Labeling Complicated Objects: Multi-View Multi-Instance Multi-Label Learning*

Cam-Tu Nguyen1,3, Xiaoliang Wang1, Jing Liu2 and Zhi-Hua Zhou1
1 National Key Laboratory for Novel Software Technology, Nanjing University, 210023, China
2 National Key Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Sciences, China
3 VNU-University of Engineering and Technology, Hanoi, Vietnam
{nguyenct, zhouzh}@lamda.nju.edu.cn, waxili@nju.edu.cn, jliu@nlpr.ia.ac.cn

Abstract
Multi-Instance Multi-Label (MIML) is a learning framework where an example is associated with multiple labels and represented by a set of feature vectors (multiple instances). In the formalization of MIML learning, instances come from a single source (single view). To leverage multiple information sources (multi-view), we develop a multi-view MIML framework based on hierarchical Bayesian Network, and derive an effective learning algorithm based on variational inference. The model can naturally deal with examples in which some views could be absent (partial examples). On multi-view datasets, it is shown that our method is better than other multi-view and single-view approaches particularly in the presence of partial examples. On single-view benchmarks, extensive evaluation shows that our method is highly competitive or better than other MIML approaches on labeling examples and instances. Moreover, our method can effectively handle datasets with a large number of labels.

Introduction
The last decade has witnessed the development of machine learning to address not only bigger but also more complicated data. Multi-instance multi-label learning (MIML) (Zhou and Zhang 2007; Zhou et al. 2012) provides a natural formulation for complicated objects, where each example is represented by a bag of instances, and associated with multiple labels simultaneously. MIML has a nice property that it allows us to discover labels for examples and instances while, during training, we only need labels for examples, not labels for individual instances. This learning setting is prevalent in practice; for example, a document is often represented by a bag of words, an image can be considered as a bag of regions, and a gene sequence can be treated as a bag of sub-sequences. Annotating these types of objects with multiple labels gives rise to many MIML problems such as image classification and annotation (Zha et al. 2008; Nguyen, Zhan, and Zhou 2013), gene pattern annotation (Li et al. 2012b), relation extraction (Surdeanu et al. 2012), etc.

The original MIML setting only deals with situations where data comes from a single feature set (single-view). For many complicated data, however, it is quite difficult for a single feature set to capture the information required to label a large number of categories. It is thus natural to consider the possibility of leveraging the usefulness of multiple feature sets (multi-view). For example, one can make use of visual and audio signals to label videos; or exploit captions and visual contents to annotate images.

Given the aforementioned scenarios, we formalize the problem of Multi-view Multi-Instance Multi-Label learning as follows. Let \( \mathcal{Y} = \{y_l | l = \ldots , L \} \) denote a set of \( L \) labels, and \( \mathcal{D} = \{(X_n, Y_n) | n = 1 \ldots N \} \) denote a training dataset, where the \( n \)-th example \( X_n \) is represented by a bag of instances from \( V \) views, and \( Y_n = \{y_{nl} | l = 1 \ldots L \} \subset \mathcal{Y} \) is the set of \( L_n \) (bag) labels of the \( n \)-th example. Here, we have \( X_n = \{x_{nv} | v = 1 \ldots V, m = 1 \ldots M_{nv} \} \) where \( M_{nv} \) indicates the number of instances in the \( v \)-th view of the \( n \)-th example, and each instance \( x_{nv} \in R^{D_v} \) is represented by a \( D_v \) dimensional feature vector of view \( v \). The goal of multi-view MIML learning is to predict bag labels \( Y_{n'} \) for an unseen example \( X_{n'} \), along with labels for its individual instances \( x_{n'v} \) in \( V \) views (see Fig. 1). Additionally, it is desirable for a multi-view approach to work on partial examples, i.e., examples with no instance in some views. This is because multi-view datasets are often corrupted and may have missing values in realistic scenarios. For instance, on sensory datasets, some input signals such as visual, auditory might be missing due to some corruption from environment.

