Multi-label Classification with Feature-aware Cost-sensitive Label Embedding

Hsien-Chun Chiu
Department of CSIE, National Taiwan University
Taipei, Taiwan
r04922004@csie.ntu.edu.tw

Hsuan-Tien Lin
Department of CSIE, National Taiwan University
Taipei, Taiwan
htlin@csie.ntu.edu.tw

***Abstract*—Multi-label classification (MLC) is an important learning problem where each instance is annotated with multiple labels. Label embedding (LE) is an important family of methods for MLC that extracts and utilizes the latent structure of labels towards better performance. Within the family, feature-aware LE methods, which jointly consider the feature and label information during extraction, have been shown to reach better performance than feature-unaware ones. Nevertheless, current feature-aware LE methods are not designed to flexibly adapt to different evaluation criteria. In this work, we propose a novel feature-aware LE method that takes the desired evaluation criterion (cost) into account during training. The method, named Feature-aware Cost-sensitive Label Embedding (FaCLE), encodes the criterion into the distance between embedded vectors with a deep Siamese network. The feature-aware characteristic of FaCLE is achieved with a loss function that jointly considers the embedding error and the feature-to-embedding error. Moreover, FaCLE is coupled with an additional-bit trick to deal with the possibly asymmetric criteria. Experiment results across different data sets and evaluation criteria demonstrate that FaCLE is superior to other state-of-the-art feature-aware LE methods and competitive to cost-sensitive LE methods.**

***Index Terms*—multi-label classification, feature-aware, cost-sensitive, label embedding**

I. Introduction

In traditional single-label learning tasks, e.g., binary and multi-class classification, one instance is associated with a single label. But in many real-world applications, one given instance is associated with a set of labels. Such a learning task is called multi-label classification (MLC). For example, for image annotation [1], [2], an image usually contains abundant semantic information such as characters, scenes, objects, actions, and colors; for document classification [3], a news article may cover numerous topics such as society, sports, entertainment, international events, and weather.

A straightforward MLC algorithm called binary relevance (BR) [4] transforms the MLC problem into binary classification sub-problems, one for each label. The binary classifier within each sub-problem of BR is trained independently, without exploiting the joint relationship across different labels, which limits BR's effectiveness in practice.

To deal with MLC problems more effectively, many Label Embedding (LE) methods have been proposed [5], [6]. LE methods aim to compute an embedding space that captures the relationship of labels. Then, the MLC problem can be reduced to a regression problem from the feature vector of an instance to the embedded vector in the embedding space. During the testing stage of LE methods, predictions are first performed in the embedding space and then decoded back to the original label space. The embedding space allows a better representation of the labels based on their relationship, therefore allowing many LE methods to perform better than the plain BR algorithm [5], [7].

One key design in modern LE methods is to consider whether the embedding space is easily "learnable"—i.e., whether the relationship between the feature vector and the embedded vector can be easily captured by a regressor. Such LE methods are called feature-aware, which take the feature information into consideration when learning the embedding space [7]–[9]. By taking both feature information and label relationship into account, feature-aware LE methods have generally reached better performance than feature-unaware ones. For instance, End-to-End Feature-aware label space Encoding (E^2FE) [9] is a state-of-the-art feature-aware LE method that jointly maximizes the recoverability of the label space and the predictability of the embedding space, when the two spaces are connected by a linear decoding matrix. E^2FE exhibits superior performance over other LE methods when being extended with the kernel trick and eigenvalue-boosted decoding matrix. But those extension tricks make E^2FE time-consuming in practice.

Most of the LE methods are not designed to flexibly adapt to different evaluation criteria, and would incur bad performance if the MLC problems are evaluated by the criterion different from what the methods optimize on. For example, in the objective function of E^2FE, every label is considered independently, which indicates that E^2FE focuses on Hamming Loss. When evaluated with other losses that are very different from the Hamming Loss, E^2FE could then suffer from unsatisfactory performance.

