mathbf{d}}_{NM_x}^{(i)} - \bar{\mathbf{d}}_{NH_x}^{(i)} \tag{12}$$

and

$$\bar{\mathbf{d}}_{NM_x}^{(i)} = \mathbf{D}_{NM_x}^{(i)} \mathbf{p}_{NM_x}^{(i)^\top} \qquad \bar{\mathbf{d}}_{NH_x}^{(i)} = \mathbf{D}_{NH_x}^{(i)} \mathbf{p}_{NH_x}^{(i)^\top} \tag{13}$$

$\mathbf{D}_{NM_x}^{(i)}$ and $\mathbf{D}_{NH_x}^{(i)}$ are matrices whose columns are absolute difference vectors with respect to $\mathbf{x}^{(i)}$:

$$\mathbf{D}_{NM_x}^{(i)} = \left(\left|\mathbf{x}^{(i)} - \mathbf{x}^{(\mathcal{M}^i(1))}\right|, \ldots, \left|\mathbf{x}^{(i)} - \mathbf{x}^{(\mathcal{M}^i(p))}\right|\right) \tag{14}$$

$$\mathbf{D}_{NH_x}^{(i)} = \left(\left|\mathbf{x}^{(i)} - \mathbf{x}^{(\mathcal{H}^i(1))}\right|, \ldots, \left|\mathbf{x}^{(i)} - \mathbf{x}^{(\mathcal{H}^i(q))}\right|\right). \tag{15}$$

$\mathcal{M}^i$ and $\mathcal{H}^i$ with cardinality of p and q denote set of all possible candidates for $\mathrm{NM}(\mathbf{x}^{(i)})$ and $\mathrm{NH}(\mathbf{x}^{(i)})$ respectively and are defined as:

$$\mathcal{M}^i = \left\{j \in \{1, \ldots, M\} \mid z^{(j)} \neq z^{(i)}\right\} \tag{16}$$

$$\mathcal{H}^i = \left\{j \in \{1, \ldots, M\} \mid z^{(j)} = z^{(i)}, j \neq i\right\}. \tag{17}$$

$\mathbf{p}_{NM_x}^{(i)}$ ($\mathbf{p}_{NH_x}^{(i)}$) in eq. (13) is an p-dimensional (q-dimensional) row vector indicates the probability of samples in $\mathcal{M}^i$ ($\mathcal{H}^i$) being $\mathrm{NM}_x(\mathbf{x}^{(i)})$ ($\mathrm{NH}_x(\mathbf{x}^{(i)})$). Within the weighted space, samples situated closer to $\mathbf{x}^{(i)}$ are more probable to be the nearest sample. So the probabilities are determined as follows:

$$\mathbf{p}_{NM_x}^{(i)} = \exp\left(\frac{-\mathbf{f}_x^\mathsf{T}\mathbf{D}_{NM_x}^{(i)}}{\sigma}\right), \tag{18}$$

$$\mathbf{p}_{NH_x}^{(i)} = \exp\left(\frac{-\mathbf{f}_x^\mathsf{T}\mathbf{D}_{NH_x}^{(i)}}{\sigma}\right), \tag{19}$$

where σ is a user settable parameter. $\mathbf{p}_{NM_x}^{(i)}$ and $\mathbf{p}_{NH_x}^{(i)}$ are then normalized to sum to one to be the probabilities utilized in (13).

Having $2 \times M$ margins of the form $\bar{\ell}_x^{(i)}(\mathbf{f}_x) = \mathbf{f}_x^\mathsf{T}\bar{\mathbf{d}}_x^{(i)}$ and $\bar{\ell}_y^{(i)}(\mathbf{f}_y) = \mathbf{f}_y^\mathsf{T}\bar{\mathbf{d}}_y^{(i)}$, it is desired to maximize all margins. Considering a logistic regression formulation, the optimization problem can be expressed as follows:

$$\max_{\mathbf{f}_x,\mathbf{f}_y} \sum_{i=1}^{M} \mathcal{G}\left(\mathbf{f}_x^\mathsf{T}\bar{\mathbf{d}}_x^{(i)}\right) + \sum_{i=1}^{M} \mathcal{G}\left(\mathbf{f}_y^\mathsf{T}\bar{\mathbf{d}}_y^{(i)}\right), \quad \text{s.t.} \quad \mathbf{f}_x, \mathbf{f}_y \geq 0, \tag{20}$$

where $\mathcal{G}(\cdot)$ is a logistic function.

$$\mathcal{G}(b) = \log\left(\frac{1}{1+\exp(-b)}\right) \tag{21}$$

$\mathcal{G}$ is a strictly increasing function, therefore maximizing $\mathcal{G}\left(\mathbf{f}_x^\mathsf{T}\bar{\mathbf{d}}_x^{(i)}\right)$ indeed implies maximizing $\mathbf{f}_x^\mathsf{T}\bar{\mathbf{d}}_x^{(i)}$. However, $\mathcal{G}$ is useful because it can take an input that can vary from negative to positive infinity whereas the output always ranges between 0 and 1. Equation (20) can be simplified as follows:

$$\min_{\mathbf{f}_x,\mathbf{f}_y} \sum_{i=1}^{M} \log\left(1+\exp\left(-\mathbf{f}_x^\mathsf{T}\bar{\mathbf{d}}_x^{(i)}\right)\right) + \log\left(1+\exp\left(-\mathbf{f}_y^\mathsf{T}\bar{\mathbf{d}}_y^{(i)}\right)\right)$$
$$\text{s.t.} \quad \mathbf{f}_x, \mathbf{f}_y \geq 0. \tag{22}$$

3.2.4 Problem Reformulation and Objective Function

We define $\mathbf{f} = \left(\mathbf{f}_x^\mathsf{T}, \mathbf{f}_y^\mathsf{T}\right)^\mathsf{T}$ and rewrite (22) in terms of $\mathbf{f}$ as follows:

$$\min_{\mathbf{f}} \sum_{i=1}^{2M} \log\left(1+\exp\left(-\mathbf{f}^\mathsf{T}\mathbf{r}^{(i)}\right)\right), \quad \text{s.t.} \quad \mathbf{f} \geq 0 \tag{23}$$

where $\mathbf{r}^{(i)}, i = 1, \ldots, 2M$ are columns of the following matrix:

$$\mathbf{R} = \begin{pmatrix} \bar{\mathbf{d}}_x^{(1)} & \cdots, & \bar{\mathbf{d}}_x^{(M)} & \mathbf{0} & \cdots, & \mathbf{0} \\ \mathbf{0} & , & \mathbf{0} & , \bar{\mathbf{d}}_y^{(1)} & , & \bar{\mathbf{d}}_y^{(M)} \end{pmatrix} \tag{24}$$

and $\mathbf{0}$ is a zero matrix of appropriate size. We also rewrite (6) in terms of $\mathbf{f}$. As shown in Appendix A, the cross-view matching error in (6) can be expressed as follows:

$$\sum_{i=1}^{M} ||\mathbf{f}_x^\mathsf{T}\mathbf{D}_x^{(i)} - \mathbf{f}_y^\mathsf{T}\mathbf{D}_y^{(i)}||_2^2 = \sum_{i=1}^{M} \mathbf{f}^\mathsf{T}\mathbf{D}^{(i)}\mathbf{f} = \mathbf{f}^\mathsf{T}\mathbf{D}\mathbf{f} \tag{25}$$

where

$$\mathbf{D}^{(i)} = \begin{pmatrix} \mathbf{D}_x^{(i)}\mathbf{D}_x^{(i)^\mathsf{T}} & -\mathbf{D}_x^{(i)}\mathbf{D}_y^{(i)^\mathsf{T}} \\ -\mathbf{D}_y^{(i)}\mathbf{D}_x^{(i)^\mathsf{T}} & \mathbf{D}_y^{(i)}\mathbf{D}_y^{(i)^\mathsf{T}} \end{pmatrix} \tag{26}$$

and $\mathbf{D} = \sum_{i=1}^{M}\mathbf{D}^{(i)}$. The final objective function consists of a terms for maximizing the sample margins as in (23), a term for minimizing the cross-view matching error as in (25) and, an additional term to encourage sparsity:

$$\min_{\mathbf{f}} \frac{1}{2M}\sum_{i=1}^{2M} \log\left(1+\exp\left(-\mathbf{f}^\mathsf{T}\mathbf{r}^{(i)}\right)\right) + \lambda\mathbf{f}^\mathsf{T}\mathbf{D}\mathbf{f} + \gamma|\mathbf{f}|_1$$
$$\text{s.t.} \quad \mathbf{f} \geq 0 \tag{27}$$

```
Input: $\mathcal{D} = \left\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, z^{(i)}\right\}_{i=1}^{M}$, $\sigma$, $\lambda$, $\gamma$
Output: $\{\mathbf{f_x}, \mathbf{f_y}\}$
Initialization: Set $\mathbf{f} = (1, \ldots, 1)^\mathsf{T}$;
repeat
    $\mathbf{f}_{\text{prev.}} = \mathbf{f}$;
    for $i \leftarrow 1$ to $M$ do
        Compute $\mathbf{D}_x^{(i)}$ and $\mathbf{D}_y^{(i)}$ as in (7) and (8);
         Compute $\mathbf{D}^{(i)}$ as in (26);
        Compute $\mathbf{D}_{NM_x}^{(i)}$ and $\mathbf{D}_{NM_y}^{(i)}$ as in (14);
        Compute $\mathbf{p}_{NM_x}^{(i)}$,$\mathbf{p}_{NM_y}^{(i)}$ using $\mathbf{f}_{prev.}$ as in (18);
        Compute $\bar{\mathbf{d}}_{NM_x}^{(i)}$ and $\bar{\mathbf{d}}_{NM_y}^{(i)}$ as in (13);
        Compute $\mathbf{D}_{NH_x}^{(i)}$ and $\mathbf{D}_{NH_y}^{(i)}$ as in (15);
        Compute $\mathbf{p}_{NH_x}^{(i)}$, $\mathbf{p}_{NH_y}^{(i)}$ using $\mathbf{f}_{prev.}$ as in (19);
        Compute $\bar{\mathbf{d}}_{NH_x}^{(i)}$ and $\bar{\mathbf{d}}_{NH_y}^{(i)}$ as in (13);
        Compute $\bar{\mathbf{d}}_x^{(i)}$ and $\bar{\mathbf{d}}_y^{(i)}$ as in (12);
    end
    Compute $\mathbf{R}$ as in (24) ;
    Compute $\mathbf{D} = \sum_{i=1}^{M}\mathbf{D}^{(i)}$ ;
    Compute $\mathbf{f}$ through solving (27);
until $\|\mathbf{f} - \mathbf{f}_{prev.}\|_\mathbf{2} < \varepsilon$;
Get $f_x$ and $f_y$ form $f$.
```

