A Reconstruction Error Based Framework for Multi-label and Multi-view Learning

Buyue Qian, Xiang Wang, Jieping Ye, and Ian Davidson

Abstract—A significant challenge to make learning techniques more suitable for general purpose use is to move beyond i) complete supervision, ii) low dimensional data, iii) a single label and single view per instance. Solving these challenges allows working with complex learning problems that are typically high dimensional with multiple (but possibly incomplete) labelings and views. While other work has addressed each of these problems separately, in this paper we show how to address them together, namely *semi-supervised dimension reduction for multi-label and multi-view learning* (SSDR-MML), which performs optimization for dimension reduction and label inference in semi-supervised setting. The proposed framework is designed to handle both multi-label and multi-view learning settings, and can be easily extended to many useful applications. Our formulation has a number of advantages. We explicitly model the information combining mechanism as a data structure (a weight/nearest-neighbor matrix) which allows investigating fundamental questions in multi-label and multi-view learning. We address one such question by presenting a general measure to quantify the success of simultaneous learning of multiple labels or views. We empirically demonstrate the usefulness of our SSDR-MML approach, and show that it can outperform many state-of-the-art baseline methods.

Index Terms—*Semi-Supervised Learning; Multi-label Learning; Multi-view Learning; Dimension Reduction; Reconstruction Error.*

1 Introduction

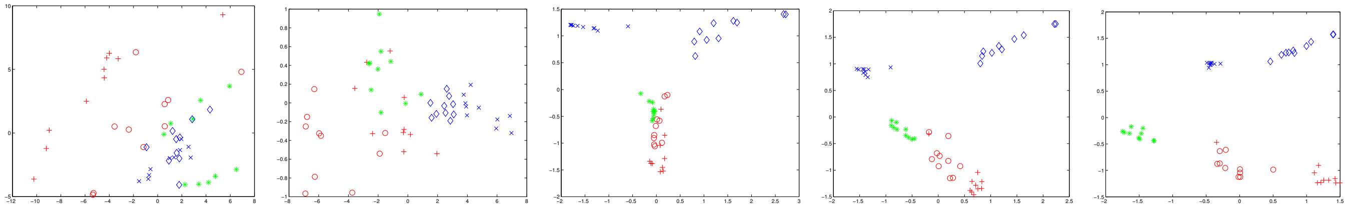
Four core challenges to making data analysis better suited to real world problems is learning from: i) partially labeled data, ii) high dimensional data, iii) multi-label, and iv) multi-view data. Whereas existing work often tackles each of these problems separately, giving rise to the fields of semi-supervised learning, dimension reduction, multi-label and multi-view learning respectively, we propose and show the benefits of addressing the four challenges together. However, this requires a problem formulation that is efficiently solvable and easily interpretable. We propose such a framework which we refer to as semi-supervised dimension reduction for multi-label and multi-view learning (SSDR-MML).

Consider this simple experiment to illustrate the weakness of solving each problem independently. We collect 50 frontal well-aligned face images of five people in ten different expressions, each of which are associated with three labels (besides name): gender, bearded, glasses (see Fig. 1). We shall project the face images into a 2D space using different techniques that perform unsupervised dimension reduction, supervised dimension reduction and finally our approach that simultaneously performs semi-supervised learning and dimension reduction for multi-label data. Fig. 2 shows the result, where each symbol denotes a different person and each color indicates a combination of labels. "Red" stands for female, unbearded, and non-glasses; "green" denotes male, unbearded, and non-glasses; and "blue" indicates male, bearded, glasses. For Principal Component Analysis (PCA) [1], an unsupervised dimension reduction technique, we see that it finds a mapping of the images into a 2D space (Fig. 2(a)) where people with different labels are not well separated. This result is not surprising given PCA does not make use of the labeled data. Using the labels for only 30% of images for supervision, we see that PCA+LDA [2] performs only marginally better in Fig. 2(b) because the missing labels can not be inferred. Our approach simultaneously infers the missing labels and performs dimension reduction for this multi-label data and, as shown in Fig. 2(c) to 2(e), produces accurate predictions and monotonic improvement. During the iterative process, images sharing similar labels gradually aggregate while dissimilar images move further apart.

Fig. 1. Sample face images

A subsequent risk in applying learning techniques to these challenging environments is that it is difficult to know if the learning approach was successful. Elegant frameworks such as structural and empirical risk minimization, though useful, have not been well extended and applied to multi-label and multi-view settings. More empirical evaluation approaches such as cross-fold validation do not work well in sparse label settings. In this work we explore explicitly modeling the mechanism to combine multiple labels and views as a data structure such that we can more clearly see how the labels are

- *B. Qian, X. Wang, and I. Davidson is with the Department of Computer Science, University Of California, Davis, CA, 95616.*
E-mail: {byqian, xiang, indavidson}@ucdavis.edu
- *Jieping Ye is with Computer Science and Engineering, Arizona State University, Tempe, AZ 85287.*
E-mail: jieping.ye@asu.edu



(a) PCA (b) PCA+LDA (c) SSDR-MML Iteration 1 (d) SSDR-MML Iteration 2 (e) SSDR-MML Iteration 3

Fig. 2. Project faces to 2D using different methods. Symbols denote different people, and colors denote attributes.

propagated and the relationship between labels and views. By examining properties of this structure we can determine the success of the learning approach.

Our proposed work makes several contributions. We create a reconstruction error based framework that can model and create classifiers for complex datasets that are multi-label and multi-view, and contain missing labels. Multi-view learning involves multiple feature sets for the same set of objects. Previous work [3], [4] shows that simply concatenating all feature sets into one single view is suboptimal and raises difficult engineering issues if the views are fundamentally different (such as binary in one view and real values in another, or dimension and normalization issues). Our reconstruction error framework makes the following contributions to the field:

- Simultaneously performs dimension reduction, multi-label propagation in a multi-view setting.
- Explicitly models and constructs a sparse graph showing which data points are related.
- Allow a quantification of how successful the learning process was by examining the properties of the graph mentioned above (see Section 7).
- Allow the domain experts to clearly understand where/how the labels were propagated and how the views are complimentary using the graph.
- Allow multi-view learning without assuming conditional independence of views and that each view (by itself) is sufficient to build a weak classifier. This is achieved as we do not serialize the learning problem, instead learning from all views simultaneously.

We begin our paper with a brief review of related studies in Section 2, and then present the general framework in Section 3. Sections 4 and 5 show specific formulations for multi-label and multi-view learning along with the optimization algorithm we use for each. Section 6 shows how to perform dimension reduction using our approach, and Section 7 presents new work that describes a method to determine how successful our approach was in a multi-label/view problem. Section 8 discusses implementation issues and the corresponding solutions. Our experimental section (Section 9) shows the results that compare our work against existing competing techniques. The new experiments include comparing against a larger set of competing algorithms and verifying the usefulness of our success measure of how well the model performs. We conclude our work in Section 10.

Differences to Conference Version. The additional work that is in this paper and not the conference version [5] is: (1) extension to handle both multi-label and multi-view learning, (2) a node regularizer to facilitate the learning with imbalanced labeling, (3) a measure to quantify the success of multi-label and multi-view learning, and (4) extensive new discussions and experiments.

2 Related Work

Our work is related to four machine learning and data mining topics: *multi-label learning*, *multi-view learning*, *multi-label dimension reduction*, and *semi-supervised learning*. Here we review some related work in the four areas.

Multi-label Learning. Multi-label learning (MLL) is motivated by the fact that a real world object naturally involves multiple related attributes, and thereby investigating them together could improve the overall learning performance. MLL learns a problem together with other related problems at the same time [6], that allows the learner to use the commonality amongst the labels. The hope is that by learning multiple labels simultaneously one can improve performance over the "*no transfer*" case. MLL has been studied from many different perspectives, such as neural networks among similar tasks [7], kernel methods and regularization networks [8], modeling task relatedness [9], [10], label set propagation [11], and probabilistic models in Gaussian process [12], [13] and Dirichlet process [14]. Although MLL techniques have been successfully applied to many real world applications, their usefulness are significantly weakened by the underlying relatedness assumption, while in practice some labels are indeed unrelated and could induce destructive information to the learner. In this work, we propose a measure to quantify the success of learning, as to benefit from related labels and reject the combining of unrelated (detrimental) labels.

Multi-view Learning. Practical learning problems often involves datasets that are naturally comprised of multiple views [15]. MVL learns a problem together with multiple feature spaces at the same time [16], that allows the learner to perceive different perspectives of the data in order to enrich the total information about the learning task at hand [17], [18]. [19] has shown that the error rate on unseen test samples can be upper bounded by the disagreement between the classification-decisions obtained from the independent characterizations of the data. Therefore, as a branch of MVL, co-training [3], [20] aims at minimizing the misclassification rate indirectly by reducing the rate of disagreement among the base classifiers. Multiple kernel learning was recently introduced by [21], where the kernels are linearly combined in a SVM framework and the optimization is performed as an semidefinite program or quadratically constrained quadratic program. [22] reformed the problems as a block l_1 formulation in conjunction with Moreau-Yosida regularization, so that efficient gradient based optimization could be performed using sequential minimal optimization techniques while still generating a sparse solution. [23] preserved the block l_1 regularization but reformulated the problem as a semi-infinite linear problem, which can be efficiently solved by recycling the standard SVM implementations and made it applicable to large scale problems. Although the successes of MVL, many existing approaches suffer from their own limitations: conditional independent assumption is important for co-training both theoretically and empirically [24] but it rarely holds in real-world applications; multi-kernel machines are limited to combining multiple kernels in linear manners, and such linear scheme sometimes induces poor data representations. In contrast, our SSDR-MML framework does not require the conditional independence assumption, and the multiple views are fused in a nonlinear fashion.

Multi-label Dimension Reduction. Various dimension reduction methods have been proposed to simplify learning problems, which generally fall into three categories: unsupervised, supervised, and semi-supervised. In contrast to traditional classification tasks where classes are mutually exclusive, the classes in multi-task/label learning are actually overlapped and correlated. Thus, two specialized multi-label dimension reduction algorithms have been proposed in [25] and [26], both of which try to capture the correlations between multiple labels. However, the usefulness of such methods is dramatically limited by requiring complete label knowledge, which is very expensive to obtain and even impossible for those extremely large dataset, e.g. web images annotation. In order to utilize unlabeled data, there are many semi-supervised multi-label learning algorithms have been proposed [27] [28], which solve learning problem by optimizing the objective function over graph or hypergraph. However, the performance of such approach is weakened by the lack of the concatenation of dimension reduction and learning algorithm. To the best of our knowledge, [29] is the first attempt to connect dimension reduction and multi-task/label learning, but it suffers from the inability of utilizing unlabeled data.

