Incorporating Label Embedding and Feature Augmentation for Multi-Dimensional Classification

Haobo Wang,[1*] Chen Chen,[1*] Weiwei Liu,[2†] Ke Chen,[1] Tianlei Hu,[1†] Gang Chen[1]
[1]Key Lab of Intelligent Computing Based Big Data of Zhejiang Province, Zhejiang University
[2]School of Computer Science, Wuhan University
{wanghaobo, cc33, htl, cg}@zju.edu.cn, liuweiwei863@gmail.com, chenk@cs.zju.edu.cn

Abstract

Feature augmentation, which manipulates the feature space by integrating the label information, is one of the most popular strategies for solving Multi-Dimensional Classification (MDC) problems. However, the vanilla feature augmentation approaches fail to consider the intra-class exclusiveness, and may achieve degenerated performance. To fill this gap, a novel neural network based model is proposed which seamlessly integrates the Label Embedding and Feature Augmentation (LEFA) techniques to learn label correlations. Specifically, based on attentional factorization machine, a cross correlation aware network is introduced to learn a low-dimensional label representation that simultaneously depicts the inter-class correlations and the intra-class exclusiveness. Then the learned latent label vector can be used to augment the original feature space. Extensive experiments on seven real-world datasets demonstrate the superiority of LEFA over state-of-the-art MDC approaches.

Introduction

Multi-Dimensional Classification (MDC) aims to deal with the problem where each data example is associated with multiple class variables. Due to its wide applications, MDC has attracted tremendous attention. For example, a piece of song can be annotated by various concepts like *emotions* and *instruments* (Turnbull et al. 2008); a document may be tagged by different types of labels such as *topic* and *mood* (Theeramunkong and Lertnattee 2002; Ortigosa-Hernández et al. 2012); a gene can be associated with different functions like *transcription* and *protein synthesis* (Barutçuoglu, Schapire, and Troyanskaya 2006). Figure 1 (Khosla et al. 2011) shows a typical multi-dimensional image classification scenario.

Formally, in an MDC problem, there are multiple class spaces $C_i = \{c_i^1, c_i^2, ..., c_i^{K_i}\}$ $(i = 1, 2, ..., d)$ where K_i is the number of possible class assignments and d is the number of classes. Then the output space is their Cartesian product $\mathcal{Y} = C_1 \times C_2 \times ... \times C_d$. Given a training dataset

Place	grassland river snowfield
House	yes no
Animal	cat dog fish
Weather	rainy cloudy sunny

Figure 1: An example of multi-dimensional classification scenario. The image is manually annotated by four class variables, each of which is multi-dimensional. The ground truth labels *snowfield*, *house (yes)*, *dog*, *sunny* are in red.

$\mathcal{D} = \{(\boldsymbol{x}_j, S_j)|1 \leq j \leq N\}$, each data point $\boldsymbol{x}_j \in \mathcal{X} \subseteq \mathbb{R}^m$ in $\mathcal{D}$ matches a set of labels $S_j = \{y_1^j, y_2^j, ..., y_d^j\} \in \mathcal{Y}$ where $y_i^j \in C_i$ $(1 \leq i \leq d)$. The goal of MDC is to build a classifier $h : \mathcal{X} \mapsto \mathcal{Y}$ which maps an instance vector to the output space. It is noteworthy that if $d = 1$, the problem degenerates to a single label classification problem. Furthermore, if $d > 1$ and $K_i = 2$ holds for all i, we obtain a multi-label classification problem.

Binary Relevance (BR) (Zhang and Zhou 2014) is one of the most popular methods for MDC problems which decomposes the multi-dimensional task into a set of multi-class classification problems. Despite its computational efficiency, BR neglects the cross correlations between class spaces. Therefore, BR works well on each single classification task but globally underperforms. Many effective techniques have been proposed to address this issue. Some methods (Bielza, Li, and Larrañaga 2011; Batal, Hong, and Hauskrecht 2013; Zhu, Liu, and Jiang 2016; Benjumeda, Bielza, and Larrañaga 2018) use a probabilistic graph model to learn a tree or graph structure of label correlations. Feature augmentation is another common strategy which models label dependencies in a manipulated feature space. Amongst them, Classifier Chains (CC) (Read et al. 2011) is the most typical one that feeds an augmented input vector

*Equal contribution.
†Corresponding authors.

$\tilde{\boldsymbol{x}}_j = [\boldsymbol{x}, y_1, y_2, ..., y_{j-1}]$ to train the j-th multi-class classifier in BR model. Nevertheless, CC is sensitive to the choice of label order and many techniques (Zaragoza et al. 2011; Liu and Tsang 2015) are proposed to alleviate this problem. Besides, a recent work KRAM (Jia and Zhang 2019) enriches the original feature space with k-Nearest Neighbor (kNN) technique. There are two main drawbacks in existing feature augmentation approaches: 1) the task of correlation extraction is completed by some simple base classifiers, such as Support Vector Machine (SVM) and Naïve Bayes. These simple classifiers are usually powerless on those datasets with complex label correlations and thus the generalization ability of these methods is limited in many applications. 2) they may wrongly learn correlations among intra-class labels, which leads to degenerated performance.

From another point of view, we can adapt specific multi-label classification techniques (Zhang and Zhou 2007; Hsu et al. 2009; Huang et al. 2016; 2018; Yeh et al. 2017) to solve multi-dimensional problems. This strategy is quite fascinating because we can reuse existing well-established multi-label classifiers. For instance, since the output space is usually highly sparse and correlative, label embedding (Yeh et al. 2017; Chen et al. 2019) might be a promising strategy. Unfortunately, these multi-label methods also fail to capture the intra-class exclusiveness in multi-dimensional settings.