In general, one can simply learn an MIML model in each view separately, and combine the outputs of the single-view

---

*This research was supported by the 973 Program (2010CB327903) and NSFC (61223003, 61370028).

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
MIML models, where the learning of single-view MIML models can rely on any previous approach (Zhou et al. 2012). However, by this way the learning in each single-view can not make full use of available information. Moreover, since MIML explores structures among both instances and labels, the effect of utilizing full information could be more significant than that in the traditional single-instance single-label scenario. As a result, it is beneficial to consider the multi-view MIML problem as a whole.

In this paper, we propose a method named Multi-Instance Multi-Label Mixture (MIMLmix) based on hierarchical Bayesian network, where labels are assumed to be sampled from "topics" that capture label relationships; and instances (from multi-views) are sampled from a mixture model where mixture components are representations of labels in multi-views. In continuous feature spaces (continuous views), labels are represented by Gaussian distributions, whereas in discrete feature spaces (discrete views), labels are represented by Multinomial distributions. The usage of Bayesian approach helps handle missing information such as the missing of instance labels, or partial examples.

The remainder of this paper is organized into 4 sections. We first revisit some related works, then present our method (MIMLmix), followed by experiments and conclusions.

**Related Work**

Zhou et al. (2007; 2012) formulated MIML (Multi-instance Multi-label) framework, proposed several algorithms and applied to image and text applications. Later on, many MIML algorithms have been proposed and many applications have been reported; to name a few, MIML algorithms based on Dirichlet-Bernoulli alignment (Yang, Zha, and Hu 2009), based on Conditional Random Fields (Zha et al. 2008), based on single-instance degeneration (Nguyen 2010), based on metric learning (Jin, Wang, and Zhou 2009), etc. MIML techniques have been found well useful in applications such as image retrieval and annotation (Nguyen et al. 2013), video annotation (Xu, Xue, and Zhou 2011), gene pattern annotation (Li et al. 2012b), relation extraction in natural language processing (Surdeanu et al. 2012), etc. A number of MIML methods that discover the relationships between bag labels and instances have been proposed in (Li et al. 2012a; Briggs, Xiaoli, and Raich 2012).

Multi-view learning deals with data in multiple views, i.e., multiple feature sets. The goal is to improve performance or reduce the sample complexity. Multi-view learning has been well studied in learning with unlabeled data. Some studies used multi-views in conjunction with semi-supervised learning (Blum and Mitchell 1998; Wang and Zhou 2010b; Zhou, Zhan, and Yang 2007), or with active-learning (Wang and Zhou 2010a). Others tried to establish a latent subspace by assuming that instances (in different views) belong to the same example are nearby after mapping into the latent subspace (White et al. 2012; Wang, Nie, and Huang 2013). To combine information from multi-views for traditional supervised learning, one can use fusion techniques at feature level, or classifier level (Atrey et al. 2010).

Almost all previous MIML studies focused on single-view setting and almost all previous multi-view learning studies focused on single-instance and/or single-label learning. To the best of our knowledge, the only exception is (Nguyen, Zhan, and Zhou 2013), where the M3LDA approach was proposed. Our approach, however, is more general and effective than M3LDA, which will be discussed with details in the next section. It is also important to highlight some related Bayesian Network structures such as Dependence-LDA (Rubin et al. 2012), GM-LDA and Corr-LDA (Blei and Jordan 2003). These methods were not designed for MIML, since Dependence-LDA works with textual documents; and GM-LDA, Corr-LDA are for unsupervised learning, i.e. labels have not been exploited.

**The MIMLmix Model**

Inspired by Bayesian Network approaches (Nguyen et al. 2010; Rubin et al. 2012; Nguyen, Zhan, and Zhou 2013), we propose the MIMLmix model (Multi-instance Multi-label Mixture model) for multi-view MIML (Fig. 2), which consists of two parts: (1) the topic-label part is a LDA topic model of K topics (Blei, Ng, and Jordan 2003), where topics capture label correlations; and (2) the label-instance part where instances are generated from a mixture of Gaussian/Multinomial distributions. The generative process is shown in Alg. 1. and in Fig. 2.

