Abstract

Label distribution covers a certain number of labels, representing the degree to which each label describes the instance. The learning process on the instances labeled by label distributions is called label distribution learning (LDL). Unfortunately, many training sets only contain simple logical labels rather than label distributions due to the difficulty of obtaining the label distributions directly. To solve this problem, we consider the label distributions as the latent vectors and infer the label distributions from the logical labels in the training datasets by using variational inference. After that, we induce a predictive model to train the label distribution data by employing the multi-output regression technique. The recovery experiment on fourteen label distribution datasets and the predictive experiment on ten multi-label learning datasets validate the advantage of our approach over the state-of-the-art approaches.

1. Introduction

Learning with ambiguity is a hot topic in recent machine learning and data mining research. A learning process is essentially building a mapping from the instances to the labels. This paper mainly focuses on the ambiguity at the label side of the mapping, i.e., one instance is not necessarily mapped to one label. Multi-label learning (MLL) (Tsoumakas & Katakis, 2006) studies the problem where each example is represented by a single instance while associated with a set of labels simultaneously, and the task is to learn a multi-label predictor which maps an instance to a relevant label set (Gibaja & Ventura, 2015; Zhang & Zhou, 2014). During the past decade, multi-label learning techniques have been widely employed to learn from data with rich semantics, such as text (Rubin et al., 2012), image (Cabral et al., 2011), audio (Lo et al., 2011), video (Wang et al., 2011), etc.

The relevance or irrelevance of a label to an instance is essentially relative in the real-world tasks. When multiple labels are associated with an instance, the relative importance among them is more likely to be different rather than exactly equal. For instance, when “sand” and “sailboat” are relevant to the two images in Fig. 1, “sand” is more significant than “sailboat” in image (a) and the opposite scenario occurs in image (b). On the other hand, the “irrelevance” of each irrelevant label may be very different. For instance, “bus” is more irrelevant than “sun” to the two images in Fig. 1 since “sun” always occurs in the view of the beach. However, in most of the supervised data, each instance is always assigned with the logical label $l_y^x \in \{0, 1\}$ to the relevant (irrelevant) label $y$, ignoring the relative importance among the relevant (irrelevant) labels.

To solve this problem, a more natural way to label an instance $x$ is to assign a real number $d_y^x$ to each possible label $y$, representing the degree to which $y$ describes $x$. Such $d_y^x$ is called the description degree of $y$ to $x$. For a particular instance, the description degrees of all the labels constitute a real-valued vector called label distribution (Geng et al., 2013). The learning process on the instances labeled by label distributions is therefore called label distribution learning (LDL) (Geng, 2016). The Label distribution is closer to the essential of the label side since the relative importance in label ambiguity is available by the label distribution.

However, the label distribution is not explicitly available in most training sets since quantifying the description de-
Variational Label Enhancement

Label distribution learning (LDL) is a novel learning paradigm, which labels an instance with a label distribution and learns a map-ping from instance to label distribution straightly. LDL has been successfully applied to many real applications, such as facial landmark detection (Su & Geng, 2019), age estimation (Gao et al., 2018; Geng et al., 2013), head pose estimation (Geng & Xia, 2014), multi-label ranking for natural scene images (Geng & Luo, 2014), zero-shot Learning (Huo & Geng, 2017) and emotion analysis from texts (Zhou et al., 2016). According to the theoretical analysis (Wang & Geng, 2019), LDL is approximate to the optimal classifier via learning on the instances labeled by the ground-truth label distributions. However, in most training sets, the label distribution is not explicitly available. There are few work to deal with this situation. One recent paper (Li et al., 2015) adopts the propagation technique to generate the label distributions without considering the correlations between the labels.

Label distribution explicitly models label ambiguity with the description degree, which is not the probability that \( y \) correctly labels \( x \), but the proportion that \( y \) accounts for in a full class description of \( x \). Therefore, label distribution can be distinguished from the previous studies on probabilistic labels (Quost & Denœux, 2009; Denœux & Zouhal, 2001; Smyth et al., 1995), where the basic assumption is that only one ‘correct’ label is assigned to each instance. Probabilistic labels are mainly used when the real label of the instance cannot be obtained with certainty. In practice, it is usually difficult to determine the probability (or confidence) of a label. In most cases, it relies on the prior knowledge of the human experts, which is a highly subjective and variable process. As a result, the problem of learning from probabilistic labels has not been extensively studied to date.

