End-to-End Probabilistic Label-Specific Feature Learning for Multi-Label Classification

Jun-Yi Hang[1,2], **Min-Ling Zhang**[1,2*], **Yanghe Feng**[3], **Xiaocheng Song**[4]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3]College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
[4]Department of Beijing Institute of Electronic Engineering, Beijing 100854, China
{hangjy, zhangml}@seu.edu.cn, fengyanghe@nudt.edu.cn, sxchitman@126.com

Abstract

Label-specific features serve as an effective strategy to learn from multi-label data with tailored features accounting for the distinct discriminative properties of each class label. Existing prototype-based label-specific feature transformation approaches work in a *three-stage framework*, where prototype acquisition, label-specific feature generation and classification model induction are performed independently. Intuitively, this separate framework is suboptimal due to its decoupling nature. In this paper, we make a first attempt towards a unified framework for prototype-based label-specific feature transformation, where the prototypes and the label-specific features are directly optimized for classification. To instantiate it, we propose modelling the prototypes probabilistically by the normalizing flows, which possess adaptive prototypical complexity to fully capture the underlying properties of each class label and allow for scalable stochastic optimization. Then, a label correlation regularized probabilistic latent metric space is constructed via jointly learning the prototypes and the metric-based label-specific features for classification. Comprehensive experiments on 14 benchmark data sets show that our approach outperforms the state-of-the-art counterparts.

Introduction

Multi-label classification deals with the problem where an instance can be associated with multiple labels simultaneously (Zhang and Zhou 2014; Liu et al. 2021). As a learning paradigm that handles objects with multiple semantics, researches on multi-label classification have been widely driven by real-world applications, such as multimedia annotation (You et al. 2020), text categorization (Tang et al. 2020), and bioinformatics analysis (Chen et al. 2017), etc.

The most straightforward strategy for tackling multi-label classification is to induce classification models with the identical representation of an instance. This strategy might be suboptimal as it fails to account for the distinct characteristics of each class label. To improve this, the strategy of label-specific features has been proposed to facilitate the discrimination of each class label by tailoring its own features (Zhang and Wu 2015; Huang et al. 2016; Zhang et al. 2018; Jia, Zhu, and Li 2020; Yu and Zhang 2021). With the basic assumption that the distinct characteristics of each class label can be captured via investigating the underlying properties of the training instances, the seminal work LIFT (Zhang and Wu 2015) proposes a three-stage framework to perform prototype-based label-specific feature transformation: firstly, clustering analysis is performed on positive/negative instances of each class label to obtain its positive/negative prototypes; then, label-specific features are generated via querying distances between the original instances and the prototypes; finally, classifiers are induced with the metric-based label-specific features. Numerous studies have been conducted to improve this three-stage framework by enhancing the process of prototype acquisition (Zhang, Fang, and Li 2015; Zhan and Zhang 2017; Zhang and Li 2021) and label-specific feature generation (Xu et al. 2016; Chen and Zhang 2019; Guo et al. 2019; Lin et al. 2021). However, the three stages still work independently, which might be suboptimal as it has no opportunity to optimize the prototypes and the label-specific features directly for classification.

A feasible way to improve this is to jointly learn the prototypes and the metric-based label-specific features in a latent metric space where an instance is close to prototypes

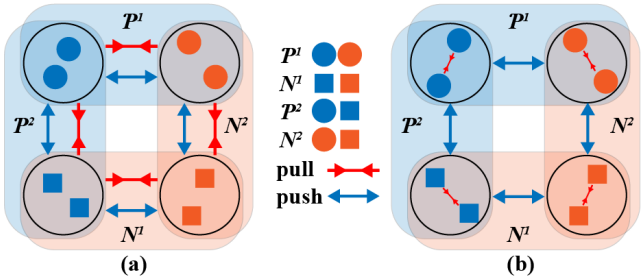


Figure 1: Intuitive illustration for the necessity of multi-prototype learning for compact latent metric space construction. For clarity, we consider a learning scenario with two labels. Label l_1 is denoted by shape (circle for positive and square for negative). Label l_2 is presented by color (blue for positive and orange for negative). (a) In single-prototype learning case, contradictory forces exist between pairwise clusters. For example, the clusters ● and ● are simultaneously pulled close and pushed away by label l_1 and label l_2 respectively. (b) In multi-prototype learning case, such contradictory forces are eliminated.

*Corresponding author

with the same category and away from prototypes with different category[1], thus highly discriminative features specific to each class label can be generated for improved classification. Due to the inherent multi-semantic properties of multi-label data, it is important to learn multiple prototypes for describing positive/negative instances of each class label when constructing such latent metric space. Intuitively, the necessity of *multi-prototype learning* for compact latent metric space construction is illustrated in Figure 1. Existing works acquire prototypes mainly by performing clustering analysis (Snell, Swersky, and Zemel 2017; Allen et al. 2019), conducting neighbor search (Liu and Tsang 2015; Rastin, Jahromi, and Taheri 2021), maintaining a prototypical memory (Zhen et al. 2020), or directly learning the parameters of prototypes (Shen et al. 2018). Nevertheless, these approaches may either fail to work when the optimal number of prototypes is unknown, or be incompatible with stochastic optimization methods, which makes the label-wise multi-prototype learning challenging.

With the above observations, we present a unified framework for prototype-based label-specific feature transformation, where prototypes and metric-based label-specific features are optimized directly for classification in an end-to-end manner. Following this framework, a novel approach named PACA, i.e. end-to-end **P**robabilistic l**A**bel-spe**C**ific fe**A**ture learning for multi-label classification, is proposed. Specifically, normalizing flows (Kobyzev, Prince, and Brubaker 2021) are exploited to conduct probabilistic modelling of prototypes, which can adaptively decide the prototypical complexity in terms of underlying properties of each class label and support stochastic optimization. Inspired by the variational inference theory, a probabilistic latent metric space is constructed via learning the prototypes and the metric-based label-specific features jointly. We further propose a label embedding-based regularizer to impose constraints on the structure of the latent metric space, which implicitly incorporates the label correlations into the prototype learning process. Comprehensive experiments on 14 benchmark data sets show that our approach performs better than well-established multi-label classification algorithms.

The rest of this paper is organized as follows. Section 2 briefly reviews related works. Section 3 presents details of the proposed PACA approach. Section 4 reports experimental results over a wide range of multi-label data sets. Section 5 concludes this paper.

