One Positive Label is Sufficient: Single-Positive Multi-Label Learning with Label Enhancement

Anonymous Author(s)
Affiliation
Address
email

Abstract

Multi-label learning (MLL) learns from the examples each associated with multiple labels simultaneously, where the high cost of annotating all relevant labels for each training example is challenging for real-world applications. To cope with the challenge, we investigate single-positive multi-label learning (SPMLL) where each example is annotated with only one relevant label and show that one can successfully learn a theoretically grounded multi-label classifier for the problem. In this paper, a novel SPMLL method named SMILE, i.e., Single-positive MultI-label learning with Label Enhancement, is proposed. Specifically, an unbiased risk estimator is derived, which could be guaranteed to approximately converge to the optimal risk minimizer of fully supervised learning and shows that one positive label of each instance is sufficient to train the predictive model. Then, the corresponding empirical risk estimator is established via recovering the latent soft label as a label enhancement process, where the posterior density of the latent soft labels is approximate to the variational Beta density parameterized by an inference model. Experiments on benchmark datasets validate the effectiveness of the proposed method.

1 Introduction

Multi-label learning (MLL) learns from examples each associated with multiple labels simultaneously and aims to derive a predictive model which can assign a set of relevant labels for the unseen instance [30, 44]. During the past decade, multi-label learning has been widely employed to learn from data with rich semantics, such as multimedia content annotation [38, 33], text categorization [29, 27], music emotion analysis [21, 35], and bioinformatics analysis [3], etc.

However, in practice, obtaining ground-truth labels for training datasets is costly due to the expensive and time-consuming manual annotations. Comparing with multi-class learning where an example is associated with only *one positive label*, multi-label learning requires the *complete positive label set* for each example. On this account, the annotation cost of multi-label learning is significantly higher than multi-class classification, which limits its application especially when the number of categories is large.

To mitigate this problem, the setting of single-positive multi-label learning (SPMLL) [5] could allow for significantly reduced annotations costs for the datasets, where each example is annotated with only one relevant label. In Figure 1, compared to fully labeled case, the SPMLL approaches on single-positive labeled examples only incur a tolerable drop in the performance but drastically reduce the amount of supervision required to train multi-label classifiers. By establishing the SPMLL methods with the learning power of DNNs, recent work [5] has also empirically validated that SPMLL

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

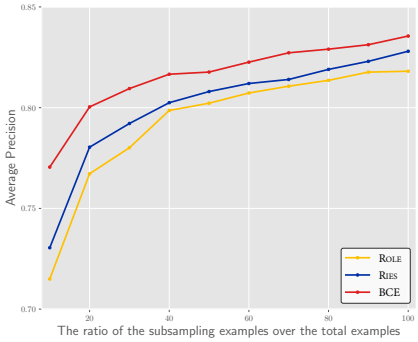

Figure 1: Test average precision on `tmc2007`. Each curve is generated by randomly subsampling the examples from the training set, where BCE is trained on fully labeled examples via binary cross-entropy loss while the SPMLL methods (ROLE [5] and the proposed SMILE) are trained on single-positive case.

would reduce the annotations costs while achieve good performance in practice. However, no method could provide theoretical insights as to why the model can converge to an ideal one.

In this paper, we propose a theoretically-guaranteed method named SMILE, i.e., Single-positive MultI-label learning with Label Enhancement. Specifically, we first derive an unbiased risk estimator, which suggests that one positive label of each instance is sufficient to train the predictive models for multi-label learning. In addition, we derive an estimation error bound which guarantees the learning consistency [25]. Then we could design a benchmark solution via recovering the soft label for each training example in the label enhancement process, which infers the posterior density of the latent soft labels via taking on the approximate Beta density parameterized by an inference model. Our contributions can be summarized as follows:

- Theoretically, we for the first time derive an unbiased risk estimator for SPMLL. Based on this, we establish an estimation error bound that guarantees the learning consistency and demonstrates that the obtained risk minimizer of single-positive multi-label learning would approximately converge to the optimal risk minimizer of fully supervised learning as the number of training data tends to infinity.
- Practically, we propose the method SMILE for SPMLL via leveraging the latent soft labels recovered in a label enhancement process. The posterior density of the latent soft label is inferred via taking on the approximate Beta density parameterized by an inference model and an evidence lower bound for the optimization is deduced.

Experiments on the benchmark datasets validate the effectiveness of the proposed method.

2 Related Work

In multi-label learning (MLL), each example is associated with a number of valid labels simultaneously. To cope with an output space which is exponential in size to the number of class labels, numerous approaches propose exploiting label correlations to improve the learning process [14, 42, 31]. The simplest one is the first-order type, which disassembles the MLL problem into a number of binary classification problems [2, 43]. The second-order approaches consider the label correlations between pairs of labels [8, 11]. The high-order approaches further focus on the label correlations among label set [26, 32]. Another line of research focus on manipulating the feature space via formalizing label-specific feature to each class label to facilitate multi-label classification [23, 15, 40]. In addition, some work focus on dealing with MLL via deep models. A directed graph over the labels is established via employing GCN to propagate information among all label nodes [4]. Transformer is leveraged to for exploring the label dependency by introducing a ternary encoding scheme to represent the state of label [20].

In practice, label information is often incomplete at training time because it can be extremely difficult to acquire exhaustive supervision. Different approaches have been proposed to address the MLL with missing labels [12], which is also termed as MLL with partial labels [7]. A transductive learning

method is proposed to concatenate features and labels and apply the matrix completion technique to it [12]. Then, the inductive learning method is proposed to exploit the structure of specific loss functions to offer efficient algorithms for learning with missing labels [39]. Wu [34] recovers the full label assignment for each sample by enforcing consistency with available label assignments and smoothness of label assignments. The global and local label correlations are exploited simultaneously [45], through learning a latent label representation and optimizing label manifolds for the missing label cases. Durand [7] empirically compare different labeling strategies to show the potential for using partial labels on multi-label datasets. Another method regularizes the cross-entropy loss with a cost function that measures the smoothness of labels and features to alleviate the overfitting issue when training data contains missing label [16].