From the conceptual point of view, it is worthwhile to distinguish description degree from the concept membership used in fuzzy classification. Membership is designed to handle the status of partial truth, which is a truth value which ranges between completely true and completely false. On the other hand, description degree reflects the ambiguity of the label description of the instance, i.e., one label may only partially describe the instance, but it is completely true that the label describes the instance. Fortunately, although the concept of membership is fundamentally different from description degree, some methods (Gayar et al., 2006; Jiang et al., 2006) which focus on generating membership can be applied to generate label distributions.

Label enhancement (LE) is a process to recover the label distributions from the logical labels in the training datasets. GLLE (Xu et al., 2018), LP (Li et al., 2015) and ML (Hou et al., 2016) are three representative algorithms of LE. They assume that the label distribution space should share similar local topological structure in the feature space. GLLE constructs a local similarity matrix to preserve the topological structure information of the feature space, LP adopts label propagation technique to propagate labeling-importance information, ML adopts the local linear embedding technique to achieve identified label degrees. Nonetheless, these methods all rely on the smoothness assumption (Zhu et al., 2005), i.e., the points close to each other are more likely to share a label. This assumption, however, might restrict modeling capacity, as graph edges need to be necessarily encoded which introduces additional bias.

In the next section, a novel label enhancement approach will be introduced. Different from existing label enhancement approaches, the generative model of the label distribution is proposed and the label distribution could be recovered via variational inference with limited assumption. The predictive model by employing multi-output regression techniques is also induced to leveraging the recovered label distributions for multi-label learning.

### 3. The LEVI Method

First of all, the main notations used in this paper are listed as follows. The instance variable is denoted by \( x \), the particular...
We construct the approximate posterior distribution.

We begin with the definition of Kullback-Leibler divergence. Applying Bayes rule:

\[ KL[q(d|l, x)||p(d|l, x)] = \mathbb{E}_{q(d|l, x)}[\log q(d|l, x) - \log p(d|l, x)] \]

Applying Bayes rule:

\[ KL[q(d|l, x)||p(d|l, x)] = \mathbb{E}_{q(d|l, x)}[\log q(d|l, x) - \log p(l|d) - \log p(x|d) + \log p(d) + \log p(l, x)] \]

Here, \( \log p(l, x) \) comes out of the expectation because it does not depend on \( d \):

\[ KL[q(d|l, x)||p(d|l, x)] = \log p(l, x) - \mathbb{E}_{q(d|l, x)}[\log p(l|d) + \log p(x|d)] + KL[q(d|l, x)||p(d)] \]

Since this KL-divergence is non-negative, we have:

\[ \log p(l, x) \geq \mathbb{E}_{q(d|l, x)}[\log p(l|d) + \log p(x|d)] - KL[q(d|l, x)||p(d)] \]

We construct the approximate posterior distribution \( q \) as an inference model, which has become a popular approach for efficient variational inference (Kingma & Welling, 2014; Rezende et al., 2014). Using an inference network, we avoid the need to compute per data point variational parameters, but can compute a set of global variational parameters instead. This allows us to amortise the cost of inference by generalizing between the posterior estimates for all latent variables through the parameters of the inference model, and allows for fast inference at both training and testing time. Then, the ELBO (Evidence Lower Bound) is written as:

\[ \mathcal{L}(x, l; \vartheta, \eta, w) = \mathbb{E}_{q_d(d|x)}[\log p_\vartheta(l|d) + \log p_\eta(x|d)] - KL[q_\varphi(d|x)||p(d)] \]

Here the inference network is introduced for \( q(d|x) \), and we parameterize them as deep neural networks whose outputs form the parameters of the distribution \( q_\varphi(d|x) \). The logical label vector \( l \) and the instance \( x \) are generated from the deep neural networks distribution \( p_\vartheta(l|d) \) and \( p_\eta(x|d) \), respectively.

3.2. Optimisation

The bound in Eq. (5) provides a unified objective function for optimisation of all the parameters \( w, \vartheta \) and \( \phi \) of the generative and inference models. By expanding the label distribution to \( d \in \mathbb{R}^c \), we assume that the prior over the latent label distribution be the centered isotropic multivariate Gaussian \( \mathcal{N}(0, I) \). We let the variational approximate posterior be a multivariate Gaussian with a diagonal covariance structure \( \mathcal{N}(\mu, \Sigma) \), where the mean and covariance matrix of the approximate posterior, \( \mu \) and \( \Sigma \), are outputs of the MLP with parameter \( w \). Then the KL divergence in the ELBO can be computed:

\[ KL[q_\varphi(d|x)||p(d)] = \frac{1}{2} \text{tr}(\Sigma) + \mu^\top \mu - k \]

where \( k \) is the dimensionality of the distribution.