Semi-supervised learning. The study of semi-supervised learning is motivated by the fact that while labeled data are often scarce and expensive to obtain, unlabeled data are usually abundant and easy to obtain. It mainly aims to address the problem where the labeled data are too few to build a good classifier by using the large amount of unlabeled data. Among various semi-supervised learning approaches, graph propagation has attracted an increasing amount of interest [30]. [31] introduces an approach based on a random field model defined on a weighted graph over both the unlabeled and labeled data; [32] proposes a classifying function which is sufficiently smooth with respect to the intrinsic structure collectively revealed by known labeled and unlabeled points. [33] extend the formulation by inducing spectral kernel learning to semi-supervised learning, as to allow the graph adaptation during the label diffusion process. Another interesting direction for semi-supervised learning is proposed in [34], where the learning with unlabeled data is performed in the context of Gaussian process. The encouraging results of many proposed algorithms demonstrate the effectiveness of using unlabeled data. A comprehensive survey on semi-supervised learning can be found in [35].

3 The Formulation

Notation. Given a set of n instances containing both labeled and unlabeled data points, we define a general learning problem on multiple labels and multiple feature spaces. In the learning problem, there are p related labels $\mathcal{T} = \{t^1, t^2, \cdots, t^p\}$, each of which can be a multi-class learning task with a given finite label set. For the k-th label t^k, we define a binary classifying function $\mathbf{F}^k \in \mathbb{B}^{n \times c^k}$ on its corresponding label set $\mathcal{C}^k = \{1, 2, \cdots, c^k\}$. Note that each instance will have p label vectors $\{\mathbf{f}^1 \ldots \mathbf{f}^p\}$, containing the label sets for each multi-class label. Then $f^k_{ij} = 1$ iff for the k-th label, the i-th instance belongs to the j-th class (the i-th instance $\in$ class $\mathcal{C}^k_j$) and $f^k_{ij} = 0$ otherwise. Without loss of generality, we assume that the points have been reordered so that for label k, the first l^k points are labeled and the remaining u^k points being unlabeled (where $n = l^k + u^k$ and typically $l^k \ll n$), and construct a prior label matrix $\mathbf{Y}^k \in \mathbb{B}^{l^k \times c^k}$ using the given labels in label t^k. Similarly, for each instance, there are q feature descriptions from different views $\{\mathbf{x}^1_i, \mathbf{x}^2_i, \cdots, \mathbf{x}^q_i\}$ with $\mathbf{x}^k_i$ being the k-th feature description/view of the i-th instance. Our aim is to create an asymmetric graph $G(V, E)$ over the n instances represented by the weight matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, where we set $w_{ii} = 0$ to avoid self-reinforcement. The diagonal node degree matrix is denoted by $\mathbf{D}$, where $d_{jj} = \sum_{i=1}^{n} w_{ij}$. To make this paper more readable, the notations used in this paper obey the following rule: the superscript is used to denote the index of a label or view, and the subscript is used to denote the index of an instance, or an entry in a vector or matrix. This article will often refer to row and column vectors of matrices, for example, the i-th row and j-th column vectors of $\mathbf{W}$ are denoted as $\mathbf{w}_{i\bullet}$ and $\mathbf{w}_{\bullet j}$, respectively. The notations are summarized in Table 1.

The Framework. The key question in multi-label or multi-view learning is how to incorporate the information carried by related labels or feature spaces into the

TABLE 1
Notation Table

Notation	Description
t^i	The i^{th} label
$\mathbf{F}^k, \mathbf{f}^k_{i\bullet}$	Binary classifying matrix, vector (i^{th} instance) for label k
$\mathbf{Y}^k, \mathbf{y}^k_{i\bullet}$	Prior label matrix, vector (i^{th} instance) for label k
$\mathbf{x}^k_i$	Feature vector of i^{th} instance at k^{th} view
$\mathbf{W}, \mathbf{w}_{i\bullet}$	Graph weight matrix, and its i^{th} row vector
$\mathbf{D}$	Graph degree matrix
$\mathbf{V}^k$	Node regularizer for label k
$\mathcal{C}^{\mathbf{k}}$	The label set for label k
$L(\mathbf{z})$	Local covariance matrix of a vector $\mathbf{z}$
n, p, q	Number of instances, labels, and views, respectively
l^k, u^k	Number of labeled, unlabeled instances for label k
I	Identity matrix
$\mathbf{1}, \mathbf{0}$	Vector with all ones (zeros) entries
$\alpha^k, \beta^k, \lambda$	Tuning parameters for view, label, and regularizer

same learning problem. In graph transduction, such kind of information encoding can be expressed in terms of partially fitting the graph to all available labels or views based on their importance or relatedness. As before, we solve the label inference problem by following the intuition that "nearby" points tend to have similar labels, and adopt *reconstruction error* [5], [36]. These nearby points are learnt using our algorithm and encoded in $\mathbf{W}$, and we can then use $\mathbf{W}$ to propagate the labels of labeled points to their neighbors. To ensure that there is no given label is overwritten, we constrain $\mathbf{F}^k_l = \mathbf{Y}^k$. Finally, to ensure that the labels are not excessively propagated, we add a regularization term $\|\mathbf{W}\|^2_{\mathcal{F}}$. The general learning framework can then be formulated as:

$$\begin{aligned}\mathcal{Q}(\mathbf{W},\mathbf{F}) &= \sum_{k=1}^{q} \alpha^k \sum_{i=1}^{n} \|\mathbf{x}^k_i - \sum_{j=1}^{n} w_{ij}\mathbf{x}^k_j\|^2_{\mathcal{F}} + \\ & \quad \sum_{k=1}^{p} \beta^k \sum_{i=1}^{n} \|\mathbf{f}^k_{i\bullet} - \sum_{j=1}^{n} w_{ij}\mathbf{f}^k_{j\bullet}\|^2_{\mathcal{F}} + \lambda\|\mathbf{W}\|^2_{\mathcal{F}} \\ & \text{s.t.} \quad \forall i,\ \mathbf{w}_{i\bullet}\mathbf{1} = 1; \quad \mathbf{F}^k_l = \mathbf{V}^k\mathbf{Y}^k. \end{aligned} \tag{1}$$

where $\|\bullet\|_{\mathcal{F}}$ denotes the Frobenius norm. The first term in Eq. 1 is the reconstruction error over the multiple feature descriptions of instances, the second term is the reconstruction error over the classifying functions for the multiple labels, and the third term is the $\mathcal{L}_2$ regularization term. Note that the two reconstruction errors share exactly the *same* weight matrix $\mathbf{W}$, which enables the multiple labels and views to help each other. The tuning parameter $0 \leq \alpha^k \leq 1$ is determined by the "importance" of feature descriptions at the k-th view, and $0 \leq \beta^k \leq 1$ is decided by the "relatedness" between the label t^k to other labels. λ is a tuning parameter of the regularization term. We will in Section 7 provide a success measure to guide the selection of these parameters. The objective function consists of the reconstruction errors of feature spaces at different views and multiple related labels, as it allows the graph weight partially fits to each of them.

Overcoming Class Imbalance. If one class is more popular than another, there is a chance that even though the labels of the less frequent class are propagated, they are ignored in favor of the popular class. To overcome this, we introduce the matrix $\mathbf{V}^k$ which is a node regularizer [37] to balance the influence of different classes in label t^k. The matrix $\mathbf{V}^k = \operatorname{diag}(\mathbf{v}^k)$ is a function of $\mathbf{Y}^k$ (given labels for label k), and $\mathbf{D}_l$ is the degree matrix of the labeled points calculated from $\mathbf{W}$:

$$\mathbf{v}^k = \sum_{i=1}^{c^k} \frac{\mathbf{y}^k_{\bullet i} \odot \mathbf{D}_l\mathbf{1}}{\left(\mathbf{y}^k_{\bullet i}\right)^T \mathbf{D}_l\mathbf{1}} \tag{2}$$

where $\odot$ denotes Hadamard product, and $\mathbf{1} = [1,1,\cdots,1]^T$. By definition, $\mathbf{V}^k\mathbf{Y}^k$ is a normalized version of the label matrix $\mathbf{Y}^k$ satisfying $\sum_i(\mathbf{V}^k\mathbf{Y}^k)_{ij} = 1$ for $\forall j$. The normalized label matrix $\mathbf{V}^k\mathbf{Y}^k$ enables the highly connected instances to contribute more during the graph diffusion/label propagation process. Since the total diffusion of each class is normalized to one, the influence of different classes is balanced even if the given labels are imbalanced. Equally balanced class generally leads to more reliable solutions. In Section 4 and 5, we will explicitly apply the general framework to multi-label and multi-view learning problems, respectively.

We shall now describe specific solutions for the multi-label and multi-view settings separately. The generalized algorithm for both settings is shown in Table 2.