Related Work

Multi-label classification has been studied extensively in the last decade (Zhang and Zhou 2014; Liu et al. 2021). Most approaches focus on modelling the label correlations to facilitate the learning process, since the output space is exponential in size to the number of class labels. In terms of the order of label correlations being considered, these approaches can be roughly grouped into three categories, namely *first-order* approaches (Boutell et al. 2004; Zhang and Zhou 2007), *second-order* approaches (Elisseeff and Weston 2001; Zhu, Kwok, and Zhou 2018) and *high-order* approaches (Tsoumakas, Katakis, and Vlahavas 2010; Feng, An, and He 2019; Xu and Guo 2021). Recently, deep learning has become a successful technique to jointly consider the label correlation exploitation and classification model induction. For example, the chain-like prediction process proposed in (Read et al. 2011) is made into a single pipeline via recurrent neural networks (Wang et al. 2016; Yazici et al. 2020) to better exploit the higher-order label dependencies for classification. Graph neural networks (Chen et al. 2019b, 2020) are employed to explicitly encode pairwise label correlations and impose constraints on the hypothesis space. Some embedding approaches (Yeh et al. 2017; Chen et al. 2019a; Bai, Kong, and Gomes 2020) resort to deep neural networks to embed and align features and labels in a latent space, where the label correlations are implicitly encoded.

Besides, label-specific features have been proven to be another effective strategy to improve multi-label classification via manipulating the input space. Generally speaking, label-specific features can be generated in two different manners, i.e. *label-specific feature selection* and *prototype-based label-specific feature transformation*.

Label-specific feature selection generates label-specific features via retaining a specific subset of the original features for each class label. As a representative work, LLSF (Huang et al. 2015, 2016) introduces the classical lasso regression for label-specific feature selection and considers pairwise label correlations to encourage feature-sharing between closely-related labels. Follow-up works extend this framework via incorporating regularized optimization into the feature selection process (Huang et al. 2018b), imposing non-sparse constraints over the selected feature subsets (Weng et al. 2020), or performing selection in an embedded feature space (Yu and Zhang 2021), etc. Under an embedded selection framework, the processes of feature selection and classification are inherently coupled. However, explicit representations of label-specific features are absent.

On the other hand, label-specific features can also be explicitly generated by treating the prototypes of each class label as the transformation bases. LIFT (Zhang and Wu 2015) proposes a three-stage framework for prototype-based label-specific feature transformation, where prototype acquisition, label-specific feature generation and classification model induction are performed successively. Several customized strategies have been proposed to enhance the three-stage framework, such as replacing the k-means clustering with spectral clustering (Zhang, Fang, and Li 2015) or clustering ensemble (Zhang and Li 2021; Zhan and Zhang 2017) to acquire more robust prototypes, removing redundant information with attribute reduction (Xu et al. 2016), enriching metric-based label-specific features with local neighbor information (Weng et al. 2018), global spatial topology information (Guo et al. 2019), or informative features from related class labels (Chen and Zhang 2019). However, each stage in the three-stage framework still works independently, with no guarantees on the optimality of the generated label-specific features for classification.

To improve this, a first attempt towards a unified framework for prototype-based label-specific feature transformation is presented in this paper. We will detail our approach in the next section.

[1]Two categories exist for each label, i.e. positive and negative.

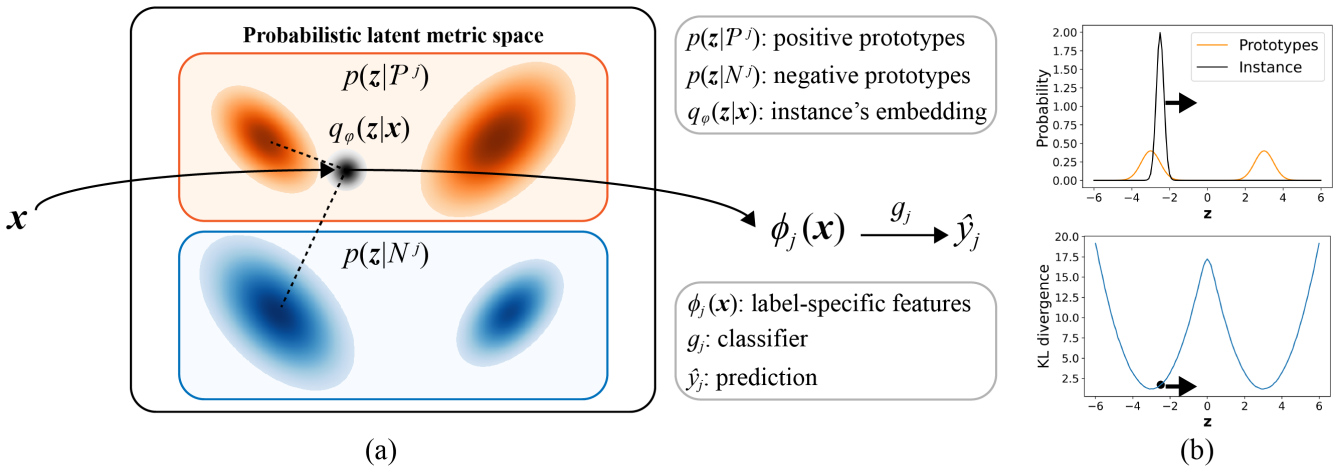


Figure 2: Illustration of the proposed PACA approach. Prototypes are modelled by normalizing flows, where each peak in the captured multimodal distribution is regarded as a prototype. (a) In a probabilistic latent metric space, label-specific features are generated by computing distances between an instance and positive, negative prototypes of each class label. Then, classification is performed on label-specific features in a label-wise manner. (b) The KL divergence is employed to measure an instance's distance to the set of postive/negative prototypes for each class label, which reflects the instance's distance to its nearest prototype in the set.

The PACA Approach

Preliminaries

Let $\mathcal{X} = \mathbb{R}^d$ denote the input space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the label space with q class labels. A multi-label example is denoted as $(\mathbf{x}, Y)$, where $\mathbf{x} \in \mathcal{X}$ is its feature vector and $Y \subseteq \mathcal{Y}$ is its set of relevant labels. Here, a q-dimensional vector $\mathbf{y} = [y_1, y_2, \ldots, y_q] \in \{0, 1\}^q$ is utilized to denote Y, where $y_k = 1$ indicates $l_k \in Y$ and $y_k = 0$ otherwise. Formally, multi-label classification aims to derive a multi-label prediction function $h : \mathcal{X} \to 2^{\mathcal{Y}}$ from a multi-label data set $\mathcal{D} = \{(\mathbf{x}_i, Y_i) | 1 \leq i \leq m\}$. Given an unseen instance $\mathbf{u} \in \mathcal{X}$, its associated label set is predicted as $h(\mathbf{u}) \subseteq \mathcal{Y}$.