Compared to multi-label learning with missing labels, single-positive multi-label learning [5] considers the hardest version of this problem, where annotators are only asked to provide a single positive label for each training example and no additional negative or positive labels. When collecting multi-label annotations, it may be more efficient to annotate the only one label rather than multiple labels for each example. To learn from SPMLL examples, an intuitive solution is "assume negative" (AN), which assumes that unobserved labels are negative and trains the predictive model with binary cross-entropy loss on observed positive label. Recent work [5] proposes some methods to reduce the damaging effects of false-negative labels. An expected positive regularization [5] is proposed to avoid the problem but the expected number of positive labels of each example should be given. Label smoothing [28] is employed to reduce the the impact of the incorrect labels (i.e. those labels incorrectly assumed to be negative). Another approach [5] online estimates of the unobserved labels throughout training and encourages the classifier predictions to match the estimated labels via binary entropy loss. However, no methods can provide theoretical insights as to why the model can converge to an ideal one.

3 Problem Setup

3.1 Multi-Label Learning

For multi-label learning, each example is associated with multiple labels, and the goal is to induce a predictive model which can assign a set of relevant labels for the unseen instance. Let $\mathcal{X} = \mathbb{R}^q$ be the q-dimensional instance space and $\mathcal{Y} = \{1, 2, ..., c\}$ be the label space with c class labels. Given the MLL training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) | 1 \leq i \leq n\}$ where $\boldsymbol{x}_i \in \mathcal{X}$ denotes the q-dimensional instance and $Y_i \in \mathcal{C}$ is the set of relevant labels associated with $\boldsymbol{x}_i$ where $\mathcal{C} = 2^{\mathcal{Y}}$. The task of multi-label learning is to induce a multi-label classifier $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ that minimizes the following classification risk:

$$R(f) = \mathbb{E}_{p(\boldsymbol{x},Y)} \left[\mathcal{L} \left(f \left(\boldsymbol{x} \right), Y \right) \right], \tag{1}$$

where $\mathcal{L} : \mathbb{R}^q \times 2^{\mathcal{Y}} \mapsto \mathbb{R}_+$ is a multi-label loss function that measures how well a classifier estimates a given label. Note that a method is risk-consistent [25] if the method possesses a classification risk estimator that is equivalent to $R(f)$ given the same classifier f.

3.2 Single-Positive Multi-Label Learning

Given the SPMLL training set $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \gamma_i) | 1 \leq i \leq n\}$ where $\gamma_i \in \mathcal{Y}$ denotes the observed single-positive label of $\boldsymbol{x}_i$. Note that $\gamma_i \in Y_i$ while its relevant label set Y_i is not directly accessible to the learning algorithms. For each SPMLL training example $(\boldsymbol{x}_i, \gamma_i)$, we use the observed single-positive vector $\boldsymbol{l}_i = [l_i^1, l_i^2, \ldots, l_i^c]^\top \in \{0, 1\}^c$ to represent whether j-th label is the observed positive label, i.e., $l_i^j = 1$ if $j = \gamma_i$, otherwise $l_i^j = 0$. The multi-label vector is denoted by $\boldsymbol{y}_i = [y_i^1, y_i^2, \ldots, y_i^c]^\top \in \{0, 1\}^c$ if the j-th label is relevant to $\boldsymbol{x}_i$ and $y_j = 0$ if the j-th label is irrelevant. The task of SPMLL is to induce a multi-label classifier $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from $\widetilde{\mathcal{D}}$, which can assign a set of relevant label set for the unseen instance.

Recent work [5] empirically validates that SPMLL would reduce the amount of supervision with a tolerable damage in classification performance. The intuitive solution AN is assuming that unobserved labels are negative, which leads to the drawback that introduces some number of false negative labels. Therefore, the SOTA approaches [5] aim to reduce the damaging effects of false negative labels via employing the learning power of DNNs to achieve good performance in practice. However, there is no method can provide theoretical insights.

4 The Proposed Method

4.1 Risk-Consistent Estimator

To deal with single-positive multi-label learning, the classification risk $R(f)$ in Eq. (1) could be rewritten as

$$\begin{aligned}
&\mathbb{E}_{p(\boldsymbol{x},Y)}\left[\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)\right]\\
=&\int_{\boldsymbol{x}}\sum_{Y\in\mathcal{C}}\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)p(Y|\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\\
=&\int_{\boldsymbol{x}}\sum_{\gamma\in\mathcal{Y}}\sum_{Y\in\mathcal{C}}\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)\frac{p(Y|\boldsymbol{x})}{p(y^{\gamma}=1|\boldsymbol{x})c}p(y^{\gamma}=1|\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\\
=&\mathbb{E}_{p(\boldsymbol{x},\gamma)}\left[\frac{1}{p(y^{\gamma}=1|\boldsymbol{x})c}\sum_{Y\in\mathcal{C}}\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)p(Y|\boldsymbol{x})\right]\\
=&R_{sp}(f).
\end{aligned}\tag{2}$$