In the label-instance part, for an example $X_n$, we set the prior for the label distribution $\theta_n \in \mathbb{R}_+^L$ as $\alpha_n = \eta' \odot \gamma_n + \xi$, where $\odot$ is an element-wise product, and $\gamma_n \in \mathbb{R}_+^L$. During training, $\xi = 0$, $y_{nl}$ equals to 1 if the l-th label is in $Y_n$ and zero otherwise, thus $\alpha_{nl}$ are zeros for labels not in $Y_n$. During testing, $\xi$ is set to a nonzero constant and all elements in $\gamma_n$ are initialized to 1 to trigger all labels for inference, the...
value of η controls how much the topic distribution affects the label distribution θ. The latent variables z represent the hidden assignments of bag labels to instances. If 𝕀 is discrete, we formalize \( p(x_{nvm}|z_{nvm} = \gamma_y, \beta_v) \) using a Multinomial distribution with parameter \( \beta_v \in R^D_v \). Here, we drop indexes \( n, m, v \) for simplicity:

\[
p(x|\beta_v) = \left( \frac{|x|_1}{\chi_1 \cdots \chi_D} \right) \prod_{i=1}^{D}(\beta_{yi})^{x_i}
\]

where \( |x|_1 = \sum_i x_i \). If the feature space \( \mathcal{X} \) is continuous, we formalize \( p(x_{nvm}|\beta_v) \) as a Gaussian distribution where \( \beta_v = \{ \mu_v, \Sigma_v \} \), \( \mu_v \in R^D_v \), \( \Sigma_v \in R^{D_v \times D_v} \):

\[
p(x|\beta_v) = \exp \left[ -\frac{1}{2} (x - \mu_v)^\top \Sigma_v^{-1} (x - \mu_v) \right] \frac{1}{(2\pi)^{D_v/2} |\Sigma_v|^{1/2}}
\]

where \( n, m, v \) have also been removed for simplicity.

As MIMLmix allows instances from discrete and continuous views, it is more general than M3LDA (Nguyen, Zhan, and Zhou 2013), which only works with discrete views. In the sequel, we will derive a training method based on variational inference for MIMLmix that is more effective than Gibbs sampling in M3LDA. Moreover, instead of a “hard” assignment of labels to instances via \( z \) (\( z \) takes value of one out of the bag label set), variational inference introduces a “soft” assignment via \( \phi \), \( \sum_y \phi_y = 1 \) - the variational variable for \( z \), which allows one instance to be associated with multiple related labels.

### Training with MIMLmix

As \( Y_n \) are observed during training, the two parts \( p(\theta', \gamma, Y|\beta', \alpha') \) and \( p(\theta, z, X|\beta, Y, \Lambda) \) can be learned independently. In the following, we show a variational inference for the label-instance part. Variational inference places a simpler family of distributions over the latent variables:

\[
q(z, \theta) = \prod_n q(\theta_n|\gamma_n) \prod_{v,m} q(z_{nvm}|\phi_{nvm})
\]

where \( \theta_n \sim \text{Dir}(\gamma_n) \) and \( z_{nvm} \sim \text{Mult}(\phi_{nvm}) \) and \( \gamma_n \in R^L \) and \( \phi_{nvm} \in R^{L} \left( \sum_y \phi_{nvm,y} = 1 \right) \). We then obtain the evidence lower bound (ELBO) \( \mathcal{L} \):

\[
\mathcal{L} = E_q[\log p(\theta, z, X|\beta, Y, \Lambda)] - E_q[\log q(z, \theta)]
\]

The training is performed by maximizing the ELBO using the EM algorithm similar to (Blei, Ng, and Jordan 2003). Here, E-step tries to assign labels in bag labels to instances by alternating the following updates:

\[
\phi_{nvm} \propto \exp \left[ E_q[\log \theta_{nvy}] + \log p(x_{nvm}|z_{nvm} = y, \beta_v) \right]
\]

\[
\gamma_{ny} = \alpha_{ny} + \sum_{v,m} \phi_{nvm}
\]

where \( E_q[\log \theta_{nvy}] = \Psi(\gamma_{ny}) - \Psi(\gamma_{ny}^\top) \) \( \Psi \) denotes the digamma function. Note that we do not consider all the labels in \( Y \) for each bag \( n \) but only the labels belonging to \( Y_n \), thus performing E-step here is efficient.

