Dual Perspective of Label-Specific Feature Learning for Multi-Label Classification

Jun-Yi Hang\textsuperscript{1,2} Min-Ling Zhang\textsuperscript{1,2}

Abstract

Label-specific features serve as an effective strategy to facilitate multi-label classification, which account for the distinct discriminative properties of each class label via tailoring its own features. Existing approaches implement this strategy in a quite straightforward way, i.e. finding the most pertinent and discriminative features for each class label and directly inducing classifiers on constructed label-specific features. In this paper, we propose a dual perspective for label-specific feature learning, where label-specific discriminative properties are considered by identifying each label’s own non-informative features and making the discrimination process immutable to variations of these features. To instantiate it, we present a perturbation-based approach \textsc{DEL}A to provide classifiers with label-specific immutability on simultaneously identified non-informative features, which is optimized towards a probabilistically-relaxed expected risk minimization problem. Comprehensive experiments on 10 benchmark data sets show that our approach outperforms the state-of-the-art counterparts.

1. Introduction

Multi-label classification allows to learn from instances associated with multiple labels simultaneously (Zhang & Zhou, 2014; Liu et al., 2021). Nowadays, researches on multi-label classification have been greatly driven by real-world applications, where multi-semantic objects widely exist, such as image annotation (You et al., 2020), text categorization (Xun et al., 2020), and bioinformatics analysis (Chen et al., 2017), etc.

The most straightforward strategy for dealing with multi-label data is to exploit the identical representation of an instance in inducing classification models. However, this strategy might be suboptimal as it fails to consider that each class label may possess its own discriminative properties. With the ability to model distinct characteristics of each class label, label-specific features have become a promising strategy to facilitate the discrimination of each class label by tailoring its own features (Zhang & Wu, 2015; Huang et al., 2016b; Zhang et al., 2018; Jia et al., 2020; Yu & Zhang, 2021).

As a seminal work, LIFT (Zhang & Wu, 2015) firstly performs clustering analysis on positive and negative instances of each class label, and then heuristically constructs label-specific features via prototype-based feature transformation. While LLSF (Huang et al., 2015) employs feature selection to obtain the most pertinent feature subset for each class label under a lasso-based framework. Recent works (Hang & Zhang, 2021; Hang et al., 2022) resort to the powerful representation learning capability of deep neural networks to learn label-specific features in an end-to-end manner. It is worth noting that existing approaches focus on finding the most pertinent and discriminative features for each class label and directly inducing classifiers on constructed label-specific features.

In this paper, we attack the problem of label-specific feature learning from a dual perspective. Instead of finding the most pertinent and discriminative features for each class label as existing approaches do, we attempt to identify each label’s own non-informative features and endow classifiers with immutability on these identified features. For example, to discriminate plane and non-plane images, existing approaches induce classifier on the most pertinent features, e.g. shape-based features. Instead, we aim to make the discrimination process immutable to variations of non-informative features, e.g. color-based features. We hypothesize that if non-informative features specific to each class label could be identified and their influence on the discrimination process could be eliminated, a more effective approach to learn from multi-label data could be achieved.
As a first attempt towards this dual strategy, a novel approach named DELA, i.e. Dual pErspective of Label-specific feAture learning for multi-label classification, is presented. Under a stochastic feature perturbation framework, DELA simultaneously identifies non-informative features and induces classifiers which are immutable to these identified features. Specifically, by selectively injecting random noise to label-specific non-informative features and inducing classifiers on these perturbed features, DELA succeeds to remove classifiers’ dependence on identified non-informative features. With the basic assumption that non-informative features should have no negative influence on the discrimination process when perturbed by noise, identification of non-informative features for each class label is casted as an expected risk minimization problem, which is further relaxed probabilistically to support end-to-end gradient-based optimization. We further justify DELA from the perspective of information theory and demonstrate that DELA actually optimizes an upper bound of the information bottleneck (Tishby et al., 1999). Comprehensive experiments on 10 benchmark data sets show that DELA 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 DELA approach. Section 4 reports experimental results over a wide range of multi-label data sets. Section 5 concludes this paper.

2. Related Works

Multi-Label Classification. In the last decade, numerous approaches have been proposed to deal with multi-label classification problem (Zhang & Zhou, 2014; Liu et al., 2021). As a feasible strategy to facilitate the learning process, modelling label correlations is one of the primary focuses in recent studies. Generally speaking, these approaches can be roughly grouped into three categories, which differ in the order of label correlations considered, namely first-order approaches (Boutell et al., 2004; Zhang & Zhou, 2007), second-order approaches (Zhu et al., 2018; Sun & Zhang, 2021) and high-order approaches (Wehrmann et al., 2018; Xu & Guo, 2021). Recent works resort to deep models, such as recurrent neural networks (Wang et al., 2016; Yazici et al., 2020) and graph neural networks (Chen et al., 2019; 2022), to jointly consider the label correlation exploitation and classification model induction. Some embedding approaches (Yeh et al., 2017; Bai et al., 2020; Dahiya et al., 2021) implicitly employ label correlations via embedding and aligning features and labels in a deep latent space.