4 Multi-label Learning

4.1 Formulation

We start with multi-label learning, where there are multiple labels over the same set of feature descriptions of instances. Our motivation is to improve the learning performance on the multiple labels by making use of the commonality among them. Intuitively, multi-label learning could greatly improve the learning performance if the multiple labels are highly correlated. On the contrary, if there is no relatedness between the multiple labels, multi-label learning cannot be beneficial and even could be detrimental. We can better understand this premise by interpreting our work as label propagation. For multi-label learning to be successful, points *with the first label being labeled*, should have this label transferred/propagated to points *with other labels being labeled* and vice-versa. If this propagation is extensive then the approach will be successful. This is the idea behind the math of identifying the success of multi-label/view learning in section 7. Under multi-label setting, the generic framework shown in Eq. (1) is reduced to:

$$\begin{aligned}\mathcal{Q}(\mathbf{W},\mathbf{F}) &= \alpha \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{j=1}^{n} w_{ij}\mathbf{x}_j\|^2_{\mathcal{F}} \\ & \quad + \sum_{k=1}^{p} \beta^k \sum_{i=1}^{n} \|\mathbf{f}^k_{i\bullet} - \sum_{j=1}^{n} w_{ij}\mathbf{f}^k_{j\bullet}\|^2_{\mathcal{F}} + \lambda\|\mathbf{W}\|^2_{\mathcal{F}} \\ & \text{s.t.} \quad \forall i,\ \mathbf{w}_{i\bullet}\mathbf{1} = 1; \quad \mathbf{F}^k_l = \mathbf{V}^k\mathbf{Y}^k. \end{aligned} \tag{3}$$

4.2 Alternating Optimization

The formulation shown in Eq. (3) is a minimization problem involving two variables to optimize. Since this objective is not convex it is difficult to simultaneously recover both unknowns. However, if we hold one unknown constant and solve the objective for the other, we have two convex problems that can be optimally solved in closed form. In the rest of this section, we propose an alternating optimization for the SSDR-MML framework, which iterates between the updates of $\mathbf{W}$ and $\mathbf{F}^k$ until $\mathbf{F}^k$ stabilized. The experimental results (see Table 4 and 6) indicate that converging to the local optima still provides good results and is better than less complex objective functions that are solved exactly.

4.2.1 Update for $\mathbf{W}$

If the classifying function $\mathbf{F}^k$ is a constant, then the weight matrix $\mathbf{W}$ can be recovered in closed form as a constrained least square problem. Since the optimal weights for reconstructing a particular point is only dependent on other points, each row of the weight matrix $\mathbf{W}$ can be obtained independently. The problem reduces to minimize the following function:

$$\begin{aligned}\min_{\mathbf{w}_{i\bullet}} \quad \mathcal{Q}(\mathbf{w}_{i\bullet}) \quad &= \quad \alpha\|\mathbf{x}_i - \mathbf{w}_{i\bullet}\mathbf{X}'_i\|^2_{\mathcal{F}} \\ &\quad + \sum_{k=1}^{p} \beta^k \|\mathbf{f}^k_{i\bullet} - \mathbf{w}_{i\bullet}\mathbf{F}^{k'}_i\|^2_{\mathcal{F}} + \lambda\|\mathbf{w}_{i\bullet}\|^2_{\mathcal{F}} \\ s.t. \qquad \mathbf{w}_{i\bullet}\mathbf{1} \quad &= \quad 1 \end{aligned} \tag{4}$$

where $\mathbf{X}'_i$ and $\mathbf{F}^{k'}_i$ denote the set difference $\{\mathbf{X} \setminus \mathbf{x}_i\}$ and $\{\mathbf{F}^k \setminus \mathbf{f}^k_{i\bullet}\}$ respectively, i.e. the set of all instances and their labels except the i^{th} instance and its labels, and as before $\| \bullet \|_{\mathcal{F}}$ denotes Frobenius norm. The derivative of the cost function with respect to $\mathbf{w}_{i\bullet}$ can be written as:

$$\begin{aligned}\nabla_{\mathbf{w}_{i\bullet}} \mathcal{Q}(\mathbf{w}_{i\bullet}) = \mathbf{w}_{i\bullet}(\alpha \left(\mathbf{1}\mathbf{x}_i - \mathbf{X}'_i\right) \left(\mathbf{1}\mathbf{x}_i - \mathbf{X}'_i\right)^T \\ + \sum_{k=1}^{p} \beta^k \left(\mathbf{1}\mathbf{f}^k_{i\bullet} - \mathbf{F}^{k'}_i\right) \left(\mathbf{1}\mathbf{f}^k_{i\bullet} - \mathbf{F}^{k'}_i\right)^T + \lambda I)\end{aligned} \tag{5}$$

To provide the solution for $\mathbf{W}$, we first introduce the local covariance matrix. Let $L(\mathbf{x}_i)$ denote the local covariance matrix of the feature description $\mathbf{x}_i$ of the i^{th} instance. The term "local" refers to the fact that the instance is used as the mean of the calculation.

$$L(\mathbf{x}_i) = \left(\mathbf{1}\mathbf{x}_i - \mathbf{X}'_i\right)\left(\mathbf{1}\mathbf{x}_i - \mathbf{X}'_i\right)^T \tag{6}$$

where as before $\mathbf{x}_i$ is a row vector, and $\mathbf{1}$ is a column vector with all one entries. Using a Lagrange multiplier to enforce the sum-to-one constraint, the update of $\mathbf{w}_{i\bullet}$ (the weights for the i^{th} instance) can be expressed in terms of the inverse local covariance matrices.

$$\mathbf{w}_{i\bullet} = \frac{\mathbf{1}^T \left(\alpha L(\mathbf{x}_i) + \sum_{k=1}^{p} \beta^k L(\mathbf{f}^k_{i\bullet}) + \lambda I\right)^{-1}}{\mathbf{1}^T \left(\alpha L(\mathbf{x}_i) + \sum_{k=1}^{p} \beta^k L(\mathbf{f}^k_{i\bullet}) + \lambda I\right)^{-1} \mathbf{1}} \tag{7}$$

where $L(\mathbf{f}^k_{i\bullet})$ is defined in the same manner as $L(\mathbf{x}_i)$ shown in Eq. (6), and I represents the identity matrix. As previously defined, α and β are the tuning parameters for the views and labels respectively, and λ controls the penalty of the L_2 norm. The optimal weight matrix consists of all the row vectors $\mathbf{w}_{i\bullet}$ for $i = 1, \cdots, n$. The L_2 norm slightly improves the sparsity of the reconstruction weights, and further sparsity can be obtained by discarding the neighbors with very small weights, since they are barely effective in the graph diffusion. Note that Eq. (7) can not guarantee the weights are non-negative. We empirically found that the negative weights are infrequent and generally have small values, and thus only have little effect to the learning performance. In our experiment, we keep both positive and negative weights, as a positive weight indicates two points are similar, and then a negative weight indicates the opposite.

4.2.2 Update for $\mathbf{F}^k$

In this step, we assume the weight matrix $\mathbf{W}$ is constant, then the goal is to fill in the missing labels in $\mathbf{F}^k_u$. For a label t^k, we relax the binary classifying function $\mathbf{F}^k$ to be real-valued, so that the optimal $\mathbf{F}^k_u$ can be recovered in closed form. Since the feature reconstruction error (first term of Eq. (3)) is a constant, we can rewrite the formulation in matrix format:

$$\begin{aligned}&\min_{\mathbf{F}^k} \quad \mathcal{Q}(\mathbf{F}^k) = \frac{1}{2} tr\left\{\left(\mathbf{F}^k\right)^T (I - \mathbf{W})^T (I - \mathbf{W})\,\mathbf{F}^k\right\} \\ &s.t. \qquad \mathbf{F}^k_l = \mathbf{V}^k\mathbf{Y}^k\end{aligned} \tag{8}$$

where $\mathbf{Y}^k$ carries the given labels of the k^{th} label, and $\mathbf{V}^k$ is the corresponding the node regularizer. To express the solution in terms of matrix operations, we assume the instances have been ordered so that the first l are labeled and the remaining u are the unlabeled instances. We can then split the weight matrix $\mathbf{W}$ and classification function $\mathbf{F}^k$ after the l^kth row and column, i.e. $\mathbf{W} = \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix}$ and $\mathbf{F}^k = \begin{bmatrix} \mathbf{F}^k_l \\ \mathbf{F}^k_u \end{bmatrix}$. Note that we do not attempt to overwrite the labeled instances. The cost function is convex, thereby allowing us to recover the optimal $\mathbf{F}^k$ by setting the derivative $\nabla_{\mathbf{F}^k}\mathcal{Q}(\mathbf{F}^k) = 0$.

$$\begin{gathered}\left(I - \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix}\right)^T \left(I - \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix}\right) \begin{bmatrix} \mathbf{F}^k_l \\ \mathbf{F}^k_u \end{bmatrix}^k = \mathbf{0} \\ \text{s.t.} \qquad \mathbf{F}^k_l = \mathbf{V}^k\mathbf{Y}^k.\end{gathered} \tag{9}$$

where $\mathbf{0}$ is a matrix with all zeros. The optimization above yields a large sparse system of linear equations that can be solved by a number of standard methods. The most straightforward one is the closed-form solution via matrix inversion. The predictions for unlabeled instances are obtained in closed form via matrix inversion:

$$\begin{aligned}\mathbf{F}^k_u = &\left(\mathbf{W}^T_{lu}\mathbf{W}_{lu} + (I_u - \mathbf{W}_{uu})^T (I_u - \mathbf{W}_{uu})\right)^{-1} \\ &\left(\mathbf{W}^T_{lu}(I_l - \mathbf{W}_{ll}) + (I_u - \mathbf{W}_{uu})^T \mathbf{W}_{ul}\right)\mathbf{V}^k\mathbf{Y}^k\end{aligned} \tag{10}$$

where I_l and I_u denote identity matrices with dimension l and u, respectively. The predictions for an unlabeled instances $\mathbf{I}_i$ can be obtained by setting $f^k_{ij} = 1$ where $j = \arg\max_j f^k_{ij}$, and other elements in $\mathbf{f}^k_{i\bullet}$ to zeros.

4.2.3 Progressive Update for $\mathbf{Y}^k$

Since there is no theoretical guarantee that our proposed alternating optimization will converge, it is possible that the prediction of the current iteration oscillates and backtracks from the predicted labellings in previous iterations. A straightforward solution to address this problem is to set up a small tolerance, but it is difficult for a practitioner to set the value of tolerance. Alternatively, we propose a progressive method to update $\mathbf{Y}^k$ incrementally instead of updating $\mathbf{F}^k$ to remove such unstable oscillations. In each iteration, we only make the most confident prediction, and treat this as the ground truth in future training. To do this, we consider the prior label matrix $\mathbf{Y}^k$ as an unknown in the cost function Eq. (8). The Choosing of the most confident prediction is guided by the direction with largest negative gradient in the partial derivative $\frac{\partial \mathcal{Q}(\mathbf{F}^k,\mathbf{Y}^k)}{\partial \mathbf{F}^k}$.