Algorithm 1: Pseudo code of the proposed algorithm.

where λ and γ are user settable parameters. Equation (27) is a constrained convex optimization problem with respect to $\mathbf{f}$. We convert it to an unconstrained problem so it can be solved via gradient descent. To reformulate the problem as an unconstrained problem, we replace $\mathbf{f}$ with a new vector $\mathbf{a}$ such that $f_j = a_j^2, \quad 1 \leq j \leq J$ where $J = J_x + J_y$. Therefore, the problem is reformulated as follows:

$$\min_{\mathbf{a}} \frac{1}{2M}\sum_{i=1}^{2M} \log\left(1+\exp\left(-\sum_{j=1}^{J} a_j^2 r^{(i)}(j)\right)\right)$$
$$+ \lambda\sum_{i,j=1}^{J} a_i^2 d_{ij} a_j^2 + \gamma\sum_{i=1}^{J} a_i^2 \tag{28}$$

$r^{(i)}(j)$ is the j-th element of vector $\mathbf{r}^{(i)}$ and d_{ij} denotes elements of matrix $\mathbf{D}$. Equation (28) can be solved via gradient decent with step size of τ and updated as follows:

$$\mathbf{a} \leftarrow \mathbf{a} - \tau\mathbf{\Delta}$$

$$\Delta = \mathbf{a} \otimes \left(\gamma\mathbf{1} + \lambda\sum_{j} a_j^2 d_{ij} - \sum_{i=1}^{2M} \frac{\exp\left(-\sum_j a_j^2 r^{(i)}(j)\right)}{1+\exp\left(-\sum_j a_j^2 r^{(i)}(j)\right)}\mathbf{r}^{(i)}\right) \tag{29}$$

where $\otimes$ is Hadamard operator and τ is the learning rate determined by the standard line search. $\mathbf{f}$ is initialized to $\mathbf{1}$, so that all weights are the same at the beginning and then will be updated using (29) until a stopping criterion is satisfied. In this study, the algorithm stops when the difference between the weights in two successive iterations is less than a threshold ε which is set to 0.01. The pseudo-code of the proposed method is presented in Algorithm 1.

The proposed method offers the potential for improving the performance in a particular view by incorporating the information from other views. The teacher view may contain additional information about the involved classes, and hence can help to guide the feature selection process of a particular view which in turn could improve the performance in that view. This has been examined through extensive experiments in section 4. Besides, the cross-view matching term of our objective function allows comparing two samples from different views using their second-order features. We will further study this in section 4.3.

4 Experimental Results

The performance of the proposed method MVSV is demonstrated by performing a large-scale experiment on ten real-world binary classification problems and is compared against eight well-known feature selection algorithms including Logo [2], FMS [41], MetaDistance [42], JMI [22], CIFE [24], ICAP [23], RELEIF [28], and KCSM [43].

Details of real-world data sets are summarized in Table 1. The total number of available samples in each case is the sum of entries in columns 2 (# train) and 3 (# test). Following [2], [44], [45], the performance of the various feature selection algorithms on each data set is evaluated using a bootstrapping algorithm. To this end, each algorithm is run 10 times on each data set. For each run, the number of data points as shown in column 2 of Table 1 is randomly selected to be the training set, and the remaining samples (whose number is indicated in the third column of Table 1) are used as test samples for that run. The proposed method is most appropriate for challenging problems where only a small number of samples are available for training. Instead of demanding more training samples, MVSV seeks to improve the generalization by incorporating additional views available during training. Since some of the data sets are imbalanced, the percentage of the minority class per data set is indicated in the last column of Table 1.

All data sets (except "DNA") presented in Table 1 are microarray data sets where in each case the number of features is significantly larger than the number of samples. Each feature variable in all the real-world data sets have been transformed into their z-score values. These real data sets represent applications where expensive feature selection methods such as an exhaustive search cannot be used directly.

To evaluate the performance of the proposed multiview feature selection method, at each run, we simulate for each data set a multiview data set by randomly selecting P% of the available features to be used as the student view (to form $\mathbb{R}^{J_x}$ discussed in Section 3), where all available J_y features are used to form the teacher view. The average performance over 10 runs, for a value of P, is recorded. For a fair comparison between feature selection algorithms, the training and test sets for each run is common for all algorithms.

TABLE 1
Characteristics of the real-world data sets used in the experiments. The last column is the percentage of the samples in the minority class.

Data set	#Train	#Test	#Features (J_y)	%Minority
ALLAML [46]	50	22	7129	35
BreastColon [47]	50	580	10936	45
BreastKidney [47]	50	554	10936	43
BreastOvary [47]	50	492	10936	37
ColonKidney [47]	50	496	10936	48
Lukemi [48]	50	72	7070	35
ProstateCancer [49]	50	136	12600	43
ProstateGE [46]	50	102	5966	49
Ovarian [50]	50	253	15154	36
DNA [43]	50	3186	280	48

The code for our comparison feature selection methods are all available on the respective author's websites, with the exception of KCSM, which was obtained directly from the author. The default settings for each algorithm are used: In FMS, the parameter γ is set to -0.5. In MetaDistance, the number of the nearest neighbour samples k and the parameter λ are set to 3 and 200 respectively. Also p is set to 1 which corresponds to l_1 distance. In Logo σ and the regularization parameter λ are set to 2 and 1 respectively. In KCSM, the regularization parameters λ and μ are set to 0.99 and 0.001 respectively.

The proposed method has three user-defined parameters: σ, λ and γ (see Section 3). The parameter σ controls the local behavior of the proposed algorithm. A large value of σ increases the effect of far samples in determining the nearest sample. If σ goes to infinity, all samples will have an equal probability to be the nearest. Therefore the nearest sample would be the average of all other samples i.e no localization. On the other side, if σ goes to zero, we experimentally realized that the algorithm will not converge. The second parameter λ controls the trade-off between maximum margin criterion and cross-matching error. A large value of λ emphasizes reducing cross-matching error while a small value of λ emphasizes maximizing margins i.e. better discrimination. The third parameter γ controls sparsity of the solutions, i.e. optimal feature weights. Generally, these parameters can be estimated through cross-validation and be tuned for each data set to provide the most accurate classification results. However, in this case, for a fair comparison, they are not tuned and set respectively to $2, 10^{-4}, 10^{-2}$, i.e. default values. These values are fixed during all the experiments discussed in this section, on all data sets. The proposed algorithm is implemented in MATLAB and executed on a desktop with an Intel Core i7-2600 CPU @ 3.4 GHz and 16 GB RAM.

4.1 Classification accuracy

An SVM classifier with an RBF kernel is used to estimate the classification accuracy corresponding to the features selected by each feature selection algorithm on each data set. To this end, after performing feature selection on the training samples, an SVM classifier with the top-α features is trained with training data and tested on the test data.

Default values for the SVM classifier are used for both the training and the test phase. In our experiments, α ranges from 1 to 30 since there is no performance improvement for larger values.

The minimum classification error and the corresponding standard deviation as determined by the bootstrapping procedure described earlier is presented in Tables 2 to 5 where the parameter P is respectively set to 2%, 3%, 10% and 25%. The classification error rate in Tables 2 to 5 is a *micro* measure i.e. misclassifications over all samples. However, for imbalanced data sets, this micro measure implicitly gives more weight to the majority class. For example, consider an imbalanced binary test set where the split of classes is 20% vs 80%. Assume that all test samples are classified as the majority class. Although the classifier has learned nothing useful, the error rate would be 20% which does not reflect the poor performance on the minority class. To reflect the performance in the presence of imbalanced data, a *macro* classification error rate which is the average of the error rate of individual classes is also reported in Tables 6 to 9. In macro error, both classes have equal weights. In the above example, the macro error rate would be (0+100)/2=50%, which for a binary classification problem means the model has learned nothing useful. This reflects the poor performance on the minority class.

Among the nine algorithms, the proposed multiview feature selection algorithm MVSV provides the best results in all data sets, where P={2%, 3%, 5% 25%}. The best result for each data set is shown in bold. The last row of Tables 2-9 shows the classification error rates averaged over all data sets followed by the win/tie/loss of each algorithm when compared to ours at the 0.05 p-value level. The results are based on Student's paired two-tailed t-test. On average the proposed method outperforms the best of the comparison methods in each experiment by 31%. For reference, the classification error rate of the SVM classifier performed on the data sets without any feature selection is also reported in Tables 10 and 11. To help SVM perform better, samples are first projected into a lower-dimensional space using PCA, normalized and then fed to SVM. We considered both lossless and lossy dimension reductions. In the lossless case, all the energy is preserved –i.e. the resulting dimension is $M - 1 = 49$. In the lossy case, only the top 5 directions corresponding to the largest eigenvalues are preserved. Our method outperforms PCA+SVM combination on average by a margin of about 10%. Note that PCA mixes the input dimensions and therefore the physical interpretation in terms of the original features is lost.

4.2 Correct feature selection

The data set "DNA", presented in the last row of Table 1, is generally used for detecting the "presence" or "absence" of a splice junction in a given deoxyribonucleic acid (DNA) sequence [43]. It has been previously shown that improved performance in most cases is observed if the attributes closest to the junctions are used [43], [51]. These attributes correspond to features indexed from 61 to 120. Furthermore, the last 100 features of this data set are artificial irrelevant features independently sampled from a zero-mean and unit-variance Gaussian distribution. We, therefore, have a good idea beforehand what the good features are, and thus have an available "ground truth" for this example [45].