To bridge these gaps, we combine Label Embedding technique and Feature Augmentation (LEFA) techniques to efficiently extract both the inter-class correlations and intra-class exclusiveness. LEFA follows a common label embedding paradigm that maps features and labels into a joint latent space to get their codewords. Firstly, we design a variant of Attentional Factorization Machine (AFM) (Xiao et al. 2017) to simultaneously extract the inter-class label correlation and preserve the intra-class exclusiveness for label embedding. Meanwhile, we present a Multi-Layer Perceptron (MLP) to encode features and propose a novel model to maximize the correlations between the resultant codewords. Finally, we manipulate the feature space by incorporating the projected labels for MDC problems.

The main contributions are summarized as follows,

- An effective deep model is proposed for multi-dimensional classification problems, which seamlessly integrates both Label Embedding and Feature Augmentation techniques (LEFA).
- Based on attentional factorization machine, we present a cross correlation aware network to simultaneously depict the inter-class dependencies and the intra-class exclusiveness for MDC tasks.
- Comprehensive experiments over seven real-world datasets demonstrate that LEFA outperforms other state-of-the-art MDC classifiers.

The rest of this paper is organized as follows. In the next section, we provide a detailed description of our network architecture. After that, the results of empirical studies are reported. Then, we discuss the related works of our method. Concluding remarks are provided in the last section.

The LEFA Approach

As mentioned in the first section, each class space is composed of multiple nominal labels which are hard to be used for computation. The most popular strategy is to transform each nominal label y_i^j in the original label sets S_j to a one-hot binary vector $\boldsymbol{z}_i^j \in \mathbb{R}^{K_i}$. Nevertheless, the transformed label space can be highly sparse and directly exploiting complex label correlations is intractable. Consequently, we combine Label Embedding and Feature Augmentation (LEFA) techniques. Firstly, based on AFM, we present a Cross Correlation Aware Network (C2AN) which projects the features and labels into a joint low-dimensional space. It is worth pointing out that C2AN can depict inter-class correlations with intra-class exclusiveness being preserved. Then, the original feature space is enriched by incorporating the embedded label vectors. Finally, the manipulated feature vectors can be fed into any off-the-shelf MDC classifiers to improve the predictive performance.

Cross Correlation Aware Network

In this subsection, we introduce the proposed C2AN model layer by layer. The detailed network architecture is shown in Figure 2.

Label Encoding Network Traditional label embedding approaches have two main limitations: 1) the encoder for label vectors are either linear model (Hsu et al. 2009; Chen and Lin 2012) or simple neural networks (Yeh et al. 2017), and hence may not be able to handle the sparse label space and complicated class space dependencies; 2) they are designed for multi-label learning tasks, and thus ignores the exclusiveness between inter-class labels.

To remedy these problems, we apply a variant of Attentional Factorization Machine to embed labels. Our AFM based model has three main advantages: 1) it is a powerful neural network based model to extract label correlations; 2) attention mechanism enables the label interactions to contribute differently to the feature augmentation; 3) as a member of Factorization Machines (FMs) (Rendle 2012) family, it works well in sparse setting. In what follows, we elaborate the design of label encoding network layer by layer.

Cross Interaction Layer Take a set of one-hot label vectors $\mathcal{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_d\}$ as the input, the first layer extracts pair-wise interactions between inter-class labels. In our model, a set of embedding matrices $\mathcal{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, ..., \boldsymbol{V}_d\}$ ($\boldsymbol{V}_i = [\boldsymbol{v}_i^1, \boldsymbol{v}_i^2, ..., \boldsymbol{v}_i^{K_i}] \in \mathbb{R}^{t \times K_i}$) are used to transform each label to a dense vector representation where t is the size of embedding vectors. Denote the p-th entry of $\boldsymbol{z}_i$ by z_i^p. Then we can get a set of embedding vectors,

$$f_{CI}(\mathcal{Z}) = \{(\boldsymbol{v}_i^p \odot \boldsymbol{v}_j^q) z_i^p z_j^q\}_{(i,j,p,q)\in\varepsilon} \tag{1}$$

where $\odot$ is the element-wise product and $\varepsilon = \{(i,j,p,q) | 1 \le p \le K_i, 1 \le q \le K_j, 1 \le i, j \le d, i \ne j\}$. Note that in the training phase, the input vectors are one hot. Hence, we can simplify the computation of embedding set by removing the zero vector,

$$\bar{f}_{CI}(\mathcal{Z}) = \{\boldsymbol{V}_i \boldsymbol{z}_i \odot \boldsymbol{V}_j \boldsymbol{z}_j\}_{(i,j,p,q)\in\varepsilon} \tag{2}$$