MLC problems that require the methods to take the criterion (cost information) into account are called cost-sensitive MLC (CSMLC) problems [10], [11]. One state-of-the-art CSMLC method is Cost-sensitive Label Embedding with Multidimensional Scaling (CLEMS) [11]. CLEMS adapts multidimensional scaling (MDS), a classic non-linear manifold learning approach, to embed the cost information as the distance measure within the embedding space. In the testing stage, CLEMS decodes every prediction as the corresponding label

vector of the nearest neighbor within a predefined candidate set. Besides, CLEMS uses a mirroring trick to solve the asymmetric cost problem, which means the costs of predicting one label-set as another and that of the reversed operation are different. CLEMS is shown to reach superior performance on CSMLC problems. Nevertheless, CLEMS maintains a dissimilarity matrix, whose size is quadratic to the size of the candidate set. The matrix results in computational difficulties for larger data sets. Moreover, CLEMS is not feature-aware and thus it is not clear whether the embedding space is easily learnable by the regressors.

In this paper we improve CLEMS by proposing a novel Feature-aware Cost-sensitive Label Embedding (FaCLE) method for CSMLC problems. FaCLE utilizes a deep Siamese network along with a sampling method of label vectors to embed the cost information as the distance between embedded vectors. The nature of sampling and deep learning structures mitigates the computation burdens within CLEMS. The asymmetric cost problem is carefully resolved with an additional-bit trick during training. Furthermore, with a feature-aware component, FaCLE successfully associates the feature space and the embedding space by jointly optimizing the embedding loss and the regression loss, and becomes the first feature-aware cost-sensitive MLC method as far as we know. The experiment results across different data sets and evaluation criteria reveal that FaCLE is superior to the state-of-the-art feature-aware LE method and competitive to cost-sensitive LE method.

II. Related work

Multi-label classification problems have attracted much interest in research. The simplest solution is binary relevance [4], which simply divides MLC into independent binary classification sub-problems for each label. Because of not considering other labels while learning on each label, BR is denounced as lacking the ability to leverage the latent structure between labels and thus reaching unsatisfactory performance.

Label embedding methods, an important family of methods for MLC, are proposed to address the problem of BR [5], [12], [13]. LE methods extract the information across different labels to learn a label embedding space, which is claimed to capture the latent label correlations. Besides, with an embedding dimension smaller than the input dimension, multi-dimensional predictions can be made in the embedding space with lower computation cost but possibly better overall performance. One example is principal label space transformation (PLST) [5], which leverages principal component analysis to find the most informative principal dimensions as the embedding dimensions, and decodes the predictions by a linear matrix.

Some LE methods take the feature information into consideration while learning the label embedding space [7]–[9], which are therefore called feature-aware LE methods. Feature-aware LE methods can appropriately associate the feature vectors and the label vectors while learning the embedding, making the embedding space more learnable for regressors, and thus improve the performance. Take CPLST [7], the conditional version of PLST, as an example. Inspired by canonical correlation analysis [14], CPLST uses singular value decomposition to jointly minimize the embedding error and the prediction error, and obtains the feature-aware embedding space. CPLST is reported to be superior to PLST, because the embedding error and the prediction error are optimized jointly instead of separately, making the embedding space more learnable in practice.

Unlike the analytical models like PLST and CPLST, C2AE [15] is the first label embedding model constructed by deep neural networks. By integrating the architectures of deep canonical correlation analysis and auto-encoder, C2AE jointly minimizes the embedding error of the auto-encoder and the regression error of canonical correlation analysis. In addition, C2AE proposes a label-correlation sensitive loss function to better decode the predictions and achieve state-of-the-art performance.

Although existing LE methods demonstrate promising performance for MLC problems, most of them are constructed to optimize on only one specific or few evaluation criteria. For example, in C2AE the label-correlation sensitive loss function is computed in a pairwise form of positive labels and negative labels, which is identical to Rank Loss. When encountering different evaluation criteria, those methods may reach bad performance. As a result, cost-sensitive MLC methods, which take the cost information (evaluation criteria) into account during training or testing, have become more important in recent days [10], [11], [16]. For example, Probabilistic Classifier Chain (PCC) [16] is proposed to make Bayes-optimal inference by estimating the probability of each label for the target criterion. But if there is no efficient inference rule designed for the criterion, PCC will encounter computational difficulties.

