Multi-View Multi-Label Learning with View-Specific Information Extraction

Xuan Wu1,2, Qing-Guo Chen2, Yao Hu2, Dengbao Wang4, Xiaodong Chang2, Xiaobo Wang2 and Min-Ling Zhang3,4

1School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
2Alibaba Group, Hangzhou, China
3Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
4College of Computer and Information Science, Southwest University, Chongqing 400715, China
wuxuan@seu.edu.cn, {qingguo.cqg, yaoohu}@alibaba-inc.com, dbwang@email.swu.edu.cn,
{tommy.cxd, yongshu.wxb}@alibaba-inc.com, zhangml@seu.edu.cn

Abstract

Multi-view multi-label learning serves an important framework to learn from objects with diverse representations and rich semantics. Existing multi-view multi-label learning techniques focus on exploiting shared subspace for fusing multi-view representations, where helpful view-specific information for discriminative modeling is usually ignored. In this paper, a novel multi-view multi-label learning approach named SIMM is proposed which leverages shared subspace exploitation and view-specific information extraction. For shared subspace exploitation, SIMM jointly minimizes confusion adversarial loss and multi-label loss to utilize shared information from all views. For view-specific information extraction, SIMM enforces an orthogonal constraint w.r.t. the shared subspace to utilize view-specific discriminative information. Extensive experiments on real-world data sets clearly show the favorable performance of SIMM against other state-of-the-art multi-view multi-label learning approaches.

1 Introduction

In many real-world applications, objects are with diverse representations and rich semantics simultaneously. For example, in image analysis, a natural scene image can often be represented by its visual features such HSV color histogram, globe feature (Gist) and scale invariant feature transform (SIFT), while it may be annotated with labels \{sky, water, cloud\}. In video annotation, the representations of a film e.g. Avengers: Infinity War are usually from multiple channels of information such as text description, audio, cover picture, frame extraction, meanwhile it can be annotated with action movie (type), America (country) and Anthony Russo and Joseph Russo (directors). The main challenge of such problems is how to integrate the multiple types of heterogeneities in an efficient and general way. Multi-view multi-label learning serves an important framework to solve the above problem.

Formally speaking, let \(X = \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \cdots \times \mathbb{R}^{d_k}\) be the feature space of \(k\) views representations, where \(d_v (1 \leq v \leq k)\) is the feature dimension of \(v\)-th view. Let \(Y = \{y_j\}_{j=1}^q\) be the label space with \(q\) class labels. Given the training set \(D = \{(x_i, Y_i) | 1 \leq i \leq m\}\), where \(x_i \in \mathcal{X}\) is the feature vector \([x_i^1, x_i^2, \cdots, x_i^v]\) and \(Y_i \subseteq Y\) is the set of relevant labels associated with \(x_i\). The task of multi-view multi-label learning is to learn a predictive model \(h : \mathcal{X} \rightarrow \mathbb{R}^q\) from \(D\) which can assign a set of proper labels for the unseen instance.

There have been some attempts for multi-view multi-label problem. In [Xing et al., 2018], the authors introduce a predictive reliability measure to select samples which are used to share label information with other views in the co-training manner. [Zhu et al., 2016; Liu et al., 2015; Luo et al., 2013] focus on the issue of multi-view image classification and map each view into a shared space to eliminate noise and redundancy while maintaining sparse, low-rank and manifold structure of image data respectively. [Zhu et al., 2018; Zhang et al., 2018] aims to further remain consensus on multi-view latent spaces by using Hilbert–Schmidt independence criterion during the mapping procedure.

Nonetheless, the above methods suffer from the following two different issues. Firstly, when trying to find shared information among all views to eliminate noise and redundancy, a common practice is to map each view into a shared subspace. However, this procedure is always done in an independent way, i.e. there is no communication among various views. It is hard to ensure that common semantic information is fully tapped. Secondly, the underlying fact that each view contains their own specific contribution to the multi-label prediction, is ignored. For example, a picture of pink rose may be tagged with two labels \{pink, flower\}, while its representation can be described by HSV and Gist. We can easily find the correspondence between them, i.e. pink is described by HSV view feature, while flower is described by Gist view feature. However, existing methods only try to find the shared information between HSV (color) and Gist (texture), while it is more reasonable to consider extracting their own specific information.
2 The Proposed Method

The goal of multi-view multi-label learning is to fully integrate various representations of a single object and assign the proper rich semantics to it. As mentioned above, information from different views usually contains the shared and specific parts. Thus, two key steps of SIMM are shared subspace exploitation and view-specific information extraction.