Additionally, we employ the widely used loss function in multi-label learning, i.e, binary cross-entropy loss, as the loss function $\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)$:

$$\begin{aligned}
\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)&=\sum_{j\in Y}\log f_j(\boldsymbol{x})+\sum_{j\notin Y}\log\left(1-f_j(\boldsymbol{x})\right)\\
&=\sum_{j\in Y}\ell^j+\sum_{j\notin Y}\bar{\ell}^j,
\end{aligned}\tag{3}$$

where $\ell^j=\log f_j(\boldsymbol{x})$ and $\bar{\ell}^j=1-f_j(\boldsymbol{x})$. Then, $\sum_{Y\in\mathcal{C}}\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)p(Y|\boldsymbol{x})$ in Eq. (2) could be calculated as[1]

$$\sum_{Y\in\mathcal{C}}\mathcal{L}\left(f\left(\boldsymbol{x}\right),Y\right)p(Y|\boldsymbol{x})=\sum_{j=1}^{c}d^j\ell^j+\left(1-d^j\right)\bar{\ell}^j.\tag{4}$$

Here, $d^j=p(y^j=1|\boldsymbol{x})\in[0,1]$ would be regarded as the soft label corresponding to class j for $\boldsymbol{x}$. By substituting Eq. (4) into Eq. (2), we obtain the following risk-consistent estimator

$$R_{sp}(f)=\mathbb{E}_{p(\boldsymbol{x},\gamma)}\left[\frac{1}{p(y^{\gamma}=1|\boldsymbol{x})c}\sum_{j=1}^{c}d^j\ell^j+(1-d^j)\bar{\ell}^j\right].\tag{5}$$

In this way, the empirical risk estimator for SPMLL can be expressed as

$$\widehat{R}_{sp}(f)=\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{p(y^{\gamma_i}=1|\boldsymbol{x}_i)c}\sum_{j=1}^{c}d_i^j\ell_i^j+\left(1-d_i^j\right)\bar{\ell}_i^j\right).\tag{6}$$

Then, we could design a benchmark solution via applying the sigmoid function on the model output $f(\boldsymbol{x}_i)$ to approximate $p(y^{\gamma_i}=1|\boldsymbol{x}_i)$ and recovering the soft label d_i^j for each training example in the label enhancement process in the following subsection.

4.2 The SMILE Approach

To recover the soft label vector $\boldsymbol{d}_i=[d_i^1,d_i^2,\ldots,d_i^c]^{\top}\in[0,1]^c$, we treat $\boldsymbol{d}_i$ as a latent vector. We assume that the prior density $p(\boldsymbol{d})$ is a Beta density with the minor values $\hat{\boldsymbol{\alpha}}=[\hat{\alpha}^1,\hat{\alpha}^2,\ldots,\hat{\alpha}^c]$ and $\hat{\boldsymbol{\beta}}=[\hat{\beta}^1,\hat{\beta}^2,\ldots,\hat{\beta}^c]$, i.e., $p\left(\boldsymbol{d}\right)=\prod_{j=1}^{c}\text{Beta}\left(d^j\mid\hat{\alpha}^j,\hat{\beta}^j\right)$. Then the prior density $p(\mathbf{D})$ of the soft label matrix $\mathbf{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_n]$ could be the product of each $p\left(\boldsymbol{d}\right)$.

Let features matrix $\mathbf{\Phi}=[\boldsymbol{\phi}_1,\boldsymbol{\phi}_2,\ldots,\boldsymbol{\phi}_n]$, adjacency matrix $\mathbf{A}$ and logical labels $\mathbf{L}=[\boldsymbol{l}_1,\boldsymbol{l}_2,\ldots,\boldsymbol{l}_n]$ be observed matrix, SMILE aims to infer the posterior density $p(\mathbf{D}|\mathbf{L},\mathbf{\Phi},\mathbf{A})$ via considering the topological information of the feature space. Here the feature vector $\boldsymbol{\phi}_i$ of each example could be

[1] More detailed calculations can be seen in Appendix A.1.

extracted from the predictive model $\boldsymbol{\theta}$ in current epoch and the adjacency matrix $\mathbf{A} = [a_{ij}]_{n\times n}$ can be obtained by

$$a_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{\phi}_i \in \mathcal{N}(\boldsymbol{\phi}_j) \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where $\mathcal{N}(\boldsymbol{\phi}_j)$ is the set for k-nearest neighbors of $\boldsymbol{\phi}_j$ and the diagonal elements of $\mathbf{A}$ are set to 1.

As the computation of the exact posterior density $p(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ is intractable, a fixed-form density $q(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ is employed to approximate the true posterior. We let the approximate posterior be the product of each Beta parameterized by $\boldsymbol{\alpha}_i = [\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^c]^\top$ and $\boldsymbol{\beta}_i = [\beta_i^1, \beta_i^2, \ldots, \beta_i^c]^\top$:

$$q_{\boldsymbol{w}}(\mathbf{D} \mid \mathbf{L}, \boldsymbol{\Phi}, \mathbf{A}) = \prod_{i=1}^{n}\prod_{j=1}^{c} \text{Beta}\left(d_i^j | \alpha_i^j, \beta_i^j\right). \tag{8}$$

Here, the parameters $\boldsymbol{\Delta} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_n]$ and $\boldsymbol{\Xi} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_n]$ are outputs of the inference model parameterized by $\boldsymbol{w}$, which is defined as a two-layer GCN [19] with adjacency matrix by $\mathbf{A}$.