Given the estimated \( \phi_n \) and \( \gamma_n \) for all \( n \), M-step updates the global variables that maximize the ELBO as follows:

### Algorithm 2 Training with MIMLmix

1. \( \triangleright \) Topic-label part
2. Train a LDA on \( Y_{1:N} \) (Blei, Ng, and Jordan 2003).
3. \( \triangleright \) Label-instance part
4. Initialize \( \beta_v \) for all view \( v \) and \( y \in Y \).
5. while relative improvement in \( \mathcal{L} < 10^{-6} \) do
6. for \( n = 1 \) to \( N \) do \{E-step\}
7. Initialize \( \gamma_n \).
8. repeat
9. for each view \( v \), ins. \( m \) and label \( y \in Y_n \) do
10. Update \( \phi_{nvm,y} \) according to Eq. [5].
11. Update \( \gamma_{ny} \) according to Eq. [6] for \( y \in Y_n \).
12. until \( 1/L \sum_y |\text{change in } \gamma_{ny}| < 10^{-6} \).
13. for each view \( v \), and \( y \in Y \) do \{M-step\}
14. Update \( \beta_v \) according to either Eq. [7] or Eqs. [8-9] depending on the view \( v \).

### Algorithm 3 Testing with MIMLmix

1. For a test bag \( X_n' \), initialize \( y_{n'l} = 1, \forall l \in Y \) and \( \alpha_n' = \eta \odot y_n' + \xi \).
2. repeat
3. Perform inference about \( \phi_{n'v} \), \( \gamma_{n'} \) like E-step of Algorithm 2 given the current value of \( \alpha_n' \).
4. Sample \( Y_{n'} \) according to a Multinomial dist. parameterized by (normalized) \( \sum_{v'y} \phi_{n'v'y} \).
5. Estimate \( \theta_{n'} \) on \( Y_{n'} \) using LDA (Blei et al. 2003)).
6. Update \( \alpha_n' = \eta \odot \beta_v^\top \theta_{n'} + \xi \).
7. until the change in \( \theta_{n'} \) is smaller than a threshold.
8. Output \( \gamma_{n'} \) and \( \phi_{n'v'y} \) for bag and instance annotation.

- If \( v \) is a discrete view, update \( \beta_v \in R^{D_v} \):
  \[
  \beta_v = \sum_{n} \sum_{m} x_{nvm} \phi_{nvm}
  \]

- If \( v \) is a continuous view, update \( \beta_v = \{ \mu_v, \Sigma_v \} \)
  \[
  \mu_v = \sum_{nvm} \phi_{nvm} x_{nvm} / \sum_{nvm} \phi_{nvm}
  \]
  \[
  \Sigma_v = \sum_{nvm} \phi_{nvm} (x_{nvm} - \mu_v)^\top x_{nvm} / \sum_{nvm} \phi_{nvm}
  \]

The training algorithm is summarized in Alg. 2.

### Testing with MIMLmix

During testing, for a new example \( X_n' \), we set \( \xi \) to a constant larger than 0 (\( \xi = 0.1 \) by default), initialize \( y_{n'l} = 1, \forall l \); thus we have \( \alpha_n' > 0, \forall l \). By doing so, we trigger all the labels for consideration. The algorithm is summarized in Alg. 3. The information is passed from the label-instance part to the topic-label part by exploiting label assignments for instances in multi-views (line 4), and from the topic-label part to label-instance part through \( \alpha_n' \), where \( y_{n'} \) is implicitly set to \( \beta_v^\top \theta_{n'} \) (line 6).