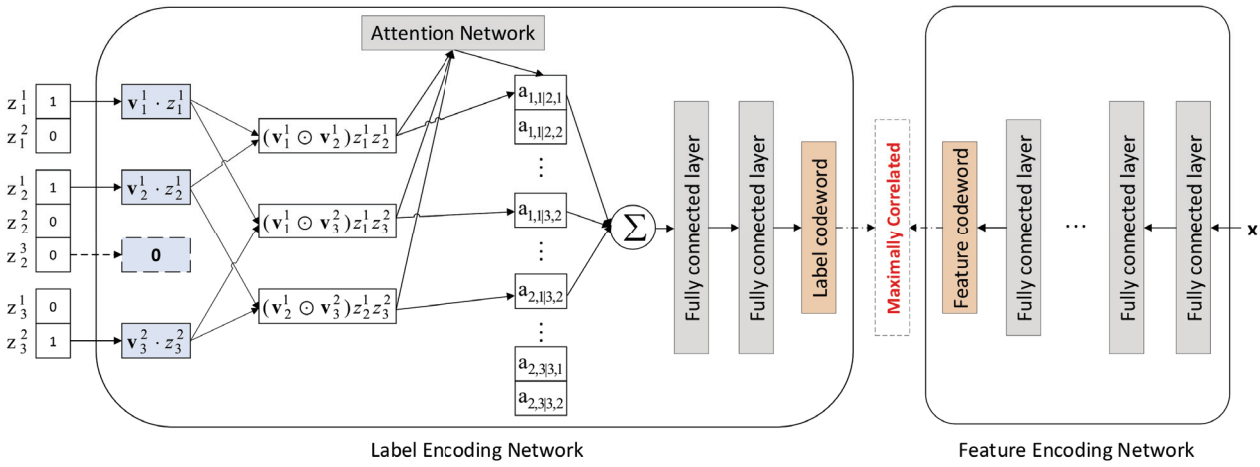


Figure 2: The neural network architecture of our proposed C2AN. For simplicity, most of the zero vectors in the cross interaction layer are omitted except $z_2^3 \cdot \boldsymbol{v}_2^3$. Note that we only perform the sum-pooling operation in the attention-based pooling layer.

Remark that existing methods like KRAM (Jia and Zhang 2019) usually model the correlations between all the labels. Our model will not involve any redundant embedding vector between the intra-class labels and thus explore only cross class space correlations.

Attention-based Pooling Layer As presented in (Xiao et al. 2017), attention mechanism has shown a promising result since it allows different correlations contribute differently when compressing the input to a single dense representation. To efficiently handle the sparse setting, AFM introduces an attention network to parameterize the attention scores. Similar to the Cross Interaction layer, LEFA preserves only the attention scores for the inter-class labels,

$$a_{i,p|j,q} = \frac{a'_{i,p|j,q}}{\sum_{(\tilde{i},\tilde{j},\tilde{p},\tilde{q})\in\varepsilon} a'_{\tilde{i},\tilde{p}|\tilde{j},\tilde{q}}} \tag{3}$$

Here the unnormalized attention weight is given by,

$$a'_{i,p|j,q} = \boldsymbol{h}^\top \sigma(\boldsymbol{W}(\boldsymbol{v}_i^p \odot \boldsymbol{v}_j^q) z_i^p z_j^q + \boldsymbol{b}) \tag{4}$$

where $\boldsymbol{h} \in \mathbb{R}^r$, $\boldsymbol{W} \in \mathbb{R}^{r\times t}$ and $\boldsymbol{b} \in \mathbb{R}^r$ are global attention parameters. r denotes hidden layer size of the attention network. After all the attention scores are learned, we apply a weighted sum-pooling operation on the embedding vectors to get a single dense vector,

$$f_{AP}(f_{CI}(\mathcal{Z})) = \sum_{(i,j,p,q)\in\varepsilon} a_{i,p|j,q}(\boldsymbol{v}_i^p \odot \boldsymbol{v}_j^q) z_i^p z_j^q \tag{5}$$

Compared with vanilla AFM, C2AN pays more attention to the label correlation extraction. Hence, the regression part is neglected to avoid introducing useless parameters.

Fully Connected Layers Above the attention-based pooling layer are two simple fully connected layers. As previous layers concentrate on pair-wise label interactions, these layers further learn higher-order correlations. Denote the output vector of attention layer by $\boldsymbol{e} = f_{AP}(f_{CI}(\mathcal{Z}))$ and we can get our unactivated latent label vector by,

$$\boldsymbol{c}_y = \hat{\boldsymbol{W}}_o \sigma(\hat{\boldsymbol{W}}\boldsymbol{e} + \hat{\boldsymbol{b}}) + \hat{\boldsymbol{b}}_o \tag{6}$$

where $\boldsymbol{c}_y \in \mathbb{R}^u$ is the label codeword. Here $\hat{\boldsymbol{W}}_o$, $\hat{\boldsymbol{W}}$ are weight matrices and $\hat{\boldsymbol{b}}_o$, $\hat{\boldsymbol{b}}$ are bias vectors.

Remark that the obtained latent label vectors not only depict the inter-class correlations, but also preserve the intra-class label exclusiveness. Hence, it can be employed to augment the original feature space with the label correlation extracted in advance.

Feature Encoding Network Inspired by (Yeh et al. 2017), we use a Multi-Layer Perceptron to encode features. Here the feature vectors are fed into a set of hidden layers and each layer can be customized to discover certain latent structures. Generally speaking, the input is linearly transformed in each layer and then activated by a non-linear function such as Rectifier (ReLU), sigmoid and so on. Assume that there are l hidden layers. Then a feature vector $\boldsymbol{x}$ is encoded by,

$$\begin{aligned} \boldsymbol{s}_1 &= \sigma_1(\boldsymbol{W}_1\boldsymbol{x} + \mathbf{b}_1) \\ &\cdots \\ \boldsymbol{s}_l &= \sigma_l(\boldsymbol{W}_l\boldsymbol{s}_{l-1} + \mathbf{b}_l) \\ \boldsymbol{c}_x &= \boldsymbol{W}_o\boldsymbol{s}_l + \boldsymbol{b}_o \end{aligned} \tag{7}$$

where $\boldsymbol{W}_i$, $\boldsymbol{b}_i$, σ_i and $\boldsymbol{s}_i$ are weight matrix, bias, activation function and output vector of i-th layer repectively. Here the codeword $\boldsymbol{c}_x \in \mathbb{R}^u$ is obtained in the last layer without activation where u is the size of latent space.