Complementary to label correlation exploitation, label-specific features have been proven to be another effective strategy to improve multi-label classification, which tackle the problem via manipulating the input space instead of the label space. Existing approaches construct the label-specific features mainly in two manners, i.e. prototype-based label-specific feature transformation and label-specific feature selection.

For the prototype-based label-specific feature transformation approaches, label-specific features are generated by treating the prototypes of each class label as the transformation bases. Under a three-stage framework, LIFT (Zhang & Wu, 2015) constructs label-specific features via querying the distances between the original instance and the cluster centers for each class label. Follow-up works enhance the three-stage framework by customized strategies, such as stabilizing the clustering process with clustering ensemble (Zhan & Zhang, 2017; Zhang & Li, 2021) or spectral clustering (Zhang et al., 2015), augmenting metric-based label-specific features with local neighbor information (Weng et al., 2018) or global topological information (Guo et al., 2019), unifying the independent three-stage framework into an end-to-end counterpart (Hang et al., 2022).

Alternatively, label-specific features can also be constructed by retaining a feature subset as the most pertinent features for each class label. LLSF (Huang et al., 2015; 2016b) presents a lasso-based framework for label-specific feature selection, where the selection process is regularized with pairwise label correlations. Subsequent studies extend this framework via imposing non-sparse constraints (Weng et al., 2020), incorporating discriminant-related regularization (Huang et al., 2018), or performing selection in a projected feature space (Yu & Zhang, 2021). Recently, CLIF (Hang & Zhang, 2021) further advances the idea to the deep learning scenario with an attractive collaborative learning strategy.

In this paper, we make a first attempt to consider label-specific discriminative properties via endowing classifiers with immutability on non-informative features, which is an unexplored direction for label-specific feature learning.

Nuisance Factor Removal. A similar concept, i.e. removal of nuisance factors, has a long history in computer vision. Early attempts include designing scale-invariant (Lowe, 1999) or rotation-invariant features (Greenspan et al., 1994), while recent approaches resort to techniques, such as data augmentation (Devries & Taylor, 2017; Cubuk et al., 2019) and representation disentangling (Tran et al., 2017; Moyer et al., 2018; Lee et al., 2021), for removal of specified factors. DELA shares the idea and generalizes it by learning to identify non-information features instead of specifying them beforehand. In domain generalization (Ahuja et al., 2020; 2021), domain-specific features are regarded as nuisance factors, which should be removed to learn domain-invariant ones for achieving good performance on unseen domains. DELA also shares similar idea but attempts to encourage the emergence of label(domain)-specific properties.
Feature Perturbation by Noise Injection. Injecting noise to perturb features has been widely applied in machine learning community. Dropout (Srivastava et al., 2014) and its extensions (Blum et al., 2015; Huang et al., 2016a; Achille & Soatto, 2018) perturb features by randomly dropping out neurons or layers during training to encourage redundancy in learned representation, which deviates from our goal to remove redundancy. Adversarial attack (Goodfellow et al., 2015; Madry et al., 2018; Duan et al., 2021) aims to find the most vulnerable directions to perturb the instance so that the training loss is maximized, while we attempt to identify and perturb non-informative features for expected risk minimization. Besides, feature perturbation can also be exploited to perform post-hoc explanation of prediction (Ribeiro et al., 2016; Fong & Vedaldi, 2017), while our approach perturbs features during the learning process for generalization.

3. The DELA Approach

3.1. Preliminaries

Let \( \mathcal{X} = \mathbb{R}^d \) denote the input space and \( \mathcal{Y} = \{l_1, l_2, \ldots, l_t\} \) denote the label space with \( t \) class labels. A multi-label example is denoted as \((x, Y)\), where \( x \in \mathcal{X} \) is its feature vector and \( Y \subseteq \mathcal{Y} \) is its set of relevant labels. Here, a \( t \)-dimensional indicator vector \( y = [y_1, y_2, \ldots, y_t] \in \{0, 1\}^t \) is utilized to denote the label, 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 \( h : \mathcal{X} \rightarrow \mathcal{Y}^t \) from a multi-label data set \( \mathcal{D} = \{(x_i, Y_i) | i \in [m]\} \), where \([m]\) denotes the set \( \{1, 2, \ldots, m\} \). Given an unseen instance \( x \in \mathcal{X} \), its associated label set is predicted as \( h(x) \subseteq \mathcal{Y} \).

3.2. Overview

DELA firstly computes a \( d_z \)-dimensional representation \( z \in \mathbb{R}^{d_z} \) through an embedding function \( e_{\phi} : \mathbb{R}^d \rightarrow \mathbb{R}^{d_z} \) parametrized by \( \phi \), which is shared among all the class labels. Then, a selective feature perturber injects additive random noise into representation \( z \) for perturbing non-informative features specific to each class label. Finally, classification is performed on the noise perturbed representations.