Specifically, existing prototype-based label-specific feature transformation approaches learn from multi-label data under the following three-stage framework. Firstly, for each class label, clustering analysis is performed on the set of positive/negative instances and resulting cluster centers are regarded as positive/negative prototypes, i.e.:

$$\begin{aligned} clustering(\mathcal{P}^j, K^+) &\to \{\mathbf{p}_1^j, \mathbf{p}_2^j, \ldots, \mathbf{p}_{K^+}^j\} \\ clustering(\mathcal{N}^j, K^-) &\to \{\mathbf{n}_1^j, \mathbf{n}_2^j, \ldots, \mathbf{n}_{K^-}^j\} \end{aligned} \quad (1)$$

where $\mathcal{P}^j = \{\mathbf{x}_i | (\mathbf{x}_i, Y_i) \in \mathcal{D}, l_j \in Y_i\}$, K^+ and $\{\mathbf{p}_1^j, \mathbf{p}_2^j, \ldots, \mathbf{p}_{K^+}^j\}$ denote the set of positive instances, the number of positive clusters and derived positive prototypes for label l_j respectively. Similarly, the other symbols denote corresponding variables of negative instances.

Then, for each class label, label-specific features are generated by querying distances between the original instances and the corresponding prototypes, formalized as:

$$\phi_j(\mathbf{x}) = [d(\mathbf{x}, \mathbf{p}_1^j), \ldots, d(\mathbf{x}, \mathbf{p}_{K^+}^j), d(\mathbf{x}, \mathbf{n}_1^j), \ldots, d(\mathbf{x}, \mathbf{n}_{K^-}^j)] \quad (2)$$

where $d(\cdot, \cdot)$ denotes the distance metric, which is generally instantiated by the Euclidean distance.

Finally, classifiers $\{g_1, g_2, \ldots, g_q\}$ are induced with the generated label-specific features. Given an unseen instance $\mathbf{u}$, its associated label set is predicted as:

$$Y = \{l_j | g_j(\phi_j(\mathbf{u})) > 0, 1 \leq j \leq q\} \quad (3)$$

Overview

The illustration of our PACA[2] is shown in Figure 2. PACA's behavior in test stage is consistent with that of existing prototype-based label-specific feature transformation approaches. An instance $\mathbf{x}$ is firstly embedded into a probabilistic latent metric space. Then, label-specific features are generated via computing distances of the instance's latent embedding against positive and negative probabilistic prototypes for each class label. Finally, classification is performed on the generated label-specific features.

Formally, the generated label-specific features are as follows:

$$\phi_j(\mathbf{x}) = [d(q_\varphi(\mathbf{z}|\mathbf{x}), p(\mathbf{z}|\mathcal{P}^j)), d(q_\varphi(\mathbf{z}|\mathbf{x}), p(\mathbf{z}|\mathcal{N}^j))] \quad (4)$$

where $q_\varphi(\mathbf{z}|\mathbf{x})$ is the instance's latent embedding in probabilistic latent metric space, $p(\mathbf{z}|\mathcal{P}^j)$ and $p(\mathbf{z}|\mathcal{N}^j)$ denote positive and negative probabilistic prototypes of label l_j respectively. Distance metric $d(\cdot, \cdot)$ is implemented by KL divergence to measure the discrepancy between two distributions. Due to strong discriminative power of the generated label-specific features, softmax-based parameter-free classifiers are employed, formalized as:

$$\begin{aligned} g_j(\phi_j(\mathbf{x})) &= [\![softmax(-\phi_j(\mathbf{x}))_1 > 0.5]\!] \\ &= [\![\frac{\exp(-d(q_\varphi(\mathbf{z}|\mathbf{x}), p(\mathbf{z}|\mathcal{P}^j)))}{\sum_{c \in \{\mathcal{P}^j, \mathcal{N}^j\}} \exp(-d(q_\varphi(\mathbf{z}|\mathbf{x}), p(\mathbf{z}|c)))} > 0.5]\!] \end{aligned} \quad (5)$$

The learning process of PACA will be described in detail, where prototypes and metric-based label-specific features are optimized jointly to construct a probabilistic latent metric space for multi-label classification.

[2]Code package is publicly available at: http://palm.seu.edu.cn/zhangml/files/PACA.rar

Probabilistic Prototypes via Normalizing Flows

Probabilistic modelling of prototypes is found to be a more informative representation of object classes compared to deterministic vectors, where prototypes of each class are treated as a class distribution. However, existing works simply assume that the class distribution is a multivariate Gaussian (Zhang et al. 2019; Zhen et al. 2020), which is far from enough to approximate the class distribution of positive/negative instances of each class label in multi-label learning scenario. Recent work (Allen et al. 2019) resorts to the mixture of Gaussians to capture complex class distributions, but the number of mixture components has to be preassigned and the multimodal distribution is fitted via clustering which is inherently incompatible with stochastic optimization methods such as SGD. To overcome these problems, we propose employing the normalizing flows to adaptively model the potentially multimodal distribution of multi-label data.

Normalizing flows provide an elegant framework for modelling complex distribution via learning a diffeomorphism $f : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$, which can transform a random variable $\mathbf{u}$ following a known base distribution p_U into a new random variable $\mathbf{z}$. With the change of variables formula, the marginal likelihood of $\mathbf{z}$ is fully determined by:

$$p_Z(\mathbf{z}) = p_U(\mathbf{u}) \cdot |\det \frac{\partial f}{\partial \mathbf{z}}| = p_U(f(\mathbf{z})) \cdot |\det \frac{\partial f}{\partial \mathbf{z}}| \quad (6)$$

where $\det \frac{\partial f}{\partial \mathbf{z}}$ denotes the determinant of f's Jacobian matrix. Typically, the base distribution p_U is chosen to be a standard normal or a uniform distribution for fast density evaluation. As a universal approximator for continuous distributions, neural autoregressive flows (NAF) (Huang et al. 2018a) are utilized in this paper. In NAF, the diffeomorphism f is expressed as monotonic neural networks. Specifically, the transformation is conducted in an autoregressive manner, i.e.

$$u_t = \tau(z_t; c(\mathbf{z}_{1:t-1})) \quad (7)$$

where $c(\cdot)$ denotes an autoregressive conditioner to parameterize the element-wise transformation $\tau(\cdot)$. And $\tau(\cdot)$ is a monotonic neural network formalized as:

$$u_t = \sigma^{-1}(\mathbf{w}^{\mathrm{T}} \cdot \sigma(\mathbf{a} \cdot z_t + \mathbf{b})) \quad (8)$$

where $\mathbf{w}, \mathbf{a}, \mathbf{b} \in \mathbb{R}^{d_\tau}$, $0 < w_i < 1$, $\sum_{i=1}^{d_\tau} w_i = 1$, $\mathbf{a} > \mathbf{0}$. All these parameters are produced by the conditioner and constraints on them are enforced via activation functions, i.e. $\mathbf{w}, \mathbf{a}$ are outputs of softmax and softplus respectively. σ is the sigmoid function and σ^{-1} is the logit function. We set the number of hidden units $d_\tau = 16$ in this paper.