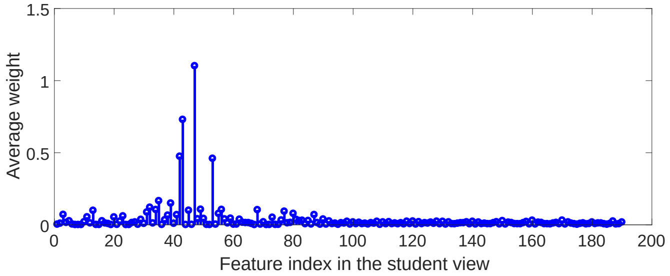

Fig. 1. Feature weights for "DNA" data set in the student view. The height corresponding to each index indicates the average of the corresponding weights over the 10 experimental trials. In this figure, indexes 1 to 89 are corresponding to features 1 to 179. Indexes 91 to 190 are corresponding to the artificially added irrelevant features.

Fig. 1 shows the result of applying our MVSV method to the data set "DNA", where the height of each feature index indicates the average of the corresponding weights over the 10 experimental runs. For illustrative purposes, features with odd indexes between 1 to 180 (i.e. 90 features) besides all the 100 artificially added irrelevant features (indexes 181 to 280) are employed in the student view while all available features $J_y = 280$ are considered for the teacher view. The results demonstrate that the proposed method mostly identifies features with indexes between 70 to 106 (i.e. indexes 35 to 53 in the student view). Thus they are well-matched to the "ground truth". The proposed method also performs very well in discarding the artificially added irrelevant features, i.e. features with indexes between 181 to 280 (i.e. indexes 91 to 190 in the student view).

The reader may be interested to know the classification performance of the proposed MVSV method on the "DNA" data set. To answer this question, the minimum classification error of the proposed MVSV and our comparison algorithms are shown in Table 12. These results indicate higher classification accuracy of the MVSV in the "DNA" data set.

4.3 Additional Experiments - Cross-view Matching

The proposed formulation involves a cross-view matching error term. Therefore, one may be interested to know if it can be used for matching samples across different views. In this section, we aim to measure the similarity between two samples situated in different views and decide whether they match? In this context, a view may refer to facial pose in cross pose face recognition, imaging spectrum in near-infrared versus visible light face recognition or heart rate in ECG recognition where heart rate of gallery and probe are different.

In the "multiview feature selection, singe-view testing" problem discussed before, we assumed that different views of a set of samples are given and the goal was to find the feature weight vectors $\mathbf{f}_x$ and $\mathbf{f}_y$. Here, we assume that $\mathbf{f}_x$ and $\mathbf{f}_y$ are given and the goal is to measure the similarity of two unseen samples (e.g. $\mathbf{x}$ and $\mathbf{y}$) situated in different views through evaluating the cross-view matching error $||\phi(\mathbf{x}) - \phi(\mathbf{y})||_2$ in the weighted spaces. More precisely, $\mathbf{f}_x$

TABLE 2
Micro classification error (in percent) of the different algorithms where 2% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	**MVSV**	**Logo**	**FMS**	**MetaDist**	**JMI**	**CIFE**	**ICAP**	**RELEIF**	**KCSM**
ALLAML	**8.2**(6.2)	13.2(11.8)	12.3(6.4)	16.4(10.3)	20.0(10.5)	22.3(10.6)	17.3(10.9)	15.0(11.3)	15.0(6.8)
BreastColon	**7.1**(1.6)	15.4(6.4)	15.2(4.4)	9.5(1.7)	9.9(3.2)	11.0(3.6)	10.2(3.2)	12.7(5.2)	7.8(0.8)
BreastKidney	**6.1**(1.0)	14.0(6.8)	17.4(4.1)	8.4(2.3)	7.5(2.2)	8.7(2.9)	8.6(3.0)	9.4(2.1)	8.1(2.8)
BreastOvary	**10.8**(2.8)	18.0(5.8)	22.6(6.8)	15.5(2.1)	14.9(6.4)	18.7(6.4)	17.5(7.1)	18.6(7.2)	13.4(3.0)
ColonKidney	**5.6**(1.7)	10.6(2.5)	12.5(3.3)	6.3(1.4)	7.6(1.8)	9.7(4.2)	8.8(3.1)	11.4(4.9)	6.6(1.6)
Lukemi	**10.9**(5.3)	16.4(4.4)	13.2(4.5)	13.6(7.1)	12.3(5.7)	13.2(5.0)	15.5(5.3)	13.2(8.1)	16.8(6.8)
ProstateCancer	**30.2**(8.7)	36.6(6.5)	38.3(6.2)	40.1(6.0)	38.1(3.3)	37.9(3.2)	38.4(5.7)	38.3(6.6)	37.7(8.3)
ProstateGE	**20.2**(3.3)	30.8(7.1)	28.3(2.7)	26.2(4.2)	26.3(6.0)	30.4(4.9)	25.4(6.2)	30.0(7.2)	26.0(4.2)
Ovarian	**9.7**(3.0)	19.1(6.3)	19.4(4.2)	22.9(3.0)	16.0(4.9)	17.4(6.6)	15.0(4.4)	13.2(6.5)	15.2(4.8)
Average **(win/tie/loss)**	**12.1** /	19.3 (0/2/7)	19.9 (0/2/7)	17.7 (0/2/7)	17.0 (0/3/6)	18.8 (0/1/8)	17.4 (0/1/8)	18.0 (0/3/6)	16.3 (0/4/5)

TABLE 3
Micro classification error (in percent) of the different algorithms where 3% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	**MVSV**	**Logo**	**FMS**	**MetaDist**	**JMI**	**CIFE**	**ICAP**	**RELEIF**	**KCSM**
ALLAML	**8.2**(9.4)	10.5(5.7)	13.2(9.2)	16.4(8.9)	15.5(12.5)	22.7(11.5)	17.7(12.2)	13.6(11.9)	12.3(9.4)
BreastColon	**7.6**(1.1)	12.6(7.0)	18.4(7.5)	18.5(3.8)	8.3(2.5)	10.9(5.4)	10.9(4.9)	9.1(3.7)	9.1(1.7)
BreastKidney	**7.5**(1.8)	14.0(5.5)	19.0(6.6)	13.1(3.3)	9.3(2.4)	13.1(4.0)	11.4(3.5)	12.3(3.2)	10.3(3.0)
BreastOvary	**10.0**(4.0)	16.9(3.6)	23.5(6.4)	22.6(4.5)	11.8(4.7)	15.7(3.4)	13.3(4.1)	15.7(6.1)	12.8(3.2)
ColonKidney	**5.3**(1.7)	11.5(6.3)	12.3(6.6)	11.1(3.4)	7.1(2.2)	11.6(2.4)	9.9(2.5)	11.6(6.2)	6.9(2.0)
Lukemi	**7.7**(5.2)	10.0(2.9)	8.2(4.7)	15.0(9.8)	11.4(2.4)	13.2(3.4)	10.9(3.2)	13.6(5.7)	11.4(5.8)
ProstateCancer	**25.5**(7.6)	29.1(7.4)	35.5(8.3)	37.1(6.0)	35.9(7.1)	36.4(8.6)	35.5(9.0)	36.0(7.7)	37.9(7.6)
ProstateGE	**17.7**(4.9)	29.2(8.5)	26.3(6.0)	21.5(3.5)	24.0(5.0)	24.0(7.7)	20.8(4.1)	25.4(2.8)	27.3(5.4)
Ovarian	**4.7**(3.0)	14.7(6.3)	18.6(4.4)	18.5(5.2)	11.2(5.4)	16.1(4.4)	15.0(6.3)	11.2(5.9)	16.5(5.4)
Average **(win/tie/loss)**	**10.5** /	16.5 (0/3/6)	19.4 (0/2/7)	19.3 (0/3/6)	14.9 (0/6/3)	18.2 (0/1/8)	16.2 (0/4/5)	16.5 (0/2/7)	16.0 (0/4/5)

TABLE 4
Micro classification error (in percent) of the different algorithms where 10% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	**MVSV**	**Logo**	**FMS**	**MetaDist**	**JMI**	**CIFE**	**ICAP**	**RELEIF**	**KCSM**
ALLAML	**2.3**(2.9)	6.8(5.8)	11.4(11.6)	12.7(8.2)	7.3(6.8)	7.7(6.4)	7.7(6.4)	10.9(14.9)	7.3(4.4)
BreastColon	**7.2**(1.6)	16.4(7.1)	18.1(5.3)	22.6(3.0)	9.8(3.5)	12.6(5.5)	12.8(5.6)	11.3(4.3)	7.8(1.8)
BreastKidney	**6.4**(0.9)	17.5(6.5)	18.2(4.8)	16.7(3.2)	7.4(2.0)	10.8(4.2)	9.9(4.0)	13.4(2.8)	7.3(2.8)
BreastOvary	**10.0**(3.3)	12.6(5.5)	26.5(5.8)	23.6(5.3)	12.2(3.6)	18.6(4.7)	16.1(5.1)	21.7(7.2)	15.2(5.3)
ColonKidney	**5.6**(1.6)	10.6(3.6)	12.4(3.8)	12.5(2.6)	7.5(1.9)	9.9(2.3)	9.5(2.2)	11.2(4.7)	7.3(1.9)
Lukemi	**6.8**(8.2)	19.5(14.2)	10.9(10.1)	23.2(9.4)	9.1(10.3)	9.1(10.3)	10.0(6.7)	10.5(7.1)	9.5(9.4)
ProstateCancer	**20.6**(7.1)	29.4(6.7)	36.6(7.6)	33.3(9.2)	36.9(9.8)	39.1(8.8)	38.5(6.9)	37.7(6.7)	36.9(9.7)
ProstateGE	**10.2**(4.1)	22.5(7.7)	20.2(9.2)	33.7(6.5)	13.8(2.8)	17.3(5.7)	17.9(5.9)	23.8(9.6)	13.3(2.5)
ovarian	**1.8**(1.2)	6.0(1.8)	9.7(3.7)	18.4(8.0)	4.3(3.4)	8.1(2.3)	7.2(2.8)	5.9(3.2)	6.0(2.0)
Average **(win/tie/loss)**	**7.9** /	15.7 (0/1/8)	18.2 (0/1/8)	21.8 (0/0/9)	12.0 (0/3/6)	14.8 (0/1/8)	14.4 (0/1/8)	16.3 (0/2/7)	12.3 (0/3/6)