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{F}^k} = \left[\begin{array}{c} \left(\frac{\partial \mathcal{Q}}{\partial \mathbf{F}^k}\right)_l \\ \left(\frac{\partial \mathcal{Q}}{\partial \mathbf{F}^k}\right)_u \end{array} \right] = (I-\mathbf{W})^T(I-\mathbf{W}) \left[\begin{array}{c} \mathbf{V}^k\mathbf{Y}^k \\ \mathbf{0} \end{array} \right] \quad (11)$$

For the label k, let $(i^*, j^*)^k$ denotes the position of the most confident prediction, which can be decided by finding the largest negative value in the partial derivative.

$$(i^*, j^*)^k = \arg\min_{i,j} \left(\frac{\partial \mathcal{Q}}{\partial \mathbf{F}^k}\right)_u \quad (12)$$

In each iteration, we locate the position $(i^*, j^*)^k$ in the matrix $\mathbf{F}_u^k$, and reset the entries in $\mathbf{f}^k_{(l^k+i^*)\bullet}$. In particular, we set the entry $f^k_{(l^k+i^*),j^*}$ to 1 and other entries in the (l^k+i^*)-th row to 0, and then update $\mathbf{Y}^k$ by:

$$\mathbf{Y}^k \Longleftarrow \left[\begin{array}{c} \mathbf{Y}^k \\ \mathbf{f}^k_{(l^k+i^*)\bullet} \end{array} \right] \quad (13)$$

After each prediction, we update l^k with $l^k \Leftarrow l^k + 1$, recompute the weight matrix $\mathbf{W}$ using Eq. (7) based on the newly obtained $\mathbf{Y}$, and update the node regularizer $\mathbf{V}$. The whole procedure repeats until all the missing labels in the p labels are inferred.

5 Multi-view Learning

5.1 Formulation

We now apply our framework to multi-view learning, where there are multiple feature descriptions obtained from different views for the same set of instances. Our goal is to improve learning performance by taking advantage of the complementary information carried by the multiple views of data. Under this setting, the general framework in Eq. (1) can be simplified to:

$$\begin{aligned} \mathcal{Q}(\mathbf{W},\mathbf{F}) \quad &= \quad \sum_{k=1}^{q} \alpha^k \sum_{i=1}^{n} \|\mathbf{x}_i^k - \sum_{j=1}^{n} w_{ij}\mathbf{x}_j^k\|_{\mathcal{F}}^2 \\ &\quad + \beta \sum_{i=1}^{n} \|\mathbf{f}_{i\bullet} - \sum_{j=1}^{n} w_{ij}\mathbf{f}_{j\bullet}\|_{\mathcal{F}}^2 + \lambda\|\mathbf{W}\|_{\mathcal{F}}^2 \\ \text{s.t.} \quad & \forall i,\ \mathbf{w}_{i\bullet}\mathbf{1} = 1; \quad \mathbf{F}_l = \mathbf{V}\mathbf{Y}. \end{aligned} \quad (14)$$

5.2 Alternating Optimization

The optimization problem can be solved using a similar approach we proposed for multi-label learning.

Update for W. While assuming the classification function $\mathbf{F}$ is fixed, the weight matrix $\mathbf{W}$ can be recovered as now a set of constrained least square problems. Given the definition of local covariance matrix shown in Eq. (6), the weights $\mathbf{w}_{i\bullet}$ for the i^{th} instance can be solved independently by applying a Lagrange multiplier.

$$\mathbf{w}_{i\bullet} = \frac{\mathbf{1}^T \left(\sum_{k=1}^{q} \alpha^k L(\mathbf{x}_i^k) + \beta L(\mathbf{f}_{i\bullet}) + \lambda I\right)^{-1}}{\mathbf{1}^T \left(\sum_{k=1}^{q} \alpha^k L(\mathbf{x}_i^k) + \beta L(\mathbf{f}_{i\bullet}) + \lambda I\right)^{-1} \mathbf{1}} \quad (15)$$

Update for F. The classifying function $\mathbf{F}$ can be recovered using exactly the same method proposed in Section 4.2.2 or 4.2.3. Specifically, the update for $\mathbf{F}$ can be calculated using the closed form solution in Eq. (10), or the progressive solution shown from Eq. (11) to Eq. (13).

6 Spectral Embedding for Dimension Reduction Step

In this section we describe an extension to our framework which allows dimension reduction to easily be performed. It can be used as an additional step in the optimization or a post-processing step after the optimization converged. It is useful as a method to more easily visualize the results of our algorithm (as done in Fig. 2(a)) or when working with high dimensional data (as done for the results shown in Fig. 4(c)).

Since the weight matrix $\mathbf{W}$ captures the intrinsic geometric relations between data points, dimension reduction can be performed using $\mathbf{W}$. Again that the spectral embedding step is unnecessary for learning purpose, it is only used to dimension reduction. Let d denote the desired dimension, the dimension reduced instance $\hat{\mathbf{x}}_i$ minimizes the embedding cost function:

$$\mathcal{Q}(\hat{\mathbf{X}}) = \sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \sum_{j=1}^{n} \mathbf{W}_{ij}\hat{\mathbf{x}}_j\|^2 \quad (16)$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{n\times d}$ is the dimension reduced data matrix. The embedding cost in Eq. (16) defines a quadratic form in the vector $\hat{\mathbf{x}}_i$. Since we want the problem well-posed and also to avoid trivial solutions, the minimization can be solved as a sparse eigen decomposition problem:

$$\min \ \mathcal{Q}(\hat{\mathbf{X}}) = tr\left(\hat{\mathbf{X}}^T\mathbf{M}\hat{\mathbf{X}}\right) \quad (17)$$

where $\mathbf{M} = (I-\mathbf{W})^T(I-\mathbf{W})$. The optimal embedding can be recovered by computing the smallest $d+1$ eigenvectors of the matrix $\mathbf{M}$, and then discard the smallest eigenvector which is an unit vector. The remaining d eigenvectors are the optimal embedding coordinates that minimize equation (16).

7 Quantifying The Success of Multi-label and Multi-view Learning

Multi-label and Multi-view learning have been successfully applied to many real-world applications, however, in many data sets the performance is no better than and sometimes even worse than if each classification problem were solved independently [38]. Consequently, avoiding destructive fusing of information is an essential element in multi-label and multi-view learning. However, very few approaches produce a measure to determine if the transfer of knowledge between labels or views was successful. In this section we outline such a measure and later in Section 9.4 empirically verify its usefulness.

Multi-label Learning. Our measure makes use of our interpretation of the SSDR-MML framework as performing label propagation and the mechanism for information combining $\mathbf{W}$. Let $\mathbf{F}$ denote a binary label matrix $\mathbf{F} \in \mathbb{B}^{n \times p}$, where p is the number of labels defined on a set of n instances. $f_{ij} = 1$ if instance i can be categorized into class j, and $f_{ij} = -1$ otherwise. After the training step we have a weight matrix $\mathbf{W}$ available built upon the set of instances, where $\mathbf{W}$ carries the information learnt from the multiple labels and views. Since $\mathbf{W}$ row-wise sums to one, then $\mathbf{W}$ can be viewed as a random walk transition matrix. To quantify the success of multi-label learning, we define a measure of cross propagation (CP):

$$\mathrm{CP}\left(\mathbf{W}\right) = \mathbf{F}^T \left(\mathbf{W}\right)^z \mathbf{F} \tag{18}$$

where z is a positive integer, indicating how many steps the labels have to propagate. The resulting $\mathrm{CP}\left(\mathbf{W}\right)$ is a $p \times p$ matrix where the entry at i, j can be interpreted as *how well label i is propagated to the instances labeled with label j*. Therefore, the values on the diagonal measure the success of *intra-label reconstruction*, and the off-diagonal values measure the success of *inter-label reconstruction*. It has been widely reported [7], [8] that multi-label learning performs better when labels are correlated, then the sum of **off-diagonal** entries denotes *how well the knowledge transfer among multiple labels has occurred*.

Multi-vew Learning. In multi-view learning, $\mathrm{CP}\left(\mathbf{W}\right)$ measures *how well the class labels are reconstructed using the knowledge carried by multiple views*. For multi-view learning, a relatively large value in $\mathrm{CP}\left(\mathbf{W}\right)$ (compared to the "single" case) implies that the joint learning of views was successful, while a smaller value indicates that detrimental view combining has occurred. The proposed success measure not only offers a way to quantify the performance of joint learning of multiple labels/views, but also can be used to guide the selection of parameters in many existing multi-labels/view learning algorithms.

8 Implementation and Pragmatic Issues

In this section, we outline issues that we believe make implementation and using of our work easier.

8.1 Uses for Multi-label and Multi-view Learning

The proposed framework is motivated by the fact that multi-label and multi-view learning may improve the learning performance over the "single" case by exploiting the complementary knowledge contained in the multiple labels or views. In multi-label learning, higher learning performance could be obtained if the multiple labels are highly correlated, while worse performance could happen if the multiple labels are irrelevant. In multi-view learning, intuitively, the multiple views are supposed to be neither too different nor too similar to each other. There would not be much gain if the multiple views are too similar. On the other hand, if the multiple views are too different, multi-view learning could even be harmful. Since we do not want to over-constrain the graph to any label or view while still absorbing knowledge from each of them, the proposed approach is in fact a moderate solution that partially fits the graph to each label or view in a weighted fashion.

In some practical cases, we may want to regard one of the multiple labels or views as the target label or view, and consider the others as the source label or view to help it. In that case, we set the β^k or α^k of the target label or view to 1, and set the weights of source labels or views to lie between 0 and 1. Then, the weights of these sources labels and views can be used to encode the relative "relatedness" and "importance" to the main label and view, respectively. For a source label or view, a weight of value 0 indicates that it is "irrelevant" or "contradictory" to the main label or view, while a weight of value 1 implies that it is "highly correlated" or "complementary" to the target label or view, respectively.