One particular LE method for CSMLC is Cost-sensitive Label Embedding with Multidimensional Scaling (CLEMS) [11], a state-of-the-art CSMLC method. CLEMS preserves the cost information in the distance of the embedded space by multidimensional scaling (MDS), and to decode every prediction as the nearest neighbor. Although CLEMS is reported to reach outstanding performance, CLEMS demands a dissimilarity matrix to be computed beforehand, whose size is quadratic to the number of unique label vectors in training data. Therefore, when handing large data sets, CLEMS could be computationally challenging as well.

In summary, current CSMLC methods can suffer from computational issues, and none of them are feature-aware. We address the two issues by proposing Feature-aware Cost-sensitive Label Embedding (FaCLE), which utilizes deep Siamese network to keep cost information as the distance of the embedded vectors, and exploits a feature-aware component to jointly optimize the embedding loss and the regression loss. Moreover, the usage of sampling method and the nature of deep learning structure make FaCLE more feasible and flexible to handle large data sets. The detail of FaCLE will be described in the following section.

III. The Proposed Approach

Denote $X \in R^{N\times d}$ as the d dimensional training feature matrix and $Y \in \{0,1\}^{N\times k}$ as the k dimensional corresponding label matrix, with N instances and the i-th row being x_i^T and y_i^T, where x_i is the feature vector and y_i is the corresponding label vector. Observing a data set $D = (X, Y)$, multi-label classification problems aim to learn a model which can make a proper prediction $\hat{y}$ of a testing instance $\hat{x}$.

In order to tackle the given evaluation criterion c of MLC directly, cost-sensitive multi-label classification (CSMLC) methods strive to leverage the information of the evaluation criterion in either the training stage or the testing stage. One precursor is cost-sensitive label embedding with multidimensional scaling (CLEMS) [11]. CLEMS utilizes multidimensional scaling to approximate the cost information with the distances of the embedded label vectors. In CLEMS, first a candidate set of label vectors P is decided as the set of label vectors appearing in the training instances. Then a dissimilarity matrix Φ is computed whose elements $\Phi_{i,j} = \delta(c(y_i, y_j))$, with y_i and y_j being the i-th and j-th vector in P and δ denoting the monotonic function. As the main step, MDS is leveraged to determine the embedded vectors u with an embedding dimension m for vectors in P by iteratively minimizing the $stress$:

$$stress = \sum_{i,j} W_{i,j}(\Delta(u_i, u_j) - \Phi_{i,j})^2 \tag{1}$$

where W and Δ denote the weight of label pairs and the distance function respectively. After that, an additional regressor is trained with feature vectors and embedded label vectors. In the testing stage, CLEMS easily decodes a prediction as the corresponding label of its nearest neighbor within embedding space. Besides, MDS requires the dissimilarity matrix to be symmetric, but some criteria are asymmetric, which means $c(y_i, y_j) \neq c(y_j, y_i)$. As a result, CLEMS proposes a $mirroring\ trick$, which views the label vectors as two roles, the ground truth role and the prediction role, and computes a symmetric dissimilarity matrix by considering those $2|P|$ label vectors. Then the regressor is trained on the truth vectors and the prediction is decoded on prediction vectors.

CLEMS is reported to be significantly better than a wide variety of state-of-the-art CSMLC methods. But one critical drawback of CLEMS is that a dissimilarity matrix needs to be computed beforehand, whose time complexity is $O(|P|^2)$, making it infeasible to be applied on large data sets. Another problem is that CLEMS does not consider feature information in training time, making the embedding space usually hard to be learned by the regressor in practice.

Motivated by CLEMS and recent developments of deep learning methods, we propose Feature-aware Cost-sensitive Label Embedding (FaCLE), which utilizes Siamese network to preserve the cost information as the distances between embedded vectors. In our method, only an assigned portion of label costs need to be computed in advance, making training on large data sets more feasible and flexible. Furthermore, we design a feature-aware component to take the feature information into consideration during the training stage, regularizing the embedding space to be more tractable for the regressor.