TABLE 5
Micro classification error (in percent) of the different algorithms where 25% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	**MVSV**	**Logo**	**FMS**	**MetaDist**	**JMI**	**CIFE**	**ICAP**	**RELEIF**	**KCSM**
ALLAML	**1.8**(3.2)	10.5(8.8)	10.5(12.7)	15.0(6.8)	10.0(6.7)	15.9(6.2)	11.4(4.9)	19.5(11.3)	4.1(5.4)
BreastColon	**6.6**(1.8)	18.8(6.6)	24.6(5.6)	25.2(8.3)	7.5(1.9)	11.2(4.8)	9.8(4.5)	14.5(4.5)	6.7(0.6)
BreastKidney	**6.4**(1.9)	15.7(6.5)	21.0(4.2)	27.9(8.7)	6.6(2.0)	11.6(3.4)	10.6(2.5)	13.7(4.1)	6.9(1.9)
BreastOvary	**11.7**(4.0)	17.6(5.0)	25.4(6.2)	34.1(2.8)	13.8(4.2)	18.8(6.5)	17.3(6.6)	22.7(6.7)	14.9(5.9)
ColonKidney	**4.3**(0.9)	11.1(3.9)	15.7(5.0)	17.0(6.9)	6.2(2.0)	10.6(3.6)	10.4(3.6)	12.7(3.9)	5.6(1.7)
Lukemi	**8.6**(4.0)	21.4(9.1)	10.5(8.6)	25.5(10.1)	9.5(6.9)	13.2(7.6)	10.9(7.2)	13.6(6.8)	9.1(6.4)
ProstateCancer	**17.4**(5.3)	29.3(7.4)	36.7(6.9)	35.3(8.2)	29.1(5.1)	30.7(5.6)	31.7(5.7)	34.5(8.1)	34.5(10.9)
ProstateGE	**7.3**(3.4)	21.7(9.7)	13.7(6.2)	42.3(5.9)	11.9(3.5)	13.3(6.2)	12.3(4.8)	12.9(6.0)	10.0(2.5)
Ovarian	**0.6**(0.6)	3.1(1.7)	8.0(8.5)	6.4(5.3)	1.8(1.6)	3.9(0.9)	4.4(1.4)	3.0(1.0)	3.4(0.9)
Average **(win/tie/loss)**	**7.2** /	16.6 (0/0/9)	18.4 (0/1/8)	25.4 (0/0/9)	10.7 (0/4/5)	14.4 (0/1/8)	13.2 (0/2/7)	16.4 (0/1/8)	10.6 (0/6/3)

TABLE 6
Macro classification error (in percent) of the different algorithms where 2% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	MVSV	Logo	FMS	MetaDist	JMI	CIFE	ICAP	RELEIF	KCSM
ALLAML	**9.5** (5.4)	15.6(10.6)	14.1(4.9)	17.5(9.1)	21.0 (11.2)	24.4(10.4)	17.3(11.8)	15.1(10.5)	16.5(5.4)
BreastColon	**7.2** (1.3)	15.6(6.6)	15.3(5.2)	10.0(1.4)	10.1(3.3)	11.3(3.9)	10.4(3.0)	12.8(6.1)	8.4 (0.9)
BreastKidney	**6.2** (0.8)	14.9(8.3)	18.3(3.5)	8.8 (2.2)	7.9(2.5)	9.6(2.9)	9.4 (3.1)	9.4(2.3)	8.5 (3.3)
BreastOvary	**11.7**(2.4)	21.1 (6.1)	27.6(6.2)	18.3(2.1)	16.4(6.0)	21.5(6.7)	20.2(8.0)	23.0(7.2)	15.9(3.1)
ColonKidney	**5.6** (2.0)	10.5(2.7)	12.4(3.7)	6.2 (1.7)	7.5(2.0)	9.6(3.5)	8.7 (3.1)	11.3(4.3)	6.5 (1.4)
Lukemi	**13.2**(4.4)	18.7 (3.8)	15.5(3.4)	15.5(7.3)	14.6(5.9)	15.7(5.9)	18.2(4.2)	15.2(8.9)	20.5(7.6)
ProstateCancer	**33.4**(7.7)	36.1 (5.9)	37.9(7.1)	41.3(4.9)	40.4(3.9)	41.5(4.1)	43.0(5.8)	39.1(5.9)	39.5(10.2)
ProstateGE	**20.8**(4.1)	29.5 (7.2)	27.8(3.2)	25.9(4.1)	25.9(5.7)	29.7(5.0)	24.8(5.0)	29.4(6.6)	25.4(3.9)
Ovarian	**10.3**(3.5)	22.9 (6.6)	23.0(4.2)	27.5(2.6)	17.9(3.7)	19.6(6.7)	17.1(4.2)	14.8(5.9)	16.5(5.0)
Average **(win/tie/loss)**	**13.1** /	20.5 (0/2/7)	21.3 (0/3/6)	19.0 (0/2/7)	18.0 (0/1/8)	20.3 (0/1/8)	18.8 (0/2/7)	18.9 (0/4/5)	17.5 (0/2/7)

TABLE 7
Macro classification error (in percent) of the different algorithms where 3% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	MVSV	Logo	FMS	MetaDist	JMI	CIFE	ICAP	RELEIF	KCSM
ALLAML	**8.7** (7.6)	10.5(5.8)	14.7(9.0)	19.6(7.7)	17.7(10.9)	25.0(10.6)	19.9(12.8)	13.7(10.4)	13.7(11.7)
BreastColon	**7.7** (1.3)	12.5(8.1)	19.1(7.5)	19.4(4.6)	8.4 (3.1)	11.4(5.8)	11.4(5.1)	9.1 (3.4)	9.4 (2.7)
BreastKidney	**7.9** (1.6)	14.4(4.7)	20.1(5.9)	12.5(2.5)	9.1 (2.5)	14.1(3.3)	12.1(3.2)	12.6(3.1)	10.6(3.1)
BreastOvary	**10.9**(3.7)	19.6(3.6)	28.1(5.2)	27.2(3.7)	13.3(5.8)	18.8(3.0)	16.0(3.9)	17.9(5.2)	14.7(2.9)
ColonKidney	**5.2** (1.3)	11.5(7.5)	12.4(6.0)	11.0(3.2)	7.1 (2.5)	11.5(2.2)	9.9 (2.2)	11.5(7.5)	6.8 (2.0)
Lukemi	**9.0** (5.1)	10.5(2.8)	9.1 (4.2)	17.0(11.8)	12.8(2.3)	14.3(3.3)	12.0(2.6)	16.0(6.7)	12.6(4.6)
ProstateCancer	**26.4**(6.5)	30.5(9.2)	37.3(7.4)	39.7(5.5)	37.6(7.5)	38.1(8.4)	36.4(11.0)	38.0(8.2)	39.8(8.5)
ProstateGE	**17.3**(5.3)	28.6(7.5)	25.7(6.4)	21.3(4.2)	23.1(3.8)	23.3(6.1)	20.5(5.0)	24.8(3.6)	26.4(4.3)
Ovarian	**5.6** (3.3)	17.9(7.5)	21.5(4.8)	22.3(6.1)	11.9(5.1)	18.2(3.8)	16.8(6.2)	12.9(5.0)	17.6(5.3)
Average **(win/tie/loss)**	**11.0** /	17.3 (0/4/5)	20.9 (0/2/7)	21.1 (0/2/7)	15.7 (0/3/6)	19.4 (0/1/8)	17.2 (0/1/8)	17.4 (0/2/7)	16.9 (0/3/6)

TABLE 8
Macro classification error (in percent) of the different algorithms where 10% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	MVSV	Logo	FMS	MetaDist	JMI	CIFE	ICAP	RELEIF	KCSM
ALLAML	**3.9** (3.2)	8.1 (5.7)	12.2(12.7)	12.3(9.1)	8.5 (7.3)	8.6 (6.3)	9.2(6.0)	11.3(11.3)	9.2 (3.6)
BreastColon	**7.4** (1.6)	17.0(7.6)	18.6(5.6)	23.1(4.2)	9.7 (3.3)	12.7(5.4)	12.9(6.7)	11.4(3.6)	8.1 (1.9)
BreastKidney	**6.5** (0.7)	18.6(7.8)	19.6(4.5)	17.6(2.5)	7.4 (2.0)	11.6(3.6)	10.5(4.4)	13.3(3.1)	7.8 (2.6)
BreastOvary	**10.6**(3.6)	14.7(6.5)	32.1(5.1)	26.7(4.8)	13.0(3.8)	20.9(4.4)	18.6(4.9)	25.9(7.8)	16.2(4.9)
ColonKidney	**5.6** (1.7)	10.6(3.0)	12.3(4.0)	12.2(2.0)	7.5 (1.9)	9.8 (1.9)	9.4(1.8)	11.3(3.7)	7.3 (2.0)
Lukemi	**7.3** (8.1)	20.9(12.4)	11.9(11.5)	25.7(8.6)	10.0(11.7)	9.7 (7.5)	11.5(6.4)	11.5(7.4)	10.1(11.1)
ProstateCancer	**22.1**(5.8)	31.6(7.1)	38.2(6.9)	33.8(10.8)	38.1(12.0)	40.6(9.4)	39.9(6.2)	40.2(7.1)	38.4(9.5)
ProstateGE	**10.0**(3.5)	21.7(9.1)	19.4(7.8)	32.4(9.5)	13.5(2.8)	17.0(5.4)	17.1(7.2)	23.2(9.5)	13.4(2.3)
ovarian	**2.0** (0.9)	6.8 (1.8)	9.9 (3.4)	20.6(8.1)	4.5 (3.1)	9.3 (2.8)	7.8(2.5)	6.3 (2.5)	6.2 (1.9)
Average **(win/tie/loss)**	**8.4** /	16.7 (0/2/7)	19.4 (0/2/7)	22.7 (0/0/9)	12.5 (0/5/4)	15.6 (0/1/8)	15.2 (0/1/8)	17.2 (0/2/7)	13.0 (0/4/5)

TABLE 9
Macro classification error (in percent) of the different algorithms where 25% of features are considered for the student view. The corresponding standard deviation (in percent) is reported in parenthesis.