In implementation, in order to reduce the randomness of the sampling step in line 4, we obtain the averaged topic
distribution $\theta'_{v}$ over all iterations, then use it to update the prior information for the final bag and instance annotation.

**Multi-Views with Unequal Importance**

Let $l_v$, be the random variable that represents the number of instances of the $v$-th view per example, and assume that $l_v$ follows Possion distribution $l_v \sim \text{Po}(\lambda_v)$. Suppose we fix the number of instances across views to a constant $\lambda$, the conditional distribution $P(l_v|\lambda)$ follows a Multinomial distribution with parameter $\rho_i = (\rho_1, \ldots, \rho_v)$, where $\rho_v = \frac{\lambda}{\lambda_v}$.

We then rewrite the joint probability $p(z_n, X_n|\theta_n, \beta)$:

$$p(z_n, X_n|\theta_n, \beta) = \prod_{v=1}^{V} p(x_{num_v}, z_{num_v}|\theta_n, \beta)^{w_v}$$

where $w_v = \lambda \rho_v / \lambda_v = \frac{1}{\sum_v w_v}$, the above equation is the same as the original one but with a new insight “in a bag of one instance of view $v$ repeated $\lambda_v$ times, instead of repeating the instance as $\lambda_v$, we repeat it with $w_v \lambda_v = \lambda \rho_v$ times”. Replacing the constraint $w_v = 1$ with $\sum_v w_v = 1$, one can change the weights of instances in different views. We modify the variational distributions similarly, and obtain the terms of the modified ELBO related to $w_v$ as follows:

$$L_{[w_v]} = \sum_i w_v \left[ E_q[log p(z_{vi}|\theta_n)] - E_q[log q(z_{vi}|\phi_{vi})] ight]$$

$$+ E_q[log p(x_{vi}|z_{vi}, \beta_{1:v})]$$

(11)

where $i$ runs over $M$ instances in the $v$-th view from all the training examples. Let $\Delta_v = 1/M_v(\partial L / \partial w_v)$, it is intuitive that $\Delta_v$ measures how likely one instance in view $v$ is generated given the bag labels. Maximizing $\sum_v \Delta_v w_v$ with $\sum_v w_v = 1$ yields an extreme result as the view with the largest $\Delta_v$ will receive all the weights while the others are zero. We then find a simple solution by setting $w_v = \frac{\Delta_v}{\sum_v \Delta_v + \tau_v}$, where $\Delta_v = \log_2(\Delta_v - \min_{i} \Delta_i + 2)$ is the scaled values of $\Delta_i$; and $\tau_v$ can be interpreted as the prior for the $v$-th view.

The updates of $\phi$ and $\beta$ are still the same as in Algorithm 2 as the view weight is eliminated due to normalization or division within each view, the update for $\gamma_{n,y}$ is changed to $\gamma_{n,y} = \alpha_{n,y} \sum_{v} w_{v} \phi_{n,v,y}$. During testing, we sample $Y_n'$ in Algorithm 3 by normalizing $\sum_v w_{v} \phi_{n,v,y}$. Note that all the experiments with multi-view datasets in the next section are conducted with this variant of MIMLmix.

**Experiments**

We perform experiments on 2 multi-view datasets and 3 single-view datasets. The summary of these datasets are given in Table 1. Citeseerx-10k contains scientific papers in two views, i.e. content (v1) and citations (v2). ImageCLEF (Müller et al. 2010) contains images with two views: visual (v1) and textual (v2). Here, we use the same subset that has been used in (Nguyen, Zhan, and Zhou 2013). Each example in the visual view is represented by a bag of segmented regions, one region is represented by a frequency vector of 1000 visual words, which are obtained by clustering Opponent SIFTs (Van de Sande, Gevers, and Snoek 2010). Citeseerx-10k has 1072 partial examples, and ImageCLEF has 2114 partial examples; most of partial examples do not have the second view. Among single-view datasets, LetterCarroll, MSRC-v2 were collected by (Briggs, Xiaoli, and Raich 2012); and IAPRTC-12 dataset was selected from (Escalante et al. 2010).