SIMM implements those two key points by using a neural network framework. Considering the network, the overall loss function naturally turns out to be the following form:

\[ L = L_{ml} + \alpha L_{shared} + \beta L_{specific} \]  

(1)

where \( \alpha \) and \( \beta \) are trade-off factors that control the interaction of the loss terms. Final multi-label classification loss \( L_{ml} \) controls the model to predict the output labels. Let \( y^j_i \) denote the ground-truth label of \( x_i \) on the \( j \)-th label. \( y^j_i = 1 \) if \( j \)-th label is the relevance label, \( y^j_i = 0 \) otherwise. Let \( \hat{y}^j_i \) denote the prediction output. The \( L_{ml} \) can be calculated as follows:

\[ L_{ml} = -\sum_{i=1}^{m} \sum_{j=1}^{q} y^j_i \log \hat{y}^j_i + (1-y^j_i) \log (1-\hat{y}^j_i) \]  

(2)

2.1 Shared Subspace Exploitation

In traditional multi-view learning, the shared subspace exploitation is often implemented in an independent way. Thus, the communication among all views is neglected. Inspired by [Liu et al., 2017], SIMM implements the mapping in a confusion way by minimizing the adversarial loss \( L_{adv} \). In particular, SIMM aims to confuse the discriminator that the coming shared subspace representation belongs to which view.

Let \( x^v \) be the \( l \) dimension shared subspace representation of original \( v \)-th view feature \( x^v \). It is extracted by a shared subspace extraction layer \( H \), i.e. \( x^v = H(M^v(x^v)) \), where \( M^v(\cdot) \) is used to map each \( v \) dimension original feature vector into the same \( l \) dimension. Let \( z_i \) be the \( k \) dimension view label vector of \( c^v_i \), where only \( z^v \) equals to 1 while others equal to 0. This indicates that \( c^v_i \) comes from the \( v \)-th view. A training set \( D_{adv} = \{(c^v_i, z_i) | 1 \leq v \leq k, 1 \leq i \leq m \} \) can be constructed for discriminator \( D \). Let \( \hat{z} \) be the prediction output by \( D \), i.e. \( \hat{z}_i = D(c^v_i) \), \( L_{adv} \) can be formed as:

\[ L_{adv} = F(-\sum_{i=1}^{m} \sum_{v=1}^{k} z^v_i \log \hat{z}^v_i) \]  

(3)

where \( F(\cdot) \) should be a monotonically decreasing function\(^1\). In this way, we confuse the discriminator so as not to recognize the true view of the coming shared subspace representation, which indicates that there is no special information in it. In other words, the incorporated representation \( c^v \) only contains the shared information of \( x^v \).

However, it might be problematic if only \( D_{adv} \) is used as noise can also confuse the discriminator easily. Therefore,

\(^1\)In this paper, \( F(x) = e^{-x} \)
we use $\mathcal{L}_{sml}$ (Shared subspace Multi-Label Loss) to guarantee that each label contains certain semantics. In particular, a training set $D_{sml} = \{(e_i^v, y_i^v) | 1 \leq v \leq k, 1 \leq i \leq m\}$ can be constructed for shared subspace representation prediction layer $G_s$. Let $\hat{y}_i^v$ be the prediction output by $G_s$, i.e. $\hat{y}_i^v = G_s(e_i^v)$. $\mathcal{L}_{sml}$ can be formulated as:

$$\mathcal{L}_{sml} = -\sum_{i=1}^{m} \sum_{v=1}^{k} \sum_{j=1}^{q} y_{ij}^v \log \hat{y}_{ij}^{t_{ij};v} + (1 - y_{ij}^v) \log (1 - \hat{y}_{ij}^{t_{ij};v})$$