Data set	MVSV	Logo	FMS	MetaDist	JMI	CIFE	ICAP	RELEIF	KCSM
ALLAML	**1.3** (3.4)	10.6(9.7)	10.5(14.3)	18.9(8.0)	11.7(7.6)	18.1(6.3)	12.4(3.9)	21.6(8.5)	3.8 (5.6)
BreastColon	**6.7** (1.4)	19.2(7.7)	24.9(4.7)	25.7(9.7)	7.6 (2.1)	11.6(5.2)	10.2(5.1)	14.9(4.1)	7.0 (0.6)
BreastKidney	**6.5** (1.9)	16.5(6.9)	22.8(4.2)	30.5(10.8)	6.6 (2.4)	12.3(2.9)	11.0(2.3)	14.2(4.1)	7.2 (1.7)
BreastOvary	**13.6**(4.8)	20.2(4.4)	30.5(6.6)	45.0(2.8)	14.8(3.8)	22.0(6.0)	20.1(6.7)	26.1(8.3)	16.4(4.9)
ColonKidney	**4.3** (0.7)	11.2(3.8)	15.6(5.8)	16.6(5.4)	6.2 (1.6)	10.6(3.3)	10.4(4.2)	12.8(3.5)	5.6 (2.0)
Lukemi	**10.5**(4.4)	22.8(7.9)	11.5(10.1)	28.6(9.2)	11.2(7.7)	15.5(6.5)	12.5(8.7)	15.3(8.1)	10.8(6.9)
ProstateCancer	**18.1**(4.3)	30.9(5.8)	38.2(7.2)	36.4(8.5)	30.8(6.3)	31.4(5.2)	32.8(5.2)	36.3(9.8)	36.8(8.2)
ProstateGE	**7.3** (2.9)	20.8(9.0)	13.4(7.6)	41.4(6.1)	11.4(3.5)	13.0(5.0)	12.2(3.8)	12.8(4.8)	10.1(3.0)
Ovarian	**0.6** (0.5)	3.6 (2.0)	9.1(8.9)	7.4 (4.0)	2.0 (1.7)	4.6 (0.9)	5.0 (1.1)	3.2 (1.1)	3.8 (0.8)
Average **(win/tie/loss)**	**7.7** /	17.3 (0/0/9)	19.6 (0/2/7)	27.8 (0/0/9)	11.4 (0/4/5)	15.4 (0/1/8)	14.0 (0/2/7)	17.5 (0/1/8)	11.3 (0/6/3)

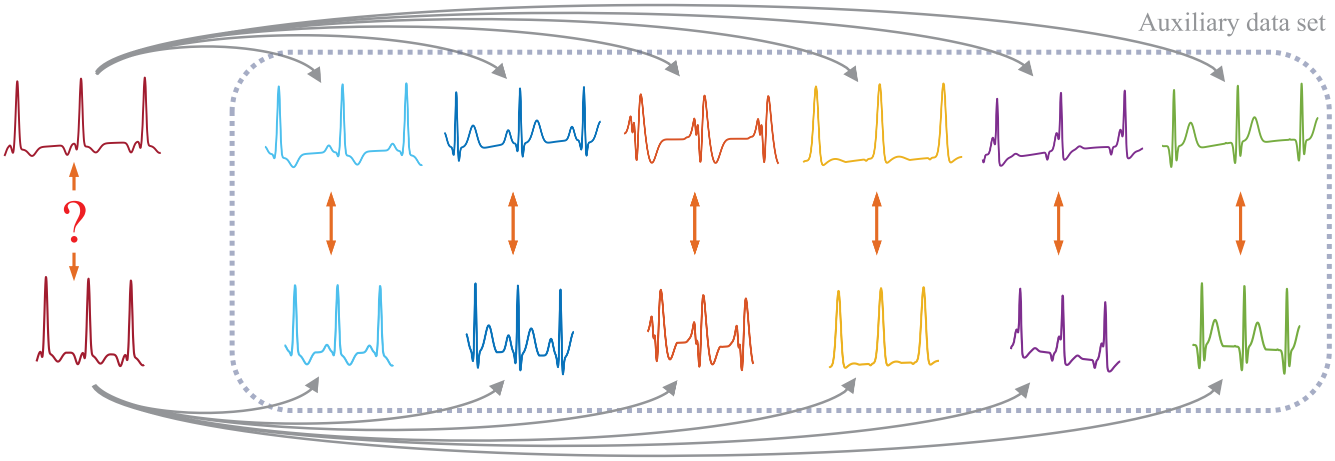

Fig. 2. Illustration of the cross-view matching problem in human recognition using ECG. Two rows correspond to heartbeats in different heart rates (views). The goal is to determine whether the two unseen test samples on the left belong to the same subject. While, direct comparison of samples from different views is not appropriate, their similarities to subjects in the auxiliary data set can be used to compare the two test samples.

TABLE 10
Micro classification error (in percent) using PCA followed by SVM for different values of P. In the second row, All refers to the case where all the energy is preserved after applying PCA, and Top5 refers to the case where only top 5 PCA directions are preserved.

	P=2%		P=3%		P=10%		P=25%	
Data set	**All**	**Top5**	**All**	**Top5**	**All**	**Top5**	**All**	**Top5**
ALLAML	35.9	23.6	35.9	22.7	35.9	20.9	35.9	15.5
BreastColon	49.0	8.3	48.7	8.2	46.0	7.3	46.0	6.9
BreastKidney	46.1	7.6	41.7	9.3	41.3	7.7	43.5	6.9
BreastOvary	36.7	15.3	36.7	14.9	36.6	14.4	36.6	13.6
ColonKidney	49.7	6.4	49.7	7.0	49.7	7.1	49.7	5.6
Lukemi	41.8	26.4	41.8	25.0	41.8	20.5	41.8	16.8
ProstateCancer	46.4	40.1	40.6	40.6	46.4	39.4	46.4	39.2
ProstateGE	51.2	28.3	52.5	31.9	50.6	23.1	51.5	26.5
Ovarian	35.6	20.8	35.6	20.1	35.6	18.1	35.6	19.2
Average	43.6	19.7	42.6	20.0	42.7	17.6	43.0	16.7

TABLE 11
Macro classification error (in percent) using PCA followed by SVM for different values of P. In the second row, All refers to the case where all the energy is preserved after applying PCA, and Top5 refers to the case where only top 5 PCA directions are preserved.

	P=2%		P=3%		P=10%		P=25%	
Data set	**All**	**Top5**	**All**	**Top5**	**All**	**Top5**	**All**	**Top5**
ALLAML	50.0	29.4	50.0	27.0	50.0	26.2	50.0	17.9
BreastColon	49.7	8.5	49.3	8.3	46.1	7.6	46.1	7.2
BreastKidney	49.9	7.5	46.2	9.1	45.8	8.4	48.6	7.6
BreastOvary	50.0	17.8	50.0	17.6	49.9	16.5	49.9	14.5
ColonKidney	50.0	6.3	50.0	6.9	50.0	7.1	50.0	5.6
Lukemi	50.0	30.9	50.0	29.7	50.0	23.9	50.0	19.2
ProstateCancer	50.0	43.5	43.1	43.1	50.0	41.6	50.0	41.4
ProstateGE	47.1	27.6	49.2	31.1	46.9	22.5	48.4	26.1
Ovarian	50.0	26.7	50.0	25.8	50.0	23.1	50.0	24.5
Average	49.6	22.0	48.6	22.1	48.8	19.7	49.2	18.2

and $\mathbf{f}_y$ are pre-computed on an auxiliary multiview data set using (27) and the resulting weights are used to verify if the two new samples situated in different views match. This can be seen as using the proposed formulation in reverse – i.e. a matcher that computes the similarity of two samples situated in different views.

We focus on a challenging scenario where testing has to be done on some new classes not seen during training. This is more challenging than the conventional classification problems where testing is done on some samples that although are new but belong to known classes that have been seen during training. This may arise in a biometric system where training is done on an auxiliary dataset consists of some generic subjects but is intended to recognize subjects not been seen during training –i.e. using the system for new subjects without retraining the system. In this section, we focus on cross-view matching for biometric recognition.

The auxiliary dataset consists of some generic subjects for which samples of all views are available. The auxiliary dataset is used to pre-compute the view-specific feature weights. Testing involves computing the distance between a pair of samples from different views and decide if they belong to the same person. Figure 2 illustrates this problem in the context of human recognition using ECG. Heartbeats in different heart rates (views) have different lengths. Therefore they cannot be directly compared across views. Instead, they can be compared using their similarities to some generic subjects in an auxiliary set in the weighted space. In addition to the cross heart rate ECG recognition, we examine two other applications: near-infrared versus visible light face recognition and cross-pose face recognition. Performance of the proposed method is compared against state-of-the-art methods in multiview dimention reduction including MULDA, MLDA-m, MULDA-m presented in [8], MvDA [6] and GMA [5].

TABLE 12
Minimum classification error (in percent) of the different algorithms on the "DNA" data set where 50% of features 1 to 180 besides 100 irrelevant features (i.e. features 181 to 280) are employed in the student view. The last column corresponds to the classification results using SVM with no feature selection.