**Evaluation:** MIML methods are evaluated from three aspects, i.e. example-pivot evaluation using hamming loss (h.l.) and average precision (a.p.), (Zhou and Zhang 2007); label-pivot evaluation using mean average precision (m.a.p) and macro-F1 (ma-f1) for labels that appear at least once in training/testing dataset (Rubin et al. 2012); and instance-pivot evaluation in terms of instance accuracy (ins-acc) (Briggs, Xiaoli, and Raich 2012). To measure h.l., ma-f1, top $L$ labels with highest decision values are selected as the annotation for each example. Here, $L$ is chosen based on the average number of labels per example. We conduct 30 times evaluation for ImageCLEF, each time we use 1000 examples for training and 1000 examples for testing; 10-fold cross-validation is conducted for the other datasets. Only single-view datasets have instance labels for ins-acc evaluation.

**Compared Methods:** On multi-view datasets, the following methods are compared and contrasted: MIMLmix, MIMLmix* (MIMLmix with $\eta = 0$); M3LDA (Nguyen, Zhan, and Zhou 2013); cs.SVM which combines decision values of single-view, cost-sensitive SVMs; and MIMLmix with individual views (MIMLmix.v1 and MIMLmix.v2). In order to train single-view SVM, we accumulate multiple instances to obtain a single instance per bag, then use one-vs-all for multi-label learning.

On single-view datasets, we compare MIMLmix, MIMLmix* with other MIML methods including RankLossSVM (Briggs et al. (2012)); MIMLSVM (Zhou and Zhang 2007); cs.MISVM (Andrews et al. (2002)) which builds a cost sensitive Multi-instance SVM for every label; and DBA (Yang et al. (2009)).

**Multi-view Datasets without Partial Examples**

We evaluate MIMLmix and the compared methods in the case without partial examples, which are obtained by removing all the partial examples from multi-view datasets. For MIMLmix methods, we set $\alpha' = 0.1$, $K = 200$ as default for both datasets, set $\eta = .3$, $\tau = 5$ for Citeseerx-10k; and $\eta = 10$ and $\tau = 0$ for ImageCLEF. M3LDA is conducted with the same setting as in (Nguyen, Zhan, and Zhou 2013) on ImageCLEF; and with $K = 200$, $\lambda = .5$, and the number of sampling iterations of 300 on Citeseerx-10k. One-vs-all cs.SVM classifiers are trained for every view, every label using LIBSVM (Chang and Lin 2011) with default parameters, except that the weights of positive and negative classes are

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#bags</th>
<th>#labels</th>
<th>#ipb</th>
<th>#dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citeseer-10K</td>
<td>10,799</td>
<td>500</td>
<td>35.7</td>
<td>2,000</td>
</tr>
<tr>
<td>(2 views)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageCLEF</td>
<td>8,000</td>
<td>78</td>
<td>18.4</td>
<td>1,000</td>
</tr>
<tr>
<td>(2 views)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LetterCarroll</td>
<td>166</td>
<td>26</td>
<td>4.3</td>
<td>16</td>
</tr>
<tr>
<td>MSRC-v2</td>
<td>591</td>
<td>23</td>
<td>2.97</td>
<td>48</td>
</tr>
<tr>
<td>IAPRTC-12</td>
<td>5,000</td>
<td>244</td>
<td>5.09</td>
<td>28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#bags</th>
<th>#labels</th>
<th>#ipb</th>
<th>#dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citeseerx-10K</td>
<td>10,799</td>
<td>500</td>
<td>35.7</td>
<td>2,000</td>
</tr>
<tr>
<td>(2 views)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageCLEF</td>
<td>8,000</td>
<td>78</td>
<td>18.4</td>
<td>1,000</td>
</tr>
<tr>
<td>(2 views)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LetterCarroll</td>
<td>166</td>
<td>26</td>
<td>4.3</td>
<td>16</td>
</tr>
<tr>
<td>MSRC-v2</td>
<td>591</td>
<td>23</td>
<td>2.97</td>
<td>48</td>
</tr>
<tr>
<td>IAPRTC-12</td>
<td>5,000</td>
<td>244</td>
<td>5.09</td>
<td>28</td>
</tr>
</tbody>
</table>