Then, we assume \( \rho(l|d) \) be multivariate Bernoulli and \( \rho_\eta(x|d) \) be multivariate Gaussian whose distribution parameters are computed from \( d \) with the MLP. Then the first part of the ELBO can be computed:

\[ \mathbb{E}_{q_\varphi(d|x)}[\log p_\vartheta(l|d) + \log p_\eta(x|d)] = \frac{1}{L} \sum_{m=1}^{L} \frac{1}{2} \|x - \rho^{(m)}(l)\|^2 \]

\[ + \frac{1}{L} \sum_{m=1}^{L} \sum_{i=1}^{c} l_i \log \tau^{(m)}_i + (1 - l_i) \cdot \log \left(1 - \tau^{(m)}_i \right) \]

where \( \rho \) is the mean of the Gaussian distribution computed from \( d \), and \( \tau \) is the Bernoulli distribution parameter computed from \( d \).
Note that back-propagate the error through a layer that samples \(d\) from \(q_{\theta}(d|l, x)\), which is a non-continuous operation and has no gradient. In order to move the sampling to an input layer, the reparameterization trick (Rezende et al., 2014) is induced to sample \(d\) by:

\[
d = \mu + \Sigma \epsilon
\]

where \(\epsilon \sim \mathcal{N}(0, I)\). In this case, Eq. (7) can be computed and differentiated.

In order to keep the sign of the label distribution corresponding to relevant label or irrelevant label, we add the least squares into the objective function. Besides, the feature embedding is adapted to alleviate the reconstruction by using a lower dimensional vector \(z\) instead of \(x\). Then, we formulate the label enhancement problem into an optimization framework over Eq. (7) and Eq. (6), and yields the target function:

\[
T(\vartheta, \eta, w) = \frac{1}{L} \sum_{m=1}^{L} \frac{1}{2} \| z - \rho^{(m)} \|_2^2 + \lambda \| d^{(m)} - l \|_2^2 \\
+ \sum_{i=1}^{c} t_i \log \tau^{(m)}_i + (1 - t_i) \cdot \log \left(1 - \tau^{(m)}_i \right) \\
+ \frac{1}{2} \{ \text{tr}(\Sigma) + \mu^\top \mu - k - \log \det(\Sigma) \}
\]

where \(\Sigma = \sum_{l}^{\mu}, \mu = \sum_{l}^{\mu}, \rho^{(m)} = \sum_{l}^{\rho^{(m)}}, d^{(m)} = \mu + \Sigma \epsilon^{(m)}, \epsilon \sim \mathcal{N}(0, I)\).

4. LEVI for Multi-label Learning

Given the recovered \(d_i\) of \(x_i\), the original training set can be transformed into \(E = \{ (x_i, d_i) | 1 \leq i \leq n \}\). We generalize a regressor to solve the multi-dimensional case. In addition, our regressor not only concerns the distance between the predicted and the real values, but also the sign consistency of them. It leads to the minimization of

\[
\Omega(\Theta, b) = \frac{1}{2} \sum_{j=1}^{c} \| \theta_j \|^2 + C_1 \sum_{i=1}^{n} \Omega_{1i} + C_2 \sum_{i=1}^{n} \Omega_{2i},
\]

where \(\Theta = [\theta^1, ..., \theta^c], b = [b^1, ..., b^c], \Omega_1\) and \(\Omega_2\) are the regression loss and the sign loss, respectively.

As shown in Eq. (10), the first term of \(\Omega(\Theta, b)\) controls the complexity of the induced model. In addition, the second term of \(\Omega(\Theta, b)\) is defined to consider all dimensions into a unique restriction and yield a single support vector for all dimensions:

\[
\Omega_{1i} = \begin{cases} 0 & r_i \leq \varepsilon \\ r_i^2 - 2r_i\varepsilon + \varepsilon^2 & r_i > \varepsilon, \end{cases}
\]

where \(r_i = \| e_i \| = \sqrt{\epsilon^2 \epsilon_i^2 + e_i d_i - \varphi(x_i)^\top \Theta - b}\). This will create an insensitive zone determined by \(\varepsilon\) around the estimate, i.e., the loss of \(r\) less than \(\varepsilon\) will be ignored. The third term is used to make the signs of the predictive output and the logical label same as much as possible:

\[
\Omega_{2i} = -\sum_{j=1}^{c} t_i \log(\varphi(x_i)^\top \theta_j + b^j).
\]

The meaning of Eq. (12) is that if the signs of the predictive output and the logical label are different, there will be some positive loss, otherwise the loss will be negative.