Data set	MVSV	Logo	FMS	MetaDist	JMI	CIFE	ICAP	RELEIF	KCSM	SVM
DNA	**16.8**	21.5	18.2	26.9	23.8	23.8	20.7	50.1	45.1	47.8

Since the goal of the experiments in this section is cross-view matching rather than feature selection, we set γ to 0 (–i.e. no sparsity constraint). The other two parameters i.e. σ and λ are set to 3 and 10^{-3} respectively for all three applications. For all comparison methods, the input dimension in each view is reduced using Principal Component Analysis (PCA) such that 95% of energy is preserved. The output dimension of the comparison methods is also tuned with a step size of 50 and the best results are reported. For GMA, as suggested in [5] and [6], μ and γ are set to 1 and the trace ratio respectively, and λ is tuned in [1 100]. For MULDA, MULDA-m and MLDA-m methods, as suggested in [8], γ is tuned among [1,5,10,15,20]. The codes for all comparison methods have been provided by the respective authors.

4.3.1 Experiments of ECG recognition

In this experiment, we use the ECG database collected at the BioSec lab at the University of Toronto [52]. There are 82 subjects that have recordings in up to 5 sessions in rest and after exercise conditions. We take the first 40 subjects as auxiliary dataset and the rest goes for enrollment and testing (i.e. test set). ECG signals were recorded using Vernier EKG sensor and Go!Link interface [53] with 12 bits resolution and a sampling rate of 200 Hz using three dry AgCl electrodes from fingertips. We consider the heart rate between 66 to 100 beats per minute which is equivalent to an RR interval of 120 to 180 samples and this interval is uniformly divided into 15 non-overlapping bins (views). We pick the middle view for enrollment and the remaining 14 views go for testing –i.e 7 views with higher heart rate and 7 views with lower heart rats compare with the enrollment heart rate. In other words, we assume that the subjects in the test set are enrolled by providing ECG samples in the middle view, and testing is performed by comparing enrollment samples against samples with heart rated other than the one used for the enrollment. Therefore, we solve 14 pairs of multiview problems on the auxiliary data set, each consists of the middle view and one of the other 14 views. There is no overlap between the auxiliary set and the test set. In the auxiliary dataset, we randomly select 15 samples to represent each bin. In the test dataset, we randomly pick 20 samples for enrollment and 5 samples for testing.

The length of samples within each view is the same. We consider a window centered at R peak where the length of the window is twice the length of the corresponding bin and compute Continues Wavelet Transform (CWT) with Daubechies 5 as mother wavelet. We also consider six frequency bands: 8-13Hz, 13-18Hz, 18-25Hz, 25-30Hz, 30-35Hz, 35-50Hz and for each band, mean of power, standard deviation of power, maximum amplitude, standard deviation of amplitude, kurtosis and skewness were computed. Maximum, standard deviation, kurtosis and skewness are also computed from the signal itself. The signal amplitude, i.e. a window centered around R peak, is also considered. Autocorrelation of the samples is also computed and the number of lags is 80% of the corresponding bin length. The above features are concatenated. The length of the feature vector of the first view i.e. the slowest heart rate is 6252 and for the last view i.e. the fastest heart rate is 4304. Features are normalized such that they have zero mean and unit variance. To account for the randomness in selecting samples in each view, experiments are repeated 5 times and the average Equal Error Rate (EER) is reported in Table 13. It can be seen that the proposed method significantly performs better than the comparison methods.

4.3.2 Experiments of near-infrared versus visible light face recognition

In this experiment, we use the NIR-VIS database in [54] which consists of 202 subjects for which both near-infrared and visible light samples are provided. We randomly select 4 samples per subject. The auxiliary dataset consists of the first 100 subjects and the testing set consists of the rest of the subjects. Face images are aligned by an affine transform on the center of eyes and center of mouth such that the distance between the center of the eyes is 100 pixels. Illumination is normalized by the method suggested in [55]. SURF features [56] are computed on 64 landmarks detected by the method in [57]. SURF features are also computed from 36 points uniformly distributed in a square area on the inner face area. Also, holistic representation is obtained by cropping and resizing the images to 41 × 36 pixels. These features are concatenated to form a final feature vector of the length 8132. Features are normalized such that they have zero mean and unit variance. The recognition rate of the proposed method and 5 comparison methods are reported in Table 14. It can be seen that the proposed method significantly performs better than the comparison methods. Note that since the recognition rate is the most widely used performance measure in face recognition, we use it in our face experiments.

4.3.3 Experiments of cross pose face recognition

In this experiment, we use the MultiPie face database [58] which consists of 337 subjects for which face images in 15 different poses, 20 illuminations, 6 expressions and 4 sessions are provided. We use the first 100 subjects for auxiliary dataset and the rest goes for testing. We consider images from 4 different sessions and 5 illuminations i.e. 1, 4, 7, 12, 17. Enrolment is done on frontal lighting i.e. illumination 7 and testing is done on the aforementioned 5 lightings. Similar to the NIR-VIS experiment, face images are aligned according to eyes and center of mouth by an affine transform and illumination is normalized by the method suggested in [55]. SURF features are computed on

TABLE 13
ECG recognition across different heart rates. EER (in percent) and standard deviation (in percent) are reported for different methods. Standard deviations are presented in parentheses.

Heart Rate	MULDA	MLDA-m	MULDA-m	GMA	MvDA	MVSV
67.4	13.0(1.3)	13.0(1.5)	13.3(1.1)	14.3(0.0)	16.6(2.4)	12.7(1.1)
69.0	16.0(0.9)	16.6(0.1)	16.5(0.4)	16.7(0.1)	22.9(1.5)	12.5(1.6)
70.6	14.5(0.8)	13.2(1.4)	13.3(1.4)	12.1(1.9)	15.8(0.8)	10.7(1.1)
72.3	13.0(1.7)	13.8(1.9)	13.9(1.7)	11.6(1.4)	13.6(2.2)	10.0(1.0)
74.1	8.2(0.7)	8.8(1.5)	8.8(1.5)	8.6(0.6)	11.9(2.4)	10.2(1.6)
76.0	11.9(1.8)	12.8(1.9)	13.0(1.8)	11.0(0.5)	12.7(2.0)	11.2(1.4)
77.9	12.8(1.9)	13.4(1.4)	13.5(1.4)	13.6(1.3)	15.6(1.2)	8.5(1.4)
82.2	16.5(0.4)	11.8(1.1)	11.6(0.9)	13.0(0.7)	15.3(0.9)	8.1(1.4)
84.5	13.9(2.2)	12.4(2.5)	12.4(2.5)	9.9(0.6)	12.1(0.4)	11.6(1.5)
87.0	16.1(1.7)	17.8(3.8)	17.2(3.8)	11.7(1.1)	14.2(1.8)	10.2(1.7)
89.6	8.8(0.6)	10.3(1.1)	10.5(1.1)	11.4(1.0)	12.6(1.1)	11.1(0.6)
92.3	17.3(1.9)	15.8(2.6)	15.0(2.6)	12.1(0.5)	18.8(5.6)	12.9(1.1)
95.3	25.1(2.4)	21.3(3.0)	21.4(3.0)	20.3(2.2)	20.2(1.7)	11.5(2.3)
98.4	24.6(1.7)	24.4(2.6)	25.0(2.6)	16.2(0.7)	12.9(1.0)	10.6(1.7)
Average	15.1	14.7	14.7	13	15.4	**10.8**
(win/tie/loss)	(2/3/9)	(0/6/8)	(0/6/8)	(1/7/6)	(0/3/11)	/

100 points uniformly distributed on the face area. Holistic features are obtained by cropping and resizing the face images to 45×56 pixels. These features are concatenated to form a final feature vector of the length 8920. Features are normalized such that they have zero mean and unit variance. The recognition rate of the proposed method and 5 comparison methods are reported in Table 15. It can be seen that the proposed method significantly performs better than the comparison methods.

4.3.4 Relation to Few-shot learning and one-shot learning

Cross-view matching problem discussed in section 4.3 is related to few-shot learning in that they both perform testing on unseen classes rather than seen classes. However, they are different in that in few-shot learning, training and testing samples of the unseen classes are in the same space whereas in the cross-view matching problem training and testing samples are in different spaces with possibly different numbers of dimensions and therefore direct comparison does not help. Zero-shot-learning is a special case of few-shot learning where there is no training sample for unseen test classes. Instead, unseen classes are described in a semantic space, usually an attribute space [59] or a continuous embedding space [60]. Therefore, there is no training sample in the space of the test samples. The method for cross-view matching discussed in section 4.3 is closer to the projection-based approaches in zero-shot learning [61], [62]. The method in [61] is based on sparse coding and [62] is based on matrix tri-factorization whereas our method is based on second-order similarity and joint feature weighting which preserves the interpretability of the input features. Also, such methods require the unlabeled test samples from the unseen classes as well as the class representation of the unseen classes to be available during training. In our experimental setup in section 4.3 neither of them is available during training.

TABLE 14
Near-infrared versus visible light face recognition. Recognition rate (in percent) for different methods are reported.

MULDA	MLDA-m	MULDA-m	GMA	MvDA	MVSV
79.26	69.33	69.33	61.14	66.05	**84.6**

5 Conclusion and Future Works

In this paper, we proposed a novel method for feature selection on multiview data such that the information in all views is used to guide the feature selection in an individual view. We realized this via a multiview feature weighting scheme such that the local margins of samples in each view are maximized and similarities of samples with respect to some reference points in different views are preserved. We also examined the application of the proposed method as a matcher where it computes the similarity of two samples situated in different views. The proposed approach has several advantages. First, it makes no assumptions about the distribution of data over the sample space. Therefore, it allows irregular and/or disjoint distributions of samples. In addition, the underlying optimization problem is convex and converges to the optimal solution regardless of the initialization point.

As a future direction of research, we would like to extend the proposed method and equip it with a built-in mechanism for handling missing data. Another direction of interest is to extend the proposed method to multi-task learning where labels for multiple tasks are available which in turn may improve the generalization on unseen data especially when the number of training samples is small.