To model probabilistic prototypes of positive/negative instances for each class label respectively, $2 \cdot q$ such autoregressive transformations are required to learn. As this is impractical for large q, we propose learning 2 conditional transformations $f_{\mathcal{P}}(\cdot; l_j), f_{\mathcal{N}}(\cdot; l_j)$ by making the conditioner $c(\cdot)$ conditioned on the class labels. To implement this, a two-layer fully-connected neural network is utilized to instantiate the conditional conditioner $c(\cdot; l_j)$ and its input is augmented with the one-hot coding of each class label. Masking trick (Germain et al. 2015) is employed to parallelize the autoregressive computations of the conditioner, thus eliminating the need for sequential recursion. Formally, the probabilistic prototypes of positive/negative instances for label l_j are modelled as:

$$\begin{aligned} p(\mathbf{z}|\mathcal{P}^j) &= p_U(f_{\mathcal{P}}(\mathbf{z}; l_j)) \cdot |\det \frac{\partial f_{\mathcal{P}}(\mathbf{z}; l_j)}{\partial \mathbf{z}}| \\ p(\mathbf{z}|\mathcal{N}^j) &= p_U(f_{\mathcal{N}}(\mathbf{z}; l_j)) \cdot |\det \frac{\partial f_{\mathcal{N}}(\mathbf{z}; l_j)}{\partial \mathbf{z}}| \end{aligned} \quad (9)$$

where the base distribution p_U is a standard normal distribution. As shown in Figure 2, each peak in these captured multimodal distributions is regarded as a prototype, and one more benefit of the probabilistic prototypes is that the distance between an instance and the set of positive/negative prototypes for each class label can be efficiently measured by the KL divergence, which reflects the instance's distance to its nearest prototype in the set to some extent.

Probabilistic Latent Metric Space Construction

We construct the probabilistic latent metric space based on the probabilistic framework of the Variational Autoencoder (VAE) (Kingma and Welling 2014). VAE assumes a generative process for the observed data point $\mathbf{x}$, which involves an unobserved latent variable $\mathbf{z}$. The process consists of two steps: (1) sample a latent $\mathbf{z}$ from some prior distribution $p_\theta(\mathbf{z})$; (2) generate a data point $\mathbf{x}$ from some conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$. Accordingly, the joint probability $p_\theta(\mathbf{x}, \mathbf{z})$ can be factorized as:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z}) \cdot p_\theta(\mathbf{x}|\mathbf{z}) \quad (10)$$

Typically, this generative process is learned via maximizing the marginal likelihood on the observed data set. To make the optimization tractable, a variational lower bound on the marginal likelihood of data point $\mathbf{x}$ is induced via introducing a variational posterior $q_\varphi(\mathbf{z}|\mathbf{x})$ to approximate the true posterior $q_\theta(\mathbf{z}|\mathbf{x})$:

$$\begin{aligned} \log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{z}) \cdot p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z} \\ &= \log \int q_\varphi(\mathbf{z}|\mathbf{x}) \cdot \frac{p_\theta(\mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{x})} \cdot p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z} \\ &\geq \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\log \frac{p_\theta(\mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{x})} \cdot p_\theta(\mathbf{x}|\mathbf{z})] = \mathcal{L}(\mathbf{x};\theta,\varphi) \end{aligned} \quad (11)$$

where the $\mathcal{L}(\mathbf{x};\theta,\varphi)$ is the derived variational lower bound, which can be rewritten as:

$$\mathcal{L}(\mathbf{x};\theta,\varphi) = \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL[q_\varphi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})] \quad (12)$$

From the perspective of autoencoder, the first term encourages the reconstruction of data point $\mathbf{x}$, and the second term acts as a constraint on the structure of the latent space to ensure a reasonable new data point can be generated when sampling from the prior distribution $p_\theta(\mathbf{z})$. Correspondingly, the $q_\varphi(\mathbf{z}|\mathbf{x})$ can be regarded as a probabilistic encoder which transforms a data point $\mathbf{x}$ into a distribution over the latent variable $\mathbf{z}$ from which the data point $\mathbf{x}$ could have been generated. And the $p_\theta(\mathbf{x}|\mathbf{z})$ can be regarded as probabilistic decoder which recovers possible values of $\mathbf{x}$ given a latent $\mathbf{z}$.

We follow the structure of VAE. Specifically, $q_\varphi(\mathbf{z}|\mathbf{x})$ is assumed to be a multivariate Gaussian with a diagonal covariance, i.e. $q_\varphi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu(\mathbf{x}), \sigma^2(\mathbf{x})\mathbf{I})$, while $p_\theta(\mathbf{x}|\mathbf{z})$ is a multivariate Gaussian (in case of real-valued data) or Bernoulli (in case of binary data). Both $q_\varphi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ are parameterized by neural networks. Based on the variational lower bound, we replace the trivial prior distribution $p_\theta(\mathbf{z})$ with a more informative prior distribution conditioned on the category (i.e. positive or negative) for each class label. Thus, the new objective function is formalized as:

$$\mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \sum_{j=1}^{q} KL[q_\varphi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|y_j)] \quad (13)$$

where $p_\theta(\mathbf{z}|y_j)$ is the introduced conditional prior distribution, which equals to the probabilistic prototypes $p(\mathbf{z}|\mathcal{P}^j)$ when $y_j = 1$, and $p(\mathbf{z}|\mathcal{N}^j)$ otherwise.

