Multi-view Weak-label Learning based on Matrix Completion*

Qiaoyu Tan† Guoxian Yu† Carlotta Domeniconi‡ Jun Wang† Zili Zhang†

Abstract

Weak-label learning is an important branch of multi-label learning; it deals with samples annotated with incomplete (weak) labels. Previous work on weak-label learning mainly considers data represented by a single view. An intuitive way to leverage multiple features obtained from different views is to concatenate the features into a single vector. However, this process is not only prone to over-fitting and often results in very high time-complexity, but also ignores the potentially useful complementary information spread across the different views. In this paper, we propose an approach based on Matrix Completion for multi-view Weak-label Learning (McWL). Matrix completion (MC) has sound theoretical properties and is robust to missing values in both feature and label spaces. Our method enforces the optimization of multiple view integration and of MC-based classification within a unified objective function. Specifically, a kernel target alignment technique and the loss function of an MC-based classifier are used to jointly and iteratively adjust the weights assigned to individual views, and to optimize the classifier. McWL can selectively integrate views and is able to assign small weights to views of low quality. Extensive experiments on a broad range of datasets validate the effectiveness of our approach against competitive algorithms.

1 Introduction

In traditional supervised learning, each sample is associated with a single label, whereas in many applications a sample is often annotated with multiple labels. For example, an article can be tagged with multiple topics given as labels, such as politics, economics, sports, and culture. Multi-label learning is a paradigm developed to handle these scenarios, and has attracted much attention in machine learning and many application domains [1, 2]. Previous studies on multi-label learning usually assume the labels associated to samples are complete, and no labels are missing. In many data mining applications, however, it is rather difficult to collect all the labels associated to a sample, and only a partial label set may be available. For instance, an image may contain ‘tiger’, ‘trees’, and ‘forest’, but we may have available only the tag *tiger* for it. This kind of multi-label learning problem is called *weak-label learning* [3, 4], and has attracted increasing attention in recent years [5, 6, 7].

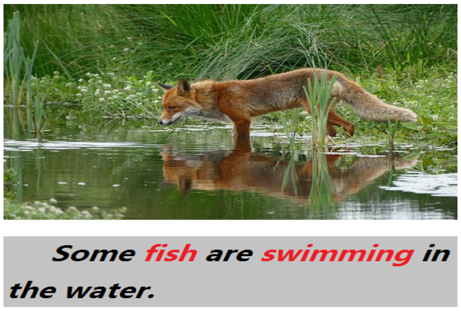
Some fish are swimming in the water.

Figure 1: An example of multi-view data with two views: image and text. By using both views, we can replenish missing labels (e.g., ‘food’), which cannot be induced by any of the single views. Courtesy: pixabay.com.

Although previous approaches achieve an excellent performance for general weak-label learning tasks, they mainly focus on single view scenarios [4, 5, 6]. However, in real-world tasks data can have multiple views. Namely, samples can be represented in several different feature spaces. For example, web images can be described using heterogenous features such as texture descriptors, shape descriptors, color descriptors, and the surrounding text [8]. An intuitive approach to utilize multiple views is to concatenate multi-view features into a single vector. But this strategy neglects the fact that features are extracted from different spaces with different statistical properties, and directly employing existing weak-label methods to multi-view learning may suffer from the over-fitting problem, especially when the dimensionality of samples is much larger than the number of samples. Besides, feature concatenation often leads to high time complexity, and this time cost may be unacceptable for many applications [9]. Feature concatenation also ignores the complementary information across different views which can help to replenish missing labels. As an example, Figure 1 contains two different views of a dataset, an image view and a text view. We can

---

[*] Supported by NSF of China (No. 61402378 and 61741217).
Corresponding Author: Guoxian Yu (gxyu@swu.edu.cn).

† College of Computer and Information Science, Southwest University, China

‡ Department of Computer Science, George Mason University, USA

easily derive three labels for the image view, namely 'fox', 'water' and 'grass', and two labels for the text view, namely 'fish' and 'swimming'. If we use the two views independently, we may obtain some of the high-level semantic concepts (labels), and miss others. For example, 'food' might become a missing label, since the image shows a fox trying to catch a fish in the water for food. Complementary information derived from different views can be used to replenish missing labels.

Recently, matrix completion (MC) has been exploited for multi-label leaning [10, 11, 12] and multi-view multi-label learning [9, 13], due to its solid mathematical foundation. However, existing MC-based methods usually model the fusion of multiple views and the prediction tasks as separate objectives. As such, they may result in an optimal multiple view integration, but not necessarily in an optimal prediction [14, 15]. In addition, they all assume that the available data labels are complete.

To address the aforementioned issues, we propose a novel multi-view weak-label learning model, termed as multi-view weak-label learning based on matrix completion (McWL). McWL effectively and simultaneously models the fusion of different kinds of features and an MC-based prediction function. McWL uses graphs to describe the relationship among samples collected from different views. To explore the complementarity of different views, a kernel target alignment technique is then used to combine the graphs into a composite graph. The composite graph is fed to an MC-based classifier under the form of constraints. Different from previous MC-based multi-view multi-label learning approaches [9, 13], McWL can jointly optimize the integration of multiple graphs and the MC-based prediction classifier in a unified objective function. In addition, McWL takes into account the unbalanced label problem which is often serious in weak-label learning problems [7], and incorporates a weighted label scheme into the unified objective function to give more emphasis to the labels with fewer related samples. Experimental results on five multi-view multi-label datasets show that the proposed McWL achieves superior performance against state-of-the-art approaches across various evaluation criteria.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 elaborates on the proposed McWL. Experimental results and conclusions are provided in Section 4 and Section 5.

2 Related Work

This work is related to three branches of studies, weak-label learning, multi-view learning, and matrix completion. Weak-label learning, as an important branch of multi-label learning, has attracted great interest and many weak-label learning algorithms have been proposed in recent years, e.g. weak-label learning algorithms under a supervised setting [3, 4, 5] and under a semi-supervised setting [7, 16, 17]. Most weak-label learning approaches assume that data is described by a single feature space (single view), ignoring the widely witnessed multi-view data.