Learning proceeds by simultaneously identifying non-informative features and making the discrimination process immutable to identified non-informative features, with the expected risk minimizing problem as following

\[
\min_{\phi, \Pi, \Theta} \mathbb{E}_{p(x, y)} \left[ \sum_{k=1}^{t} \mathcal{L}(f_k(g_k(e_{\phi}(x); \pi_k); \theta_k), y_k) \right],
\]

where \( \Pi = \{\pi_1, \ldots, \pi_t\} \), \( \Theta = \{\theta_1, \ldots, \theta_t\} \) are the sets to parametrize the selective feature perturber \( g_k : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z} \) and the classifier \( f_k : \mathbb{R}^{d_z} \rightarrow \mathbb{R} \) for each class label respectively. We will describe the ingredients of DELA in detail in the next subsection.

3.3. Selective Feature Perturber

The goal of the selective feature perturber is to identify non-informative features in representation \( z \) and perturb them via injecting random noise in a label-wise manner, so that immutability of classification on the non-informative features can be gradually enhanced.

To instantiate the selective feature perturber, we formalize the perturbation process as

\[
g_k(z; \pi_k) = z + i_{S_k} \odot \epsilon, \quad \text{with} \quad \epsilon \sim p_\vartheta(\epsilon),
\]

where \( S_k \subseteq \{d_z\} \) denotes a subset of identified non-informative features for label \( l_k \) which is determined by parameter \( \pi_k \) and \( i_{S_k} \in \{0, 1\}^{d_z} \) is the indicator vector of set \( S_k \). \( \epsilon \) is a random noise variable shared among all the class labels, which is treated as an instance-dependent Gaussian one, i.e. \( p_\vartheta(\epsilon) = \mathcal{N}(0, \sigma^2_\vartheta(x)I) \).

With the Hadamard product \( \odot \), additive random noise is selectively injected into the identified non-informative features.

Substituting Eq. (2) into Eq. (1), the expected risk minimizing problem becomes

\[
\min_{\phi, \Pi, \Theta} \mathbb{E}_{p(x, y) p_\vartheta(\epsilon)} \left[ \sum_{k=1}^{t} \mathcal{L}(f_k(e_{\phi}(x)+i_{S_k} \odot \epsilon; \theta_k), y_k) \right],
\]

where \( \mathcal{S} = \{S_1, \ldots, S_t\} \). The above problem is hard to solve, since the optimization over the discrete subsets of non-informative features \( \{S_1, \ldots, S_t\} \) is intractable, whose choices grow exponentially in \( d_z \). Furthermore, constraint on the level of noise is necessary to prevent collapse, i.e. insufficient perturbation, so as to endow classifiers with immutability on non-informative features. We will describe our considerations towards these two problems in detail.

3.3.1. Differentiable Subset Selection

To make Eq. (3) tractable, we introduce Bernoulli gates to substitute the indicator vector \( i_{S_k} \in \{0, 1\}^{d_z} \). These Bernoulli gates can be represented by a random vector \( b_k \in \{0, 1\}^{d_z} \), whose entries are independent and satisfy \( P[b_k = 1] = p_k \), \( i \in \{d_z\} \). Then, the expected risk minimizing problem can be rewritten as

\[
\min_{\phi, \Pi, \Theta} \mathbb{E}_{p(x, y) p_\vartheta(\epsilon)} \left[ \sum_{k=1}^{t} \mathbb{E}_{p[b_k]} \mathbb{E}_{p_\vartheta(\epsilon)} \mathcal{L}(f_k(e_{\phi}(x)+b_k \odot \epsilon; \theta_k), y_k) \right].
\]

(\( \vartheta \) parametrizes the standard deviation function, which is shared among all the class labels. We can now denote \( \pi_k = [S_k, \vartheta] \).
By introducing Bernoulli gates, the original intractable subset selection problem is converted to an optimization problem in terms of Bernoulli distribution parameters \( P = \{p_1, \ldots, p_l\} \). Nonetheless, the discrete property of the sampling from Bernoulli distribution prevents gradients from flowing through the discrete random nodes \( b_k \), thus making the problem unable to be optimized end-to-end via gradient descent.

A feasible way to circumvent this is to exploit Gumbel-Softmax trick (Jang et al., 2017; Maddison et al., 2017) to smooth the sampling process, where a Bernoulli random variable \( b \sim \text{Bern}(p) \) is relaxed by its continuous alternative, i.e. a binary Concrete variable \( c \sim \text{BinConcrete}(p, \tau) \), which can be reparameterized as

\[
c = \frac{1}{1 + \exp[-(\log \alpha + l)/\tau]},
\]

where \( \alpha = \frac{p}{1-p} \), \( l \) is a sampling from Logistic distribution, and \( \tau > 0 \) is a temperature parameter. In the limit \( \tau \to 0 \), a binary Concrete variable smoothly converges to its Bernoulli counterpart.