In the above equation, the label-wise KL divergence terms encourage the probabilistic representation of an instance to match the probabilistic prototypes of the same category for each class label. However, it is not enough to construct a discriminative latent metric space by merely optimizing the distance between an instance and prototypes with the same category. Therefore, we propose to extend the objective function as follows:

$$\begin{aligned}\mathcal{L}_{latent} &= \mathcal{L}_{rec} + \mathcal{L}_{cls} \\ &= \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \\ &+ \sum_{j=1}^{q} \log \frac{\exp(-KL[q_\varphi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|y_j)])}{\sum_{c\in\{0,1\}} \exp(-KL[q_\varphi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|c)])}\end{aligned} \quad (14)$$

where an instance is encouraged to stay close to prototypes with the same category and keep away from prototypes with different category. We formalize this contrastive process as the parameter-free classification on the label-specific features $\phi_j(\mathbf{x})$, which can keep the training and prediction behaviors of our model consistent.

Latent Space Regularization

Intuitively, instances with the same label vectors should be close to each other in the latent space. To incorporate this intuition into the learning process, we propose a label embedding-based regularizer defined as:

$$\mathcal{L}_{reg} = \mathbb{E}_{q(\mathbf{z}|\mathbf{y})}[\log p(\mathbf{y}|\mathbf{z})] - KL[q_\varphi(\mathbf{z}|\mathbf{x})||q(\mathbf{z}|\mathbf{y})] \quad (15)$$

As shown in the above equation, another probabilistic autoencoder is utilized to embed the label vectors into the probabilistic latent space, where $q(\mathbf{z}|\mathbf{y})$ and $p(\mathbf{y}|\mathbf{z})$ are assumed to be a multivariate Gaussian with a diagonal covariance and a Bernoulli respectively. Then, KL divergence between the probabilistic representations of an instance's features and label vector encourages instances to cluster close to their labels.

From the perspective of prototype learning, this regularizer provides guidance for generating more semantic peaks in the captured multimodal distributions and also incorporates the label correlations into the learning process since labels sharing more instances will share more peaks in their captured distributions.

Dataset	$\lvert\mathcal{S}\rvert$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$LCard(\mathcal{S})$	Domain
CAL500	502	68	174	26.044	Music[1]
Image	2000	294	5	1.236	Images[2]
scene	2407	294	6	1.074	Images[1]
yeast	2417	103	14	4.237	Biology[1]
corel5k	5000	499	374	3.522	Images[1]
rcv1-s1	6000	944	101	2.880	Text[1]
Corel16k-s1	13766	500	153	2.859	Images[1]
delicious	16105	500	983	19.020	Text[1]
iaprtc12	19627	1000	291	5.719	Images[3]
espgame	20770	1000	268	4.686	Images[3]
mirflickr	25000	1000	38	4.716	Images[3]
tmc2007	28596	981	22	2.158	Text[1]
mediamill	43907	120	101	4.376	Video[1]
bookmarks	87856	2150	208	2.028	Text[1]

Table 1: Characteristics of the experimental data sets.

The overall objective function to maximize is given as follows:

$$\mathcal{L} = \mathcal{L}_{rec} + \alpha \cdot \mathcal{L}_{cls} + \gamma \cdot \mathcal{L}_{reg} \quad (16)$$

where α and γ are two trade-off parameters.

Experiments

Experimental Setup

Data Sets In this paper, fourteen benchmark multi-label data sets with diversified multi-label properties are employed for comprehensive performance evaluation. Table 1 summarizes characteristics of each experimental data set S, including the number of examples ($|S|$), number of features ($dim(S)$), number of class labels ($L(S)$), label cardinality ($LCard(S)$, i.e. average number of labels per instance). Following (Zhang and Wu 2015), we perform dimensionality reduction for rcv-s1 and tmc2007 by retaining the top 2% features with highest document frequency. For iaprtc12, espgame and mirflickr, the local descriptor *DenseSift* is used.

Evaluation Metrics For performance evaluation, we use six widely-used evaluation metrics for multi-label classification, including *Average precision*, *Macro-averaging AUC*, *Hamming loss*, *One-error*, *Coverage* and *Ranking loss*. Detailed definitions on these metrics can be found in (Zhang and Zhou 2014).

Implementation Details We employ a fully-connected neural network with hidden dimensionality $[256]$ to instantiate the conditional conditioner $c(\cdot; l_j)$ in the normalizing flows. The probabilistic autoencoders of PACA are parameterized by fully-connected neural networks with ReLU activations, where the hidden dimensionalities of the encoder and the decoder are both set to $[256, 512, 256]$. To compute the overall objective function in Eq. (16), we conduct Monte Carlo sampling to estimate the expectations in the first two terms with sampling number $L = 1$ and analytically calculate the third term as it is the KL divergence between two Gaussian distributions. For network optimization, Adam

[1] http://mulan.sourceforge.net/datasets.html

[2] http://palm.seu.edu.cn/zhangml/

[3] http://lear.inrialpes.fr/people/guillaumin/data.php

Data sets	*Average precision* ↑						
	ML-KNN	LIFT	LLSF	WRAP	C2AE	MPVAE	PACA
CAL500	0.4928±0.0127	0.5004±0.0150	0.5110±0.0161	0.5204±0.0144	0.4782±0.0124	0.5094±0.0133	**0.5246±0.0170**
Image	0.7882±0.0239	0.8238±0.0185	0.7536±0.0229	0.7812±0.0209	0.8249±0.0217	0.8212±0.0193	**0.8561±0.0173**
scene	0.8612±0.0172	0.8857±0.0162	0.8470±0.0159	0.8350±0.0208	0.8834±0.0212	0.8749±0.0225	**0.9048±0.0161**
yeast	0.7685±0.0182	0.7695±0.0171	0.7634±0.0140	0.7615±0.0138	0.7524±0.0167	0.7626±0.0193	**0.7717±0.0176**
corel5k	0.2466±0.0101	0.2879±0.0111	0.3006±0.0117	0.3285±0.0110	0.2940±0.0110	0.3299±0.0129	**0.3339±0.0126**
rcv1-s1	0.4676±0.0137	0.5957±0.0103	0.6195±0.0101	0.6337±0.0135	0.6083±0.0187	0.6415±0.0112	**0.6444±0.0113**
Corel16k-s1	0.2860±0.0058	0.3196±0.0050	0.3459±0.0066	0.3571±0.0062	0.3290±0.0088	0.3679±0.0050	**0.3717±0.0068**
delicious	0.3352±0.0045	0.3782±0.0049	0.3621±0.0047	0.3741±0.0049	0.3682±0.0026	0.4062±0.0057	**0.4129±0.0046**
iaprtc12	0.3837±0.0050	0.3459±0.0045	0.3680±0.0053	0.3853±0.0058	0.3950±0.0069	0.4375±0.0078	**0.4430±0.0053**
espgame	0.2459±0.0041	0.2835±0.0045	0.2772±0.0042	0.2906±0.0051	0.2815±0.0051	0.3103±0.0043	**0.3146±0.0039**
mirflickr	0.6083±0.0055	0.6354±0.0030	0.6510±0.0058	0.6549±0.0055	0.6690±0.0062	0.6944±0.0054	**0.7022±0.0058**
tmc2007	0.7263±0.0065	0.8148±0.0028	0.8148±0.0031	0.8057±0.0031	0.8000±0.0070	0.8310±0.0029	**0.8322±0.0036**
mediamill	0.7562±0.0032	0.7301±0.0032	0.7281±0.0026	0.7325±0.0030	0.7368±0.0033	0.7697±0.0043	**0.7864±0.0033**
bookmarks	0.3896±0.0039	0.4916±0.0035	0.5007±0.0020	0.4825±0.0029	0.4772±0.0043	**0.5153±0.0021**	0.5126±0.0027