Optimization Now we present our objective function which enables the codewords of features and labels to be maximally correlated. Given a pair of codewords $(\boldsymbol{c}_x, \boldsymbol{c}_y)$, the maximization problem can be expressed as the following,

$$\underset{\Theta}{\operatorname{argmax}} \frac{\boldsymbol{c}_x^\top \boldsymbol{c}_y}{||\boldsymbol{c}_x|| \cdot ||\boldsymbol{c}_y||} \tag{8}$$

where Θ is the set of parameters and $||\cdot||$ is l_2-norm. Since scaling the codewords will not change the result, we can reformulate it by adding two constraints and converting it to a

minimization problem,

$$\begin{aligned} &\underset{\Theta}{\operatorname{argmin}} \frac{1}{2}||\boldsymbol{c}_x - \boldsymbol{c}_y||^2 \\ &s.t. \quad ||\boldsymbol{c}_x|| = 1, \quad ||\boldsymbol{c}_y|| = 1 \end{aligned} \tag{9}$$

Here the objective is modified because $\operatorname{argmax}_\Theta \boldsymbol{c}_x^\top \boldsymbol{c}_y = \operatorname{argmin}_\Theta -2\boldsymbol{c}_x^\top \boldsymbol{c}_y = \operatorname{argmin}_\Theta ||\boldsymbol{c}_x - \boldsymbol{c}_y||^2$, since $||\boldsymbol{c}_x|| = ||\boldsymbol{c}_y|| = 1$. However, it is intractable to directly solve this optimization problem with two hard constraints. Following (Yeh et al. 2017), we relax the constraints and involve two penalty terms to get our final objective function,

$$\begin{aligned} \mathcal{L} = &\sum_{i=1}^{N} (\frac{1}{2}||\boldsymbol{c}_x^i - \boldsymbol{c}_y^i||^2 \\ &+ \lambda_1(||\boldsymbol{c}_x^i|| - 1)^2 + \lambda_2(||\boldsymbol{c}_y^i|| - 1)^2) + \eta \sum_{\theta \in \Theta} ||\theta||^2 \end{aligned} \tag{10}$$

where λ_1 and λ_2 are multipliers associated with the penalty terms. Here we also add a regularization term with the trade-off parameter η to prevent overfitting.

Since the objective function is clearly defined, we can effectively optimize our model using gradient based techniques such as Stochastic Gradient Descent (SGD).

Feature Augmentation

In the second stage, we manipulate the feature space by combining the projected labels and original features. The augmented dataset $\tilde{\mathcal{D}}$ can be formulated as follows,

$$\tilde{\mathcal{D}} = \{(\tilde{\boldsymbol{x}}_j, S_j)|1 \leq j \leq N\}, \tilde{\boldsymbol{x}}_j = [\boldsymbol{x}_j, \boldsymbol{c}_y^j] \tag{11}$$

Thereafter, we induce an MDC predictive function $f : \tilde{\mathcal{X}} \mapsto \mathcal{Y}$ from the new dataset where the manipulated feature space $\tilde{\mathcal{X}} \subseteq \mathbb{R}^{m+u}$ is the Cartesian Production of the original feature space and the latent label space. As a meta strategy for MDC, we can employ any off-the-shelf MDC algorithm in LEFA, e.g. Binary Relevance (Zhang and Zhou 2014), Classifier Chains (Read et al. 2011; Liu, Tsang, and Müller 2017), and so on.

There is still one main concern that in the testing phase, the ground truth labels are invisible. Inspired by KRAM (Jia and Zhang 2019), LEFA utilizes kNN technique to estimate a set of label vectors $\mathcal{Z}^* = \{\boldsymbol{z}_1^*, \boldsymbol{z}_2^*, ..., \boldsymbol{z}_d^*\}$ for an testing instance $\boldsymbol{x}^*$. Assume that $\mathcal{N}(\boldsymbol{x}^*)$ is the index set of k-nearest neighbors identified for $\boldsymbol{x}^*$ among $\mathcal{D}$. After that, a weighted average strategy is used to obtain each label vector,

$$\boldsymbol{z}^* = \sum_{j \in \mathcal{N}(\boldsymbol{x}^*)} (1 - \frac{d(\boldsymbol{x}^*, \boldsymbol{x}_j)}{\sum_{k \in \mathcal{N}(\boldsymbol{x}^*)} d(\boldsymbol{x}^*, \boldsymbol{x}_k)}) \cdot \boldsymbol{z}^j \tag{12}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the original feature space. Then we feed it into the label encoding network to obtain a embedded label vector $\boldsymbol{c}_y^*$. Note that in the testing phase, we have to compute all the embedding vectors $f_{CI}(\mathcal{Z})$ since the input label vectors are dense. Since we have removed all the redundant embedding interaction operation between intra-class labels, their exclusiveness can be fully preserved. Though the estimation phase is at a coarse granularity, our latent label vector can provide finer-grained semantic information about the labels. By concatenating $\boldsymbol{x}^*$ and $\boldsymbol{c}_y^*$, we can get the augmented vector $\tilde{\boldsymbol{x}}^*$ and predict the corresponding label set $S^* = f(\tilde{\boldsymbol{x}}^*)$.