We relax the Bernoulli gates \( b_k \) to the above binary Concrete gates \( c_k \) so that the gradients w.r.t. the distribution parameters \( \{p_1, \ldots, p_l\} \) are well-defined by the chain rule. In the forward pass we discretize the continuous samplings from binary Concrete gates by rounding, i.e. \( d_k = \text{round}(c_k) \), and in the backward pass we use a straight-through gradient estimator to approximate \( \nabla_{p_k} c_k \approx \nabla_{p_k} d_k \).

### 3.3.2. Constraint on Noise Distribution

Let \( z_k = c_k(x) + \text{round}(c_k) \circ \epsilon, \epsilon \sim p(\epsilon) \) and \( c_k \sim p(c_k) \), we can regard the above equation as the reparameterization form of the random variable \( z_k \), i.e. label \( l_k \)'s perturbed stochastic representation, which follows an implicit distribution \( \mathbb{E}_{p(c_k)}[p_{\varphi, \theta}(z_k | x, c_k)] \).

From a probabilistic perspective, we propose to constrain the noise distribution \( p(\epsilon) \) by penalizing the expected discrepancy between \( p_{\varphi, \theta}(z_k | x, c_k) \) and an instance-agnostic prior distribution \( q(z_k) \), which can be formalized as

\[
\mathbb{E}_{p(c_k)}[\text{KL}(p_{\varphi, \theta}(z_k | x, c_k) || q(z_k))],
\]

where \( \text{KL}(\cdot || \cdot) \) denotes the KL-divergence. The conditional distribution \( p_{\varphi, \theta}(z_k | x, c_k) \) describes the extent to which the original representation is perturbed by noise conditioning on currently identified non-informative features, and the prior distribution \( q(z_k) \) reflects the target level of noise which is sufficient to remove classifiers’ dependence on non-informative features. We set \( q(z_k) \) as a standard Gaussian in this paper.

Substituting the reparameterization form of the perturbed stochastic representation and the constraint on noise distribution for each class label, the overall objective is defined as

\[
\min_{\phi, \vartheta} \mathbb{E}_{p(x, y)} \sum_{k=1}^{l} \mathbb{E}_{p(c_k)} \left[ \mathbb{E}_{p(z_k | x, c_k)} [\mathcal{L}(f_k(z_k; \theta_k), y_k)] + \beta \cdot \text{KL}(p(z_k | x, c_k) || q(z_k)) \right],
\]

where \( \beta \) is a trade-off parameter and \( p_{\phi, \vartheta}(z_k | x, c_k) \) is abbreviated to \( p(z_k | x, c_k) \) for clarity.

### 3.4. Information Theory Explanation

We further provide an information theoretic insight of DELA and demonstrate that DELA actually optimizes towards an upper bound of the information bottleneck when the risk function \( \mathcal{L}(\cdot, \cdot) \) is instantiated by cross entropy loss.

The information bottleneck defines a optimal information transportation process \( x \to h \to y \) by

\[
\min -I(h; y) + \beta \cdot I(h; x),
\]

where \( h \) is the internal representation of a model (e.g. neural networks) to predict the target variable \( y \) based on the input variable \( x \), and \( I(\cdot \mid \cdot) \) denotes the mutual information operator. The goal of the information bottleneck is to learn an optimal representation \( h \) which is maximally expressive about the target variable \( y \) and minimally compressive about the input variable \( x \). In other words, any information irrelevant to target prediction will be dropped during the information transportation process \( x \to h \). In DELA, we perform label-specific feature learning by making the discrimination process immutable to non-informative features with explicit noise injection during the above information transportation process, which shares motivation with the information bottleneck.

To show the connection theoretically, we firstly derive an upper bound for the information bottleneck.

**Theorem 3.1.** For any random variant \( c \sim p(c) \), the information bottleneck can be upper bounded as follows

\[
- I(h; y) + \beta \cdot I(h; x) \leq \mathbb{E}_{p(x, y)} \mathbb{E}_{p(h \mid x)} \left[ - \log q(y \mid h) \right] + \beta \cdot \text{KL}(p(h \mid x, c) || q(h)) \].

Proof of Theorem 3.1 can be found in the appendix A. Extending it into multi-label classification scenario, the upper bound of label-wise information bottlenecks can be formal-
AUC We implement $\sigma$ deviation function dimensionalities are set to $[256; 512; 256]$ neural network with ReLU activations, where the hidden and decoder as $M$.

Detailed definitions on these metrics can be found in (Zhang & Wu, 2015), we perform dimensionality reduction for rcv-s1 and tmc2007 by retaining the top 2% properties of each data set are employed in this paper. Table 1 summarizes detailed multi-label data sets with diversified multi-label properties.

For comprehensive performance evaluation, ten benchmark classification approaches with parameter configurations suggested by (Maddison et al., 2017).

To parametrize the binary Concrete gates, we employ a two-layer fully-connected neural network to produce the distribution parameters $\alpha, \beta \in \{0, 1\}$.