Multi-view learning deals with data represented by multiple feature views [18]. Existing approaches have considered multi-views in conjunction with subspace learning [9, 19], with co-training [20], or with multiple kernel learning [21, 15, 22]. Almost all previous multi-view learning studies assume a complete annotation for training samples. The only exception is LabelMe [8], but this method treats each view equally, and may result in performance degradation when low quality views exist.

Matrix completion (MC) tackles the problem of recovering a low-rank matrix from a limited number of observed entries [23]. It has been recently exploited for multi-label learning [11, 24] and weak-label learning [12] due to its solid mathematical theory. Although these methods perform well in single view scenarios, they cannot be directly used with multi-view data because they neglect the potential feature correlation between different views and may cause over-fitting. In order to make use of available heterogeneous features from multiple views, some methods have applied matrix completion to multi-view learning [9, 13]. However, these techniques usually work under the complete label assumption, or do not explicitly consider the widely spread weak-label scenarios. Furthermore, they model the fusion of multiple views and the MC-based prediction tasks as separate objectives, which may lead to a suboptimal solution.

In this study, we design an MC-based multi-view weak-label learning model, called McWL. McWL jointly optimizes the fusion of multiple views and the MC-based prediction in a unified objective function. It is worth to note that McWL differs from LabelMe [8] in that it can selectively assign weights to views and is able to assign smaller weights to noisy views. To the best of our knowledge, this is the first study that addresses multi-view weak-label learning using matrix completion.

3 The McWL Approach

In this section, we first introduce the problem statement and the used notations. We then describe the general framework, named McWL, which simultaneously integrates the fusion of multiple views and the MC-based prediction problem for weak-label learning. Finally, we develop an optimization method to iteratively optimize the multiple view integration and the MC-based classifier.

3.1 Data Representation and Notation Suppose $\mathcal{X} = \{\mathbf{X}^v\}_{v=1}^m$ represents a dataset with n samples and m views, where $\mathbf{X}^v \in \mathbb{R}^{n \times d_v}$ is the feature space of the v-th view. $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T \in \{-1, 1\}^{n \times q}$ is the corresponding weak-label matrix, where $\mathbf{y}_i \in \{-1, 1\}^q$ is the label vector of $\mathbf{x}_i$ and q is the number of labels considered. $\mathbf{y}_{ic} = 1$ $(c = 1, ..., q)$ indicates that the c-th label is a proper label for $\mathbf{x}_i$, while $\mathbf{y}_{ic} = -1$ gives us no information (i.e., the absence of a specific label does not imply that it is not appropriate for the sample). Without loss of generality, we assume that, out of n samples, the first l are partially labeled samples, and the remaining u are unlabeled samples, where $n = l + u$.

How to define a suitable approach to efficiently capture the intrinsic structure of samples across different views is still a challenging problem. In this paper, we adopted the popular k nearest neighbor (kNN) approach to construct a graph for each view, where each node represents a sample, and the weighted edge between two nodes represents their similarity. In this way, we not only can efficiently describe the feature-level similarity among samples and facilitate the integration of multiple views, but also can leverage the information concerning unlabeled and labeled samples to train a semi-supervised weak-label learning classifier. The weighted adjacency matrix of the kNN graph is defined as follows:

(3.1)
$$\mathbf{W}_{ij}^v = \begin{cases} 1, \text{ if } \mathbf{x}_i^v \in kNN(\mathbf{x}_j^v) \text{ or } \mathbf{x}_j^v \in kNN(\mathbf{x}_i^v) \\ 0, \text{ otherwise} \end{cases}$$

where $\mathbf{W}^v \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix of n samples in the v-th view, $\mathbf{W}_{ij}^v$ represents the weight of the edge between samples i and j. $\mathbf{x}_i^v \in kNN(\mathbf{x}_j^v)$ is one of the k nearest neighbors of $\mathbf{x}_j^v$, and the neighborhood relationship between samples is determined using the Euclidean distance. For simplicity, we use binary weights in Eq.(3.1), but other formulations are also possible. Let $\mathcal{W} = \{\mathbf{W}^v\}_{v=1}^m$ denotes the set of adjacency matrices of m views. Our goal is to use the m graphs defined by $\mathcal{W}$ to train an MC-based multi-view weak-label learning classifier.

3.2 Integrating multiple views In the past years, many multi-view approaches have been proposed to use the complementary information of heterogenous views, but most of them often treat each view equally [13] and may result in performance degradation when low quality (noisy) views exist [15, 18]. As such, to avoid the influence of noisy views, we try to assign different weights to different views, and resort to a linear regression problem as follows:

(3.2)
$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} ||\mathbf{K} - \mathbf{W}||_F^2, \; s.t. \; \mathbf{W} = \sum_{v=1}^{m} \boldsymbol{\theta}_v \mathbf{W}^v, \; \boldsymbol{\theta}_v \geq 0$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_m]$; $\mathbf{W}$ is the composite weighted graph learned by combining the m individual graphs $\mathcal{W}$; $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the target kernel induced from labels and defined as $\mathbf{K} = \sum_{c=1}^q \mathbf{K}^c$, where $\mathbf{K}^c$ is the c-th label-induced target kernel; $|| \cdot ||_F$ represents the Frobenius norm, we use this norm just for its simplicity and wide applications, other norms like $l_{2,1}$ can also be applied. Eq.(3.2) has close connection with kernel-target alignment [25], and the target aligned kernel can often enhance the performance of kernel-based classifier and regression. The c-th label-induced target kernel $\mathbf{K}^c$ is defined as follows:

(3.3)
$$\mathbf{K}_{ij}^c = \begin{cases} \frac{(n_c^+)^2}{l^2}, \text{ if } \mathbf{y}_{ic} = \mathbf{y}_{jc} = 1 \\ \frac{n_c^+ n_c^-}{l^2}, \text{ if } \mathbf{y}_{ic}\mathbf{y}_{jc} = -1, \; \mathbf{y}_{ic} + \mathbf{y}_{jc} = 0, \; i, j \leq l \\ 0, \text{ otherwise} \end{cases}$$

where n_c^+ (n_c^-) is the number of samples currently (not) annotated with the c-th label. Since a label often has less positive samples than negative ones, that is $n_c^+ < n_c^-$, it follows $(\frac{n_c^-}{l})^2 > \frac{n_c^+ n_c^-}{l^2}$. From definition Eq.(3.3), the more labels two samples have in common, the larger the weight of the edge connecting them in the target label graph is. Mostafavi *et al.* [14] adapted this idea to define the target kernel and to reconstruct the kernel label association. They set the weight, corresponding to the edge between two samples such that one has the c-th label and the other does not, in the c-th target graph equal to $-\frac{n_c^+ n_c^-}{n^2}$. However, this setup assumes that $\mathbf{y}_{ic} = -1$ means that the i-th sample is not associated with the c-th label. As discussed above, this assumption is often violated in weak-label scenarios. Furthermore, if sample i is a weak-label sample and $\mathbf{W}(i, j)$ is large, then sample j is likely to share some labels with the i-th sample. Given this, we set the corresponding entry as $\frac{n_c^+ n_c^-}{l^2}$ instead.

By minimizing Eq.(3.2), we aim to credit large weights to views in which more similar samples share more labels, and small weights to views in which similar samples share few (or no) labels. As a result, we assign larger weights to views that are coherent with the labels. This is consistent with the widely used smoothness assumption [26], which implies that similar samples should have similar labels.

Based on the fact that $tr(\mathbf{KW}) = vec(\mathbf{K})^T vec(\mathbf{W})$, where $vec(\mathbf{K})$ is the vector operator that stacks the columns of $\mathbf{K}$ together, we can rewrite Eq.(3.2) as a non-negative quadratic programming problem:

$$\begin{aligned} \boldsymbol{\theta} &= \arg\min_{\boldsymbol{\theta}} ||vec(\mathbf{K}) - vec(\mathbf{W})\boldsymbol{\theta}^T||_F^2 \\ &\quad s.t. \; \boldsymbol{\theta}_v \geq 0, \; 1 \leq v \leq m \end{aligned} \tag{3.4}$$

where $vec(\mathbf{W}) = [vec(\mathbf{W}^1), ..., vec(\mathbf{W}^m)] \in \mathbb{R}^{n^2 \times m}$.

3.3 Matrix completion based multi-view classification

Recently, some MC based methods have been proposed for single view weak-label learning tasks [12] and multi-view learning tasks [9, 13]. Although two existing MC-based multi-view algorithms [9, 13] can be used in missing label scenarios, neither explicitly considers the widely witnessed missing labels and they both need to estimate a matrix of size $(n \times (d+q))$ (d is the dimensionality of samples), which may lead to unacceptable high time complexity for many applications. Xu *et al.* [24] recently proposed a speedup matrix completion approach (Maxide) by using the feature matrix as side information. The formulation of Maxide is as follows:

$$\min_{\mathbf{Z}\in\mathbb{R}^{d\times q}} \mathcal{L}(\mathbf{Z}) = \alpha||\mathbf{Z}||_{tr} + \frac{1}{2}||\mathcal{R}_\Omega(\mathbf{X}^0\mathbf{Z} - \mathbf{Y})||_F^2 \tag{3.5}$$

where $\mathbf{Z} \in \mathbb{R}^{d\times q}$ is the target recovery matrix, and $\mathbf{X}^0 \in \mathbb{R}^{n\times d}$ is the feature matrix. $\alpha \geq 0$ balances the importance of the two terms, $\mathcal{R}_\Omega(\mathbf{Y})$ is a linear operator, where $\mathcal{R}_\Omega(\mathbf{Y})_{i,j} = \mathbf{Y}_{i,j}$ if (i,j) is an observed entry in $\mathbf{Y}$; $\mathcal{R}_\Omega(\mathbf{Y})_{i,j} = 0$, otherwise. Solving Eq.(3.5) requires to search for an optimal matrix $\mathbf{Z}$ of size $d \times q$. The main assumption made by the Maxide approach is that $d \ll n$. This usually holds in low-dimensional single view learning problems, but it does not hold in many multi-view data mining applications. To transform multi-view learning tasks into a single view learning problem, one can concatenate the multi-view features into a single vector; unfortunately, this transformation may not only result in impractical time complexity for recovering the matrix $\mathbf{Z}$ (being $d \gg n$), but also result in over-fitting the data. As such, how to effectively and efficiently mine multi-view data remains a difficult challenge.

To address this issue, we substitute $\mathbf{X}^0$ in Eq.(3.5) with the composition graph $\mathbf{W}$ as side information, and update the multi-view weak-label learning as follows:

$$\min_{\mathbf{Z}\in\mathbb{R}^{n\times q}} \mathcal{L}(\mathbf{Z}) = \alpha||\mathbf{Z}||_{tr} + \frac{1}{2}||\mathcal{R}_\Omega(\mathbf{W}\mathbf{Z} - \mathbf{Y})||_F^2 \tag{3.6}$$

$\mathbf{Y} \in \mathbb{R}^{n\times q}$ is the target label matrix, where the first l data are partially labeled samples and the remaining are unlabeled samples. It is important to observe that, unlike existing MC-based multi-view learning approaches (e.g., [9] and [13]), which need to estimate a matrix of size $n \times (d+q)$, Eq.(3.6) estimates a significantly smaller matrix of size $n \times q$.

An inherent property of learning with multi-label data is class-imbalance among labels, and this issue has not been addressed in Eq.(3.6). Class-imbalance has long been regarded as one fundamental threat that can compromise the performance of standard data mining algorithms [27]. To address this limitation, we modify $\mathbf{y}_{ic}$ into $\tilde{\mathbf{y}}_{ic} = \mathbf{y}_{ic} \log \frac{\hat{n}}{n_c^+}$, where $\hat{n} = \sum_{c=1}^{q} n_c^+$, and n_c^+ represents the number of samples tagged with the c-th label. This modification has the effect of putting more emphasis on labels with fewer relevant samples and forces the optimizer to focus on these labels. Setting $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, ..., \tilde{\mathbf{y}}_n]^T$, Eq.(3.6) can be rewritten as follows:

$$\min_{\mathbf{Z}\in\mathbb{R}^{n\times q}} \mathcal{L}(\mathbf{Z}) = \alpha||\mathbf{Z}||_{tr} + \frac{1}{2}||\mathcal{R}_\Omega(\mathbf{W}\mathbf{Z} - \tilde{\mathbf{Y}})||_F^2 \tag{3.7}$$