It is worth noting that the most recent work KRAM enriches the original feature space by simply counting statistics on the class membership of neighboring MDC examples (the unaveraged $\boldsymbol{z}_i^*$) both in the training and testing phases. Compared to KRAM, LEFA has three main advantages. Firstly, the latent vector can provide preciser semantic information than the label set induced by kNN in the testing phase. Secondly, the label correlations are extracted in the augmentation phase instead of directly being induced by the predictive function f. In the meantime, a powerful AFM based neural

Algorithm 1 The pseudo-code of LEFA

Input: Training dataset $\mathcal{D} = \{(\boldsymbol{x}_j, S_j)|1 \leq j \leq N\}$, dimension parameters u, r and t, penalty parameters λ_1 and λ_2, number of nearest neighbors k, testing instance $\boldsymbol{x}^*$

Output: the predicted label set S^* for $\boldsymbol{x}^*$

Transform each label set S_j to a set of binary one hot vectors $\mathcal{Z}_j$
Initialize all the trainable parameters in C2AN with random values from a given Gaussian distribution
repeat
  Randomly sample an example $(\boldsymbol{x}, \mathcal{Z})$ from $\mathcal{D}$
  Feed the instance vector $\boldsymbol{x}$ into the feature encoding network to get its codeword $\boldsymbol{c}_x$
  Feed $\mathcal{Z}$ into the Cross Interaction Layer to get the output set $\bar{f}_{CI}(\mathcal{Z})$ by Eq. (2)
  Feed the embedding vectors into the attention network to get the attention scores $a_{i,p|j,q}$ by Eq. (3) and (4)
  Feed $\bar{f}_{CI}(\mathcal{Z})$ and attention scores $a_{i,p|j,q}$ into the pooling layer to get the dense vector $\boldsymbol{e}$ by Eq. (5)
  Feed $\boldsymbol{e}$ into the fully connected layers to get the label codeword $\boldsymbol{c}_y$ by Eq. (6)
  Calculate the correlation loss $\mathcal{L}$ by Eq. (10)
  Update the trainable parameters with SGD algorithm
until *Converge*
for $j = 1$ to N **do**
  Feed $\mathcal{Z}_j$ into the label encoding network to get the codeword $\boldsymbol{c}_y^j$
  Set $\tilde{\boldsymbol{x}}_j = [\boldsymbol{x}_j, \boldsymbol{c}_y^j]$
end for
Form the augmented MDC training dataset $\tilde{\mathcal{D}} = \{(\tilde{\boldsymbol{x}}_j, S_j)|1 \leq j \leq N\}$
Induce an MDC predictive function $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\tilde{\mathcal{D}}$
Estimate a set of label vector $\mathcal{Z}^*$ of $\boldsymbol{x}^*$ from its k-nearest neighbors according to Eq. (12)
Feed $\mathcal{Z}^*$ into the label encoding network to get the codeword $\boldsymbol{c}_y^*$ and get the augmented feature by $\tilde{\boldsymbol{x}}^* = [\boldsymbol{x}^*, \boldsymbol{c}_y^*]$
Return $S^* = f(\tilde{\boldsymbol{x}}^*)$

network is employed. Thus, LEFA can deal with complex label correlation hierarchy and highly sparse label space. Finally, the intra-class exclusiveness is depicted. Empirical study also demonstrates that LEFA outperforms state-of-the-art MDC approaches.

Experiments

In this section, we evaluate the performance of the proposed method on seven real-world datasets. All the computations are performed on a same workstation with an i7-5930K CPU, a TITAN Xp GPU and 64GB main memory running Linux platform.

Datasets

For comprehensive performance evaluation, a total of seven datasets are employed. The first four datasets are collected from UCI repository (Dheeru and Karra Taniskidou 2017):

- Bridges estimates bridge properties from specific constraints, e.g. the used materials, span properties, bridge types and so on.
- Water Quality determines the plant and animal species in Slovenian rivers. Each object is equipped with 14 labels. The first 7 labels focuses on the plant types and the others concentrate on the animal species.
- Flare deals with the problem of predicting the number of times that certain types of solar flare occurred within 24 hours period. There are two different datasets Flare1 and Flare2, both of which has 3 class variables. Here Flare2 contains more instances and also more labels in each class space.

Unfortunately, there are not many publicly available standardised MDC datasets yet. Following the experimental setting in (Read, Bielza, and Larrañaga 2014), we boost our collection by introducing three real-world multi-label datasets, including two medium-sized datasets Emotions (Trohidis et al. 2008) , Scene (Boutell et al. 2004) and a large-scale dataset TMC2007 (Srivastava and Zane-Ulman 2005). Here, each class space corresponds to a binary-valued variable that indicates whether a label is relevant to the example or not.

In this paper, we conduct 5-fold cross-validation on these datasets and the mean metric values with standard deviations are reported. The statistics of the seven datasets are summarized in Table 1.

Table 1: Statistics of the experimental datasets.

Datasets	N	d	K	$m^\dagger$
Bridges	108	5	2-6	7x
Water Quality	1,060	14	4	16n
Flare1	323	3	2-4	10x
Flare2	1,066	3	3-8	10x
Emotions	593	6	2	72n
Scene	2,407	6	2	294n
TMC2007	28,596	22	2	500b

† n, b and x denote numeric, binary, and nominal features respectively.