It is a piecewise quadratic problem whose optimum can be integrated as solving a system of linear equations for \(j = 1, ..., c\):

\[
\left[ C_1 \Phi^\top F \Phi + I \right] \left[ \begin{array}{c} \theta_j \\ \eta_j \end{array} \right] = \left[ C_1 \Phi^\top F d^i + C_2 \Phi^\top U \right] \left[ \begin{array}{c} \theta_j \\ \eta_j \end{array} \right],
\]

where \(\Phi = [\varphi(x_1), ..., \varphi(x_n)]^\top, \eta = [a_1, ..., a_n]^\top, F^k = a_i \delta^k_i (\delta^k_i\text{ is the Kronecker’s delta function}), \text{ and } U = [u_1, ..., u_n]^\top\). Then, the direction of the optimal solution of Eq. (13) is used as the descending direction for the optimization of \(\Omega(\Theta, b)\), and the solution for the next iteration \((\Theta^{(k+1)} \text{ and } b^{(k+1)})\) is obtained via a line search algorithm along this direction.

LEVII proceeds to predict the set of proper labels for \(x\) via virtual label Bipartition (Li et al., 2015). An extra virtual label \(y_0\) is added into the original label set, i.e., the extended original label set \(\tilde{Y} = Y \cup \{y_0\} = \{y_0, y_1, ..., y_c\}\). In this paper, the origin value \(L\) is set to 0.5. Once the recovered label distribution and the predictive model have been learned on the extended original label set, the extended label distribution \(d^*\) corresponding to the test instance \(x^*\) can be predicted. Then, the predicted label set for \(x^*\) is determined as:

\[
f(x) = \{ y_j \mid d^*_{x_j} > d^*_{x_k}, 1 \leq j \leq c \}
\]

5. Experiments

5.1. Recovery Experiment

We consider the following learning setting. With each instance, a label distribution is associated. The training set, however, contains for each instance not the actual distribution, but logical labels. As shown in Fig. 2, we recover the label distributions from the logical labels via the LE algorithms, and then compare the recovered label distributions with the ground-truth label distributions. The label set includes the labels with the highest weights in the distribution, and is the smallest set such that the sum of these weights exceeds a given threshold. This setting can model, for instance, the way in which users label images or add
The logical labels in the datasets can be binarized from logical labels to the labels in \( Y \) into \( \{0, 1\} \). This process continues until \( H = \sum_{j=1}^{\hat{c}} d_{j} \) is less than a predefined threshold \( T \), where \( \hat{c} \) is the set of the current relevant labels. If \( H \) is less than a predefined threshold \( T \), we continue finding the greatest description degree among other labels excluded from \( Y^+ \) and select the label corresponding to the greatest description degree into \( Y^+ \). This process continues until \( H > T \). Finally, the logical labels to the labels in \( Y^+ \) are set to 1, and other logical labels are set to 0. In our experiments, \( T = 0.5 \).

5.1.1. Datasets

There are in total one artificial dataset and 13 real-world label distribution datasets\(^1\). These real-world datasets (Geng, 2016) collected from biological experiments on the yeast genes, facial expression images, natural scene images and movies, respectively. Some basic statistics about these 14 datasets are given in Table 1.

The artificial dataset is generated to show in a visual way whether the LE algorithms can recover the label distributions from the logical labels. In this dataset, the instance \( x \) is of three-dimensional and there are three labels. In order to show the results of LE algorithms in a direct and visual way, the examples of the toy dataset are selected from a certain manifold in the feature space. The first two components of the instance \( x \), \( x_1 \) and \( x_2 \), are located at a grid of the interval 0.04 within the range \([-1, 1]\), and there are in total \( 51 \times 51 = 2601 \) instances. The third component \( x_3 \) is calculated by

\[
x_3 = \sin((x_1 + x_2) \times \pi)
\]

The label distribution \( \mathbf{d} = [d_{x_1}^{x_1}, d_{x_2}^{x_1}, d_{x_3}^{x_1}] \) of \( x = [x_1, x_2, x_3]^T \) is created to deliberately make the description degree of one label depend on those of other labels (Geng, 2016).