(4)

The shared subspace exploitation by SIMM is no longer conducted in an independent way as $H$ and $D$ meet the information from all views with $G_s$ maintaining semantics. Combining the above losses together, $\mathcal{L}_{shared}$ can be written as:

$$\mathcal{L}_{shared} = \mathcal{L}_{adv} + \mathcal{L}_{sml}$$

(5)

### 2.2 View-Specific Information Extraction

In SIMM, helpful view-specific information is further considered. Actually, it is difficult to define what is the specific information of a particular view. However, we can extract it from original information by eliminating shared information. This can be implemented by enforcing an orthogonal constraint. Let $s^v$ be the $l$ dimension feature vector extracted by specific information extraction layer $E^v$, i.e. $s^v = E^v(x^v)$. Let $c$ be the $l$ dimension feature vector containing shared information among all views. $c$ is performed by element-wise addition on each individual $e^v$, i.e. $c = \sum_{v=1}^{k} e^v$. The $\mathcal{L}_{specific}$ encourages orthogonality between $s^v$ and $c$:

$$\mathcal{L}_{specific} = ||s^v c||_2^2$$

(6)

where $||\cdot||_2^2$ is the squared $L_2$-norm. $\mathcal{L}_{specific}$ encourages $s^v$ extracted from the original $v$-th view feature vector $x^v$ to be as different from $c$ as possible.

### 2.3 The Whole Framework

The whole framework of SIMM is shown in Figure 1. The goal of training is to minimize the whole loss function (1) with respect to parameters $\Theta = \{\theta_E, \theta_M, \theta_H, \theta_D, \theta_{G_s}, \theta_{G_c}\}$. Each unit is a neural network framework and we use popular optimization algorithm Adam [Kingma and Ba, 2015] to do the back propagation to update $\Theta$ simultaneously in the training phase. In the testing phase, given an unseen instance $x^*$, its prediction output $\hat{y}^*$ can be obtained as:

$$\hat{y}^* = G_s([E^1(x^*), \cdots, E^v(x^*), \cdots, E^k(x^*), c])$$

(7)

where $c = \sum_{v=1}^{k} H(M^v(x^*))$.

## 3 Related Work

In the past years, multi-label learning has been widely used in many kinds of applications, such as text categorization [Ueda and Saito, 2003], bioinformatics [Zhang and Zhou, 2006], web mining [Tang et al., 2009], etc. Following [Zhang and Zhou, 2014], existing multi-label methods can be categorized into two groups, i.e. problem transformation methods and algorithm adaptation methods. Problem transformation methods aim to tackle multi-label learning problem by transforming it into other well-established learning scenarios, such as Binary Relevance [Boutell et al., 2004] transforms the task of multi-label learning into the task of binary classification while Calibrated Label Ranking [F{"u}rnkranz et al., 2008] transforms the task of multi-label learning into the task of label ranking. Algorithm adaptation methods tackle multi-label learning problem by adapting popular learning techniques to deal with multi-label data directly, such as ML-kNN [Zhang and Zhou, 2007] adapts lazy learning technique while ML-DT [Clare and King, 2001] adapts decision tree technique.

In multi-view learning, the information in some views can help handle the weakness of other views. Furthermore, multi-view learning can be embedded into multi-label learning tasks naturally to further improve the classification performance. The most important part in multi-view multi-label learning is how to utilize and communicate heterogeneous information among all views under the multi-label framework. [Xing et al., 2018] focuses on selecting reliable samples from one view and passing them to other views. However, the communication among views is only on the label level. [Zhu et al., 2016] introduces a block-row regularizer to reduce the noise and redundancy. The method in [Liu et al., 2015] seeks a low-dimensional common representation of all views and constructs the classifier based on matrix completion. [Luo et al., 2013] integrates multiple features under multi-view vector-valued manifold regularization. [Zhu et al., 2018; Zhang et al., 2018] try to remain latent semantic when studying the low-dimensional common representation.
Comparing Methods | Emotions | Year | Corel5k | EspGame | Pascal | Mirflickr | Youku15w | Youku3w | Yeast | SIMM | Benchmark | COMMON | ML-KNN (C) | ML-KNN (B) | LSA/ML | F2L1F | F2L21F | F2L21T | IMM | F2L1F | F2L21F | Benchmark | COMMON | ML-KNN (C) | ML-KNN (B) | LSA/ML | F2L1F | F2L21F | F2L21T | IMM | F2L1F | F2L21F
**Table 2:** Predictive performance of each method. (mean ± std. deviation) on the multi-view multi-label data sets.