By following the Variational Bayes techniques [18], the evidence lower bound (ELBO) on the marginal likelihood of the model is derived which ensures that $q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ is as close as possible to $p(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L},\boldsymbol{\Phi},\mathbf{A})}[\log p(\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A}|\mathbf{D})] - \text{KL}[q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})||p(\mathbf{D})]. \tag{9}$$

As the first part of Eq. (9) is intractable, we employ the implicit reparameterization trick [10] to approximate it by Monte Carlo (MC) estimation. Inspired by [19], we simply drop the dependence on $\boldsymbol{\Phi}$:

$$\begin{aligned} p(\mathbf{L} \mid \mathbf{A}, \mathbf{D}) &= \prod_{i=1}^{n} p\left(\boldsymbol{l}_i \mid \mathbf{A}, \mathbf{D}\right), \\ p(\mathbf{A} \mid \mathbf{D}) &= \prod_{i=1}^{n}\prod_{j=1}^{n} p\left(a_{ij} \mid \boldsymbol{d}_i, \boldsymbol{d}_j\right), \end{aligned} \tag{10}$$

where $p\left(a_{ij} = 1 \mid \boldsymbol{d}_i, \boldsymbol{d}_j\right) = s\left(\boldsymbol{d}_i^\top \boldsymbol{d}_j\right)$ and $s(\cdot)$ is the logistic sigmoid function. We further assume that $p\left(\boldsymbol{l}_i|\mathbf{A}, \mathbf{D}\right)$ is a multivariate Bernoulli with probabilities $\boldsymbol{\tau}_i$. In order to simplify the observation model, $\mathbf{T}^{(m)} = [\boldsymbol{\tau}_1^j(m), \boldsymbol{\tau}_2^{(m)}, \ldots, \boldsymbol{\tau}_n^{(m)}]$ is computed from m-th sampling $\mathbf{D}^{(m)}$ with a three-layer MLP parameterized by $\boldsymbol{\eta}$. Then the first part of Eq. (9) can be tractable as

$$\begin{aligned} \mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L},\boldsymbol{\Phi},\mathbf{A})}[\log p(\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A}|\mathbf{D})] = &\operatorname{tr}\left(\mathbf{L}^\top \log \mathbf{T}^{(m)}\right) - \|\mathbf{A} - S\left(\mathbf{D}^{(m)}\mathbf{D}^{(m)\top}\right)\|_F^2 \\ &+ \frac{1}{M}\sum_{m=1}^{M} \operatorname{tr}\left((\mathbf{I} - \mathbf{L})^\top \log\left(\mathbf{I} - \mathbf{T}^{(m)}\right)\right) \end{aligned} \tag{11}$$

Note that we can use only one MC sample in Eq. (11) during the training process as suggested in [18, 37, 36].

The second part of Eq. (9) can be analytically calculated as

$$\begin{aligned} \text{KL}\left(q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})\|p(\mathbf{D})\right) = &\sum_{i=1}^{n}\sum_{j=1}^{c} \log\frac{\Gamma(\alpha_i^j + \beta_i^j)}{\Gamma(\alpha_i^j)\Gamma(\beta_i^j)} + (\alpha_i^j - \hat{\alpha}_i^j)\psi(\alpha_i^j) \\ &-(\alpha_i^j - \hat{\alpha}_i^j + \beta_i^j - \hat{\beta}_i^j)\psi(\alpha_i^j + \beta_i^j) + (\beta_i^j - \hat{\beta}_i^j)\psi(\beta_i^j). \end{aligned} \tag{12}$$

Here, $\Gamma(\cdot)$ and $\psi(\cdot)$ are Gamma function and Digamma function, respectively.

In addition, SMILE improves the label enhancement by enforcing that the recovered soft label should not be completely different from the confidence estimated by current prediction $f_j(\boldsymbol{x}_i; \boldsymbol{\theta})$ with sigmoid function:

$$T_C = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} f_j(\boldsymbol{x}_i)\log d_i^j + (1 - f_j(\boldsymbol{x}_i))\left(1 - \log d_i^j\right) \tag{13}$$

Algorithm 1 SMILE Algorithm

Input: The SPMLL training set $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \gamma_i)\}_{i=1}^n$, epoch T and iteration I;

```
Initialize the predictive model θ by warm-up training via using AN solution, the reference model w and observation model η;
Extract the features Φ from predictive model θ and calculate the adjacency matrix A;
for t = 1, ..., T do
    Shuffle training set D̃ = {(x_i, γ_i)}_{i=1}^n into I mini-batches;
    for k = 1, ..., I do
        Update w and η by forward computation and back-propagation by Eq. (14);
        Obtain the soft label d_i for each example x_i by Eq. (8);
        Apply the sigmoid function on the model output f(x_i) to approximate p(y^{γ_i} = 1|x_i);
        Update θ by forward computation and back-propagation by Eq. (6);
    end for
end for
```

Output: The predictive model $\boldsymbol{\theta}$.

Then we can easily get the objective of label enhancement T_{LE} as follows

$$T_{LE} = -\lambda \mathcal{L}_{ELBO} + T_C \tag{14}$$

where λ is a hyper-parameter. Note that the implicit reparameterization gradient [10] is employed, which avoids the inversion of the standardization function, which makes the gradients can be computed analytically in backward pass.

SMILE first initializes the predictive network by warm-up training with AN solution, which would attain a reasonable predictive network before it starts fitting noise [41]. Then we could sample the label matrix $\mathbf{D}$ from $q(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ after label enhancement and apply the sigmoid function on the model output $f(\boldsymbol{x}_i)$ to approximate $p(y^{\gamma_i} = 1|\boldsymbol{x}_i)$ to make the empirical risk estimator in Eq. (6) accessible, and would train the predictive model $\boldsymbol{\theta}$ by minimizing the risk estimator. Note that SMILE implements label enhancement and classifier training alternately in every epoch. The algorithmic description of SMILE is shown in Algorithm 1.