5.1.2. Evaluation Measures

The output of LE algorithm is label distribution rather than logical output of clustering or classification, which makes some commonly used measures inapplicable. As suggested in (Geng, 2016), we select six measures, i.e., Chebyshev distance (Cheb), Clark distance (Clark), Canberra metric (Canber), Kullback-Leibler divergence (KL), cosine coefficient (Cosine) and intersection similarity (Intersec), which belong to the Minkowski family, the \( \chi^2 \) family, the \( L_1 \) family, the Shannon’s entropy family, the inner product family, and the intersection family, respectively. The first four are

\[
\text{Measure} \quad \text{Formula}
\]

\[
\text{Chebyshev} \downarrow \quad D_{IS1}(\mathbf{d}, \hat{\mathbf{d}}) = \max_{j} |d_{j} - \hat{d}_{j}|
\]

\[
\text{Clark} \downarrow \quad D_{IS2}(\mathbf{d}, \hat{\mathbf{d}}) = \sqrt{\sum_{j=1}^{c} (d_{j} - \hat{d}_{j})^2}
\]

\[
\text{Canberra} \downarrow \quad D_{IS3}(\mathbf{d}, \hat{\mathbf{d}}) = \sum_{j=1}^{c} \frac{|d_{j} - \hat{d}_{j}|}{d_{j} + \hat{d}_{j}}
\]

\[
\text{Kullback-Leibler} \downarrow \quad D_{IS4}(\mathbf{d}, \hat{\mathbf{d}}) = \sum_{j=1}^{c} d_{j} \ln \left( \frac{d_{j}}{\hat{d}_{j}} \right)
\]

\[
\text{cosine} \uparrow \quad \text{Sim}_1(\mathbf{d}, \hat{\mathbf{d}}) = \frac{\sum_{j=1}^{c} d_{j} \hat{d}_{j}}{\sqrt{\sum_{j=1}^{c} d_{j}^2 \sum_{j=1}^{c} \hat{d}_{j}^2}}
\]

\[
\text{intersection} \uparrow \quad \text{Sim}_2(\mathbf{d}, \hat{\mathbf{d}}) = \min(d_{j}, \hat{d}_{j})
\]

---

\(^1\)http://palm.seu.edu.cn/xgeng/LDL/index.htm
Variational Label Enhancement

distance measures and the last two are similarity measures.

Suppose the real label distribution is \( d = [d_1, d_2, ..., d_c] \),
the predicted label distribution is \( \hat{d} = [\hat{d}_1, \hat{d}_2, ..., \hat{d}_c] \),
then the formulae of the six measures are summarized in Table 2,
where the “↓” after the distance measures indicates “the smaller the better”,
and the “↑” after the similarity measures indicates “the larger the better”.
Considering that the selected measures all come from different families, the
selected measures are significantly different in both syntax and semantics.

5.1.3. Methodology

The five LE algorithms, i.e., FCM (Gayar et al., 2006), KM (Jiang et al., 2006), LP (Li et al., 2015), ML (Hou et al., 2016), GLLE (Xu et al., 2018), and our LEVI are all applied
to the 14 real-world datasets shown in Table 1. For each
compared algorithm, we adopt the suggested configuration
in their literature, i.e., the parameter \( \alpha \) in LP is set to 0.5,
the number of neighbors \( K \) for ML is set to \( c + 1 \), the
parameter \( \beta \) in FCM is set to 2, and the kernel function
in KM is Gaussian kernel. For GLLE, the parameter \( \lambda_1 \)
and \( \lambda_2 \) are chosen among \( \{10^{-2}, 10^{-1}, ..., 100\} \), and the
number of neighbors \( K \) is set to \( c + 1 \). The kernel function
in GLLE is Gaussian kernel. For LEVI, the MLPs were
constructed with two hidden layer, each with 500 hidden
units and softplus activation functions.

5.1.4. Recovery Performance

In order to visually show the results of the LE algorithms
on the artificial dataset, the description degrees of the three
labels are regarded as the three color channels of the RGB
color space, respectively. In this way, the color of a point in
the feature space will visually represent its label distribution.
Thus, the label distribution recovered by the LE algorithms
can be compared with the ground-truth label distribution
through observing the color patterns on the manifold. For
easier comparison, the images are visually enhanced by
applying a decorrelation stretch process. The results are
shown in Fig. 3. It can be seen that LEVI recovers almost
identical color patterns with the ground-truth. GLLE, LP,
ML can also recover similar color patterns with the ground-truth.
However, FCM, KM fails to obtain a reasonable result.