However, for most existing methods the mapping from original views to the shared subspace is conducted in an independent way where there is only limited communication among each view. Furthermore, the specific characteristics of individual view is ignored. SIMM serves as the first attempt towards jointly enhancing communication during shared subspace exploitation and remaining view-specific information.

### 4 Experiment

#### 4.1 Experimental Setting

**Data Sets**

A total of eight multi-view multi-label data sets are employed for performance evaluation including six benchmark data sets^{2} and two real-world video annotation data sets.

**Emotions** has two feature views: 8 rhythmic attributes and 64 timbre attributes. Yeast also has two views including the concatenation of the genetic expression (79 attributes)


#### 4.2 Evaluation Metric

**Evaluation metric** $F_P$ critical value

<table>
<thead>
<tr>
<th>Hamming Loss</th>
<th>$F_P$</th>
<th>critical value</th>
<th>Average Precision</th>
<th>One error</th>
<th>Ranking Loss</th>
<th>Coverage</th>
<th>Micro-F1</th>
</tr>
</thead>
</table>

**Table 3:** Friedman statistics $F_P$ in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms $k = 7$, # data sets $N = 8$).
Youku data set consists 148,089 videos in 114 classes collected from the Video Application YOUKU. Each object has four views: textual description in the video title, the audio information, cover picture and video frame extraction. The dimensions are 100, 128, 2048, 2048 respectively. For textual view, we extract the weights of each word in the title. Using the pre-trained word embedding, title embedding vector can be obtained by weighting the word embedding. For audio view, we follow [Gemmeke et al., 2017] and reduce dimension by using Principle Component Analysis (PCA). For image view, we use Inception-v3 proposed in [Szegedy et al., 2016]. For video view, we extract video frames every two seconds and use the same procedure as image view to obtain frame-level features. Then, these frame-level features are aggregated into video-level by mean-pooling and L2-normalization. Label information is obtained by manual labeling, from which we choose the most frequent 114 ones.

Given the data set \( D \), we use \(|D|, V(D), V Dim(D), CL(D)\) to represent its number of samples, number of views, dimension of each view, class labels. Moreover, several other properties of multi-label data sets are denoted as [Zhang and Zhou, 2014]:

**Label cardinality:** \( LCard(D) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{q} I(y_{ij} = +1) \) which counts the average number of relevant labels per example;

**Label density:** \( LDen(D) = \frac{LCard(D)}{CL(D)} \) which normalizes label cardinality by the number of class labels;

**Distinct label sets:** \( DL(D) = |\{ y \exists x : (x, y) \in D \}| \) which counts the number of distinct label vectors (relevant label set) existing in \( D \);

**Proportion of distinct label sets:** \( PDL(D) = \frac{DL(D)}{|D|} \) which normalizes \( DL(D) \) by the number of examples.

Table 1 summarizes detailed characteristics of the experimental data sets.

**Comparing Algorithms**

The performance of SIMM is compared against six algorithms, including two baselines related to SIMM, a multi-label algorithm ML-kNN with two types of feature inputs and two multi-view multi-label algorithms.

**Benchmark:** A multi-layer perceptron where the number of parameters is no less than SIMM. The input is the concatenation of all view features.

**COMMON:** The prediction results are only obtained by shared information, i.e. \( \hat{y} = \frac{1}{k} \sum_{k=1}^{k} \hat{y}^{(k)} \).