4.3 Estimation Error Bound

In this subsection, we establish an estimation error bound of Eq.(6) to demonstrate its learning consistency. The empirical risk estimator according to Eq.(6) can be rewritten as:

$$\widehat{R}_{sp}(f) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{L} \left(w_i^j \ell_i^j + \bar{w}_i^j \bar{\ell}_i^j \right), \tag{15}$$

where $w_i^j = \frac{d_i^j}{p(y^\gamma = 1|\boldsymbol{x}_i)c}$ and $\bar{w}_i^j = \frac{1-d_i^j}{p(y^\gamma = 1|\boldsymbol{x}_i)c}$. Then the loss function $\mathcal{L}_{sp}$ is

$$\mathcal{L}_{sp} = \sum_{j=1}^{L} \left(w_i^j \ell_i^j + \bar{w}_i^j \bar{\ell}_i^j \right). \tag{16}$$

We define a function space as:

$$\mathcal{G}_{sp} = \left\{ (\boldsymbol{x}, y) \mapsto \sum_{j=1}^{L} \left(w^j \ell^j + \bar{w}^j \bar{\ell}^j \right) | f \in \mathcal{F} \right\}. \tag{17}$$

Let $\widetilde{\mathfrak{R}}_n(\mathcal{G}_{sp})$ be the expected Rademacher complexity [1] of $\mathcal{G}_{sp}$, i.e.,

$$\widetilde{\mathfrak{R}}_n(\mathcal{G}_{sp}) = \mathbb{E}_{\boldsymbol{x}, y, \boldsymbol{\sigma}} \left[\sup_{g \in \mathcal{G}_{sp}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i g\left(\boldsymbol{x}_i, y_i\right) \right], \tag{18}$$

where $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ is n Rademacher variables with σ_i independently uniform variable taking value in $\{+1, -1\}$. Then we have

Table 1: Predictive performance of each comparing approach (mean±std) in terms of *Average precision* ↑. The best performance (the larger the better) is shown in bold face.

Datasets	SMILE	AN	AN-LS	WAN	ROLE	GLOCAL	MLML	D2ML
CAL500	**0.409±0.024**	0.381±0.012	0.308±0.014	0.402±0.007	0.399±0.006	0.227±0.002	0.233±0.000	0.223±0.001
image	**0.773±0.004**	0.743±0.039	0.666±0.043	0.756±0.011	0.703±0.042	0.771±0.003	0.652±0.001	0.274±0.003
scene	**0.827±0.003**	0.760±0.057	0.744±0.030	0.789±0.025	0.771±0.019	0.825±0.001	0.814±0.000	0.285±0.002
yeast	0.744±0.007	**0.751±0.004**	0.746±0.002	0.729±0.006	0.714±0.010	0.646±0.002	0.456±0.002	0.323±0.001
corel5k	**0.308±0.003**	0.301±0.002	0.280±0.004	0.304±0.001	0.267±0.002	0.218±0.001	0.072±0.001	0.028±0.001
rcv1-s1	**0.598±0.003**	0.593±0.005	0.562±0.003	0.594±0.003	0.580±0.005	0.229±0.000	0.221±0.003	0.053±0.001
corel16k-s1	**0.347±0.001**	0.345±0.002	0.322±0.001	0.346±0.001	0.311±0.002	0.029±0.001	0.081±0.001	0.029±0.004
delicious	**0.334±0.002**	0.328±0.001	0.265±0.006	0.333±0.003	0.240±0.004	0.027±0.001	0.086±0.001	0.028±0.001
iaprtc12	**0.313±0.005**	0.268±0.003	0.273±0.002	0.310±0.002	0.258±0.005	0.035±0.002	0.126±0.001	0.026±0.001
espgame	**0.237±0.006**	0.223±0.002	0.206±0.003	0.215±0.002	0.227±0.004	0.038±0.000	0.086±0.002	0.038±0.001
mirflickr	**0.627±0.004**	0.609±0.014	0.605±0.004	0.604±0.009	0.547±0.035	0.476±0.000	0.253±0.003	0.132±0.002
tmc2007	**0.824±0.001**	0.822±0.001	0.807±0.001	0.818±0.002	0.803±0.001	0.649±0.000	0.415±0.000	0.161±0.001

Lemma 1 *Suppose the loss function $\mathcal{L}_{sp}$ are bounded by M, i.e., $M = \sup_{\boldsymbol{x}\in\mathcal{X}, f\in\mathcal{F}, y\in\mathcal{Y}} \mathcal{L}_{sp}\left(f(\boldsymbol{x}), y\right)$, then for any $\delta > 0$, with probability at least $1-\delta$,*

$$\sup_{f\in\mathcal{F}} \left| R_{sp}(f) - \widehat{R}_{sp}(f) \right| \leq 2\widetilde{\mathfrak{R}}_n(\mathcal{G}_{sp}) + \frac{M}{2}\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

The proof of Lemma 1 is provided in Appendix A.2.

Lemma 2 *Assume the loss function $\ell\left(f(\boldsymbol{x}), y\right)$ and $\bar{\ell}\left(f(\boldsymbol{x}), y\right)$ are ρ^+-Lipschitz and ρ^--Lipschitz with respect to $f(\boldsymbol{x})$ $(0 < \rho^+ < \infty$ and $0 < \rho^- < \infty)$ for all $y \in \mathcal{Y}$, and w^j and $\bar{w}^j$ are both bounded in $[0, \pi]$. Then, the following inequality holds:*

$$\widetilde{\mathfrak{R}}_n(\mathcal{G}_{sp}) \leq \sqrt{2}\pi c(\rho^+ + \rho^-) \sum_{j=1}^{c} \mathfrak{R}_n(\mathcal{H}_{y_j}),$$

where $\mathcal{H}_y = \{h : \boldsymbol{x} \mapsto f_y(\boldsymbol{x}) | f \in \mathcal{F}\}$ and $\mathfrak{R}_n(\mathcal{H}_y) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_y} \frac{1}{n}\sum_{i=1}^{n} h\left(\boldsymbol{x}_i\right)\right]$.

The proof of Lemma 2 is provided in Appendix A.3.