3.4 The unified objective function

Eq.(3.4) can be used to compute $\boldsymbol{\theta}$ for the individual views (kernels) and fuse these views into a composite graph. An MC-based classifier can then be applied on the composition graph $\mathbf{W}$ to predict labels. However, this widely adopted paradigm may result in a suboptimal solution, since the optimization of multiple views (resources) is decomposed into two separate objectives, as done in two MC-based approaches (lrMVL [9], MVMC-LS [13]), and the optimized composite graph may not be optimal for the follow-up predictor [15]. To avoid this risk, we integrate the two objectives defined in Eq.(3.4) and Eq.(3.7) into a unified function as follows:

$$\begin{aligned}\boldsymbol{\theta} &= \arg\min_{\boldsymbol{\theta},\mathbf{Z}} ||vec(\mathbf{K}) - \sum_{v=1}^{m} \boldsymbol{\theta}_v vec(\mathbf{W}^v)||_F^2 \\ &\quad + \lambda(\alpha||\mathbf{Z}||_{tr} + \frac{1}{2}||\mathcal{R}_\Omega(\sum_{v=1}^{m} \boldsymbol{\theta}_v vec(\mathbf{W}^v)\mathbf{Z} - \tilde{\mathbf{Y}})||_F^2) \\ &\quad s.t.\ \boldsymbol{\theta}_v \geq 0,\ 1 \leq v \leq m\end{aligned} \tag{3.8}$$

where $\lambda \geq 0$ is used to control the importance of multi-view integration and MC-based classification. By combining the objectives of MC-based classification and of target graph alignment in a unified objective, we can therefore enforce the composition graph to be coherently optimal with respect to both objectives.

3.5 Optimization

Two vector variables ($\boldsymbol{\theta}$ and $\mathbf{Z}$) need to be optimized in Eq.(3.8). Since the problem cannot be solved directly, we develop an EM-style [28] algorithm to find the optimal solution.

3.5.1 Z update ($\boldsymbol{\theta}$ fixed)

We initially consider all views as equally relevant, and initialize $\boldsymbol{\theta}_v = 1$ ($v = 1, 2, ..., m$) with $\mathbf{W}$ fixed. Our goal is to minimize the matrix $||\mathbf{Z}||_{tr}$, where $\mathcal{R}_\Omega(\mathbf{W}\mathbf{Z}) = \mathcal{R}_\Omega(\tilde{\mathbf{Y}})$. As in the Singular Vector Thresholding (SVT) method [29], we can approximate the problem of finding the optimal $\mathbf{Z}$ in Eq.(3.7) with an unconstrained optimization problem. Eq.(3.7) can be efficiently solved using Maxide [24], which only needs to estimate a matrix of size $n \times q$.

3.5.2 $\boldsymbol{\theta}$ update (Z fixed)

Given $\mathbf{Z}$, the subproblem of optimizing Eq.(3.8) with respect to $\boldsymbol{\theta}$ can be rewritten as follows:

$$H(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} -2\boldsymbol{\theta}^T vec(\mathbf{W})^T vec(\mathbf{K}) + \boldsymbol{\theta}^T vec(\mathbf{W})^T vec(\mathbf{W})\boldsymbol{\theta} + vec(\mathbf{K})^T vec(\mathbf{K}) + \lambda(-2\boldsymbol{\theta}^T \boldsymbol{\mu} + \boldsymbol{\theta}^T \boldsymbol{\Theta}\boldsymbol{\theta}) \quad s.t.\ \boldsymbol{\theta}_v \geq 0,\ 1 \leq v \leq m \tag{3.9}$$

where $\boldsymbol{\Theta}$ is an $m \times m$ matrix with $\boldsymbol{\Theta}(v', v'') = tr(\mathcal{R}_\Omega(\mathbf{W}^{v'}\mathbf{Z})^T \mathcal{R}_\Omega(\mathbf{W}^{v''}\mathbf{Z}))$, and $\boldsymbol{\mu}$ is an $m \times 1$ vector with $\boldsymbol{\mu}_v = tr(\mathcal{R}_\Omega(\tilde{\mathbf{Y}})^T \mathcal{R}_\Omega(\mathbf{W}^v\mathbf{Z}))$. Since $\mathbf{K}$, $\mathbf{W}$ and $\mathbf{Z}$ are known, the above equation gives a quadratic programming problem and is convex with respect to $\boldsymbol{\theta}$. Taking the derivative of $H(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, we can obtain the following solution:

$$\boldsymbol{\theta} = (vec(\mathbf{W})^T vec(\mathbf{W}) + \lambda\boldsymbol{\Theta})^{-1}(vec(\mathbf{W})^T vec(\mathbf{K}) + \lambda\boldsymbol{\mu}) \tag{3.10}$$

It's easy to see that when $\lambda = 0$, only the kernel target alignment criterion is used to optimize $\boldsymbol{\theta}$, and McWL degenerates into two separate optimization objectives.

The learning procedure of McWL is summarized in Algorithm 1.

Algorithm 1 Multi-view weak-label learning based on Matrix Completion (McWL)

```
Input: Multi-view feature matrices $\mathcal{X}$, $\mathbf{Y}$, $\alpha$, $\lambda$ and $k$.
Output: Predicted likelihood score matrix $\hat{\mathbf{Y}}$.
Initialize $\boldsymbol{\theta}_v = 1 (1 \leq v \leq m)$.
Construct multiple graphs $\mathcal{W}$ using Eq.(3.1).
while convergence is not reached do
    update Z using Eq.(3.7).
    update $\boldsymbol{\theta}$ using Eq.(3.10).
end while
Return the predicted likelihood score matrix $\hat{\mathbf{Y}} = \mathbf{WZ}$.
```