For quantitative analysis, table 3 tabulates the results of the
five LE algorithms on all real-world the datasets, and the
best performance on each dataset is highlighted by boldface.
For each evaluation metric, ↓ indicates the smaller the better
while ↑ indicates the larger the better. Note that since each
LE algorithm only runs once, there is no record of standard
deviation. we can find that our method achieves optimal
average rank in terms of all the six evaluation metrics.

5.2. Predictive Experiment

In this experiment, the effective performance of LEVI for
MLL prediction can be validated.

5.2.1. Datasets

There are ten MLL datasets\(^2\) used in the experiments. Some
basic statistics about these datasets are given in Table 4. The
MLL datasets cover a broad range of cases with diversified
multi-label properties and thus serve as a solid basis for
thorough comparative studies.

5.2.2. Evaluation Measures

Five widely-used MLL evaluation metrics are selected in
this experiment, i.e., Hamming loss, One-error, Coverage,
Ranking loss and Average precision (Zhang & Zhou, 2014).
Note that for all the five multi-label metrics, their values
vary between [0,1]. Furthermore, for average precision,
the larger the values the better the performance; While for
the other four metrics, the smaller the values the better
the performance. These metrics serve as good indicators

\(^2\)mulan.sourceforge.net/datasets.html
Table 3. Recovery Results (value(rank)) Evaluated by Six LLD Measures

<table>
<thead>
<tr>
<th>Comparator algorithm</th>
<th>SF</th>
<th>spcm</th>
<th>spo</th>
<th>dtl</th>
<th>cold</th>
<th>heat</th>
<th>Chls.</th>
<th>cpe</th>
<th>spq</th>
<th>tnl</th>
<th>ccl</th>
<th>alpha</th>
<th>DiffE</th>
<th>Mov</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCM</td>
<td>0.1324</td>
<td>0.2354</td>
<td>0.1624</td>
<td>0.0973</td>
<td>0.1441</td>
<td>0.1895</td>
<td>0.1304</td>
<td>0.1246</td>
<td>0.0524</td>
<td>0.0514</td>
<td>0.0464</td>
<td>0.1315</td>
<td>0.2393</td>
<td>4.08</td>
<td></td>
</tr>
<tr>
<td>KM</td>
<td>0.2146</td>
<td>0.4680</td>
<td>0.2778</td>
<td>0.2570</td>
<td>0.2520</td>
<td>0.1756</td>
<td>0.1532</td>
<td>0.0706</td>
<td>0.0636</td>
<td>0.2386</td>
<td>0.2354</td>
<td>6.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>0.1073</td>
<td>0.1463</td>
<td>0.1143</td>
<td>0.1264</td>
<td>0.1373</td>
<td>0.0863</td>
<td>0.0903</td>
<td>0.0903</td>
<td>0.0643</td>
<td>0.0423</td>
<td>0.0403</td>
<td>0.1232</td>
<td>0.1613</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>ML</td>
<td>0.1895</td>
<td>0.4050</td>
<td>0.2757</td>
<td>0.2445</td>
<td>0.2425</td>
<td>0.1654</td>
<td>0.1715</td>
<td>0.1485</td>
<td>0.0725</td>
<td>0.0715</td>
<td>0.0575</td>
<td>0.2335</td>
<td>0.1644</td>
<td>4.85</td>
<td></td>
</tr>
<tr>
<td>GLLE</td>
<td>0.0872</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.0982</td>
<td>0.1267</td>
<td>0.1267</td>
<td>2.06</td>
<td></td>
</tr>
<tr>
<td>LEVI</td>
<td>0.0751</td>
<td>0.0631</td>
<td>0.0671</td>
<td>0.0511</td>
<td>0.0511</td>
<td>0.0431</td>
<td>0.0431</td>
<td>0.0431</td>
<td>0.0431</td>
<td>0.0431</td>
<td>0.0431</td>
<td>0.1099</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Statistics of the 10 datasets used in MLL predictive experiment

<table>
<thead>
<tr>
<th>No.</th>
<th>Dataset</th>
<th>#Examples</th>
<th>#Features</th>
<th>#Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cal500</td>
<td>502</td>
<td>68</td>
<td>174</td>
</tr>
<tr>
<td>2</td>
<td>emotion</td>
<td>593</td>
<td>72</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>medical</td>
<td>978</td>
<td>1,449</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>1log</td>
<td>1,460</td>
<td>1,004</td>
<td>75</td>
</tr>
<tr>
<td>5</td>
<td>enron</td>
<td>1,702</td>
<td>1,001</td>
<td>53</td>
</tr>
<tr>
<td>6</td>
<td>msra</td>
<td>1,868</td>
<td>898</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>image</td>
<td>2,000</td>
<td>294</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>scene</td>
<td>2,407</td>
<td>294</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>slashdot</td>
<td>3,782</td>
<td>1,079</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>core5k</td>
<td>5,000</td>
<td>499</td>
<td>374</td>
</tr>
</tbody>
</table>