Appendix

$$\begin{aligned}
||\mathbf{f}_x^\top \mathbf{D}_x^{(i)} - \mathbf{f}_y^\top \mathbf{D}_y^{(i)}||_2^2 &= \mathbf{f}_x^\top \mathbf{D}_x^{(i)} {\mathbf{D}_x^{(i)}}^\top \mathbf{f}_x - \mathbf{f}_x^\top \mathbf{D}_x^{(i)} {\mathbf{D}_y^{(i)}}^\top \mathbf{f}_y \\
&\quad - \mathbf{f}_y^\top \mathbf{D}_y^{(i)} {\mathbf{D}_x^{(i)}}^\top \mathbf{f}_x + \mathbf{f}_y^\top \mathbf{D}_y^{(i)} {\mathbf{D}_y^{(i)}}^\top \mathbf{f}_y \\
&= \begin{pmatrix} \mathbf{f}_x^\top & \mathbf{f}_y^\top \end{pmatrix} \begin{pmatrix} \mathbf{D}_x^{(i)} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} {\mathbf{D}_x^{(i)}}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{f}_x \\ \mathbf{f}_y \end{pmatrix} \\
&\quad - \begin{pmatrix} \mathbf{f}_x^\top & \mathbf{f}_y^\top \end{pmatrix} \begin{pmatrix} \mathbf{D}_x^{(i)} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & {\mathbf{D}_y^{(i)}}^\top \end{pmatrix} \begin{pmatrix} \mathbf{f}_x \\ \mathbf{f}_y \end{pmatrix}
\end{aligned}$$

TABLE 15
Cross pose face recognition. Recognition rate (in percent) for different methods are reported. ah refers to the two additional cameras (08-1 and 19-1) located above the subject, simulating a typical surveillance camera view.

Angle	MULDA	MLDA-m	MULDA-m	GMA	MvDA	MVSV
$-ah$	22.1	19.7	19.6	32.0	47.0	59.6
-90	10.1	9.4	9.5	15.2	22.2	26.5
-75	13.1	12.5	12.6	20.3	34.3	37.1
-60	16.0	15.3	15.3	28.3	45.7	58.6
-45	32.2	29.2	29.0	38.2	62.0	78.6
-30	53.1	47.9	47.5	55.6	80.2	81.8
-15	59.1	55.3	55.2	66.2	87.6	93.6
15	64.5	59.9	60.0	70.4	85.0	93.4
30	38.0	37.0	36.9	47.1	74.2	85.3
45	27.0	24.6	24.5	35.3	60.1	76.1
60	19.5	18.3	18.2	25.8	44.7	62.7
75	14.1	12.3	12.4	17.7	32.2	42.0
90	10.0	9.3	9.0	16.2	24.5	33.8
ah	25.6	22.7	22.8	31.8	47.4	64.6
Average	28.9	26.7	26.6	35.7	53.4	**63.8**

$$\begin{aligned}
&- \begin{pmatrix} \mathbf{f}_x^\top & \mathbf{f}_y^\top \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{D}_y^{(i)} \end{pmatrix} \begin{pmatrix} \mathbf{D}_x^{(i)^\top} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{f}_x \\ \mathbf{f}_y \end{pmatrix} \\
&+ \begin{pmatrix} \mathbf{f}_x^\top & \mathbf{f}_y^\top \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{D_y^{(i)}} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{D}_y^{(i)^\top} \end{pmatrix} \begin{pmatrix} \mathbf{f}_x \\ \mathbf{f}_y \end{pmatrix} \\
&= \mathbf{f}^\top \mathbf{D}_{xx}^{(i)} \mathbf{f} - \mathbf{f}^\top \mathbf{D}_{xy}^{(i)} \mathbf{f} - \mathbf{f}^\top \mathbf{D}_{yx}^{(i)} \mathbf{f} + \mathbf{f}^\top \mathbf{D}_{yy}^{(i)} \mathbf{f} \\
&= \mathbf{f}^\top \mathbf{D}^{(i)} \mathbf{f} \qquad (30)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{f} &= \begin{pmatrix} \mathbf{f}_x^\top & \mathbf{f}_y^\top \end{pmatrix}^\top, \\
\mathbf{D}_{xx}^{(i)} &= \begin{pmatrix} \mathbf{D}_x^{(i)} \mathbf{D}_x^{(i)^\top} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \\
\mathbf{D}_{xy}^{(i)} &= \begin{pmatrix} \mathbf{0} & \mathbf{D}_x^{(i)} \mathbf{D}_y^{(i)^\top} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \\
\mathbf{D}_{yx}^{(i)} &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{D}_y^{(i)} \mathbf{D}_x^{(i)^\top} & \mathbf{0} \end{pmatrix}, \\
\mathbf{D}_{yy}^{(i)} &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_y^{(i)} \mathbf{D}_y^{(i)^\top} \end{pmatrix}, \\
\mathbf{D}^{(i)} &= \mathbf{D}_{xx}^{(i)} - \mathbf{D}_{xy}^{(i)} - \mathbf{D}_{yx}^{(i)} + \mathbf{D}_{yy}^{(i)} \\
&= \begin{pmatrix} \mathbf{D}_x^{(i)} \mathbf{D}_x^{(i)^\top} & -\mathbf{D}_x^{(i)} \mathbf{D}_y^{(i)^\top} \\ -\mathbf{D}_y^{(i)} \mathbf{D}_x^{(i)^\top} & \mathbf{D}_y^{(i)} \mathbf{D}_y^{(i)^\top} \end{pmatrix}.
\end{aligned}$$

ACKNOWLEDGMENTS

This research was supported by Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] L. J. Van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. Van Der Kooy, M. J. Marton, A. T. Witteveen *et al.*, "Gene expression profiling predicts clinical outcome of breast cancer," *nature*, vol. 415, no. 6871, p. 530, 2002.

[2] Y. Sun, S. Todorovic, and S. Goodison, "Local-learning-based feature selection for high-dimensional data analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1610–1626, 2010.

[3] Y. Wang, J. G. Klijn, Y. Zhang, A. M. Sieuwerts, M. P. Look, F. Yang, D. Talantov, M. Timmermans, M. E. Meijer-van Gelder, J. Yu *et al.*, "Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer," *The Lancet*, vol. 365, no. 9460, pp. 671–679, 2005.

[4] Y. Ma, S. Lao, E. Takikawa, and M. Kawade, "Discriminant analysis in correlation similarity measure space," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 577–584.

[5] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2160–2167.

[6] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 188–194, 2016.

[7] S. Eleftheriadis, O. Rudovic, and M. Pantic, "Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition," *IEEE transactions on image processing*, vol. 24, no. 1, pp. 189–204, 2015.

[8] S. Sun, X. Xie, and M. Yang, "Multiview uncorrelated discriminant analysis," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 3272–3284, 2016.

[9] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Subspace clustering guided unsupervised feature selection," *Pattern Recognition*, vol. 66, pp. 364–374, 2017.

[10] X. Zhu, S. Zhang, R. Hu, Y. Zhu *et al.*, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 517–529, 2018.

[11] W. Shao, L. He, C.-T. Lu, X. Wei, and S. Y. Philip, "Online unsupervised multi-view feature selection," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 1203–1208.

[12] F. Nie, W. Zhu, X. Li *et al.*, "Unsupervised feature selection with structured graph optimization." in *AAAI*, 2016, pp. 1302–1308.

[13] W. Yang, Y. Gao, Y. Shi, and L. Cao, "Mrm-lasso: A sparse multiview feature selection method via low-rank analysis," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 11, pp. 2801–2815, 2015.

[14] Y. El-Manzalawy, T.-Y. Hsieh, M. Shivakumar, D. Kim, and V. Honavar, "Min-redundancy and max-relevance multi-view feature selection for predicting ovarian cancer survival using multi-omics data," *bioRxiv*, p. 317982, 2018.

[15] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Adaptive unsupervised feature selection with structure regularization," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 944–956, 2018.

[16] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, and S. Zhang, "Graph self-representation method for unsupervised feature selection," *Neurocomputing*, vol. 220, pp. 130–137, 2017.

[17] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *Cybernetics, IEEE Transactions on*, vol. 44, no. 6, pp. 793–804, 2014.

TABLE 16
List of mathematical symbols

Symbol	Description
$\mathcal{D}$	The data set
M	Number of samples
C	Number of classes
V	Number of available views
$\mathbf{v}_j^{(i)}$	View j of the i-th sample
$\mathcal{X}$, $\mathcal{Y}$	Student and teacher views respectively
$\mathcal{Z}$	Set of class labels
$\mathbf{z}^{(i)}$	Class label of the i-th sample
$\mathbf{x}^{(i)}$	Student view of the i-th sample
$\mathbf{y}^{(i)}$	Teacher view of the i-th sample
J_x, J_y	Dimension of the $\mathcal{X}$ and $\mathcal{Y}$ views
$\phi(\mathbf{x}^{(i)})$	Second-order representation of $\mathbf{x}^{(i)}$
$\mathbf{r}^{(n)}$	Reference point n
N	Number of reference points
$d_x^{(i,n)}$	Distance between $\mathbf{x}^{(i)}$ and $\mathbf{r}^{(n)}$
$\mathbf{d}_x^{(i,n)}$	The absolute difference vector between $\mathbf{x}^{(i)}$ and $\mathbf{r}^{(n)}$
$\mathbf{D}_x^{(i)}$	Difference matrix between $\mathbf{x}^{(i)}$ and reference points
$\mathbf{f}_x$, $\mathbf{f}_y$	Feature weight vector in $\mathcal{X}$ and $\mathcal{Y}$ views
$\ell_x^{(i)}$	Margin of the i-th sample in view $\mathcal{X}$
$\mathbf{d}_x^{(i)}$	Difference vector between $\mathbf{d}_{NM_x}^{(i)}$ and $\mathbf{d}_{NH_x}^{(i)}$
$\mathbf{d}_{NM_x}^{(i)}$	Difference vector between $\mathbf{x}^{(i)}$ and $NM(x^{(i)})$
$\mathbf{d}_{NH_x}^{(i)}$	Difference vector between $\mathbf{x}^{(i)}$ and $NH(x^{(i)})$
$NM(\mathbf{x})$	The nearest miss of $\mathbf{x}$
$NH(\mathbf{x})$	The nearest hit of $\mathbf{x}$
$\bar{\mathbf{d}}_x^{(i)}$	Estimated $\mathbf{d}_x^{(i)}$ in view $\mathcal{X}$
$\bar{\ell}_x^{(i)}$	Estimated margin $\ell_x^{(i)}$ in view $\mathcal{X}$
$\bar{\mathbf{d}}_{NH_x}^{(i)}$	Estimated $\mathbf{d}_{NH_x}^{(i)}$
$\mathbf{D}_{NM_x}^{(i)}$	Difference matrix between $\mathbf{x}^{(i)}$ and $\mathcal{M}^i$ set
$\mathbf{D}_{NH_x}^{(i)}$	Difference matrix between $\mathbf{x}^{(i)}$ and $\mathcal{H}^i$set
$\mathcal{M}^i$	Set of all possible candidates for NM($\mathbf{x}^{(i)}$)
$\mathcal{H}^i$	Set of all possible candidates for NH($\mathbf{x}^{(i)}$)
p, q	Number of samples in $\mathcal{M}^i$ and $\mathcal{H}^i$ respectively
$\mathbf{p}_{NM_x}^{(i)}$	Probability of being the nearest miss of $\mathbf{x}^{(i)}$
$\mathbf{p}_{NH_x}^{(i)}$	Probability of being the nearest hit of $\mathbf{x}^{(i)}$
σ	Kernel width
$\mathcal{G}(\cdot)$	Logistic function
$\mathbf{f}$	Concatenation of $\mathbf{f}_x$ and $\mathbf{f}_y$
λ	The hyperparameter that controls the trade-off between the margin and cross-view terms
γ	The hyperparameter that controls the sparsity of $\mathbf{f}$
ϵ	Stopping threshold
$\mathbf{r}^{(i)}$	i-th column of matrix $\mathbf{R}$
$r^{(i)}(j)$	j-th element of $\mathbf{r}^{(i)}$
$\mathbf{R}$	See equation (24)
$\mathbf{D}^{(i)}$	See equation (26)
$\mathbf{D}$	Sum of $\mathbf{D}^{(i)}$ matrices
a_j	j-th element of $\mathbf{a}$
$\mathbf{a}$	Defined such that a_j^2 is equal to the j-th element of f