4.2. Comparative Studies

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

- **LIFT** (Zhang & Wu, 2015): A prototype-based label-specific feature transformation approach under independent three-stage framework. $[r = 0.1]$
- **LLSF** (Huang et al., 2016b): LLSF performs label-specific feature selection in a lasso-based framework with feature-sharing between closely-related labels. [grid search for $\alpha, \beta \in \{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$ and $\gamma = 0.01$]
- **C2AE** (Yeh et al., 2017): A deep label embedding approach, which jointly embeds features and labels via integrating deep canonical correlation analysis and

is a four-layer fully-connected neural network, which shares the first three layers with $e_\phi$. Classifiers $f_k, k \in [t]$ are implemented as three-layer fully-connected neural networks, where the hidden dimensionalities are set to [256; 512] and the first two layers are shared among all the class labels. To parametrize the binary Concrete gates, we employ a two-layer fully-connected neural network to produce the distribution parameters $\{p_1, \ldots, p_t\}$ and use $\tau = 2/3$ as suggested by (Maddison et al., 2017).

In all experiments, we consider cross entropy loss to instantiate the risk function $L(e, \cdot)$, as it allows to build connection between DELA and the information bottleneck. To compute the overall objective in Eq. (7), we conduct Monte Carlo sampling to estimate the expectations in terms of $p(c_k), p(z_k|x, c_k)$ with sampling number $L = 1$ and analytically calculate the KL-divergence term between two Gaussian distributions. For network optimization, Adam with a batch size of 128, weight decay of $10^{-4}$, momentum of 0.999 and 0.9 is employed.
autoencoder. [search for \( \alpha \in \{0.1, 1, 2, 5, 10\} \)]

- **MPVAE** (Bai et al., 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. \([\lambda_1 = \lambda_2 = 0.5, \lambda_3 = 10, \beta = 1.1]\)

- **CLIF** (Hang & Zhang, 2021): A deep approach for label-specific feature learning, which finds the most discriminative features for each class label with the guidance of collaboratively learned label semantics. [grid search for \( \lambda \in \{10^{-5}, 10^{-4}, \ldots, 1, 2, 5, 10\} \) and \( d_e \in \{64, 128, 256\} \)]

- **PACA** (Hang et al., 2022): A prototype-based deep label-specific feature transformation approach, which learns prototypes, label-specific features and classifiers in a unified probabilistic framework. [grid search for \( \alpha \in \{1, 2, 5, 10, 20, 50\} \) and \( \lambda \in \{10^{-4}, 10^{-3}, \ldots, 10\} \)]

For our DELA approach, the trade-off parameter \( \beta \) is searched in \( \{10^{-5}, 10^{-4}, \ldots, 10\} \). 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 schedule. We take out 10% examples in each data set as hold-out validation set for hyperparameter searching and perform ten-fold cross validation on the remaining 90% examples to evaluate above approaches.

Table 2 and Table 3 report detailed experimental results in terms of each evaluation metric. Table 5 further reports results of the Wilcoxon signed-ranks test (Wilcoxon, 1992) at 0.05 significance level to analyze whether DELA performs statistically better than other comparing algorithms. Based on these results, it is impressive to observe that:

- Across all evaluation metrics, DELA achieves the best performance in 92% cases over all the 10 data sets.
- As shown in Table 5, DELA significantly outperforms deep label embedding approaches C2AE and MPVAE in all evaluation metrics. The superior performance of DELA against C2AE and MPVAE indicates that it is a