Table 5. Tabulates the results of the four LE based algorithms (LEVI, GLLE, LP and ML) and the four state-of-the-art algorithms (BR, CLR, ECC and RAKEL) on the ten MLL datasets evaluated by five evaluation metrics, and the best for comprehensive comparative studies as they evaluate the performance of the learned models from various aspects.

5.2.3. METHODOLOGY

In this paper, we choose to compare the performance of LEVI against four well established multi-label learning algorithms, including Binary Relevance (BR) (Boutell et al., 2004), Calibrated Label Ranking (CLR) (F{"u}rnkranz et al., 2008), Ensemble of Classifier Chains (ECC) (Read et al., 2011), Random k-labelsets (RAKEL) (Tsoumakas et al., 2011). For ECC, the ensemble size is set to 30. For RAKEL, the ensemble size is set to be 2q with k = 3 as suggested in the literature (Tsoumakas et al., 2011). Note that some work (Li et al., 2015; Hou et al., 2016; Xu et al., 2019) validate the effectiveness of LP, ML and GLLE in MLL, LEVI is also compared with them. A deep model (MLP) is trained with logical labels, which has the same structure as the encoder of the VAE. For each compared algorithm, we adopt the suggested configuration in their literature. For the predictive model in LEVI, the parameters $C_1$ and $C_2$ are set to 1 and 10.

5.2.4. PREDICTIVE PERFORMANCE
Table 5. Predictive performance of each algorithm (mean±std(rank)) measured by five MLL measures. We add results of a MLP model which has the same structure as the encoder of the our model.