Data sets	*Macro-averaging AUC* ↑						
	ML-KNN	LIFT	LLSF	WRAP	C2AE	MPVAE	PACA
CAL500	0.5098±0.0123	0.5176±0.0108	0.5786±0.0159	0.5803±0.0324	0.4850±0.0198	0.5488±0.0159	**0.5832±0.0272**
Image	0.8288±0.0208	0.8583±0.0152	0.7926±0.0212	0.8220±0.0237	0.8506±0.0226	0.8644±0.0164	**0.8778±0.0128**
scene	0.9317±0.0116	0.9480±0.0087	0.9210±0.0108	0.9110±0.0112	0.9390±0.0146	0.9415±0.0105	**0.9546±0.0104**
yeast	0.6888±0.0170	0.6752±0.0186	0.6937±0.0164	0.6890±0.0222	0.6658±0.0152	0.7000±0.0169	**0.7048±0.0250**
corel5k	0.5526±0.0155	0.7173±0.0128	0.6618±0.0169	0.7209±0.0117	0.7071±0.0142	0.7589±0.0113	**0.7655±0.0119**
rcv1-s1	0.6364±0.0201	0.9262±0.0069	0.9117±0.0094	0.9315±0.0090	0.9036±0.0108	0.9408±0.0056	**0.9427±0.0053**
Corel16k-s1	0.5244±0.0093	0.6875±0.0084	0.7100±0.0055	0.7539±0.0066	0.7200±0.0083	0.7865±0.0067	**0.7900±0.0052**
delicious	0.6461±0.0060	0.7819±0.0041	0.7659±0.0054	0.7780±0.0049	0.7822±0.0027	0.8284±0.0038	**0.8297±0.0043**
iaprtc12	0.7073±0.0076	0.7978±0.0049	0.8159±0.0045	0.8279±0.0045	0.8354±0.0049	0.8768±0.0026	**0.8773±0.0035**
espgame	0.5964±0.0036	0.7608±0.0074	0.7384±0.0058	0.7623±0.0076	0.7383±0.0142	0.7949±0.0053	**0.7997±0.0051**
mirflickr	0.7310±0.0068	0.7970±0.0051	0.8210±0.0046	0.8234±0.0067	0.8241±0.0061	0.8533±0.0043	**0.8567±0.0046**
tmc2007	0.8125±0.0052	0.9230±0.0030	0.9232±0.0036	0.9193±0.0036	0.9053±0.0023	**0.9332±0.0029**	0.9325±0.0027
mediamill	0.7802±0.0096	0.7743±0.0109	0.7780±0.0041	0.8510±0.0084	0.8202±0.0081	0.8698±0.0071	**0.8737±0.0087**
bookmarks	0.6756±0.0036	0.8939±0.0018	0.8818±0.0033	0.8742±0.0031	0.8507±0.0030	0.9128±0.0019	**0.9164±0.0019**

Table 2: Predictive performance of each comparing approach (mean±std. deviation). ↑ (↓) indicates the larger (smaller) the value, the better the performance. Best and second best results are shown in boldface and underlined respectively.

with a batch size of 128, weight decay of 10^{-5}, momentums of 0.999 and 0.9 is employed.

Comparative Studies

PACA is compared against six well-established multi-label classification approaches with parameter configurations suggested in respective literatures:

- ML-KNN (Zhang and Zhou 2007): A kNN-based approach with Bayesian inference. [$k = 10$]
- LIFT (Zhang and Wu 2015): A prototype-based label-specific feature transformation approach under independent three-stage framework. [$r = 0.1$]
- LLSF (Huang et al. 2016): LLSF performs label-specific feature selection in a lasso-regression-like framework with feature-sharing between closely-related labels. [grid search for $\alpha, \beta \in \{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$ and $\gamma = 0.01$]
- WRAP (Yu and Zhang 2021): WRAP performs label-specific feature selection within an embedded feature space and considers pairwise label correlation regularization. [grid search for $\lambda_1, \lambda_2 \in \{0, 1, \ldots, 10\}$]
- C2AE (Yeh et al. 2017): A deep label embedding approach, which jointly embeds features and labels via integrating deep canonical correlation analysis and autoencoder. [search for $\alpha \in \{0.1, 1, 2, 5, 10\}$]
- MPVAE (Bai, Kong, and Gomes 2020): MPVAE employs a variational autoencoder to align features and labels in a probabilistic latent space and explicitly learns a shared covariance matrix to model the label correlations. [$\lambda_1 = \lambda_2 = 0.5$, $\lambda_3 = 10$, $\beta = 1.1$]

For the proposed PACA approach, we search the trade-off parameters α, λ in $\{1, 2, 5, 10, 20, 50\}$ and $\{10^{-4}, 10^{-3}, \ldots, 10\}$ respectively. For fair comparison, all deep approaches share the same neural network structure. Grid search is conducted to find the best learning rate and learning rate decay ratio. We employ ten-fold cross validation to evaluate above approaches on the 14 data sets.