8.2 Parameter Selection

TABLE 2
SSDR-MML Algorithm (progressive)

```
Input:
  feature descriptions $\mathbf{x}_i^k$, for $i \in \{1, \cdots, n\}$ and $k \in \{1, \cdots, q\}$
  prior labels matrix $\mathbf{Y}^k$, for labeled instances and $k \in \{1, \cdots, p\}$
  regularization parameters $\alpha^k$, $\beta^k$ and $\lambda$.
Training Stage:
    initialize: count $m = 0$, $(\mathbf{Y}^k)^0 = \mathbf{Y}^k$, $\mathbf{F}_l^k = (\mathbf{Y}^k)^0$.
    do{
        compute $\mathbf{w}_{i\bullet}^m = \frac{\mathbf{1}^T \left(\sum_{k=1}^q \alpha^k L(\mathbf{x}_i^k) + \sum_{k=1}^p \beta^k L(\mathbf{f}_{i\bullet}^k) + \lambda I\right)^{-1}}{\mathbf{1}^T \left(\sum_{k=1}^q \alpha^k L(\mathbf{x}_i^k) + \sum_{k=1}^p \beta^k L(\mathbf{f}_{i\bullet}^k) + \lambda I\right)^{-1} \mathbf{1}}$;
        update $\mathbf{D}^m$, $d_{jj}^m = \sum_{i=1}^n w_{ij}^m$;
        update $(\mathbf{v}^k)^m = \sum_{i=1}^{c^k} \frac{\left(\mathbf{y}_{\bullet i}^k\right)^m \odot \mathbf{D}^m \mathbf{1}}{\left(\left(\mathbf{y}_{\bullet i}^k\right)^m\right)^T \mathbf{D}^m \mathbf{1}}$;
        locate $(i^*, j^*)^k = \arg\min_{i,j} \left(\frac{\partial \mathcal{Q}}{\partial \mathbf{F}^k}\right)_u$;
        set $f_{(l^k+i^*)j^*}^k = 1$, and $f_{(l^k+i^*)j}^k = 0$ for $j \neq j^*$;
        update $\left(\mathbf{Y}^k\right)^{m+1} = \begin{bmatrix} \left(\mathbf{Y}^k\right)^m \\ \mathbf{f}_{(l^k+i^*)\bullet}^k \end{bmatrix}$;
        update $\mathbf{F}_l^k = \left(\mathbf{Y}^k\right)^{m+1}$, and remove the $(l^k + i^*)^{th}$ instance from $\mathbf{F}_u^k$;
        update $m = m + 1$, $l^k = l^k + 1$;
    }while($\mathbf{F}_u^k != \phi$)

Output:
  weight matrix $\mathbf{W}$, label predictions $\mathbf{Y}_u^k$ for $k \in \{1, \cdots, p\}$.
```

The general solution for the proposed algorithm can be implemented as shown in Table 2. There are three parameters in our proposed framework α^i (the weight of view i), β^j (the weight of label j), and λ (the regularization parameter). The regularization term has two advantages, one is to simplify the learning model, and the other one is to avoid excessive label propagation. Empirically, we found that the learning performance of our approach is not sensitive to λ, which also implies that the excessive label propagation can be avoided if λ is not too small. For the parameters α^k and β^k, we wish to show that the performance of the algorithm does not fluctuate greatly with respect to minor changes in the parameter values. We shall empirically show the stability of the performance of our framework with respect to the parameters (α^k and β^k) in Section 9.3. Though the algorithm's performance is stable, how to set the parameters is still an open question as is the case for most learning algorithms. Two alternatives are to use the knowledge from domain experts to guide or constrain the selection of parameters, e.g. it is well known that the color feature should be more emphasized in the image classification problem "tomato v.s. washing machine". Alternatively, our previously defined measure for success of learning $\mathrm{CP}(\mathbf{W})$ in section 7 could also be used to guide the selection of parameters which is the approach we use in our experimental section.

8.3 Computational Complexity

The standard implementation of the algorithm shown in Table 2 takes $\mathcal{O}(mn^3)$ time, where m denotes the number of missing labels, and n denotes the number of instances. Since the fact $m < n$, the computation complexity of our algorithm is dominated by the matrix inversion, where standard methods require $\mathcal{O}(n^3)$ time. Fig. 3 reports the empirical runtime of our method on SIAM TMC 2007 dataset. The reported time is calculated on a laptop equipped with Intel i7 2.60 GHz CPU, and 16 GB memory. The details of the dataset and experimental settings can be found in Section 9. We see from the result that the runtime decreases significantly along with the decrease of the number of missing labels.

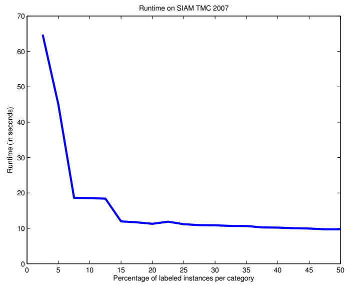


Fig. 3. Runtime w.r.t. varying number of missing labels.

To speed up the computation, the weight matrix $\mathbf{W}$ can be recovered by solving a linear system of equations as it is much more efficient. In Eq. (7) or Eq. (15) we see that the denominator of the fraction is simply a constant which rescales the sum of i-th row of $\mathbf{W}$ to 1. Therefore, in practice, a more efficient way to recover the optimal $\mathbf{w}_{i\bullet}$ is to solve a linear system of equations, and then rescale the sum of weights to one. Let $\mathbf{L}_i$ denote the mixed local covariance matrix $\left(\sum_{k=1}^{q} \alpha^k L(\mathbf{x}_i^k) + \sum_{k=1}^{p} \beta^k L(\mathbf{f}_{i\bullet}^k) + \lambda I\right)$, $\mathbf{w}_{i\bullet}$ can be recovered by solving $\mathbf{L}_i \mathbf{w}_{i\bullet} = \mathbf{1}$, and then rescale the sum of $\mathbf{w}_{i\bullet}$ to 1. When a local covariance matrix is singular, the linear system of equations can be conditioned by adding a small multiple of the identity matrix $\mathbf{L}_i \leftarrow \mathbf{L}_i + \frac{\xi tr(\mathbf{L}_i)}{k} I$, where k denotes the number of neighbors for each instance, and ξ is a small value ($\xi \ll 0.1$).

8.4 Extensions for Noisy Prior Labels

We propose an extension to our method that allows some incorrect labels to be ignored. Since we previously assumed that all the initial labels are accurate, the solution of $\mathbf{F}^k$ provided in Eq. (10) or Eq. (11) suffers from the problem that there may be considerable noise scattered in labeled data. A reasonable solution to address this is to relax the inference objective function by replacing the constraint on the given labels with an inconsistent penalty term, namely local fitting penalty [32] allowing partial neglect of the given labels. We now expand the prior label matrix $\mathbf{Y}^k$ to be a $n \times c^k$ matrix, and fill in the missing label locations with zeros. To relax the inference problem shown in Eq. (8), we add in the inconsistent penalty term and rewrite the cost function as

$$\begin{aligned} \min_{\mathbf{F}^k} \quad \mathcal{Q}(\mathbf{F}^k, \mathbf{Y}^k) = \frac{1}{2} tr\{ \left(\mathbf{F}^k\right)^T (I - \mathbf{W})^T (I - \mathbf{W}) \mathbf{F}^k \\ + \gamma \left(\mathbf{F}^k - \mathbf{V}^k \mathbf{Y}^k\right)^T \left(\mathbf{F}^k - \mathbf{V}^k \mathbf{Y}^k\right) \} \end{aligned} \tag{19}$$

where γ (> 0) is a tuning parameter to balance the influence of label reconstruction error and local fitting penalty. If we set $\gamma = \infty$, the cost function will reduce to Eq. (8). The cost function is convex and unconstrained, then the update for $\mathbf{F}^k$ (Eq. (10)) can be rewritten as

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial \mathbf{F}^k} = (I - \mathbf{W})^T (I - \mathbf{W}) \mathbf{F}^k + \gamma \left(\mathbf{F}^k - \mathbf{Y}^k\right) = 0 \quad \Longrightarrow \\ \mathbf{F}^k = \left(\frac{1}{\gamma} (I - \mathbf{W})^T (I - \mathbf{W}) + I\right)^{-1} \mathbf{V}^k \mathbf{Y}^k \end{aligned} \tag{20}$$

Accordingly, in this relaxed version of label inference, the progressive update for $\mathbf{Y}^k$ will also change. Since the prior label matrix $\mathbf{Y}^k$ is included in the cost function Eq. (19), the optimization problem is now over both the classifying function $\mathbf{F}^k$ and the prior label matrix $\mathbf{Y}^k$, mathematically $\min_{\mathbf{F}^k, \mathbf{Y}^k} \mathcal{Q}(\mathbf{F}^k, \mathbf{Y}^k)$. Therefore, we replace $\mathbf{F}^k$ in the cost function Eq. (19) with its optimal solution shown in Eq. (20), let $A = \left(\frac{1}{\gamma} (I - \mathbf{W})^T (I - \mathbf{W}) + I\right)^{-1}$, the optimization problem over $\mathbf{Y}^k$ can be formulated as

$$\begin{aligned} \mathcal{Q}(\mathbf{Y}^k) &= \frac{1}{2} tr\{ \left(\mathbf{A}\mathbf{V}^k \mathbf{Y}^k\right)^T (I - \mathbf{W})^T (I - \mathbf{W}) \left(\mathbf{A}\mathbf{V}^k \mathbf{Y}^k\right) \\ &\quad + \gamma \left(\mathbf{A}\mathbf{V}^k \mathbf{Y}^k - \mathbf{V}^k \mathbf{Y}^k\right)^T \left(\mathbf{A}\mathbf{V}^k \mathbf{Y}^k - \mathbf{V}^k \mathbf{Y}^k\right) \} \\ &= \frac{1}{2} tr\{ \left(\mathbf{V}^k \mathbf{Y}^k\right)^T [\mathbf{A}^T (I - \mathbf{W})^T (I - \mathbf{W}) \mathbf{A} \\ &\quad + \gamma (\mathbf{A} - I)^T (\mathbf{A} - I)] \left(\mathbf{V}^k \mathbf{Y}^k\right) \} \end{aligned} \tag{21}$$

Let $\mathbf{B} = \mathbf{A}^T(I - \mathbf{W})^T\left(I - \mathbf{W}\right)\mathbf{A} + \gamma(\mathbf{A} - I)^T\left(\mathbf{A} - I\right)$, we write the gradient of $\mathcal{Q}$ with respect to $\mathbf{V}^k\mathbf{Y}^k$ as show in Eq. (22), which substitutes Eq. (11).