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Average precision (\uparrow)</th>
<th>Macro-averaging AUC (\uparrow)</th>
<th>Hamming loss (\downarrow)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CLIP</td>
<td>PACA</td>
<td>DELA</td>
</tr>
<tr>
<td>corel5k</td>
<td>0.2911±0.0017</td>
<td>0.2996±0.0133</td>
<td>0.2915±0.0100</td>
</tr>
<tr>
<td>rcv1-s1</td>
<td>0.5921±0.0145</td>
<td>0.6129±0.0116</td>
<td>0.6147±0.0130</td>
</tr>
<tr>
<td>Corel16k-s1</td>
<td>0.3168±0.0059</td>
<td>0.3428±0.0053</td>
<td>0.3297±0.0042</td>
</tr>
<tr>
<td>delicious</td>
<td>0.3833±0.0061</td>
<td>0.3857±0.0079</td>
<td>0.3648±0.0075</td>
</tr>
<tr>
<td>iaprtc12</td>
<td>0.3833±0.0058</td>
<td>0.3597±0.0041</td>
<td>0.3900±0.0061</td>
</tr>
<tr>
<td>espgame</td>
<td>0.2808±0.0072</td>
<td>0.2770±0.0046</td>
<td>0.2741±0.0072</td>
</tr>
<tr>
<td>mirfllickr</td>
<td>0.6516±0.0041</td>
<td>0.6477±0.0039</td>
<td>0.6627±0.0064</td>
</tr>
<tr>
<td>tmc2007</td>
<td>0.8207±0.0048</td>
<td>0.8130±0.0050</td>
<td>0.7977±0.0048</td>
</tr>
<tr>
<td>mediannull</td>
<td>0.7417±0.0058</td>
<td>0.7275±0.0051</td>
<td>0.7266±0.0051</td>
</tr>
<tr>
<td>bookmarks</td>
<td>0.5119±0.0044</td>
<td>0.4920±0.0035</td>
<td>0.4707±0.0046</td>
</tr>
</tbody>
</table>
### Table 3. Predictive performance of each comparing approach (mean±std. deviation) in terms of One-error, Coverage and Ranking loss. ↑ (↓) indicates the larger (smaller) the value, the better the performance. Best results are highlighted in **boldface**.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>LIFT</th>
<th>LLSF</th>
<th>C2AE</th>
<th>MVPVAE</th>
<th>CLIP</th>
<th>PACA</th>
<th>DELA</th>
</tr>
</thead>
<tbody>
<tr>
<td>corel5k</td>
<td>0.6804±0.0251</td>
<td>0.4640±0.0231</td>
<td>0.6462±0.0145</td>
<td>0.6292±0.0339</td>
<td>0.6320±0.0246</td>
<td>0.6280±0.0279</td>
<td><strong>0.6178±0.0260</strong></td>
</tr>
<tr>
<td>rcv1-s1</td>
<td>0.4106±0.0194</td>
<td>0.4220±0.0161</td>
<td>0.4383±0.0210</td>
<td>0.4078±0.0330</td>
<td>0.4102±0.0192</td>
<td>0.4026±0.0239</td>
<td><strong>0.4006±0.0215</strong></td>
</tr>
<tr>
<td>Corel16k-s1</td>
<td>0.6764±0.0126</td>
<td>0.6398±0.0092</td>
<td>0.6442±0.0090</td>
<td>0.6331±0.0155</td>
<td>0.6420±0.0126</td>
<td>0.6283±0.0168</td>
<td><strong>0.6268±0.0099</strong></td>
</tr>
<tr>
<td>Corel16k-s2</td>
<td>0.3339±0.0153</td>
<td>0.3537±0.0145</td>
<td>0.3374±0.0178</td>
<td>0.3070±0.0183</td>
<td>0.3194±0.0180</td>
<td>0.3091±0.0156</td>
<td><strong>0.3061±0.0138</strong></td>
</tr>
<tr>
<td>delicious</td>
<td>0.4649±0.0182</td>
<td>0.4827±0.0118</td>
<td>0.4684±0.0145</td>
<td>0.4279±0.0134</td>
<td>0.4275±0.0097</td>
<td>0.4228±0.0114</td>
<td><strong>0.4104±0.0105</strong></td>
</tr>
<tr>
<td>iaprtc12</td>
<td>0.4659±0.0211</td>
<td>0.6371±0.0140</td>
<td>0.6511±0.0122</td>
<td>0.6070±0.0118</td>
<td>0.6033±0.0173</td>
<td>0.5933±0.0101</td>
<td><strong>0.5936±0.0113</strong></td>
</tr>
<tr>
<td>espgame</td>
<td>0.3076±0.0106</td>
<td>0.3025±0.0089</td>
<td>0.2848±0.0110</td>
<td>0.2740±0.0105</td>
<td>0.2702±0.0083</td>
<td>0.2695±0.0113</td>
<td><strong>0.2622±0.0089</strong></td>
</tr>
<tr>
<td>mirflkr</td>
<td>0.2125±0.0076</td>
<td>0.2254±0.0094</td>
<td>0.2196±0.0082</td>
<td>0.2031±0.0072</td>
<td>0.2003±0.0036</td>
<td>0.2013±0.0093</td>
<td><strong>0.1945±0.0067</strong></td>
</tr>
<tr>
<td>tmc2007</td>
<td>0.1757±0.0122</td>
<td>0.1590±0.0040</td>
<td>0.1643±0.0072</td>
<td>0.1422±0.0046</td>
<td>0.1421±0.0060</td>
<td>0.1339±0.0043</td>
<td><strong>0.1288±0.0038</strong></td>
</tr>
<tr>
<td>bookmarks</td>
<td>0.5115±0.0044</td>
<td>0.5319±0.0054</td>
<td>0.5408±0.0070</td>
<td>0.5165±0.0068</td>
<td>0.5337±0.0050</td>
<td>0.5242±0.0050</td>
<td><strong>0.5079±0.0042</strong></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>DELA against</th>
<th>DELA-sn</th>
<th>DELA-nn</th>
<th>DELA-fs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average precision</strong></td>
<td><strong>win [0.0020]</strong></td>
<td>win [0.0020]</td>
<td>win [0.0039]</td>
<td></td>
</tr>
<tr>
<td><strong>Macro-averaging AUC</strong></td>
<td>tie [0.1055]</td>
<td><strong>win [0.0098]</strong></td>
<td>win [0.0371]</td>
<td></td>
</tr>
<tr>
<td><strong>Hamming loss</strong></td>
<td><strong>win [0.0078]</strong></td>
<td>win [0.0469]</td>
<td>tie [0.0996]</td>
<td></td>
</tr>
<tr>
<td><strong>One error</strong></td>
<td><strong>win [0.0032]</strong></td>
<td>win [0.0020]</td>
<td>win [0.0098]</td>
<td></td>
</tr>
<tr>
<td><strong>Coverage</strong></td>
<td>win [0.0098]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td></td>
</tr>
<tr>
<td><strong>Ranking loss</strong></td>
<td>win [0.0039]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td></td>
</tr>
</tbody>
</table>