Comparison Approaches

In this paper, we compare LEFA with three well-established MDC methods and two state-of-the-art embedding based multi-label methods:

- Binary Relevance (BR) (Zhang and Zhou 2014): BR is the most intuitive way for MDC tasks, which predicts each class variable by decomposing the MDC task to a set of independent multi-class problems.
- Ensemble Classifier Chains (ECC) (Read et al. 2011): To alleviate the problem of label order sensitivity in Classifier Chains (CC), ECC generates several different chains with randomly reordered labels. Then, the class variables are predicted by voting.
- KRAM (Jia and Zhang 2019): By utilizing the popular kNN techniques, KRAM enriches the feature space with specific counting statistics on the class membership of neighboring MDC examples.
- CPLST (Chen and Lin 2012): CPLST is a popular label embedding approach, which combines the concepts of principal component analysis and canonical correlation analysis for better correlation extraction.
- C2AE (Yeh et al. 2017): C2AE is the first neural network based label embedding approach for multi-label problems, which integrates the autoencoder and the deep canonical analysis techniques.

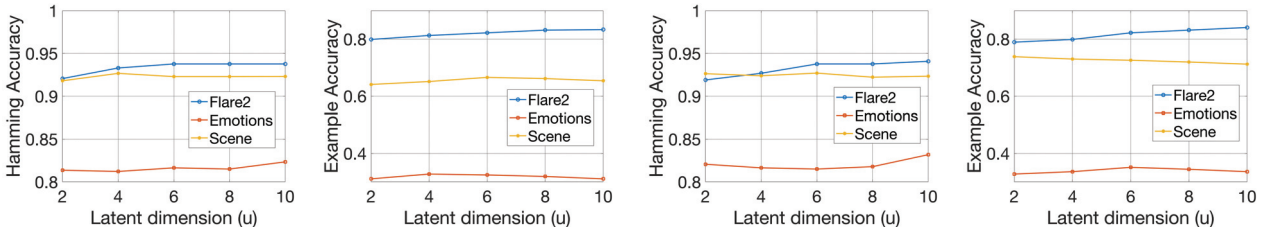
It is noteworthy that KRAM and LEFA are meta approaches, which can integrate any off-the-shelf MDC methods to improve the generalization ability. As a result, BR and ECC are used to instantiate practical models. The resultant approaches are denoted by KRAM-BR, KRAM-ECC, LEFA-BR and LEFA-ECC respectively.

For our proposed method, the latent dimension is empirically set as 10 and the size of embedding vectors is fixed to 64. The hidden dimensions of the attention, label and feature networks are 8, 64, and $[64, 64]$ respectively. We use ReLU as our activation function in C2AN. Both the penalty parameters λ_1 and λ_2 are empirically set as 1. The regularization parameter is set as $\eta = 0.01$. Moreover, we set the learning rate of gradient descent algorithm as 0.001.

For baselines, we use Support Vector Machine (SVM), which is implemented by Scikit-learn (Pedregosa et al. 2011), as the base classifier for BR, ECC and our methods. Specifically, for the medium-sized datasets, the radial basis function kernel is used to handle the non-linear separable cases. As for the large scale dataset TMC2007, linear kernel and stochastic gradient descent algorithm are employed. We set the number of nearest neighbors as $k = 10$ for all the kNN based approaches. For CPLST, we take the first 5 principle components. The loss leveraging parameter of C2AE is set as $\alpha = 2$. Other parameters in the baselines are set to their recommended values. Finally, since CPLST and C2AE are multi-label classifiers, we adapt them to the multi-dimensional setting by preserving the labels with maximum scores in each class as the output labels.

Table 2: Predictive performance comparison on seven real-world datasets.

Datasets	Hamming Accuracy↑							
	BR	KRAM-BR	LEFA-BR	ECC	KRAM-ECC	LEFA-ECC	CPLST	C2AE
Bridges	.785±.032	**.840**±.041	.816±.015	**.691**±.016	.675±.022	.684±.024	.618±.056	.722±.030
Water Quality	.644±.006	.643±.009	**.658**±.008	.618±.030	.636±.021	**.651**±.005	.642±.005	.641±.012
Flare1	.917±.003	.936±.009	**.947**±.009	.917±.013	.926±.017	**.948**±.020	.919±.014	.904±.010
Flare2	.922±.003	.929±.011	**.938**±.005	.928±.004	.928±.008	**.941**±.009	.920±.005	.924±.006
Emotions	.791±.008	.786±.010	**.818**±.012	.789±.006	.791±.016	**.809**±.013	.791±.010	.577±.025
Scene	.910±.006	.918±.003	**.924**±.003	.915±.004	.916±.003	**.923**±.003	.889±.002	.723±.009
TMC2007	.939±.001	.941±.002	**.942**±.001	.940±.002	**.942**±.003	.941±.001	.934±.001	.901±.010
Datasets	Example Accuracy↑							
	BR	KRAM-BR	LEFA-BR	ECC	KRAM-ECC	LEFA-ECC	CPLST	C2AE
Bridges	.327±.081	**.445**±.073	.427±.061	.127±.075	.100±.038	**.136**±.056	.109±.069	.246±.110
Water Quality	**.010**±.006	.010±.007	.009±.003	**.012**±.005	.008±.003	.011±.002	.008±.005	.007±.003
Flare1	.809±.017	.834±.012	**.883**±.020	.800±.030	.825±.023	**.873**±.039	.815±.021	.792±.008
Flare2	.784±.010	.813±.025	**.833**±.010	.799±.016	.808±.021	**.841**±.021	.785±.018	.802±.026
Emotions	.274±.041	.266±.030	**.306**±.028	.286±.048	.304±.033	**.321**±.042	.252±.049	.077±.066
Scene	.612±.032	.652±.010	**.654**±.012	.690±.014	.698±.020	**.712**±.016	.481±.014	.155±.028
TMC2007	.291±.004	**.304**±.004	.303±.001	.309±.002	.314±.004	**.316**±.000	.207±.003	.054±.029



(a) LEFA-BR (b) LEFA-ECC

Figure 3: Performance of LEFA changes as the latent dimension u changes from 2 to 10 on three datasets with different base classifiers.