Let $\widehat{f}_{\mathrm{sp}} = \min_{f\in\mathcal{F}} \widehat{R}_{\mathrm{sp}}(f)$ be lthe empirical risk minimizer and $f^\star = \min_{f\in\mathcal{F}} R(f)$ be the true risk minimizer. Based one Lemma 1 and 2, we have the following theorem:

Theorem 1 *Assume the loss function $\ell\left(f(\boldsymbol{x}), y\right)$ and $\bar{\ell}\left(f(\boldsymbol{x}), y\right)$ are ρ^+-Lipschitz and ρ^--Lipschitz with respect to $f(\boldsymbol{x})$ $(0 < \rho^+ < \infty$ and $0 < \rho^- < \infty)$ for all $y \in \mathcal{Y}$ and the loss function $\mathcal{L}_{sp}$ are bounded by M, i.e., $M = \sup_{\boldsymbol{x}\in\mathcal{X}, f\in\mathcal{F}, y\in\mathcal{Y}} \mathcal{L}_{sp}\left(f(\boldsymbol{x}), y\right)$, with probability at least $1-\delta$,*

$$R(\widehat{f}_{sp}) - R(f^*) \leq 4\sqrt{2}\pi c(\rho^+ + \rho^-) \sum_{j=1}^{c} \mathfrak{R}_n(\mathcal{H}_y) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

The proof is provided in Appendix A.4. Theorem 1 shows that the empirical risk minimizer f_{sp} converges to the true risk minimizer $f^\star$ as $n \to \infty$ and $\mathfrak{R}_n\left(\mathcal{H}_y\right) \to 0$ for all parametric models with a bounded norm.

5 Experiments

5.1 Experimental Configurations

We adopt twelve widely-used MLL datasets [13] in the experiments. 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. To evaluate the performance on SPMLL, we generate the single positive training data by randomly selecting one positive label to keep for each training example in the MLL datasets. For each dataset, we run the comparing methods with 80%/10%/10% train/validation/test split. The validation and test sets are always fully labeled. The detailed descriptions of these datasets are

Table 2: Predictive performance of each comparing approach (mean±std) in terms of *One-error* ↓. The best performance (the smaller the better) is shown in bold face.

Datasets	SMILE	AN	AN-LS	WAN	ROLE	GLOCAL	MLML	D2ML
CAL500	**0.313±0.068**	0.492±0.068	0.500±0.087	0.513±0.065	0.383±0.129	0.843±0.011	0.839±0.032	0.833±0.003
image	0.415±0.045	0.422±0.062	0.527±0.060	0.427±0.043	0.493±0.060	0.365±0.012	**0.200±0.023**	0.600±0.019
scene	0.358±0.066	0.408±0.093	0.420±0.038	0.366±0.049	0.384±0.030	0.286±0.024	**0.167±0.011**	0.667±0.023
yeast	**0.221±0.018**	0.238±0.009	0.241±0.002	0.244±0.003	0.258±0.010	0.276±0.032	0.285±0.003	0.500±0.022
corel5k	**0.635±0.005**	0.635±0.005	0.670±0.013	0.684±0.007	0.661±0.008	0.764±0.011	0.947±0.005	0.987±0.003
rcv1-s1	**0.450±0.009**	0.465±0.014	0.481±0.012	0.491±0.013	0.457±0.012	0.810±0.029	0.782±0.002	0.941±0.004
corel16k-s1	0.639±0.004	0.643±0.005	0.655±0.002	**0.637±0.003**	0.655±0.001	0.989±0.003	0.830±0.001	0.987±0.003
delicious	**0.394±0.007**	0.416±0.008	0.438±0.010	0.411±0.014	0.458±0.008	0.996±0.003	0.804±0.011	0.967±0.003
iaprtc12	**0.578±0.009**	0.641±0.013	0.580±0.005	0.590±0.008	0.602±0.017	0.997±0.000	0.605±0.012	0.897±0.006
espgame	**0.692±0.012**	0.718±0.008	0.710±0.004	0.747±0.005	0.705±0.009	0.995±0.000	0.699±0.003	0.734±0.004
mirflickr	**0.325±0.011**	0.364±0.020	0.355±0.007	0.408±0.020	0.501±0.084	0.670±0.054	0.447±0.011	0.816±0.007
tmc2007	**0.204±0.003**	0.206±0.002	0.217±0.002	0.217±0.005	0.220±0.002	0.313±0.001	0.227±0.002	0.409±0.008

Table 3: Summary of the Wilcoxon signed-ranks test for SMILE against other comparing approaches at 0.05 significance level. The p-values are shown in the brackets.

SMILE against	AN	AN-LS	WAN	ROLE	GLOCAL	MLML	D2ML
Average precision	**win** [0.0068]	**win** [0.0010]	**win**[0.0005]	**win** [0.0005]	**win** [0.0005]	**win** [0.0005]	**win** [0.0005]
One-error	**win** [0.0033]	**win** [0.0005]	**win** [0.0010]	**win** [0.0005]	**win** [0.0034]	**tie** [0.0640]	**win** [0.0005]
Ranking loss	**win** [0.0066]	**win** [0.0005]	**win** [0.0163]	**win** [0.0005]	**win** [0.0005]	**win** [0.0015]	**win** [0.0005]
Hamming loss	**win** [0.0160]	**win** [0.0180]	**win** [0.0068]	**win** [0.0273]	**tie** [0.1282]	**win** [0.0180]	**win** [0.0051]
Coverage	**win** [0.0207]	**win** [0.0005]	**win** [0.0164]	**win** [0.0005]	**win** [0.0005]	**win** [0.0024]	**win** [0.0010]

provided in Appendix A.5. Five popular multi-label metrics *Ranking loss*, *Hamming loss*, *One-error*, *Coverage*, and *Average precision* [44] are employed for performance evaluation. 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.