Due to page limit, Table 2 reports detailed experimental results in terms of two evaluation metrics. Results on other metrics can be found in the supplementary material. To analyze whether PACA performs statistically better than other comparing algorithms, the *Wilcoxon signed-ranks test* (Wilcoxon 1992) at 0.05 significance level is further conducted. Table 3 summarizes the p-value statistics on each evaluation metric. Based on these results, it is impressive to observe that:

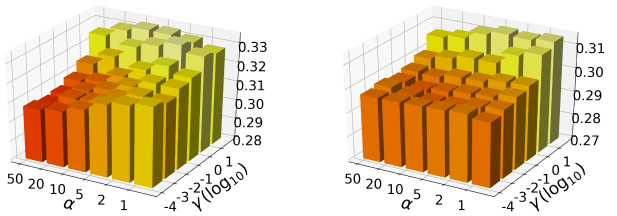
- Across all evaluation metrics, PACA achieves the best performance in 81% cases over all the 14 data sets.
- As shown in Table 3, PACA significantly outperforms other deep approaches. Note that PACA and MPVAE both

PACA against	ML-KNN	LIFT	LLSF	WRAP	C2AE	MPVAE
Average precision	**win** [0.0001]	**win** [0.0001]	**win** [0.0001]	**win** [0.0001]	**win** [0.0001]	**win** [0.0004]
Macro-averaging AUC	**win** [0.0001]	**win** [0.0001]	**win** [0.0001]	**win** [0.0001]	**win** [0.0001]	**win** [0.0004]
Hamming loss	**win** [0.0247]	**tie** [0.0588]	**win** [0.0063]	**win** [0.0287]	**win** [0.0001]	**win** [0.0127]
One-error	**win** [0.0002]	**win** [0.0001]	**win** [0.0001]	**win** [0.0002]	**win** [0.0002]	**win** [0.0009]
Coverage	**win** [0.0002]	**win** [0.0002]	**win** [0.0002]	**win** [0.0002]	**win** [0.0001]	**win** [0.0040]
Ranking loss	**win** [0.0002]	**win** [0.0002]	**win** [0.0001]	**win** [0.0001]	**win** [0.0001]	**win** [0.0067]

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

PACA against	PACA-sp	PACA-nr
Average precision	**win** [0.0001]	**win** [0.0009]
Macro-averaging AUC	**win** [0.0023]	**win** [0.0067]
Hamming loss	**win** [0.0112]	**win** [0.0195]
One error	**win** [0.0004]	**win** [0.0010]
Coverage	**win** [0.0017]	**win** [0.0171]
Ranking loss	**win** [0.0004]	**win** [0.0103]

Table 4: Summary of the Wilcoxon signed-ranks test for PACA against its variants at 0.05 significance level. p-values are shown in the brackets.



(a) corel5k **(b) espgame**

Figure 3: Performance of PACA with varying parameter configurations in terms of *Average precision*.

tackle the problem of multi-lable classification under a deep probabilistic framework. The superior performance of PACA against MPVAE provides another strong evidence for the effectiveness of label-specific features to facilitate multi-label classification.

- Meantime, PACA achieves much better performance against other approaches based on label-specific features. Specifically, PACA is statistically superior to LIFT in terms of all evaluation metrics except Hamming loss. These impressive results demonstrate the effectiveness of our unified framework for prototype-based label-specific feature transformation.

Further Analyses

Multi-Prototype Learning vs. Single-Prototype Learning We have provided an intuitive explanation for the necessity of the multi-prototype learning in Figure 1. Here, ablation study is further conducted to validate the superiority of the multi-prototype learning against single-prototype learning on all the 14 data sets with ten-fold cross validation. We implement a variant named PACA-sp, which models the positive/negative prototype of each class label by a multivariate Gaussian distribution. Wilcoxon signed-ranks test in Table 4 shows PACA is statistically superior to PACA-sp in terms of all evaluation metrics. Detail results can be found in the supplementary material[3].

Effectiveness of the Latent Space Regularization We implement a variant named PACA-nr by removing all the network structures related to the latent space regularization and setting the trade-off parameter γ in Eq. (16) to 0. Table 4 shows the proposed latent space regularization, which incorporates the label correlations into the prototype learning process, is statistically effective.

Parameter Sensitivity Figure 3 gives an illustrative example on how the performance of PACA changes when the values of the trade-off parameters α and λ change. The performance of PACA is quite sensitive to the value of λ, which demonstrates again the effectiveness of the latent space regularization. Similar results can be observed on other data sets.

More Analyses More analyses of PACA, including algorithmic complexity and running time comparisons, can be found in the supplementary material.

Conclusion

In this paper, a first attempt towards a unified framework for prototype-based label-specific feature transformation is presented. Different from the existing three-stage pipeline, we propose a novel approach PACA which learns the prototypes and the metric-based label-specific features jointly for multi-label classification. To allow for scalable stochastic optimization, PACA employs the normalizing flows to model prototypes probabilistically, which enables the prototype learning process adaptive to the underlying properties of each class label. A probabilistic latent metric space is constructed to generate more discriminative label-specific features for classification, with further regularization from label correlations. Comprehensive experiments show that PACA outperforms other well-established multi-label classification approaches. Despite the demonstrated effectiveness of PACA, it only considers fixed prototypes for each class label, which may be further improved via more elaborate strategies for prototype modelling, such as instance-specific prototypes to account for the distinct characteristics of each instance and each label simultaneously.

[3]http://palm.seu.edu.cn/zhangml/files/PACASupplement.pdf

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science Foundation of China (62176055) and the China University S&T Innovation Plan Guided by the Ministry of Education. We thank the Big Data Center of Southeast University for providing the facility support on the numerical calculations in this paper.

References

Allen, K. R.; Shelhamer, E.; Shin, H.; and Tenenbaum, J. B. 2019. Infinite mixture prototypes for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*, 232–241. Long Beach, CA.

Bai, J.; Kong, S.; and Gomes, C. P. 2020. Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 4313–4321. Yokohama, Japan.

Boutell, M.; Luo, J.-B.; Shen, X.-P.; and Brown, C. 2004. Learning multi-label scene classification. *Pattern Recognition*, 37(9): 1757–1771.

Chen, C.; Wang, H.-B.; Liu, W.-W.; Zhao, X.-Y.; Hu, T.-L.; and Chen, G. 2019a. Two-stage label embedding via neural factorization machine for multi-label classification. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 3304–3311. Honolulu, HI.

Chen, D.; Xue, Y.; Fink, D.; Chen, S.; and Gomes, C. P. 2017. Deep multi-species embedding. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 3639–3646. Melbourne, Australia.

Chen, T.; Lin, L.; Hui, X.; Chen, R.; and Wu, H. 2020. Knowledge-guided multi-label few-shot learning for general image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen, Z.-M.; Wei, X.-S.; Wang, P.; and Guo, Y.-W. 2019b. Multi-label image recognition with graph convolutional networks. In *Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, 5177–5186. Long Beach, CA.

Chen, Z.-S.; and Zhang, M.-L. 2019. Multi-label learning with regularization enriched label-specific features. In *Proceedings of the 11th Asian Conference on Machine Learning*, 411–424. Nagoya, Japan.