3.6 Complexity analysis $\mathcal{W} = \{\mathbf{W}^v\}_{v=1}^m$ can be computed before the iterative process. The time complexity to compute $tr(\mathbf{Z}^T\mathbf{W}^T\mathbf{W}\mathbf{Z})$ is $O(qn^2)$. $\boldsymbol{\Theta}$ is an $m \times m$ symmetric matrix and m is usually smaller than 10. In each iteration there are $m(m+1)/2$ elements to be computed, so the time complexity for $\boldsymbol{\Theta}$ is $O(m(m+1) \times qn^2)$. The complexity of matrix completion for $\mathbf{Z}$ in Eq.(3.5) is $O(r(n+q)\ln(n+q)\ln n)$, where r is the rank of the matrix $\mathbf{Z}$ to be estimated. Since the cost of computing the matrix inverse in Eq.(3.10) and the cost for $\boldsymbol{\mu}$ in Eq.(3.9) are smaller than $\boldsymbol{\Theta}$, the overall time complexity of McWL is $max\{O(m^2Tqn^2), O(T(r(n+q)\ln(n+q)\ln n))\}$, which is $O(m^2Tqn^2)$ since $n \gg q$ and $n^2 \gg (n\ln(n+q)\ln n)$. T is the number of iterations to reach convergence. In practice, T does not exceed 20. In our study, the adjacency matrix of the individual view and the composition graph are all sparse, with $O(n)$ nonzero elements. For this reason, the actual time costs of the above operations can be further reduced. In fact, our runtime comparisons on multi-view datasets show that McWL generally runs much faster than other MC-based multi-view learning methods [9, 13].

4 Experiments

4.1 Experimental setup Five multi-view datasets used in the experiments are summarized in Table 1. Core15k, Pascal07 and ESPGame are three multi-view image datasets obtained from [30] and each image is represented by six representative feature views: HUE, SIFT, GIST, HSV, LAB and RGB. The dimensionality of HUE, SIFT, GIST, and the last three feature views are 100, 1000, 512, and around 4000, respectively. Yeast is a biological dataset with two views [31], namely the genetic expression and the phylogenetic profile of a gene. Emotions is a music dataset with two views [31], where the first view is obtained by extracting periodic changes from a beat histogram, and the second one is obtained from the mel frequency cepstral coefficients and from the spectral centroid, spectral rolloff, and spectral flux-extracted short-term Fourier transform. For each dataset, we randomly sample 70% data for training and use the remaining 30% data for testing (unlabeled data). No label assignment is provided for any test data. To create weak-label scenarios for the training data, we followed the protocol in [24]: for each label c we expose the label assignment of c for $\omega\%$ randomly sampled positive training data, and keep the c label assignment as unknown for the rest of the training data. For example, if $\omega\% = 30\%$ and the number of samples annotated with label c is 100, then we randomly sample 30 points as positive training data for label c and keep the c label assignment for the remaining 70 samples as unknown by changing $y_{\cdot c} = 1$ to $y_{\cdot c} = -1$.

Table 1: Datasets used in the experiments. n is the number of samples, m is the number of views, q is the number of distinct labels, and Avg is the average number of labels per sample.

Data sets	n	m	q	Avg
Emotions	593	2	6	1.869
Yeast	2417	2	14	4.237
Core15k	4999	6	260	3.396
Pascal07	9963	6	20	1.465
ESPGame	20770	6	268	4.686

4.1.1 Methods We compared McWL against five state-of-the-art methods: lrMVL [9], MVMC-LS [13], LabelMe [8], MLAN [22], and MLR-GL [4]. McWL, lrMVL, and MVMC-LS are three matrix completion based multi-view multi-label learning methods; LabelMe

Table 2: Results on different datasets with $\omega\% = 30\%$. In addition, ●/○ indicates whether McWL is statistically superior/inferior to the comparing algorithms under a particular evaluation metric (pairwise t-test at 0.05 significance level).