1 collected from http://citeseerx.ist.psu.edu/index
set to \#pos+\#neg and \#pos+\#neg, where \#pos and \#neg are the number of positive and negative bags, respectively. We combine decision values of single-view cs.SVM using the rule \((.3\times v1+.7\times v2)\) on ImageCLEF, and \((.6\times v1+.4\times v2)\) on Citeseerx, these parameters are selected by trying different values of combination.

The experimental results are represented in Table 2. It can be seen that MIMLmix outperforms other multi-view methods including MIMLmix* in most of the cases. Particularly, the performance of M3LDA is not satisfactory on Citeseerx dataset, mostly due to the small number of sampling iterations that we set to meet the time constraint. More details about computational comparison will be discussed later in this section. MIMLmix outperforms MIMLmix* by a large gap on ImageCLEF where labels are highly correlated. Compared to single view MIMLmix models, MIMLmix is significantly better on both multiview datasets. This validates the importance of combining multi-views to obtain better performance.

MIMLmix is significantly better than cs.SVM in most of the cases except for m.a.p on ImageCLEF. Interestingly, on ImageCLEF dataset, cs.SVM achieves much worse ma-f1 than MIMLmix although it has higher m.a.p. By examining the combined decision values of cs.SVM, we see that although cs.SVM obtains good ranking of examples with regards to some rare labels, the values for rare labels are not large enough to meet the cut of ma-f1 evaluation. This leads to the fact that a lot of rare labels have zero recalls, consequently low values of ma-f1.

### Multi-view Datasets with Partial Examples

This section compares MIMLmix with other methods in the presence of partial examples. Experimental settings are the same as in the previous section, except that we do not remove partial examples from multi-view datasets. The results on Citeseerx-10k and ImageCLEF with partial examples are shown in Fig. 3(a) and Fig. 3(b), respectively. Here, we only show the results of a.p and m.a.p metrics as the results of h.l. (resp. ma-f1) changes in the similar way with a.p (resp. m.a.p) but with smaller magnitudes. Also, we do not show the result of MIMLmix.v1 as most of partial examples have the second view missing.

Fig. 3 shows that most methods degenerate in the presence of partial examples. Nevertheless, MIMLmix outperforms all compared methods on both multi-view datasets. On Citeseerx-10k, the difference in the degeneration magnitude is not so obvious among these methods, probably due to the small rate of partial examples. On ImageCLEF, where there are more partial examples, it is not surprising to see that MIMLmix.v2 suffers more than its multi-view counterparts (MIMLmix, MIMLmix*) on both a.p and m.a.p metrics. M3LDA has comparably low degeneration because it also follows the Bayesian Network approach.

From Fig. 3(b), we can observe some interesting behaviors of cs.SVM on ImageCLEF dataset. It is shown that cs.SVM in the complete case has a.p metric worse than it is in the partial case, where there exist examples without textual view. This is indeed not difficult to understand as textual view tends to be more useful towards rare labels, and a naive
combination of the decision values can hurt frequent labels, resulting in the lower value of example-pivot evaluation metrics, which give more credits to frequent labels. On the other hand, we can see that the degeneration of cs.SVM on m.a.p metrics, which give more credits to frequent labels. On the other hand, we can see that the degeneration of cs.SVM on m.a.p results in the lower value of example-pivot evaluation metrics. However, this ratio parameter of MIMLSVM is set to $\eta > 0$, we may have to trade off between label-pivot for example-pivot/instance-pivot evaluations on datasets without strong label relationships. Nevertheless, MIMLmix still has better label-pivot evaluation compared to other MIML methods. Particularly on IAPRTC-12, the gap in m.a.p and ma-f1 between MIMLmix and other MIML methods becomes significant. In terms of ins-acc, MIMLmix obtains best results on two datasets and slightly worse than cs.MISVM on IAPRTC-12. MIMLSVM transforms MIML problem into SIML problem, and thus it cannot assign labels to instances, consequently ins-acc is not available.