[18] M. Banerjee and N. R. Pal, "Unsupervised feature selection with controlled redundancy (ufescor)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3390–3403, 2015.

[19] E. Alpaydin, *Introduction to machine learning*. MIT press, 2004.

[20] J. C. H. Hernandez, B. Duval, and J.-K. Hao, "A genetic embedded approach for gene selection and classification of microarray data," in *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 2007, pp. 90–101.

[21] K. Nag and N. R. Pal, "A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification," *IEEE transactions on cybernetics*, vol. 46, no. 2, pp. 499–510, 2016.

[22] H. H. Yang and J. E. Moody, "Data visualization and feature selection: New algorithms for nongaussian data." in *NIPS*, vol. 99. Citeseer, 1999, pp. 687–693.

[23] A. Jakulin, "Machine learning based on attribute interactions," Ph.D. dissertation, Univerza v Ljubljani, 2005.

[24] D. Lin and X. Tang, "Conditional infomax learning: An integrated framework for feature extraction and fusion," in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, vol. 3951, pp. 68–82.

[25] P. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Applications of Evolutionary Computing*. Springer Berlin / Heidelberg, 2006, vol. 3907, pp. 91–102.

[26] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.

[27] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection-theory and algorithms," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 43.

[28] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the ninth international workshop on Machine learning*. Morgan Kaufmann Publishers Inc., 1992, pp. 249–256.

[29] Y. Sun, "Iterative relief for feature weighting: algorithms, theories, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1035–1051, 2007.

[30] B. Chen, H. Liu, J. Chai, and Z. Bao, "Large margin feature weighting method via linear programming," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 10, pp. 1475–1488, 2009.

[31] B. Liu, B. Fang, X. Liu, J. Chen, Z. Huang, and X. He, "Large margin subspace learning for feature selection," *Pattern Recognition*, vol. 46, no. 10, pp. 2798–2806, 2013.

[32] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[33] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1083–1092.

[34] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 270–278.

[35] Y. Feng, J. Xiao, Y. Zhuang, and X. Liu, "Adaptive unsupervised multi-view feature selection for visual concept recognition," in *Asian conference on computer vision*. Springer, 2012, pp. 343–357.

[36] M. Qian and C. Zhai, "Unsupervised feature selection for multi-view clustering on text-image web news data," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. ACM, 2014, pp. 1963–1966.

[37] K. W. Wangila, K. Gao, P. Zhu, Q. Hu, and C. Zhang, "Mixed sparsity regularized multi-view unsupervised feature selection," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1930–1934.

[38] H. Liu, H. Mao, and Y. Fu, "Robust multi-view feature selection," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 281–290.

[39] P. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Coupled dictionary learning for unsupervised feature selection." in *AAAI*, 2016, pp. 2422–2428.

[40] W.-Y. Lin, S. Liu, J.-H. Lai, and Y. Matsushita, "Dimensionality's blessing: Clustering images by underlying distribution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5784–5793.

[41] Q. Cheng, H. Zhou, and J. Cheng, "The fisher-markov selector: fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 6, pp. 1217–1233, 2011.

[42] Z. Liu, W. Hsiao, B. L. Cantarel, E. F. Drábek, and C. Fraser-Liggett, "Sparse distance-based learning for simultaneous multiclass classification and feature selection of metagenomic data," *Bioinformatics*, vol. 27, no. 23, pp. 3242–3249, 2011.

[43] L. Wang, "Feature selection with kernel class separability," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 9, pp. 1534–1546, 2008.

[44] N. Armanfard, J. P. Reilly, and M. Komeili, "Logistic localized modeling of the sample space for feature selection and classifi-

cation," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1396–1413, 2018.

[45] ——, "Local feature selection for data classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 6, pp. 1217–1227, 2016.

[46] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $l_{2,1}$-norms minimization," in *Advances in Neural Information Processing Systems*, 2010, pp. 1813–1821.

[47] G. Stiglic and P. Kokol, "Stability of ranked gene lists in large microarray analysis studies," *BioMed Research International*, vol. 2010, 2010.

[48] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *The Journal of Machine Learning Research*, vol. 13, pp. 27–66, 2012.

[49] R. Hewett and P. Kijsanayothin, "Tumor classification ranking from microarray data," *BMC genomics*, vol. 9, no. 2, p. S21, 2008.

[50] T. J. Abrahamsen, *Kernel Methods for Machine Learning with Life Science Applications*. DTU Compute, 2013.

[51] G. John. Dna dataset (statlog version) - primate splice-junction gene sequences (dna) with associated imperfect domain theory. [Online]. Available: https://www.sgi.com/tech/mlc/db/DNA.names

[52] Medical biometric databases. http://www.comm.utoronto.ca/ biometrics/databases. Last accessed December 9, 2018.

[53] Vernier ekg sensor. www.vernier.com. Last accessed December 9, 2018.

[54] S. Z. Li, Z. Lei, and M. Ao, "The hfb face database for heterogeneous face biometrics research," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE, 2009, pp. 1–8.

[55] H. Wang, S. Z. Li, Y. Wang, and J. Zhang, "Self quotient image for face recognition," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 2. IEEE, 2004, pp. 1397–1400.

[56] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[57] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[58] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[59] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[60] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2927–2936.

[61] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2452–2460.

[62] X. Xu, F. Shen, Y. Yang, D. Zhang, H. Tao Shen, and J. Song, "Matrix tri-factorization with manifold regularizations for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3798–3807.

Majid Komeili Majid Komeili received his Ph.D. degree in Electrical and Computer Engineering from the University of Toronto, Toronto, ON, Canada in 2017. He is currently an Assistant Professor in the School of Computer Science and Institute for Data Science at Carleton University, Ottawa, ON, Canada. He performs fundamental and applied research in machine learning.

Narges Armanfard Narges Armanfard received her Ph.D. degree in Electrical and Computer Engineering from McMaster University, Hamilton, ON, Canada, in 2016. She is currently an Assistant Professor at the department of Electrical and Computer Engineering at McGill University, Montreal, QC, Canada. Her current research interests include machine learning, pattern recognition and their applications.

Dimitrios Hatzinakos Dimitrios Hatzinakos received the Diploma degree from the University of Thessaloniki, Greece, in 1983, the M.A.Sc degree from the University of Ottawa, Canada, in 1986 and the Ph.D. degree from Northeastern University, Boston, MA, in 1990, all in Electrical Engineering. In September 1990 he joined the Department of Electrical and Computer Engineering, University of Toronto, where now he holds the rank of Professor with tenure. He has served as Chair of the Communications Group of the Department during the period July 1999 to June 2004. He has been the holder of the Bell Canada Chair in Mutimedia from 2004-2014, at the University of Toronto. Also, he is the co-founder and since 2009 the Director of the Identity, Privacy and Security Institute (IPSI) at the University of Toronto. His research interests and expertise are in the areas of Multimedia Signal Processing, Multimedia Security, Multimedia Communications and Biometric Systems. He is author/co-author of more than 350 papers in technical journals and conference proceedings, he has contributed to 18 books and he has 8 patents in his areas of interest. He is a Fellow of IEEE, a Fellow of the Engineering Institute of Canada since February 2012 and recipient of the 2012 University of Toronto Inventor of the year award. He served as an Associate Editor for the IEEE Transactions on Mobile Computing, from 2008-2013. Also, he has served as an Associate Editor for the IEEE Transactions on Signal Processing from 1998 till 2002 and Guest Editor for the special issue of Signal Processing, Elsevier, on Signal Processing Technologies for Short Burst Wireless Communications which appeared in October 2000. He was a member of the IEEE Statistical Signal and Array Processing Technical Committee (SSAP) from 1992 till 1995 and Technical Program co-Chair of the 5th Workshop on Higher-Order Statistics in July 1997. He has been co-chair of the International Symposium on Smart Data organized at the University of Toronto in May 2012. He is a member of the Professional Engineers of Ontario (PEO), and the Technical Chamber of Greece.