Metric	MLAN	MLR-GL	LabelMe	lrMVMC	MVMC-LS	McWL-En	McWL($\lambda = 0$)	McWL
				Emotions				
1-HL	0.566 ± 0.006●	0.602 ± 0.001●	0.629 ± 0.009○	0.585 ± 0.003●	0.530 ± 0.003●	0.621 ± 0.005	0.677 ± 0.006○	0.623 ± 0.022
1-RL	0.535 ± 0.009●	0.678 ± 0.003○	0.557 ± 0.004	0.639 ± 0.008○	0.554 ± 0.001●	0.587 ± 0.006○	0.683 ± 0.005○	0.566 ± 0.043
AP	0.529 ± 0.008●	0.567 ± 0.003●	0.586 ± 0.003	0.544 ± 0.009●	0.489 ± 0.000●	0.583 ± 0.004	0.665 ± 0.005○	0.586 ± 0.030
AUC	0.586 ± 0.006●	0.624 ± 0.004●	0.605 ± 0.008●	0.607 ± 0.007●	0.541 ± 0.001●	0.665 ± 0.008○	0.746 ± 0.004○	0.640 ± 0.035
				Yeast				
1-HL	0.634 ± 0.002●	0.725 ± 0.002●	0.645 ± 0.005●	0.720 ± 0.001●	0.716 ± 0.002●	0.597 ± 0.002●	0.700 ± 0.002●	0.759 ± 0.001
1-RL	0.730 ± 0.002●	0.787 ± 0.002●	0.775 ± 0.003●	0.784 ± 0.002●	0.760 ± 0.001●	0.571 ± 0.002●	0.729 ± 0.003●	0.810 ± 0.004
AP	0.501 ± 0.002●	0.700 ± 0.002●	0.644 ± 0.009●	0.701 ± 0.002●	0.650 ± 0.002●	0.488 ± 0.001●	0.637 ± 0.004●	0.735 ± 0.005
AUC	0.630 ± 0.002●	0.798 ± 0.001●	0.788 ± 0.005●	0.793 ± 0.002●	0.786 ± 0.001●	0.605 ± 0.003●	0.746 ± 0.002●	0.823 ± 0.003
				Core15k				
1-HL	0.946 ± 0.000●	0.951 ± 0.001●	0.952 ± 0.000●	0.954 ± 0.000●	0.947 ± 0.000●	0.955 ± 0.000●	0.961 ± 0.000●	0.963 ± 0.000
1-RL	0.726 ± 0.003●	0.866 ± 0.003●	0.776 ± 0.001●	0.860 ± 0.001●	0.756 ± 0.003●	0.647 ± 0.002●	0.787 ± 0.002●	0.885 ± 0.001
AP	0.281 ± 0.001●	0.418 ± 0.010●	0.348 ± 0.005●	0.437 ± 0.001●	0.317 ± 0.002●	0.265 ± 0.003●	0.382 ± 0.002●	0.454 ± 0.001
AUC	0.786 ± 0.002●	0.818 ± 0.003●	0.800 ± 0.001●	0.811 ± 0.001●	0.760 ± 0.003●	0.654 ± 0.001●	0.792 ± 0.002●	0.889 ± 0.001
				Pascal07				
1-HL	0.847 ± 0.000●	0.882 ± 0.000●	0.882 ± 0.000●	0.882 ± 0.000●	0.845 ± 0.000●	0.857 ± 0.000●	0.878 ± 0.000●	0.893 ± 0.000
1-RL	0.649 ± 0.002●	0.767 ± 0.001●	0.764 ± 0.001●	0.765 ± 0.002●	0.693 ± 0.000●	0.589 ± 0.002●	0.735 ± 0.002●	0.816 ± 0.001
AP	0.424 ± 0.002●	0.485 ± 0.001●	0.485 ± 0.001●	0.483 ± 0.003●	0.398 ± 0.001●	0.306 ± 0.001●	0.444 ± 0.002●	0.535 ± 0.002
AUC	0.730 ± 0.001●	0.787 ± 0.001●	0.785 ± 0.001●	0.786 ± 0.001●	0.699 ± 0.000●	0.610 ± 0.002●	0.754 ± 0.002●	0.839 ± 0.001
				ESPGame				
1-HL	0.965 ± 0.000●	0.964 ± 0.000●	0.964 ± 0.000●	0.970 ± 0.000●	0.965 ± 0.000●	0.969 ± 0.000●	0.971 ± 0.000●	0.974 ± 0.000
1-RL	0.567 ± 0.002●	0.493 ± 0.015●	0.576 ± 0.000●	0.779 ± 0.001○	0.528 ± 0.000●	0.585 ± 0.001●	0.688 ± 0.001●	0.768 ± 0.001
AP	0.071 ± 0.001●	0.034 ± 0.003●	0.025 ± 0.001●	0.185 ± 0.001●	0.052 ± 0.001●	0.141 ± 0.000●	0.213 ± 0.001●	0.307 ± 0.001
AUC	0.589 ± 0.001●	0.489 ± 0.014●	0.555 ± 0.000●	0.784 ± 0.000○	0.556 ± 0.000●	0.586 ± 0.001●	0.693 ± 0.001●	0.771 ± 0.000

and MLR-GL are weak-label learning methods, and MLAN is a multi-view learning method. MLAN was initially proposed for single label classification; we adapt it for a multi-label scenario by assigning multiple labels instead of a single one to unlabeled data. To further investigate the benefit of simultaneously optimizing the fusion of multiple views and the MC-based classification, we introduce McWL($\lambda = 0$) and McWL-En. McWL($\lambda = 0$) isolates multiple view fusion from MC-based prediction; McWL-En trains multiple MC-based classifiers using Eq.(3.7) for individual views and then combines these base classifiers into an ensemble classifier.

We adapted the original code of lrMVL, MVMC-LS, LabelMe, MLAN and MLR-GL for our experiments. The code was downloaded online or provided by the authors. Five-fold cross validation is used to select the optimal parameter values for each competitive method. For lrMVL, we set the parameter $\mu = 0.25\sigma_1$ (σ_1 is the largest singular value of the recovery matrix), and decreases it using a factor of 0.25 in the continuation steps until $\mu = 10^{-12}$; parameter λ is tuned using the set of values $\{10^i | i = -4, \cdots, 3\}$. For MVMC-LS, the parameter η is tuned using the set $\{10^i | i = -2, \cdots, 5\}$. For LabelMe, the two parameters θ_1 and θ_2 are tuned in $[0.1, 1]$. For MLAN, the parameter λ is initialized to a random positive value between 1 and 30, as suggested in the original paper. In our experiments, parameters α, λ and k for McWL are tuned in $\{2^i | i = -5, \cdots, 5\}$, $[0.1, 1]$ and $[1, 20]$, respectively, and finally we set $\alpha = 2^3$, $\lambda = 0.5$ and $k = 15$ for experiments. All the experiments are independently repeated ten times under each fixed setting, and both the mean and standard deviation are reported. *The source code of McWL is publicly available at* `http://mlda.swu.edu.cn/codes.php?name=McWL`.

4.1.2 Evaluation Four popular evaluation metrics for multi-label learning are adopted for performance comparisons: Hamming Loss (HL), Ranking Loss (RL), average precision (AP), and adapted AUC. The formal definition of the first three metrics can be found in reference [1]. The adaptive AUC is suggested in [4]. To maintain consistency with other evaluation metrics, we report 1-HL and 1-RL instead of HL and RL, respectively. Thus, as for the other metrics, the higher the value of 1-HL and 1-RL, the better the performance is. These metrics evaluate multi-label classification from different points of view, and therefore it is unlikely that a single method outperforms all the other techniques on all the metrics.

4.2 Results on All Datasets

The results obtained for all the methods on five datasets across four evaluation metrics are presented in Table 2. A self-test with different ratios of missing labels ($\omega\%$) is carried out to see the effect on performance, with $\omega\%$ varying between 30% and 70%, with a step-size of 20%. McWL, in general, outperforms other competitive methods in most cases when $\omega\% = 30\%$ (50% or 70%). For space limitation, we only report the results for $\omega\% = 30\%$. From Table 2 we can observe that McWL achieves the best (or comparable to the best) performance on several datasets across four evaluation metrics. MLR-GL, LabelMe, and McWL are weak-label learning methods, but McWL frequently outperforms the former two methods across four evaluation metrics. The main reason is that the former two methods treat each view equally, and cannot selectively assign weights to different views. As previously discussed, an equal weight assignment may result in performance degradation when low quality views exist. This comparison justifies our motivation to give different weights to views. Both MLAN and McWL are multi-view learning methods; they can jointly optimize the composite graph and the classifier on the composite graph and give different weights to different views, but McWL almost always outperforms MLAN. The reason is threefold: (i) MLAN assumes the available labels are complete, ignoring the widely spread weak-label scenarios; (ii) it does not take into account the unbalanced label problem, which may be the cause of performance degradation; and (iii) McWL is an MC-based classifier, which is more robust to missing values as suggested in [9] and in [13].