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{V}^k\mathbf{Y}^k} = \left(\mathbf{B}^T\mathbf{B}\right)\mathbf{V}^k\mathbf{Y}^k \tag{22}$$

Base on $\frac{\partial \mathcal{Q}}{\partial \mathbf{Y}^k} = \frac{\partial \mathbf{V}^k\mathbf{Y}^k}{\partial \mathbf{Y}^k}\frac{\partial \mathcal{Q}}{\partial \mathbf{V}^k\mathbf{Y}^k}$ and basics in algebra, we know that the optimization of $\mathcal{Q}$ with respect to $\mathbf{Y}^k$ is equivalent to the optimization over $\mathbf{V}^k\mathbf{Y}^k$. Consequently, the most confident prediction is located at the position shown in Eq. (23), which substitutes Eq. (12).

$$(i^*, j^*)^k = \arg\min\nolimits_{i,j}\left(\frac{\partial \mathcal{Q}}{\partial \mathbf{V}^k\mathbf{Y}^k}\right)_u \tag{23}$$

9 Empirical Study

In this section, we empirically evaluate our framework SSDR-MML and its success measure on several real-world applications under multi-label/view settings.

9.1 Multi-label Learning

9.1.1 Experimental Settings

We compare the performance of our SSDR-MML approach against five baseline methods: (1) **RankSVM** [39], a state-of-the-art supervised multi-label classification algorithm based on ranking the results of support vector machine (SVM); (2) **ML-GFHF**, the multi-label version (two-dimensional optimization) of the harmonic function [27]; (3) **Regularized MTL** [8], a regularized multi-task learning method, which assumes all predicting functions come from a Gaussian distribution; (4) **AdaBoost-MH** [40], which is an extension of AdaBoost for multi-label data that tries to minimize hamming loss; (5) **BR-RDT** [41], Binary Relevance based Random Decision Tree, which is an ensemble method. In RankSVM, we choose RBF kernel function (σ is selected using cross validation), and fix the penalty coefficient $C = 1000$. For ML-GFHF, we construct a k-*NN* ($k = 15$) graph similarity via RBF kernel function with length scale σ selected using cross validation. For AdaBoost-MH, the number of boosting rounds parameter is set to 100. For BR-RDT, there are 100 trees constructed in total, the maximum depth allowed to be the half of the number of features, and the minimal instances on a leaf node is 10. In our SSDR-MML approach, we set the regularization parameter $\lambda = 1$ and determine the importance of each label by maximizing the learning success measure, $\beta^i = \arg\max_{\beta^i} \mathrm{CP}(\mathbf{W})$. For fairness, the parameters (λ_i) in Regularized MTL are set to the values that are equivalent to the parameter setting of SSDR-MML. We adopt micro-averaged F_1 measure (F_1 Micro) [42] to evaluate the relative performance, which is a standard evaluation method for multi-label learning.

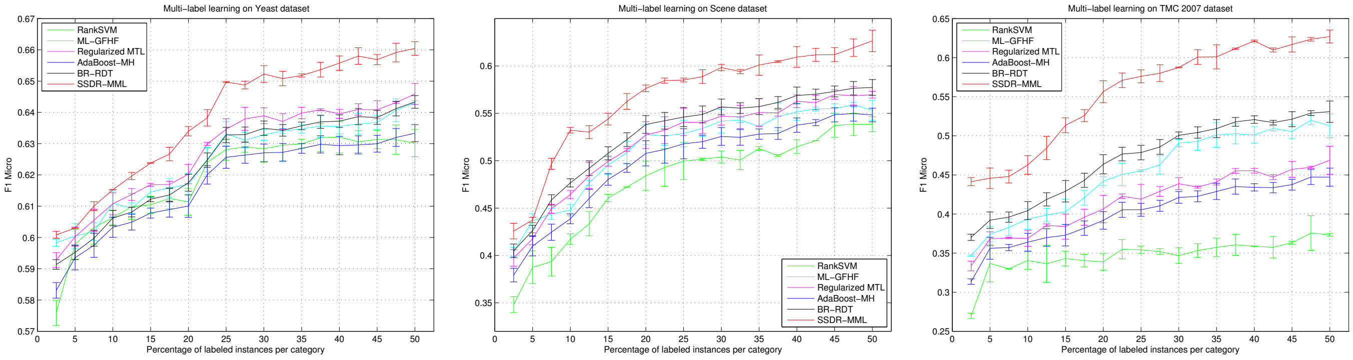
Dataset. We evaluate the performance of our multi-label method on three different types of real world datasets.

- **Yeast** [43]: consists of $2,417$ gene samples, each of which belongs to one or several of 14 distinct functional categories, such as transcription, cell communication, protein synthesis, Ionic Homeostasis, and etc. The feature descriptions of *Yeast* dataset are extracted by different sequence recognition algorithms, and each gene sample is represented in a 103 dimensional space. The tasks on *Yeast* dataset are to predict the localization site of protein, where each sample is associated with 4.24 labels on average.
- **Scene**: image dataset consists of $2,407$ natural scene images, each of which is represented as a 294-dimensional feature vector and belongs to one or more (1.07 in average) of 6 categories (beach, sunset, fall foliage, field, mountain, and urban).
- **SIAM TMC 2007**: text dataset for SIAM Text Mining Competition 2007 consisting of $28,596$ text samples, each of which belongs to one or more (2.21 in average) of 22 categories. To simplify the problem, in our experiments we randomly select a subset containing $3,000$ samples from the original dataset, then use binary term frequencies and normalize each instance to unit length ($30,438$-dimensional vector).

We chose these three multi-label datasets because that represent a range of situations, most instances in *Yeast* have more than one label, while most instances in *Scene* have only one label, and *SIAM TMC 2007* data is in a very high dimensional space.

9.1.2 Empirical Result

To comprehensively compare our proposed algorithm with the five baseline methods, we train the algorithms on data with varying numbers of labeled instances. In each trial, for each label we randomly select a portion of instances from the dataset as the training data, while the remaining unlabeled ones are used for testing. The portion of labeled data gradually increases from 2.5% to 50% with a step size of 2.5%. In Fig. 4, we report the average F_1 Micro scores and standard deviations, all of which are based on the average over 50 trials. For the three real-world datasets that we explored in the experiments, when all experimental results (regardless of step size) are pooled together, we see that the proposed approach SSDR-MML performs significantly better than the competing algorithms in terms of both lower error rate and standard deviation. Compared to the result in Fig. 4(a) and 4(b), we observe from Fig. 4(c) that the performance of our method is especially good on the high-dimensional data SIAM TMC 2007. This implies that on high-dimensional data typical multi-label methods would fail, due to the lack of the connection between dimension reduction and learning algorithm. By the promising performance of the proposed method shown in Fig. 4(c), where we iteratively perform learning and spectral embedding on the *SIAM TMC 2007* dataset, we demonstrate the effectiveness of connecting dimension reduction and learning. This is especially useful when the data are in the high-dimensional space.



(a) Yeast (b) Scene (c) SIAM TMC 2007

Fig. 4. Learning performance measured by F_1 Micro score w.r.t. different numbers of labeled instances

9.2 Multi-view Learning

9.2.1 Experimental Settings

We evaluate our framework for multi-view learning on two applications: (1) image classification on Caltech [44] dataset; and (2) a set of UCI benchmarks. We compare the performance of our SSDR-MML approach against three baseline models: (1) *SVM*, standard support vector machine [45] on each view of the data separately. (2) *SKMsmo*, multiple kernel learning [22] solved by sequential minimal optimization; (3) *Bayesian Co-Training* (BCT) [20], a Gaussian process consensus learner. To further understand our framework, we also compare SSDR-MML against two of its variates, i.e. (i) SSDR-MML Simple: SSDR-MML on single-viewed data which is obtained by simply joining the two view features into a single view; and (ii) SSDR-MML nonsparse, our SSDR-MML implementation without sparsity enforcement, which means excessive label propagation could happen. For both SVM and SKMsmo, the kernel is constructed using RBF with length scale $\sigma = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} \parallel \mathbf{x}_i - \mathbf{x}_j \parallel$, and the penalty coefficient $C0$ is fixed to 100. For Bayesian Co-Training (BCT), we choose a standard Gaussian process (GP) setting: zeros mean function, Gaussian likelihood function, and isotropic squared exponential covariance function without latent scale. The hyperparameters in GP models can be adapted to the training data, thus the predictions of Bayesian Co-Training can be readily obtained by consenting the predictions of GP from different views using the learned noise variances. To avoid overfitting, the number of function evaluations in gradient-based optimization is limited to a maximum of 100. In our SSDR-MML approach, we set $\lambda = 1$, and the importance of each view is decided by finding the maximal success measure, $\alpha^i = \arg\max_{\alpha^i} \mathrm{CP}(\mathbf{W})$.

9.2.2 Color-aided SIFT Matching

Caltech-256 image dataset [44] consists of $30,608$ images in 256 categories, where each category contains around 100 images on average. We conduct the experiments on ten binary labels that are randomly selected from Caltech-256, as summarized in Table 3. Since images are difficult to describe, there are several standard methods to extract features from an image, such as edge direction, and color or visual word histogram. However, there is no straightforward answer to tell what kind of features can outperform others, as the performance of each type of features highly depends on the specific applications. Thus, it could be desirable if we can make use of multiple image descriptors to help the learner. In the experiment, we exploit two image descriptions for each image: (1) color histogram, a representation of the distribution of colors in an image; and (2) visual word histogram, which is based on the successful image descriptor SIFT [46]. Although SIFT can accurately detect and describe interesting points in an image, it suffers from the limitation that information carried by color is ignored. Consequently, higher learning accuracy can be expected if we are able to perform learning on both SIFT and color simultaneously. In the preprocessing of the data, we construct the color histogram (80 bins in HSV color space) by counting the number of pixels that have colors in each of a fixed list of color ranges that span the color space. To produce the visual word histogram, we first build a visual vocabulary (800 visual words) by clustering all the SIFT descriptions collected from the image dataset, and then the histograms can be obtained by mapping images to the visual codebook, which is often called bag-of-word.

TABLE 3
Ten binary labels selected from Caltech-256

Label	Binary Label	Data Size
1	binocular vs killer-whale	216 vs 91
2	breadmaker vs telephone-box	141 vs 84
3	eyeglasses vs skyscraper	82 vs 95
4	fireworks vs self-propelled lawnmower	100 vs 120
5	mars vs saturn	155 vs 92
6	pci-card vs sunflower	105 vs 80
7	swiss-army-knife vs telephone-box	109 vs 84
8	tennis-ball vs zebra	98 vs 96
9	tomato vs washing machine	103 vs 84
10	video-projector vs watermelon	97 vs 93

We compare the performance of our approach against

TABLE 4
Mean error rates and standard deviations (in percentage) of the ten binary labels from Caltech-256. The best result is shown in bold.