In this paper, SMILE is compared against four well-established SPMLL approaches including 1) AN [5] which assumes unobserved labels are negative and employs binary entropy loss for the training examples with the modified labels, 2) AN-LS [5] which assumes unobserved labels are negative and employs label smoothing [28] to reduce the impact of the incorrect labels (i.e. those labels incorrectly assumed to be negative), 3) WAN [5] which reduces the impact of false negatives by employing the down-weight terms in the loss corresponding to negative labels, and 4) ROLE [5] which online estimates of the unobserved labels throughout training and encourages the classifier predictions to match the estimated labels via binary entropy loss.

In addition, three learning approaches for MLL with missing labels (also termed as MLL with partial labels) are adopted as compared approaches, as SPMLL could be regarded as a hardest version of MLL with missing labels including 1) GLOCAL [45] which exploits global and local label correlations simultaneously, through learning a latent label representation and optimizing label manifolds for the missing label cases, 2) MLML [34] which recovers the full label assignment for each sample by enforcing consistency with available label assignments and smoothness of labels, and 3) D2ML [24] which utilizes both local low-rank label structures and label discriminant information for learn from missing labels.

For all the DNN based approaches (AN, AN-LS, WAN, ROLE and SMILE), we adopt three-layer MLP as the predictive model for fair comparisons and use the Adam optimizer [17] with the mini-batch size set to 16 and the number of epochs set to 25. The learning rate and weight decay are selected from $\{10^{-4}, 10^{-3}, 10^{-2}\}$ to maximize the accuracy on a validation set and other hyper-parameters for all the comparing methods are also selected based on the validation. All the comparing methods run 5 trials (with 80%/10%/10% train/validation/test split) on each datasets.

5.2 Experimental Results

Tables 1 and 2 report the results of all approaches on *Average precision* and *One-error*, respectively. For each evaluation metric, "↑" indicates "the smaller the better" while "↓" indicates "the larger the better". The results on other metrics are similar and could be seen in Appendix A.5. In addition, Wilcoxon signed-ranks test [6] is employed to show whether SMILE has a significant performance than other comparing approaches. Table 3 reports the p-values for the corresponding tests and the statistical test results at 0.05 significance level.

Table 4: Predictive performance of SMILE and its variant (mean±std) in terms of *Average Precision*, *One-error*, and *Ranking loss*.

Datasets	*Average precision*↑		*One-error*↓		*Ranking loss*↓	
	SMILE	SMILE-SI	SMILE	SMILE-SI	SMILE	SMILE-SI
CAL500	**0.409±0.024**	0.406±0.013	**0.313±0.068**	0.396±0.037	0.226±0.026	**0.223±0.017**
image	**0.773±0.004**	0.732±0.003	**0.415±0.045**	0.430±0.049	**0.161±0.009**	0.193±0.005
scene	**0.827±0.003**	0.730±0.002	**0.358±0.066**	0.471±0.053	**0.099±0.041**	0.144±0.026
yeast	**0.744±0.007**	0.461±0.002	**0.221±0.018**	0.513±0.023	**0.163±0.002**	0.531±0.024
corel5k	**0.308±0.003**	0.277±0.009	**0.635±0.005**	0.710±0.014	**0.109±0.004**	0.111±0.002
rcv1-s1	**0.598±0.003**	0.527±0.007	**0.450±0.009**	0.522±0.025	**0.044±0.001**	0.074±0.001
corel16k-s1	**0.347±0.001**	0.337±0.002	**0.639±0.004**	0.657±0.007	**0.132±0.007**	0.136±0.002
delicious	**0.334±0.002**	0.263±0.005	**0.394±0.007**	0.533±0.005	**0.121±0.007**	0.533±0.003
iaprtc12	0.313±0.005	**0.322±0.007**	0.578±0.009	**0.535±0.011**	**0.129±0.001**	0.131±0.001
espgame	0.237±0.006	**0.246±0.004**	0.692±0.012	**0.682±0.016**	**0.160±0.003**	0.162±0.002
mirflickr	**0.627±0.004**	0.623±0.001	**0.325±0.011**	0.388±0.019	0.125±0.001	**0.122±0.001**
tmc2007	**0.824±0.001**	0.788±0.002	**0.204±0.003**	0.291±0.005	**0.043±0.001**	0.046±0.001

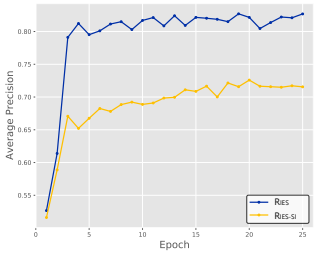

Figure 2: *Average precision*.

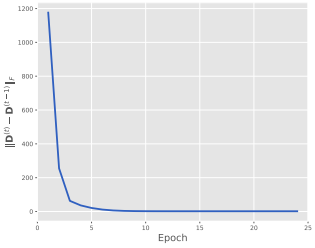

Figure 3: Convergence of $\mathbf{D}$.

Table 5: Wilcoxon signed ranks test(at 0.05 significance level).

Evaluation metric	SMILE against SMILE-SI	
	performance	p-value
Average precision	**win**	0.0092
One-error	**win**	0.0049
Ranking loss	**win**	0.0210
Hamming loss	**win**	0.0051
Coverage	**win**	0.0122

Across all evaluation metrics, SMILE achieves the best performance in 81.67% cases over all the 12 datasets on all evaluation metrics. Table 3 shows that SMILE achieves superior performance against all the comparing approaches on all evaluation metrics (except on *Hamming loss* where SMILE achieves comparable performance against GLOCAL and *One-error* where SMILE achieves comparable performance against MLML). The superior performance of SMILE provides a strong evidence for the effectiveness of risk-consistent estimator for SPMLL. Tables 1 and 2 show that the performance advantage of SMILE over comparing approaches is stable under varying the number of class labels. In summary, these experimental results clearly validate the effectiveness of SMILE.