### 4.3. Further Analyses

#### 4.3.1. Ablation Studies

In ablation studies, we employ ten-fold cross validation on all the 10 data sets to validate the superiority of DELA against its variants. Table 4 summarizes the p-value statistics of the Wilcoxon signed-ranks test at 0.05 significance level and Table 6 shows the detailed experimental results in terms of Average precision.

Promising direction to facilitate multi-label classification with the strategy of label-specific features.

- Meanwhile, DELA achieves much better performance against other approaches based on label-specific features. Specifically, DELA is statistically superior to deep approach CLIF in terms of all evaluation metrics, and achieves statistically superior or at least comparable performance against PACA. These consistently better results demonstrate the effectiveness of our dual perspective for label-specific feature extraction.

#### Consideration on label-specific discriminative properties.

DELA accounts for each label's own discriminative properties via perturbing label-specific non-informative features in the shared representation z and inducing classifiers on these perturbed representations. To validate the effectiveness of the above consideration, we implement two variants named DELA-sn and DELA-nn. DELA-sn removes the identification process of label-specific non-informative features and merely perturbs z with a noise ε shared among all class labels, while DELA-nn further removes the noise and directly induces classifiers on the shared representation z. Results
Table 5. Summary of the Wilcoxon signed-ranks test for DELA against other comparing approaches at 0.05 significance level. \( p \)-values are shown in the brackets.

<table>
<thead>
<tr>
<th>DELA against</th>
<th>LIFT</th>
<th>LLSF</th>
<th>C2AE</th>
<th>MPVAE</th>
<th>CLIF</th>
<th>PACA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average precision</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
</tr>
<tr>
<td>Macro-averaging AUC</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>tie [0.0059]</td>
</tr>
<tr>
<td>Hamming loss</td>
<td>win [0.0352]</td>
<td>win [0.0313]</td>
<td>win [0.0020]</td>
<td>win [0.0078]</td>
<td>win [0.0117]</td>
<td>win [0.0020]</td>
</tr>
<tr>
<td>One-error</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0039]</td>
</tr>
<tr>
<td>Coverage</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
</tr>
<tr>
<td>Ranking loss</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
<td>win [0.0020]</td>
</tr>
</tbody>
</table>

Table 6. Predictive performance of DELA and its variants (mean±std. deviation) in terms of Average precision.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Average precision ± std. deviation</th>
<th>DELA</th>
<th>DELA-nn</th>
<th>DELA-nn</th>
<th>DELA-fs</th>
</tr>
</thead>
<tbody>
<tr>
<td>corel5k</td>
<td>0.3382 ± 0.0103</td>
<td>0.3338±0.0117</td>
<td>0.321±0.0135</td>
<td>0.3313±0.0124</td>
<td></td>
</tr>
<tr>
<td>rcv1-s1</td>
<td>0.6391 ± 0.0153</td>
<td>0.6365±0.0149</td>
<td>0.6114±0.0132</td>
<td>0.6355±0.0122</td>
<td></td>
</tr>
<tr>
<td>Corel16k-s1</td>
<td>0.3675 ± 0.0062</td>
<td>0.3627±0.0048</td>
<td>0.3610±0.0048</td>
<td>0.3635±0.0055</td>
<td></td>
</tr>
<tr>
<td>delicious</td>
<td>0.4082 ± 0.0053</td>
<td>0.4079±0.0057</td>
<td>0.3872±0.0066</td>
<td>0.3987±0.0065</td>
<td></td>
</tr>
<tr>
<td>iapricl2</td>
<td>0.4490 ± 0.0050</td>
<td>0.4401±0.0046</td>
<td>0.4255±0.0057</td>
<td>0.4257±0.0062</td>
<td></td>
</tr>
<tr>
<td>espagame</td>
<td>0.3162 ± 0.0059</td>
<td>0.3129±0.0049</td>
<td>0.3078±0.0059</td>
<td>0.3099±0.0056</td>
<td></td>
</tr>
<tr>
<td>mirlflickr</td>
<td>0.6960 ± 0.0043</td>
<td>0.6951±0.0069</td>
<td>0.6900±0.0057</td>
<td>0.6922±0.0061</td>
<td></td>
</tr>
<tr>
<td>tmc2007</td>
<td>0.8363 ± 0.0037</td>
<td>0.8300±0.0050</td>
<td>0.8302±0.0049</td>
<td>0.8304±0.0045</td>
<td></td>
</tr>
<tr>
<td>mediawiki</td>
<td>0.7883 ± 0.0049</td>
<td>0.7815±0.0055</td>
<td>0.7849±0.0040</td>
<td>0.7908±0.0071</td>
<td></td>
</tr>
<tr>
<td>bookmarks</td>
<td>0.5191 ± 0.0036</td>
<td>0.5104±0.0038</td>
<td>0.5067±0.0033</td>
<td>0.5123±0.0039</td>
<td></td>
</tr>
</tbody>
</table>