Methods	Label 1	Label 2	Label 3	Label 4	Label 5	Label 6	Label 7	Label 8	Label 9	Label 10
SVM-SIFT	14.05±3.75	13.77±3.97	23.80±8.00	7.11±2.47	24.54±6.02	11.77±4.77	16.39±2.72	8.91±1.81	25.27±6.70	18.95±6.68
SVM-color	11.45±3.16	15.44±3.86	**19.81±3.76**	5.55±1.91	29.72±4.96	15.04±3.98	13.45±3.48	23.50±5.22	17.80±4.54	12.94±4.38
SKMsmo	11.26±3.20	15.20±3.89	19.84±3.58	5.52±1.89	25.17±3.13	14.55±3.93	13.40±3.50	22.56±5.06	17.76±4.72	12.57±4.17
BCT	11.63±3.32	14.78±3.88	19.85±3.28	**4.88±1.93**	24.26±4.20	13.43±4.48	14.36±4.45	21.29±4.98	17.42±4.74	15.24±6.14
SSDR-MML Simple	11.73±3.63	13.95±3.91	21.27±5.21	5.21±2.01	27.58±4.79	13.92±4.38	15.14±3.36	20.25±4.07	19.61±5.13	16.43±5.02
SSDR-MML nonsparse	15.09±5.37	16.31±4.07	23.83±8.22	5.39±3.84	31.75±6.45	16.08±5.03	16.77±4.89	14.43±5.21	17.98±6.36	16.27±6.18
SSDR-MML	**9.14±2.55**	**12.08±3.15**	19.87±3.26	4.99±2.14	**21.80±3.13**	**11.33±3.93**	**13.48±3.03**	**10.10±2.13**	**14.56±4.66**	**12.14±4.90**

the three baseline techniques on the ten predefined binary labels. In each trial, we randomly select 10% of the images as the training set, and the rest of the images are used as the test set. The experiment are repeated 50 times, and the mean error rates and standard deviations are reported in Table 4. It can be seen that in general the proposed SSDR-MML approach outperforms all the competing techniques. Moreover, it performs statistically significantly better than baseline models with both lower mean error rate and standard deviation. In addition, as we can observe from the result, the multi-view learning algorithms generally perform better than any of the single-view learners. This confirms the motivation of multi-view transfer that using the knowledge carried by all available feature descriptions to improve the performance of the learner. As expected, SIFT is more discriminative on some labels, while color is more discriminative on the other labels. Sometimes, the performances of SIFT and color feature are dramatically different. For example, in the binary label "tennis-ball vs zebra" color feature is much reliable than SIFT, while in the binary label "video-projector vs watermelon" SIFT performs significantly better than color. This demonstrate the necessity of multi-view transfer, as it is generally difficult for the users to tell which kind of features can outperform others. By comparing the performance of SSDR-MML with its two variates, we see that (1) simply placing features from multiple views into a single view does not work well due to the increased dimension and normalization issues and (2) Sparsity need to be enforced in semi-supervised settings since excessive label propagation is generally detrimental and unreliable.

9.2.3 UCI Benchmarks

The second evaluation of multi-view learning is carried out on a set of *UCI benchmarks* [47], namely (1)*Adult* (subset): extracted from the census bureau database; (2) *Diabetes*: contains the distribution for 70 sets of data recorded on diabetes patients; (3) *Ionosphere*: radar data was collected by a system that consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts; (4) *Liver Disorders*: attributes are collected from blood tests which are thought to be sensitive to liver disorders; (5) *Sonar*: contains signals obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions.; and (6) *SPECT Heart*: describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. We chose these six datasets since they have been widely used in the evaluations of various learning algorithms. To create two views for each of the datasets, we equally divide the features of each dataset into two disjoint subsets such that they are related but different, and thus each subset can be considered as one view. The details of the six UCI benchmarks are summarized in Table 5.

TABLE 5
Statistics of UCI benchmarks

Dataset Name	Instance No.	View 1 Feature No.	View 2 Feature No.
Adult (subset)	1, 605	60	59
Diabetes	768	4	4
Ionosphere	351	17	17
Liver Disorders	345	3	3
Sonar	208	30	30
SPECT Heart	270	7	6

We follow the previously stated methodology, where 10% of the samples are used for training and the remaining 90% for testing, to evaluate the performance of our SSDR-MML framework on multi-view setting. The resulting mean error rates and standard deviations on the six UCI benchmarks, which are based on 50 random trials, are reported in Table 6. It can be observed that the multiple view learning techniques generally outperform the single view classifiers, which substantiates the benefits of learning with multiple views. Among all techniques evaluated in the experiment, our SSDR-MML approach performs statistically significantly better than all competitors with both lower misclassification rates and standard deviations. The performance of the proposed framework on multi-view learning tasks not only demonstrates the effectiveness of our approach, but also validates the advantage of simultaneous learning of multiple feature descriptions. On the two variates of our approach, we also see that excessive label propagation and simply joining the multiple views into one could be harmful to the learning performance, which confirms the conclusion made in Section 9.2.2.

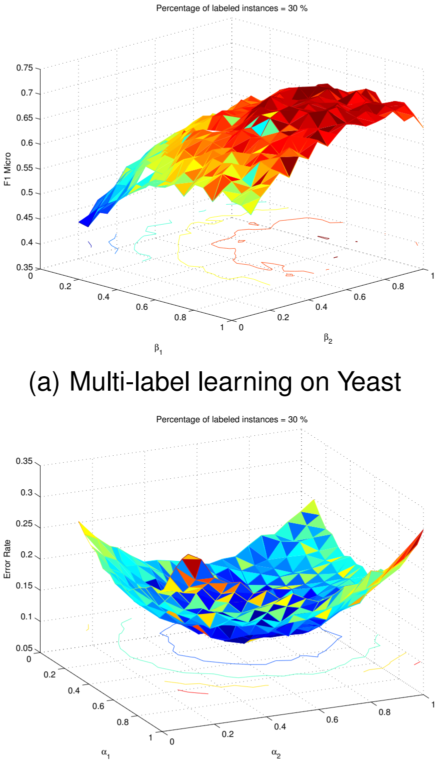
TABLE 6
Mean error rates and standard deviations (in percentage) on UCI benchmarks. The best result is shown in bold.

Methods	Adult(subset)	Diabetes	Ionosphere
SVM-SIFT	22.37±1.50	35.22±2.04	10.70±4.73
SVM-color	24.23±1.55	35.84±3.14	19.57±5.33
SKMsmo	21.24±1.32	32.71±2.88	12.55±4.86
BCT	**19.98±1.68**	29.27±2.54	17.41±4.75
SSDR-MML Simple	21.98±1.61	31.57±2.93	13.76±4.97
SSDR-MML nonsparse	23.38±2.73	36.62±3.04	15.29±5.21
SSDR-MML	20.40±1.16	**27.19±2.35**	**11.48±4.51**
Methods	Liver Disorders	Sonar	SPECT Heart
SVM-SIFT	45.87±3.43	32.91±4.97	36.47±6.66
SVM-color	45.05±3.83	34.91±3.64	26.77±4.75
SKMsmo	45.47±3.66	33.18±4.90	29.07±4.22
BCT	58.20±0.82	34.17±6.84	31.42±4.28
SSDR-MML Simple	46.01±3.41	33.35±4.26	29.69±4.52
SSDR-MML nonsparse	50.32±4.39	34.58±5.97	29.49±5.18
SSDR-MML	**40.99±2.75**	**31.40±4.19**	**25.83±4.18**

Based on the promising experiment results, we conclude that the proposed SSDR-MML framework is advantageous when (1) there are multiple feature descriptions available for each instance that are neither too different nor too similar; or (2) the multiple labels are highly related; or (3) the data is sparsely labeled and in high dimensional space.

9.3 Stability to Parameters

We empirically show the stability of the performance of our framework with respect to the parameters (α^k and β^k) in Fig. 5. We first pick two highly related labels, Cell Growth, Cell Division, DNA synthesis v.s. Transcription, from Yeast dataset (gene dataset with multiple labels), on which we show the learning performance of our framework in Fig. 5(a) using F_1 Micro score w.r.t. different settings of β^k $(0.1 \leq \beta^k \leq 1)$, where β^1 and β^2 are two parameters used to balance the influence of the two labels. We then choose a subset (swiss-army-knife v.s. telephone-box) from Caltech-256 image dataset, and construct two views from the images: (1) visual word histogram using SIFT visual feature where color is ignored by definition; and (2) color histogram. Fig. 5(b) shows the error rate of our framework on the binary label learning w.r.t. different settings of α^k $(0.1 \leq \alpha^k \leq 1)$, where α^1 and α^2 are two parameters used to balance the influence of the two views. In both Fig. 5(a) and 5(b), we see that learning performance surface is relatively smooth.