Performance Measurements

Following the experimental setting in (Jia and Zhang 2019), we consider two popular metrics to evaluate the predictive performance of all the methods:

- Hamming Accuracy:

$$\text{HAccuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{|S_i \cap S_i^*|}{d} \tag{13}$$

Here S_i^* is the predicted label set for the i-th data example. The hamming accuracy computes the classification accuracy (Wang et al. 2018; 2019b) on each class variable and takes the average. Here $|\cdot|$ denotes the cardinality of a set and $\cap$ is the intersection of two sets.

- Example Accuracy:

$$\text{EAccuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(S_i, S_i^*) \tag{14}$$

The example accuracy regards the label set as a single classification problem that is either fully correct or incorrect. Here $\mathbb{I}(A, B)$ is an indicator function returning 1 if $A = B$ and 0 otherwise.

Experimental Results

Table 2 summarizes the predictive performance of all the methods on four multi-dimensional datasets and three multi-label datasets. Figure 3 reports the parameter sensitivities of our methods to the latent dimensions u.

From the experimental results, we observe that:

- LEFA generally achieves the best performance. Take the Flare2 dataset as an example, in terms of hamming accuracy and example accuracy, LEFA-BR improves the best results of baselines (except LEFA-ECC) by 1.0%, 2.5%, and LEFA-ECC improves the best results of baselines (except LEFA-BR) by 1.3%, 3.4% respectively. These results demonstrate the superiority of LEFA.
- BR underperforms other methods due to the lack of exploiting class space correlations.
- BR and ECC are much inferior to their KRAM and LEFA couterparts, which indicates the effectiveness of feature augmentation.
- KRAM and LEFA are the most successful on these datasets. However, LEFA obtain a better performance for two reasons: 1) C2AN preserves the exclusiveness between intra-class labels; 2) LEFA extracts label correla-

tions before augmentation. Thus, LEFA enables simple MDC classifiers like BR and ECC to handle complicated label correlation hierarchy.

- C2AE and CPLST show the worst performance on some datasets. Because they neglect the exclusiveness between the intra-class labels and hence, they are unsuitable for MDC tasks.
- LEFA achieves relatively stable performance with different values of latent dimension u.

Related Work

Multi-Label Learning

In multi-label learning, each data example is equipped with a set of binary labels. Existing MLL algorithms can be roughly categorized into three groups based on the thought of *degree of label correlations*. First-order methods, e.g. binary relevance (Zhang and Zhou 2014) and ML-kNN (Zhang and Zhou 2007), are the most straightforward that assume independencies among labels. Second-order methods usually involve ranking technique (Liu et al. 2018), or learn a pair-wise correlation matrix (Huang et al. 2016; 2018). High-order approaches can fully utilize label correlations through various ways. For example, label powerset based algorithms (Tsoumakas, Katakis, and Vlahavas 2011; Liu and Tsang 2017) transforms the MLL task to some multi-class classification problems by label combination; feature augmentations based methods (Read et al. 2011; Wang et al. 2019a; Liu, Tsang, and Müller 2017) augment the original feature space by previous elicited labels; label embedding approaches (Hsu et al. 2009; Yeh et al. 2017; Chen et al. 2019) jointly embeds the features and labels to a same latent space.

Multi-Dimensional Classification

Multi-dimensional classification (MDC) aims to assign each object to multiple class spaces. It is a generalization of multi-label learning that allows each class variable to have more than two values. Transforming the MDC task to a multi-label learning problem is an appealing strategy. However, compared to MLL problems, the label correlations in MDC are more complicated, because inter-class labels can be correlated to each other, but intra-class labels are exclusiveness to each other. To cope with this issue, many works are proposed. For instance, probabilistic graph models (Bielza, Li, and Larrañaga 2011; Batal, Hong, and Hauskrecht 2013; Zhu, Liu, and Jiang 2016; Benjumeda, Bielza, and Larrañaga 2018) usually learn a tree or graph structure of label correlations across the class spaces. Feature augmentation (Zaragoza et al. 2011; Jia and Zhang 2019) is another effective and has been adopted by many approaches. One recent work (Ma and Chen 2018) also explores to learn a distance metric for MDC problems. The main drawback of existing feature augmentation based MDC methods is that they employ simple base classifiers to extract label correlations, which leads to degenerated performance.

Conclusion

Recently, multi-dimensional classification problems have attracted huge attention from the research community. In this work, we propose a novel deep model LEFA which seamlessly integrate the Label Embedding and Feature Augmentation techniques for MDC tasks. Based on attentional factorization machine, a cross-correlation aware network is presented which maps the features and labels into a joint low-dimensional space such that they are maximally correlated. Due to the peculiarity of AFM, the embedded labels not only depict the inter-class label correlation, but preserve the exclusiveness of intra-class labels. Then, we augment the original feature space using the latent label vectors, which can provide discriminative information to the original feature space. Empirical study on seven real-world datasets shows that the proposed method generally outperforms other state-of-the-art MDC approaches.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 61976161 and the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies). The author Haobo Wang completed this work when he is in ZJUINA.