Elisseeff, A.; and Weston, J. 2001. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, 681–687. Vancouver, Canada.

Feng, L.; An, B.; and He, S. 2019. Collaboration based multi-label learning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 3550–3557. Honolulu, HI.

Germain, M.; Gregor, K.; Murray, I.; and Larochelle, H. 2015. MADE: Masked autoencoder for distribution estimation. In *Proceedings of the 32nd International Conference on Machine Learning*, 881–889. Lille, France.

Guo, Y.; Chung, F.; Li, G.; Wang, J.; and Gee, J. C. 2019. Leveraging label-specific discriminant mapping features for multi-label learning. *ACM Transactions on Knowledge Discovery from Data*, 13(2): 24:1–24:23.

Huang, C.; Krueger, D.; Lacoste, A.; and Courville, A. C. 2018a. Neural autoregressive flows. In *Proceedings of the 35th International Conference on Machine Learning*, 2083–2092. Stockholm, Sweden.

Huang, J.; Li, G.; Huang, Q.; and Wu, X. 2015. Learning label specific features for multi-label classification. In *Proceedings of the 15th IEEE International Conference on Data Mining*, 181–190. Atlantic City, NJ.

Huang, J.; Li, G.; Huang, Q.; and Wu, X. 2016. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 28(12): 3309–3323.

Huang, J.; Li, G.; Huang, Q.; and Wu, X. 2018b. Joint feature selection and classification for multilabel learning. *IEEE Transactions on Cybernetics*, 48(3): 876–889.

Jia, X.-Y.; Zhu, S.-S.; and Li, W.-W. 2020. Joint label-specific features and correlation information for multi-label learning. *Journal of Computer Science and Technology*, 35(2): 247–258.

Kingma, D. P.; and Welling, M. 2014. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*. Banff, Canada.

Kobyzev, I.; Prince, S.; and Brubaker, M. 2021. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11): 3964–3979.

Lin, Y.; Hu, Q.; Liu, J.; Zhu, X.; and Wu, X. 2021. MULFE: Multi-label learning via label-specific feature space ensemble. *ACM Transactions on Knowledge Discovery from Data*, 16(1): 1–24.

Liu, W.; Shen, X.; Wang, H.; and Tsang, I. W. 2021. The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, W.; and Tsang, I. W. 2015. Large margin metric learning for multi-label prediction. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2800–2806. Austin, TX.

Rastin, N.; Jahromi, M. Z.; and Taheri, M. 2021. A generalized weighted distance k-nearest neighbor for multi-label problems. *Pattern Recognition*, 114: 107526.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine Learning*, 85(3): 333–359.

Shen, X.; Liu, W.; Luo, Y.; Ong, Y.-S.; and Tsang, I. W. 2018. Deep binary prototype multi-label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2675–2681. Stockholm, Sweden.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30*, 4077–4087. Long Beach, CA.

Tang, P.; Jiang, M.; Xia, B. N.; Pitera, J. W.; Welser, J.; and Chawla, N. V. 2020. Multi-label patent categorization with non-local attention-based graph convolutional network. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 9024–9031. New York, NY.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2010. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7): 1079–1089.

Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. CNN-RNN: A unified framework for multi-label image classification. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, 2285–2294. Las Vegas, NV.

Weng, W.; Chen, Y.-N.; Chen, C.-L.; Wu, S.; and Liu, J. 2020. Non-sparse label specific features selection for multi-label classification. *Neurocomputing*, 377: 85–94.

Weng, W.; Lin, Y.; Wu, S.; Li, Y.; and Kang, Y. 2018. Multi-label learning based on label-specific features and local pairwise label correlation. *Neurocomputing*, 273: 385–394.

Wilcoxon, F. 1992. *Individual Comparisons by Ranking Methods*, 196–202. Berlin, Germany: Springer.

Xu, M.; and Guo, L.-Z. 2021. Learning from group supervision: the impact of supervision deficiency on multi-label learning. *Science China Information Sciences*, 64(3): Article 130101.

Xu, S.-P.; Yang, X.-B.; Yu, H.-L.; Yu, D.-J.; Yang, J.-Y.; and Tsang, E. 2016. Multi-label learning with label-specific feature reduction. *Knowledge-Based Systems*, 104: 52–61.

Yazici, V. O.; Gonzalez-Garcia, A.; Ramisa, A.; Twardowski, B.; and van de Weijer, J. 2020. Orderless recurrent models for multi-label classification. In *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition*, 13437–13446. Seattle, WA.

Yeh, C.; Wu, W.; Ko, W.; and Wang, Y. 2017. Learning deep latent spaces for multi-label classification. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2838–2844. San Francisco, CA.

You, R.; Guo, Z.; Cui, L.; Long, X.; Bao, Y.; and Wen, S. 2020. Cross-modality attention with semantic graph embedding for multi-label classification. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 12709–12716. New York, NY.

Yu, Z.-B.; and Zhang, M.-L. 2021. Multi-label classification with label-specific feature generation: A wrapped approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhan, W.; and Zhang, M.-L. 2017. Multi-label learning with label-specific features via clustering ensemble. In *Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics*, 129–136. Tokyo, Japan.

Zhang, C.; and Li, Z. 2021. Multi-label learning with label-specific features via weighting and label entropy guided clustering ensemble. *Neurocomputing*, 419: 59–69.

Zhang, J.; Fang, M.; and Li, X. 2015. Multi-label learning with discriminative features for each label. *Neurocomputing*, 154: 305–316.

Zhang, J.; Li, C.; Cao, D.; Lin, Y.; Su, S.; Dai, L.; and Li, S. 2018. Multi-label learning with label-specific features by resolving label correlations. *Knowledge-Based Systems*, 159: 148–157.

Zhang, J.; Zhao, C.; Ni, B.; Xu, M.; and Yang, X. 2019. Variational few-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 1685–1694. Seoul, Korea.

Zhang, M.; and Zhou, Z. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7): 2038–2048.

Zhang, M.-L.; and Wu, L. 2015. LIFT: Multi-label learning with label-specific features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1): 107–120.

Zhang, M.-L.; and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8): 1819–1837.

Zhen, X.; Du, Y.; Xiong, H.; Qiu, Q.; Snoek, C.; and Shao, L. 2020. Learning to learn variational semantic memory. In *Advances in Neural Information Processing Systems 33*, 9122–9134. virtual.

Zhu, Y.; Kwok, J. T.; and Zhou, Z. 2018. Multi-label learning with global and local label correlation. *IEEE Transactions on Knowledge and Data Engineering*, 30(6): 1081–1094.