**Time Cost Comparison**

Table 4 shows training times of MIML methods on 3 large datasets on the same computer (CPU of 3.3Gz, 4GB memory). MIMLmix is more effective than other MIML methods, particularly in contrast to M3LDA, where sampling method is used for training; and RankLossSVM where all labels in $Y$ are ranked for every example. Note that RankLossSVM is particularly time consuming when $Y$ is large.

**Conclusion**

This paper proposes MIMLmix, a model based on hierarchical Bayesian network for the general problem of multi-view MIML in the presence of partial examples. Extensive evaluation on 5 datasets suggests that (1) MIMLmix can naturally deal with multiple view MIML data with partial examples; (2) our method less suffers from the problem of label-imbalance; and (3) our training method is effective particularly on datasets with a large number of labels.

For the future work, stochastic variational inference (Hoffman et al. (2010)) can be applied to further reduce the computational complexity in training large datasets. It is also interesting to extend MIMLmix to a deep model for a larger representation capacity within each view.

**Acknowledgements:** We thank anonymous reviewers for suggestions, and Yang Yu for reading our draft.

*Table 3: Performance on single-view datasets; Here, o/e means a method is worse/better than MIMLmix with 95% t-test.*

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MIMLmix</th>
<th>MIMLmix*</th>
<th>DBA</th>
<th>cs.MISVM</th>
<th>MIMLSVM</th>
<th>RankL.SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.p.↑</td>
<td>.770 ± .040</td>
<td>.744 ± .037</td>
<td>.308 ± .025</td>
<td>.740 ± .048</td>
<td>.443 ± .059</td>
<td>.672 ± .057</td>
</tr>
<tr>
<td>h.l.↓</td>
<td>.126 ± .014</td>
<td>.130 ± .010</td>
<td>.246 ± .016</td>
<td>.098 ± .014</td>
<td>.139 ± .011</td>
<td>.137 ± .017</td>
</tr>
<tr>
<td>m.a.p↑</td>
<td>.761 ± .067</td>
<td>.771 ± .072</td>
<td>.388 ± .050</td>
<td>.627 ± .034</td>
<td>.397 ± .030</td>
<td>.632 ± .037</td>
</tr>
<tr>
<td>ma-f1↑</td>
<td>.556 ± .062</td>
<td>.568 ± .046</td>
<td>.218 ± .042</td>
<td>.372 ± .041</td>
<td>.227 ± .067</td>
<td>.442 ± .045</td>
</tr>
<tr>
<td>ins-acc↑</td>
<td>.623 ± .053</td>
<td>.604 ± .049</td>
<td>.122 ± .033</td>
<td>.571 ± .052</td>
<td>N/A</td>
<td>.493 ± .048</td>
</tr>
</tbody>
</table>

*Table 4: Training time in seconds (here, #examples is the number of training examples in one evaluation).*

<table>
<thead>
<tr>
<th></th>
<th>#examples</th>
<th>MIMLmix</th>
<th>M3LDA</th>
<th>cs.MISVM</th>
<th>MIMLSVM</th>
<th>RankL.SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CiteSeer (2 views)</td>
<td>9,719</td>
<td>3,448</td>
<td>255,000</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>ImageCLEF (2 views)</td>
<td>1,000</td>
<td>150</td>
<td>10,000</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>IAPRTC-12 (1 view)</td>
<td>4,500</td>
<td>915</td>
<td>N/A</td>
<td>3,969</td>
<td>9,545</td>
<td>35,976</td>
</tr>
</tbody>
</table>