Whereas lrMVMC, MVMC-LS and McWL are all MC-based multi-view multi-label learning methods, which aim at integrating multiple views for prediction, McWL still outperforms the former two methods in many cases. This is because both lrMVMC and MVMC-LS are two-phase methods, and they consider the optimization of multi-view integration and of the MC-based classification as separate objectives. In addition, they assume the available labels of samples are complete. As discussed above, this assumption is often violated in practice.

McWL($\lambda = 0$) is a degenerate case of McWL obtained by isolating the fusion of multiple views from the MC-based classification. McWL($\lambda = 0$) is almost always outperformed by McWL on these datasets, confirming the advantage of optimizing the two objectives simultaneously. When the objectives are treated separately, an optimal multiple view integration maybe achieved but may not be optimal for the follow-up prediction [15]. These results corroborate our motivation to jointly optimize the two objectives. Classifier ensembles are widely-used and can effectively integrate multiple views. Nevertheless, McWL-En is outperformed by McWL, which only takes advantage of a single classifier. The possible reason is that McWL can identify noisy views, and discard or assign smaller weights to them, while McWL-En combines multiple views with equal weights and ignores the impact of noisy (or low quality) ones. Another related cause is that the performance of the base classifiers of McWL-En may be poor, since they operate on separate views, and therefore the ensemble cannot be effective. These comparisons justify once again our motivation to unify the optimization of the multiple view fusion and of the MC-based classification.

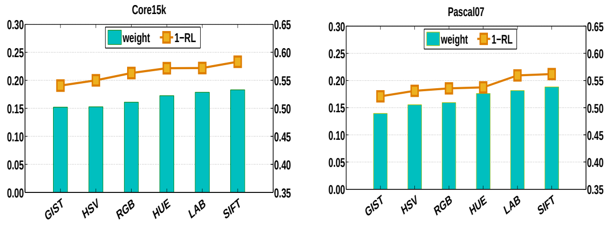

Figure 2: View weights learned by McWL, along with the 1-RL values of each view. The left y-axis of the biaxial represents the weight, and the right y-axis gives the 1-RL. The weights and the 1-RL scores are in general agreement.

4.3 Analysis of assigned weights In Figure 2, we report the weight coefficients $\boldsymbol{\theta}$ learned by McWL, and the corresponding 1-RL values by using MC for each view. For a fair comparison and a better visualization, we scale these weights in the [0,1] interval via $\boldsymbol{\theta}_v / \sum_{i=1}^{m} \boldsymbol{\theta}_i$. Due to space limitation, we only show the results on Core15k and Pascal07; we have similar observations on the other datasets. From Figure 2, we can observe that the trend of the weight values is consistent with the corresponding performance, i.e., the views with higher classification performance usually receive larger weights. According to both weight coefficients and 1-RL scores, the rank of the six views is SIFT>LAB>HUE>RGB>HSV>GIST. These comparisons demonstrate the effectiveness of McWL in combining multiple views.

4.4 Sensitivity Analysis of Parameters In this section we test the sensitivity of McWL w.r.t λ and α. The tested ranges for λ and α are [0.1,1] and $\{2^i | i = -5, \cdots, 5\}$ respectively. For brevity, we only report the 1-RL and AUC results on Yeast in Figure 3; however, similar results were obtained for the other datasets as well. From the results, we can see that McWL achieves a stable and good performance for a wide range of λ and α values. In addition, as we can see from both evaluation metrics, the performance of McWL tends to decrease when λ is close to 0. These results corroborate our motivation to jointly optimize the two objectives.

In addition, we also conduct experiments to investigate the sensitivity of McWL w.r.t k. Figure 4 gives the 1-RL values of McWL on Yeast and Core15k datasets when k varies from 1 to 20. The performance trend for the other datasets are similar to those reported in Figure 4. From the Figure we can see that the performance of McWL on both datasets increases as k increases, and it achieves stable performance when k is between 12 and 20. This is mainly because too small k can not well capture the geometric structure of samples. These results

Table 3: Runtime comparison (in seconds).

	Emotions	Yeast	Core15k	Pascal07	ESPGame	Total
McWL	1.05	4.52	43.07	894.95	2966.77	3910.36
MLAN	1.04	38.24	198.83	1500.93	7377.86	9116.89
MLR-GL	0.23	9.29	131.48	335.04	1986.11	2462.16
LabelMe	0.54	0.96	1308.61	6929.49	4542.15	12781.74
lrMVMC	2.34	1.94	4476.23	7725.56	9415.02	21621.08
MVMC-LS	3.74	21.76	21292.15	17278.55	26713.39	65309.59

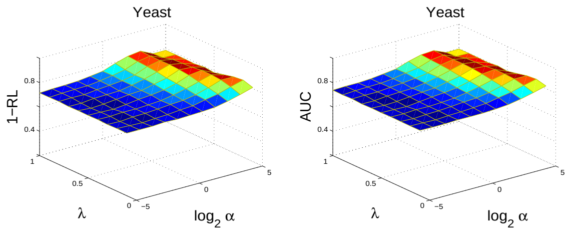

Figure 3: Performance of McWL on Yeast under different combinations of α and λ.

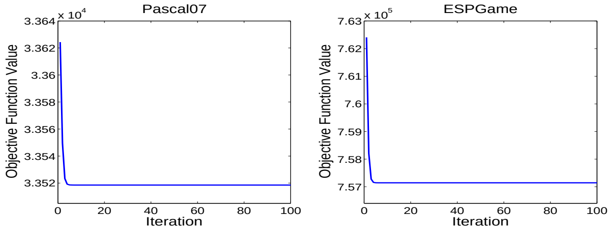

Figure 5: Convergence trend analysis.