<table>
<thead>
<tr>
<th>Comparing algorithm</th>
<th>cmcS</th>
<th>encren</th>
<th>medical</th>
<th>Log</th>
<th>cross</th>
<th>image</th>
<th>sscene</th>
<th>state</th>
<th>stable</th>
<th>rank</th>
<th>Avg. Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Algorithm</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEVI</td>
<td>0.272 ±0.041(3)</td>
<td>0.152 ±0.048(3)</td>
<td>0.452 ±0.046(3)</td>
<td>0.154 ±0.050(3)</td>
<td>0.989 ±0.051(3)</td>
<td>0.642 ±0.050(3)</td>
<td>0.962 ±0.050(3)</td>
<td>0.926 ±0.050(3)</td>
<td>0.699 ±0.050(3)</td>
<td>0.112 ±0.050(3)</td>
<td>1.50 ±0.050(3)</td>
</tr>
<tr>
<td>GLE</td>
<td>0.179 ±0.032(2)</td>
<td>0.190 ±0.031(2)</td>
<td>0.257 ±0.032(2)</td>
<td>0.014 ±0.030(2)</td>
<td>0.960 ±0.032(2)</td>
<td>0.647 ±0.030(2)</td>
<td>0.960 ±0.030(2)</td>
<td>0.925 ±0.030(2)</td>
<td>0.697 ±0.030(2)</td>
<td>0.113 ±0.030(2)</td>
<td>3.00 ±0.030(2)</td>
</tr>
<tr>
<td>MLP</td>
<td>0.322 ±0.048(3)</td>
<td>0.105 ±0.048(3)</td>
<td>0.452 ±0.046(3)</td>
<td>0.124 ±0.050(3)</td>
<td>0.989 ±0.051(3)</td>
<td>0.642 ±0.050(3)</td>
<td>0.962 ±0.050(3)</td>
<td>0.926 ±0.050(3)</td>
<td>0.699 ±0.050(3)</td>
<td>0.112 ±0.050(3)</td>
<td>3.00 ±0.050(3)</td>
</tr>
<tr>
<td>LP</td>
<td>0.190 ±0.032(3)</td>
<td>0.235 ±0.032(3)</td>
<td>0.525 ±0.032(3)</td>
<td>0.062 ±0.030(3)</td>
<td>0.960 ±0.032(3)</td>
<td>0.647 ±0.030(3)</td>
<td>0.960 ±0.030(3)</td>
<td>0.925 ±0.030(3)</td>
<td>0.697 ±0.030(3)</td>
<td>0.113 ±0.030(3)</td>
<td>3.00 ±0.030(3)</td>
</tr>
<tr>
<td>ML</td>
<td>0.190 ±0.032(4)</td>
<td>0.255 ±0.032(4)</td>
<td>0.525 ±0.032(4)</td>
<td>0.164 ±0.030(4)</td>
<td>0.960 ±0.032(4)</td>
<td>0.647 ±0.030(4)</td>
<td>0.960 ±0.030(4)</td>
<td>0.925 ±0.030(4)</td>
<td>0.697 ±0.030(4)</td>
<td>0.113 ±0.030(4)</td>
<td>3.00 ±0.030(4)</td>
</tr>
<tr>
<td>BR</td>
<td>0.258 ±0.038(8)</td>
<td>0.233 ±0.038(8)</td>
<td>0.690 ±0.038(8)</td>
<td>0.323 ±0.070(8)</td>
<td>0.313 ±0.080(8)</td>
<td>0.314 ±0.080(8)</td>
<td>0.229 ±0.080(8)</td>
<td>0.368 ±0.080(8)</td>
<td>0.240 ±0.080(8)</td>
<td>0.416 ±0.080(8)</td>
<td>8.30 ±0.080(8)</td>
</tr>
<tr>
<td>CLR</td>
<td>0.232 ±0.038(7)</td>
<td>0.272 ±0.038(7)</td>
<td>0.752 ±0.038(7)</td>
<td>0.432 ±0.087(7)</td>
<td>0.312 ±0.097(7)</td>
<td>0.312 ±0.097(7)</td>
<td>0.288 ±0.097(7)</td>
<td>0.266 ±0.097(7)</td>
<td>0.114 ±0.097(7)</td>
<td>6.30 ±0.097(7)</td>
<td></td>
</tr>
<tr>
<td>ECT</td>
<td>0.205 ±0.036(6)</td>
<td>0.272 ±0.036(6)</td>
<td>0.752 ±0.036(6)</td>
<td>0.327 ±0.076(6)</td>
<td>0.312 ±0.086(6)</td>
<td>0.312 ±0.086(6)</td>
<td>0.213 ±0.086(6)</td>
<td>0.268 ±0.086(6)</td>
<td>0.113 ±0.086(6)</td>
<td>6.70 ±0.086(6)</td>
<td></td>
</tr>
<tr>
<td>RAKEL</td>
<td>0.445 ±0.039(8)</td>
<td>0.254 ±0.039(8)</td>
<td>0.692 ±0.039(8)</td>
<td>0.411 ±0.080(8)</td>
<td>0.241 ±0.090(8)</td>
<td>0.241 ±0.090(8)</td>
<td>0.253 ±0.090(8)</td>
<td>0.272 ±0.090(8)</td>
<td>0.672 ±0.090(8)</td>
<td>8.10 ±0.090(8)</td>
<td></td>
</tr>
</tbody>
</table>

**Performance on each dataset is highlighted by boldface. For each evaluation metric, ↓ indicates the smaller the better while ↑ indicates the larger the better. All the algorithms are tested via ten-fold cross validation. The ranks are given in the parentheses right after the performance values. The average rank of each algorithm over all the datasets is also calculated and given in the last row of each table.**

When looking at the average ranks over all the ten real-world datasets, LEVI achieves rather competitive performance over other algorithms. Besides, the rankings of each LE based algorithm on five measures are higher than the four state-of-the-art MLL algorithms. When compared with the state-of-the-art MLL algorithms, LEVI ranks 1st in 84.0% cases and ranks 2nd in 10.0% cases. Thus, LEVI based MLL algorithm achieves rather superior performance over the state-of-the-art multi-label learning algorithms across all the evaluation measures.

**6. Conclusion**

Label enhancement can recover the label distributions from the logical labels in the training sets, which reinforces the supervision information in the training sets. By induce the generative model of the label distribution and adopt the variational inference technique, we give a lower bound of the label distribution and propose a novel LE approach called Label Enhancement via Variational Inference (LEVI) to infer the label distributions from the logical labels. Extensive comparative studies clearly validate the advantage of LEVI against other LE algorithms and the effectiveness of MLL after LE pre-process on the logical-labels database.
Acknowledgements

This research was supported by the National Key Research & Development Plan of China (No. 2017YFB1002801), the National Science Foundation of China (61622203), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Collaborative Innovation Center of Wireless Communications Technology.

References