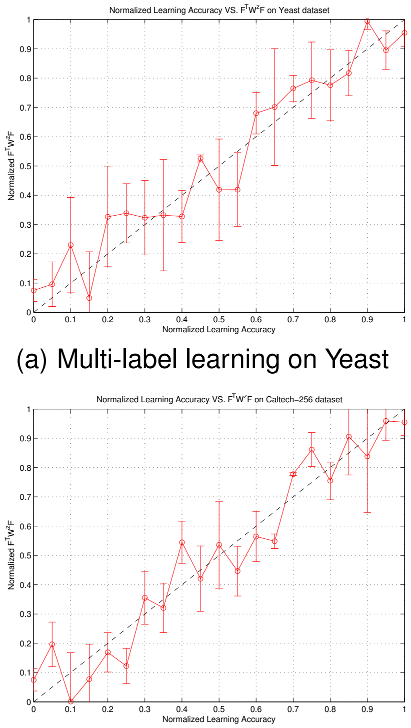
(a) Multi-label learning on Yeast

(b) Multi-view learning on Caltech-256

Fig. 5. Learning performance of SSDR-MML w.r.t. different parameter settings

9.4 Empirical Evaluation of Our Success Measure

We shall now illustrate the use of the proposed success measure. We apply our approach to two real-world datasets: Yeast gene dataset and Caltech-256 image dataset. Since our algorithms are deterministic, we can obtain a variety of experiments by changing the parameters α^k or β^k. We collect a set of learning accuracies and the corresponding values of $\mathrm{CP}(\mathbf{W})$ under these different parameterizations, and normalize all of them to lie between 0 and 1. In this experiment we adopt two labels (DNA synthesis and transcription) for multi-label learning on Yeast, and two views (visual word and color) for multi-view learning on Caltech-256. Fig. 6(a) shows the learning accuracy v.s. the value of $\mathbf{F}^T\mathbf{W}^z\mathbf{F}$ (sum of off-diagonal values) under multi-label setting on Yeast dataset with respect to different β^k's. Fig. 6(b) shows the normalized learning accuracy v.s. the value of $\mathbf{F}^T\mathbf{W}^z\mathbf{F}$ (naturally a single value) under multi-view setting on Caltech-256 dataset with respect to different parameterizations of α^k. The results shown in the two figures are quantified and averaged from 100 random trials (per parameterization). The red solid curve shows the results obtained using our SSDR-MML algorithm, while the black dash line marks the perfect proportional relation. Observing the results, in both settings we see that the value of $\mathbf{F}^T\mathbf{W}^z\mathbf{F}$ is approximately linear to the learning accuracy. Therefore, we conclude that a parameterization with relatively higher value of $\mathbf{F}^T\mathbf{W}^z\mathbf{F}$ will generally lead to a better learning result, and thus α^k or β^k should be set to the one which corresponds to the highest value of $\mathbf{F}^T\mathbf{W}^z\mathbf{F}$.



(a) Multi-label learning on Yeast

(b) Multi-view learning on Caltech-256

Fig. 6. Two examples showing the approximately proportional relation between the learning accuracy and the value of $\mathrm{CP}\left(\mathbf{W}\right)$.

10 Conclusion

As applications in machine learning and data mining move towards demanding domains, they must move beyond the restrictions of complete supervision, single-label, single-view and low-dimensional data. In this paper we present a joint learning framework based on reconstruction error, which is designed to handle both multi-label and multi-view learning settings. It can be viewed as simultaneously solving for two sets of unknowns: filling in missing labels and identifying projection vectors that makes points with similar labels close together and points with different labels far apart. As to improve the learning performance, the underlying objective is to enable the learner to combine knowledge from multiple labels and views by partially fitting the graph to each of them. Empirically, our proposed approach was shown to give more reliable results on real world applications with both lower error rate and standard deviation compared to the baseline models, as shown in Section 9. Perhaps the most useful part of our approach is that since the mechanism for combining knowledge is explicit it also offers the ability to measure the success of learning.

Acknowledgments

The authors gratefully acknowledge support of this research from ONR grants N00014-09-1-0712, N00014-11-1-0108 and NSF Grant NSF IIS-0801528.

References

[1] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York: Springer-Verlag, 2002.

[2] P. N. Belhumeur, J. P. Hespanha, P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[3] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, ser. COLT' 98. New York, NY, USA: ACM, 1998, pp. 92–100. [Online]. Available: http://doi.acm.org/10.1145/279943.279962

[4] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ser. ACL '95. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995, pp. 189–196. [Online]. Available: http://dx.doi.org/10.3115/981658.981684

[5] B. Qian and I. Davidson, "Semi-supervised dimension reduction for multi-label classification," in *AAAI'10: In Proceedings of the 24rd National Conference on Artificial Intelligence*. Menlo Park, California, USA: The AAAI Press, 2010, pp. 569–574.

[6] J. Chen, J. Zhou, and J. Ye, "Integrating low-rank and group-sparse structures for robust multi-task learning," in *KDD*, 2011, pp. 42–50.

[7] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, 1997, pp. 41–48.

[8] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2004, pp. 109–117.

[9] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *COLT*, 2003, pp. 567–580.

[10] H. Fei and J. Huan, "Structured feature selection and task relationship inference for multi-task learning," in *ICDM*, 2011, pp. 171–180.

[11] X. Kong, M. K. Ng, and Z.-H. Zhou, "Transductive multilabel learning via label set propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 704–719, 2013.

[12] K. Yu, V. Tresp, and A. Schwaighofer, "Learning gaussian processes from multiple tasks," in *ICML*, 2005, pp. 1012–1019.

[13] E. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 153–160.

[14] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with dirichlet process priors," *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, May 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248659.1248661

[15] B. Qian, X. Wang, J. Ye, and I. Davidson, "Multi-objective multi-view spectral clustering via pareto optimization," in *SDM*, 2013, pp. 234–242.

[16] X. Liu, S. Ji, W. Glänzel, and B. D. Moor, "Multiview partitioning via tensor methods," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 1056–1069, 2013.

[17] G. Li, K. Chang, and S. C. H. Hoi, "Multiview semi-supervised learning with consensus," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2040–2051, 2012.

[18] X. Wang, B. Qian, and I. Davidson, "Improving document clustering using automated machine translation," in *CIKM '12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 645–653.

[19] S. Dasgupta and M. Littman, "Pac generalization bounds for co-training," in *Neural Information Processing Systems*, 2002.

[20] S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and R. B. Rao, "Bayesian co-training," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 1665–1672.

[21] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research (JMLR)*, vol. 5, pp. 27–72, December 2004. [Online]. Available: http://portal.acm.org/citation.cfm?id=1005332.1005334

[22] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 6–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015424

[23] S. Sonnenburg, G. Rätsch, and C. Schäfer, "A General and Efficient Multiple Kernel Learning Algorithm," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 1273–1280.

[24] C. X. Ling, J. Du, and Z.-H. Zhou, "When does co-training work in real data?" in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, ser. PAKDD '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 596–603.

[25] Y. Zhang and Z.-H. Zhou, "Multi-label dimensionality reduction via dependence maximization," in *AAAI'08: Proceedings of the 23rd National Conference on Artificial Intelligence*, 2008, pp. 1503–1505.

[26] K. Yu, S. Yu, and V. Tresp, "Multi-label informed latent semantic indexing," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 258–265.

[27] G. Chen, Y. Song, F. Wang, and C. Zhang, "Semi-supervised multi-label learning by solving a sylvester equation," in *Proceedings of the SIAM International Conference on Data Mining*, 2008, pp. 410–419.

[28] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 668–676.

[29] S. Ji and J. Ye, "Linear dimensionality reduction for multi-label classification," in *Proceedings of the 21st international jont conference on Artifical intelligence*, 2009, pp. 1077–1082.

[30] B. Qian, X. Wang, F. Wang, H. Li, J. Ye, and I. Davidson, "Active learning from relative queries," in *IJCAI*, 2013.

[31] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[32] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, 2004, pp. 321–328.

[33] W. Liu, B. Qian, J. Cui, and J. Liu, "Spectral kernel learning for semi-supervised classification," in *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, pp. 1150–1155.

[34] N. D. Lawrence and M. I. Jordan, "Semi-supervised learning via gaussian processes," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 753–760.

[35] X. Zhu, "Semi-Supervised Learning Literature Survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep., 2005.

[36] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[37] J. Wang, T. Jebara, and S. fu Chang, "Graph transduction via alternating minimization," in *Proc. 25th ICML*, 2008.

[38] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *In NIPS05 Workshop, Inductive Transfer: 10 Years Later*, 2005.

[39] A. E. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *In Advances in Neural Information Processing Systems 14*, 2001, pp. 681–687.

[40] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based systemfor text categorization," *Mach. Learn.*, vol. 39, no. 2-3, pp. 135–168, May 2000.

[41] X. Zhang, Q. Yuan, S. Zhao, W. Fan, W. Zheng, and Z. Wang, "Multi-label Classification without the Multi-label cost," in *Proceedings of the Tenth SIAM International Conference on Data Mining*, Apr. 2010.

[42] Y. Yang, "An evaluation of statistical approaches to text categorization," *Journal of Information Retrieval*, vol. 1, pp. 67–88, 1999.

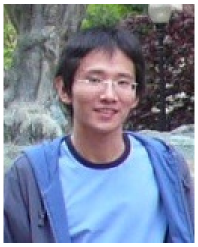
[43] A. Elisseeff and J. Weston, "A Kernel Method for Multi-Labelled Classification," in *Annual ACM Conference on Research and Development in Information Retrieval*, 2005, pp. 274–281.

[44] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

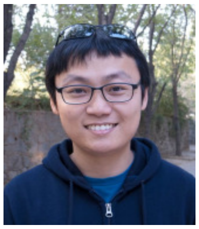
[45] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm

[46] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, November 2004.

[47] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml



Buyue Qian is currently a research scientist at IBM T. J. Watson Research. He received his PhD in 2013 from Computer Science Department, University of California at Davis. Before that, he received Master of Science (2009) from Columbia University, and BS in Information Engineering (2007) from Xi'an Jiaotong University. The major awards he received include Yahoo! Research Award, IBM Eminence & Excellence Award, and SIAM Data Mining'13 Best Research Paper Runner Up.



Xiang Wang is currently a research scientist at IBM T. J. Watson Research. He received his PhD in 2013 from Computer Science Department, University of California at Davis. Before that, he received Master of Software Engineering (2008) and BS in Mathematics (2004) from Tsinghua University. The major awards he received include SIAM Data Mining'13 Best Research Paper Runner Up, IBM Invention Achievement Award, and UC Davis Best Graduate Researcher Award.

Jieping Ye is an Associate Professor of Computer Science and Engineering at Arizona State University. He received his Ph.D. in Computer Science from University of Minnesota, Twin Cities in 2005. His research interests include machine learning, data mining, and biomedical informatics. He won the outstanding student paper award at ICML in 2004, the SCI Young Investigator of the Year Award at ASU in 2007, the SCI Researcher of the Year Award at ASU in 2009, the NSF CAREER Award in 2010, and the KDD best research paper award honorable mention in 2010.

Ian Davidson grew up in Australia and obtained his Ph.D. at Monash University, Melbourne. He joined the University of California Davis in 2007 and is now a Professor of Computer Science. He directs the Knowledge Discovery and Data Mining Lab and his work is supported by grants from NSF, ONR, OSD Google and Yahoo!. He is a senior member of the IEEE.