5.3 Further Analysis

To show the helpfulness of label enhancement to SMILE, a vanilla variant about SMILE (named as SMILE-SI) is adopted. Here, label enhancement is replaced by approximating d_i^j with the confidence of current model $f_j(\boldsymbol{x}_i)$, which is a widely-used technique [9, 22] to approximate the soft label in weakly supervised learning. Tables 4 reports detailed experimental results in terms of *Average precision*, *One-error*, and *Ranking loss*, respectively. The detailed experimental results in terms of other metrics are reported in Appendix A.5. Besides, the performance of each approach with the number of epoch on `delicious` is shown in Figure 2. Wilcoxon signed-ranks test [6] in Table 5 shows that SMILE achieves superior performance against SMILE-SI on all evaluation metrics, which clearly validates the usefulness of label enhancement. Figure 3 illustrates the recovered soft label matrix $\mathbf{D}$ over all training examples converges as the number of epoch on `delicious`. We can see that the recovered soft label matrix converges fast with the increasing number of epoch.

6 Conclusion

In this paper, the problem of single-positive multi-label learning is studied where a novel approach SMILE is proposed. We derive an unbiased risk estimator, which suggests that one positive label of each instance is sufficient to train predictive models for multi-label learning, and design a benchmark solution via recovering the soft label for each training example in the label enhancement process. The effectiveness of the proposed method is validated on benchmark datasets. If single-positive multi-label learning methods become very effective, the employment for data annotators might be decreased which could lead to a negative societal impact.

References

[1] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[2] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[3] Di Chen, Yexiang Xue, Shuo Chen, Daniel Fink, and Carla Gomes. Deep multi-species embedding. *arXiv preprint arXiv:1609.09353*, 2016.

[4] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

[5] Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 933–942, 2021.

[6] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[7] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 647–657, 2019.

[8] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14 (NIPS 2002)*, pages 681–687, Vancouver, British Columbia, Canada, 2002.

[9] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. *Advances in Neural Information Processing Systems*, 2020.

[10] Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. *Advances in Neural Information Processing Systems*, 2018.

[11] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[12] Andrew B. Goldberg, Xiaojin Zhu, Ben Recht, Jun-Ming Xu, and Robert D. Nowak. Transduction with matrix completion: Three birds with one stone. In *Advances in Neural Information Processing Systems 23*, pages 757–765. Curran Associates, Inc., 2010.

[13] Jun-Yi Hang and Min-Ling Zhang. Collaborative learning of label semantics and deep label-specific features for multi-label classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[14] Thomas Hartvigsen, Cansu Sen, Xiangnan Kong, and Elke Rundensteiner. Recurrent halting chain for early multi-label classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1382–1392, 2020.

[15] Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE transactions on knowledge and data engineering*, 28(12):3309–3323, 2016.

[16] Dat Huynh and Ehsan Elhamifar. Interactive multi-label cnn learning with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9423–9432, 2020.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, 2015, San Diego, CA*, 2015.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, Banff, AB, Canada, 2014.

[19] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[20] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16478–16488, 2021.

[21] Hung-Yi Lo, Ju-Chiang Wang, Hsin-Min Wang, and Shou-De Lin. Cost-sensitive multi-label learning for audio tag annotation and retrieval. *IEEE Transactions on Multimedia*, 13(3):518–529, 2011.

[22] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, pages 6500–6510. PMLR, 2020.

[23] Jianghong Ma and Tommy WS Chow. Topic-based instance and feature selection in multilabel classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[24] Zhongchen Ma and Songcan Chen. Expand globally, shrink locally: Discriminant multi-label learning with missing labels. *Pattern Recognition*, 111:107675, 2021.

[25] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[26] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333, 2011.

[27] Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine learning*, 88(1-2):157–208, 2012.

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[29] Pingjie Tang, Meng Jiang, Bryan Ning Xia, Jed W Pitera, Jeffrey Welser, and Nitesh V Chawla. Multi-label patent categorization with non-local attention-based graph convolutional network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9024–9031, 2020.

[30] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2006.

[31] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

[32] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089, 2011.

[33] Alexis Vallet and Hiroyasu Sakamoto. A multi-label convolutional neural network for automatic image annotation. *Journal of information processing*, 23(6):767–775, 2015.

[34] Baoyuan Wu, Zhilei Liu, Shangfei Wang, Bao-Gang Hu, and Qiang Ji. Multi-label learning with missing labels. In *2014 22nd International Conference on Pattern Recognition*, pages 1964–1968. IEEE, 2014.

[35] Bin Wu, Erheng Zhong, Andrew Horner, and Qiang Yang. Music emotion recognition by multi-label multi-layer multi-instance multi-view learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 117–126, 2014.

[36] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[37] Ning Xu, Jun Shu, Yun-Peng Liu, and Xin Geng. Variational label enhancement. In *Proceedings of the International Conference on Machine Learning*, pages 10597–10606, Vienna, Austria, 2020.

[38] Renchun You, Zhiyao Guo, Lei Cui, Xiang Long, Yingze Bao, and Shilei Wen. Cross-modality attention with semantic graph embedding for multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12709–12716, 2020.

[39] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *International conference on machine learning*, pages 593–601. PMLR, 2014.

[40] Ze-Bang Yu and Min-Ling Zhang. Multi-label classification with label-specific feature generation: A wrapped approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[41] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations*, Toulon, France.

[42] Jing Zhang and Xindong Wu. Multi-label inference for crowdsourcing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2738–2747, 2018.

[43] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[44] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.

[45] Yue Zhu, James T Kwok, and Zhi-Hua Zhou. Multi-label learning with global and local label correlation. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1081–1094, 2017.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]