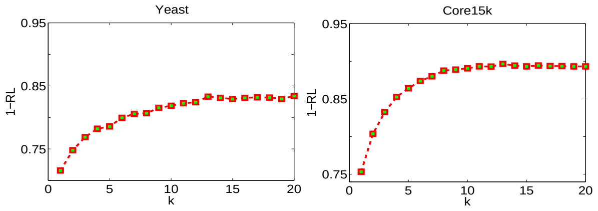

Figure 4: Sensitivity analysis of parameter k.

confirm the robustness of McWL with respect to k.

4.5 Runtime and Convergence Analysis

The runtime of the comparing methods on the five datasets is given in Table 3. The experiments are conducted on CentOS 7 with Inter(R) Xeon(R) E5-2678 and 256GB RAM, and the methods are implemented in MATLAB 2013a. We can see that the total runtime of McWL ranks 2nd among all the methods. MLR-GL is a supervised learning method; it relaxes the convex-concave optimization problem into a Second Order Cone Programming (SOCP) [4] problem, and overall is the fastest. LabelMe and MLAN do not utilize an MC-based classifier for prediction, and thus they are faster than lrMVMC and MVMC-LS. An interesting observation is that although McWL is also an MC-based classifier, its runtime is superior to that of MLAN and LabelMe in most cases. This is because, instead of estimating a matrix of size $n \times (d+q)$ as in lrMVMC and MVMC-LS, McWL estimates a matrix of much smaller size $n \times q$.

Figure 5 reports the convergence curve of McWL on Pascal07 and ESPGame. As we can see, for both datasets, the algorithm converges in less than 10 iterations. We also observe similar convergence trend on the other datasets.

5 Conclusion

Weak-label learning has attracted great attention in many data mining and pattern recognition tasks. Current efforts mainly focus on performing weak-label learning on a single view, despite the abundance of multiple view features in many real-world data mining tasks. In this paper, we proposed a multi-view weak-label learning algorithm based on matrix completion (McWL). McWL differs from previous MC-based multi-view learning methods in that it integrates the fusion of multiple views and the MC-based classifier into a unified objective function, and accounts for weak-label scenarios. Furthermore, McWL is able to assign smaller weights to views of low quality. Our experimental results show that McWL outperforms other competitive methods. Improving the efficiency of our method and exploiting label correlations of labels in matrix completion remain an interesting future pursue.

References

[1] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[2] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, p. 52, 2015.

[3] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou, "Multi-label learning with weak label," in *24th AAAI Conference on Artificial Intelligence*, 2010, pp. 1862–1868.

[4] S. S. Bucak, R. Jin, and A. K. Jain, "Multi-label learning with incomplete class assignments," in *24th IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2801–2808.

[5] X. Kong, Z. Wu, L.-J. Li, R. Zhang, H. Wu, and W. Fan, "Large-scale multi-label learning with incomplete label assignments." in *SIAM International Conference on Data Mining*, 2014, pp. 920–928.

[6] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," in *31st International Conference on Machine Learning*, 2014, pp. 593–601.

[7] B. Wu, S. Lyu, and B. Ghanem, "Constrained submodular minimization for missing labels and class imbalance in multi-label learning." in *30th AAAI Conference on Artificial Intelligence*, 2016, pp. 2229–2236.

[8] W. Zhang, K. Zhang, P. Gu, and X. Xue, "Multi-view embedding learning for incompletely labeled data." in *23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 1910–1916.

[9] M. Liu, Y. Luo, D. Tao, C. Xu, and Y. Wen, "Low-rank multi-view learning in matrix completion for multi-label image classification," in *29th AAAI Conference on Artificial Intelligence*, 2015, pp. 2778–2784.

[10] R. S. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for multi-label image classification." in *Advances in Neural Information Processing Systems*, vol. 201, no. 1, 2011, p. 2.

[11] A. Goldberg, B. Recht, J. Xu, R. Nowak, and X. Zhu, "Transduction with matrix completion: Three birds with one stone," in *Advances in Neural Information Processing Systems*, 2010, pp. 757–765.

[12] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 121–135, 2015.

[13] Y. Luo, T. Liu, D. Tao, and C. Xu, "Multiview matrix completion for multilabel image classification," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2355–2368, 2015.

[14] S. Mostafavi and Q. Morris, "Fast integration of heterogeneous data sources for predicting gene function with limited annotation," *Bioinformatics*, vol. 26, no. 14, pp. 1759–1765, 2010.

[15] G.-X. Yu, H. Rangwala, C. Domeniconi, G. Zhang, and Z. Zhang, "Protein function prediction by integrating multiple kernels." in *23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 1869–1875.

[16] F. Zhao and Y. Guo, "Semi-supervised multi-label learning with incomplete labels," in *24th International Conference on Artificial Intelligence*, 2015, pp. 4062–4068.

[17] G. Yu, C. Domeniconi, H. Rangwala, and G. Zhang, "Protein function prediction using dependence maximization," in *23rd Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 574–589.

[18] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, pp. 43–54, 2017.

[19] S. Li, M. Shao, and Y. Fu, "Multi-view low-rank analysis for outlier detection," in *15th SIAM International Conference on Data Mining*, 2015, pp. 748–756.

[20] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *11th Annual Conference on Computational Learning Theory*, 1998, pp. 92–100.

[21] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 2211–2268, 2011.

[22] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *31st AAAI Conference on Artificial Intelligence*, 2017, pp. 2408–2414.

[23] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, 2012.

[24] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Advances in Neural Information Processing Systems*, 2013, pp. 2301–2309.

[25] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel target alignment," *Innovations in Machine Learning*, pp. 205–256, 2006.

[26] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, 2003, pp. 321–328.

[27] M.-L. Zhang, Y.-K. Li, and X.-Y. Liu, "Towards class-imbalance aware multi-label learning," in *24th International Joint Conference on Artificial Intelligence*, 2015, pp. 4041–4047.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[29] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[30] M. Guillaumin, J. Verbeek, and C. Schmid, "Multi-modal semi-supervised learning for image classification," in *23rd IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 902–909.

[31] E. L. Gibaja, J. M. Moyano, and S. Ventura, "An ensemble-based approach for multi-view multi-label classification," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 251–259, 2016.