4.3.2. Parameter Sensitivity

Figure 1 gives an illustrative example on how the performance of DELA changes when the value of the trade-off parameter \( \beta \) changes. Degraded performance is witnessed in Table 4 show the consideration on label-specific discriminative properties is statistically effective.

Figure 2. Visualization of identified non-informative features in DELA on tmc2007. The \( k^{th} \) row denotes the indicator vector of the subset of non-informative features for label \( l_k \), where the blue one denotes a non-informative feature and the white one denotes a pertinent feature.

4.3.3. Visualization

Figure 2 gives an illustrative example on identified non-informative features for each class label. As can be seen, the subsets of non-informative features are quite different among class labels, which is essential to fully consider distinct discriminative properties of each class label. It is appealing to explore how to incorporate label correlations into the identification process of non-informative features, e.g. letting closely-related labels share more features (Huang et al., 2016b), which will be left for future work.

4.3.4. Complexity Analyses

Let \( b \) be the batch size and \( \hat{d} \) denote a proxy of the hidden dimensionalities of the network, the time complexity of DELA corresponds to \( O(bt\hat{d}^2) \) with \( t \) class labels. Figure B.1 illustrates the empirical training and test time of each comparing approach, which shows that DELA is comparable to existing approaches in time overhead.

5. Conclusion

In this paper, we propose to tackle the problem of label-specific feature learning for multi-label classification from a novel dual perspective, where distinct discriminative prop-

\[ \beta \]...
roperties of each class label are considered by endowing classifiers with immutability on identified label-specific non-informative features. Following this dual perspective, we present a perturbation-based approach DELA which learns to simultaneously identify non-informative features and make the discrimination process immutable to variations of these identified features via solving a probabilistically-relaxed expected risk minimization problem. Theoretical justification from an information theoretic view and comprehensive empirical studies against other well-established multi-label classification algorithms show the superiority of our approach. A nature direction for future work is to incorporate label correlations into the identification process of non-informative features and it is also interesting to explore alternative implementations towards the promising dual perspective for label-specific feature learning.

References


Dual Perspective of Label-Specific Feature Learning for Multi-Label Classification


Dual Perspective of Label-Specific Feature Learning for Multi-Label Classification

A. The Proof of Theorem 3.1

Theorem 3.1. For any random variant $c \sim p(c)$, the information bottleneck can be upper bounded as follows

$$-I(h; y) + \beta \cdot I(h; x) \leq \mathbb{E}_{p(x, y)} \mathbb{E}_{p(c)} \left[ \mathbb{E}_{p(h|x, c)} [- \log q(y|h)] + \beta \cdot KL(p(h|x, c)||q(h)) \right].$$

Proof.

$$-I(h; y) + \beta \cdot I(h; x) \leq \mathbb{E}_{p(x, y)} \left[ \mathbb{E}_{p(h|x)} [- \log q(y|h)] + \beta \cdot KL(p(h|x)||q(h)) \right]$$

$$= \mathbb{E}_{p(x, y)} \left[ \mathbb{E}_{p(c)p(h|x, c)} [- \log q(y|h)] + \beta \cdot \mathbb{E}_{p(c)p(h|x, c)} \left[ \log \frac{p(h|x)}{q(h)} \right] \right]$$

$$= \mathbb{E}_{p(x, y)} \left[ \mathbb{E}_{p(c)p(h|x, c)} [- \log q(y|h)] + \beta \cdot \mathbb{E}_{p(c)p(h|x, c)} \left[ KL(p(h|x, c)||q(h)) - KL(p(h|x, c)||p(h|x)) \right] \right]$$

where the first inequality is derived based on the variational approximation (Alemi et al., 2017) to the information bottleneck and the last inequality is derived based on non-negativity of the KL-divergence.

B. Empirical Running Time Comparison

Empirical running time of each comparing approach considered in the Comparative Studies part of the main body is further reported here for comprehensive evaluation. Figure B.1 illustrates the empirical training and test time of each comparing approach, which shows that DELA is comparable to existing approaches in time overhead.

*Figure B.1.* Running time (training/test) of each comparing approach on five benchmark data sets. For histogram illustration, the y-axis corresponds to the logarithm of running time.