**ML-kNN(C) & ML-kNN(B)** [Zhang and Zhou, 2007]: A lazy multi-label algorithm with two types feature inputs. (C) stands for the concatenation of all views. (B) stands for the best performance obtained from the best single view. [configuration: \( k = 10 \)]

**F2L21F** [Zhu et al., 2016]: A multi-view multi-label algorithm using block-row regularizer to reduce the overlap, noise, and redundancy in multiple views. [configuration: \( \lambda_1 = 10, \lambda_2 = 10 \)]

**LSAMML** [Zhang et al., 2018]: A multi-view multi-label algorithm using Hilbert-Schmidt Independence Criterion to enhance the dependence between different pairs of views. [configuration: \( r \in \{2,3,4,5\}, \beta \) and \( \gamma \in \{0.01,0.1,1,10,100\} \)]

For SIMM, in order to make the model more elegant and lightweight, we set each module to be only a fully connected layer without hidden layer. \( l \) is fixed to 64. In light of comparison to COMMON, \( \alpha \) is fixed to 1. \( \beta \) is selected from \{0.1, 0.01, 0.001, 0.0001\}.

**Experimental Protocol**

In this paper, six widely-used multi-label metrics are employed for performance evaluation, including Hamming Loss, Average Precision, One Error, Ranking Loss, Coverage and Micro-F1, which consider the performance of multi-label predictor from various aspects. Concrete metric definitions can be found in [Zhang and Zhou, 2014], and Coverage metric is normalized by the number of class labels (i.e. \( q \)). For Average Precision and Micro-F1, the larger the values the better the performance. For the other four metrics, the smaller the values the better the performance.

For each data set, ten-fold cross-validation is performed where the mean metrics results and standard deviations are recorded for all comparing approaches.
4.2 Experimental Results

Detailed experimental results are reported in Table 2, where the best performance on each data set is shown in boldface. To analyze the relative performance among the comparing approaches in a systematic manner, Friedman test is employed as the statistical test for performance comparison.

Table 3 reports Friedman statistics $F_F$ and the corresponding critical values in terms of each evaluation metric. It is obvious that the null hypothesis of equal performance is rejected at 0.05 significance level. Accordingly, post-hoc Boferoni-Dunn test is performed to compare the relative performance among the comparing approaches. The CD diagrams are presented in Figure 2, where the average rank of each approach is marked along the axis (the smaller the better).

Based on the reported experimental results, the following observations can be made: (a) Among the 48 configurations (8 data sets × 6 evaluation metrics), SIMM ranks 1st and 2nd in 87.5% and 10.4% cases respectively. (b) It is remarkable that SIMM achieves best performance in all cases on Hamming Loss, Ranking Loss and Coverage. (c) It is noticeable that SIMM outperforms all the comparing algorithms on Emotions, Pascal, Mirflickr and Youku15w on all metrics while outperforms all the comparing algorithms on Espgame and Youku3w on all metrics except on Micro-F1.

4.3 Further Analysis

In order to examine the effectiveness of $L_{\text{shared}}$ and $L_{\text{specific}}$, we remain the basic structure of our model and remove these two constraints by setting $\alpha$ and $\beta$ to zero. Figure 3 shows the comparison results on Pascal and Youku15w data sets. The curves show the change of the metric results with the number of epoch. It is clearly shown that the model performed worse without $L_{\text{shared}}$ and $L_{\text{specific}}$ (SIMM-NL). In other words, it is not enough to fuse the information from different views directly. SIMM is a good help to separate the shared and specific information.

5 Conclusion

In this paper, a novel neural network based approach is proposed to solve the multi-view multi-label problem. Specifically, we enhance the communication among views while remain individual specific characteristics through two steps, i.e. shared subspace exploitation and view-specific information extraction. Effectiveness of the proposed approach is validated via comprehensive experiments on real-world multi-view multi-label data sets. In the future, it is interesting to extend the structure of each neural network module. Meanwhile, note that $s^v$ and $c$ are fixed with the same dimension due to the orthogonality constraint and therefore a more general way of specific information extraction can be explored.

Acknowledgments

This work was completed while Xuan Wu was conducting her internship at Alibaba Group. The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Key R&D Program of China (2018YFB1004300), the National Science Foundation of China (61573104), and partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization.
References