References

Barutçuoglu, Z.; Schapire, R. E.; and Troyanskaya, O. G. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics* 22(7):830–836.

Batal, I.; Hong, C.; and Hauskrecht, M. 2013. An efficient probabilistic framework for multi-dimensional classification. In *CIKM*, 2417–2422.

Benjumeda, M.; Bielza, C.; and Larrañaga, P. 2018. Tractability of most probable explanations in multidimensional bayesian network classifiers. *Int. J. Approx. Reasoning* 93:74–87.

Bielza, C.; Li, G.; and Larrañaga, P. 2011. Multi-dimensional classification with bayesian networks. *Int. J. Approx. Reasoning* 52(6):705–727.

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

Chen, Y., and Lin, H. 2012. Feature-aware label space dimension reduction for multi-label classification. In *NeurIPS*, 1538–1546.

Chen, C.; Wang, H.; Liu, W.; Zhao, X.; Hu, T.; and Chen, G. 2019. Two-stage label embedding via neural factorization machine for multi-label classification. In *AAAI*, 3304–3311.

Dheeru, D., and Karra Taniskidou, E. 2017. UCI machine learning repository.

Hsu, D. J.; Kakade, S.; Langford, J.; and Zhang, T. 2009. Multi-label prediction via compressed sensing. In *NeurIPS*, 772–780.

Huang, J.; Li, G.; Huang, Q.; and Wu, X. 2016. Learning label-specific features and class-dependent labels for

multi-label classification. *IEEE Trans. Knowl. Data Eng.* 28(12):3309–3323.

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

Jia, B.-B., and Zhang, M.-L. 2019. Multi-dimensional classification via knn feature augmentation. In *AAAI*.

Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Fei-Fei, L. 2011. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*.

Liu, W., and Tsang, I. W. 2015. On the optimality of classifier chain for multi-label classification. In *NeurIPS*, 712–720.

Liu, W., and Tsang, I. W. 2017. Making decision trees feasible in ultrahigh feature and label dimensions. *Journal of Machine Learning Research* 18:81:1–81:36.

Liu, Y.; Wen, K.; Gao, Q.; Gao, X.; and Nie, F. 2018. SVM based multi-label learning with missing labels for image annotation. *Pattern Recognition* 78:307–317.

Liu, W.; Tsang, I. W.; and Müller, K. 2017. An easy-to-hard learning paradigm for multiple classes and multiple labels. *Journal of Machine Learning Research* 18:94:1–94:38.

Ma, Z., and Chen, S. 2018. Multi-dimensional classification via a metric approach. *Neurocomputing* 275:1121–1131.

Ortigosa-Hernández, J.; Rodríguez, J. D.; Alzate, L.; Lucania, M.; Inza, I.; and Lozano, J. A. 2012. Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. *Neurocomputing* 92:98–115.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

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

Read, J.; Bielza, C.; and Larrañaga, P. 2014. Multi-dimensional classification with super-classes. *IEEE Trans. Knowl. Data Eng.* 26(7):1720–1733.

Rendle, S. 2012. Factorization machines with libfm. *ACM TIST* 3(3):57:1–57:22.

Srivastava, A. N., and Zane-Ulman, B. 2005. Discovering recurring anomalies in text reports regarding complex space systems. In *Aerospace conference, 2005 IEEE*, 3853–3862.

Theeramunkong, T., and Lertnattee, V. 2002. Multi-dimensional text classification. In *COLING*.

Trohidis, K.; Tsoumakas, G.; Kalliris, G.; and Vlahavas, I. P. 2008. Multi-label classification of music into emotions. In *ISMIR*, 325–330.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. P. 2011. Random k-labelsets for multilabel classification. *IEEE Trans. Knowl. Data Eng.* 23(7):1079–1089.

Turnbull, D.; Barrington, L.; Torres, D. A.; and Lanckriet, G. R. G. 2008. Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio, Speech & Language Processing* 16(2):467–476.

Wang, J.; Tian, F.; Yu, H.; Liu, C. H.; Zhan, K.; and Wang, X. 2018. Diverse non-negative matrix factorization for multiview data representation. *IEEE Trans. Cybernetics* 48(9):2620–2632.

Wang, H.; Liu, W.; Zhao, Y.; Zhang, C.; Hu, T.; and Chen, G. 2019a. Discriminative and correlative partial multi-label learning. In *IJCAI*, 3691–3697.

Wang, J.; Suzuki, A.; Xu, L.; Tian, F.; Yang, L.; and Yamanishi, K. 2019b. Orderly subspace clustering. In *AAAI*, 5264–5272.

Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; and Chua, T. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *IJCAI*, 3119–3125.

Yeh, C.; Wu, W.; Ko, W.; and Wang, Y. F. 2017. Learning deep latent space for multi-label classification. In *AAAI*, 2838–2844.

Zaragoza, J. H.; Sucar, L. E.; Morales, E. F.; Bielza, C.; and Larrañaga, P. 2011. Bayesian chain classifiers for multidimensional classification. In *IJCAI*, 2192–2197.

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

Zhang, M., and Zhou, Z. 2014. A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.* 26(8):1819–1837.

Zhu, M.; Liu, S.; and Jiang, J. 2016. A hybrid method for learning multi-dimensional bayesian network classifiers based on an optimization model. *Appl. Intell.* 44(1):123–148.