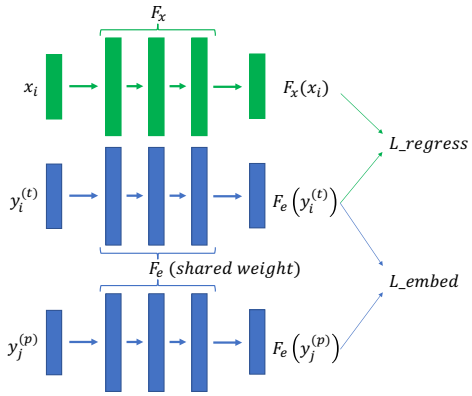

Fig. 1. The training architecture of FaCLE. The blue part learns a nonlinear label embedding function F_e by Siamese network, and the green part is the feature-aware component, which learns a nonlinear feature transformation function F_x making the training process also considering feature information.

As depicted in Figure 1, the overall training architecture of FaCLE comprises two parts: deep cost-sensitive label embedding (the blue part) and feature-aware component (the green part). The former learns a label embedding function F_e by Siamese network such that the distance of each embedded label pairs is monotonic to their cost, and the latter learns a feature transformation function F_x which associates the feature space with the label embedding space. Therefore, we formulate the objective function of FaCLE as:

$$L_{FaCLE}(F_e, F_x) = \min_{F_e, F_x} L_{embed}(F_e) + \alpha\ L_{regress}(F_e, F_x) \tag{2}$$

where L_{embed}, $L_{regress}$, and α denote the embedding loss, the regression loss, and the balancing parameter respectively. It is worth noticing that FaCLE jointly optimizes the embedding loss and the regression loss instead of separately. The details will be discussed in the following two subsections.

A. Deep Cost-sensitive Label Embedding

Siamese network is a special learning structure which has been exploited to learn decent representations in many research problems [17], [18]. With shared weights in the twin networks, Siamese network is usually aligned with the contrastive loss to enlarge the distance of similar pairs and reduce that of dissimilar pairs.

Our method starts from an idea that label pairs can be the inputs and the cost information of them can be regarded as the similarity measure in Siamese network to learn a cost-sensitive label embedding. Therefore, we propose a kind of restricted version of contrastive loss such that we not merely increase/decrease the distance of dissimilar/similar pairs, but force the distance close to their similarity measure as much as possible. Consequently, we formulate the embedding loss,

which is the same as the $stress$ in CLEMS, as follows:

$$L_{embed}(F_e) = \sum_{y_i, y_j} (\Delta(F_e(y_i), F_e(y_j)) - \delta(c(y_i, y_j)))^2 \quad (3)$$

But we can notice two problems. First, it is infeasible to optimize on all the label pairs, especially on large data sets. Accordingly, we suggest simply using random sampling for training label pairs, which give us the feasibility and the flexibility to adjust the training time according to how many computation resources we have. Second, the asymmetric cost problem, which is also discussed in CLEMS. The asymmetric cost, e.g., the rank loss, implies $c(y_i, y_j) \neq c(y_j, y_i)$ with the same input pairs $(y_i, y_j) = (y_j, y_i)$ in Siamese network's view. One intuitive solution is to distinguish the label pairs as two different roles, the ground truth and the prediction. Thus, we propose an Additional-Bit trick (AB-trick), which appends an additional bit 0 to label vectors serving as the ground truth role and an additional bit 1 to label vectors serving as the prediction role. In addition, we both append 0.5 while dealing with symmetric cost to avoid overfitting on the meaningless additional bit. The embedding loss is now modified as:

$$L_{embed}(F_e) = \sum_{(y_i, y_j) \in S} (\Delta(F_e(y_i^{(t)}), F_e(y_j^{(p)})) - \delta(c(y_i, y_j)))^2 \quad (4)$$

where $^{(t)}$, $^{(p)}$, and S denote the truth role, the prediction role, and the sampling respectively.

B. Feature-aware Component

Feature-awareness has been proved effective in label embedding methods [7]–[9]. Empirically, the label embedding space usually becomes intractable for the regressor if it is trained independently. As a consequence, we regularize our deep cost-sensitive label embedding by allying $F_e(y_i^{(t)})$ to the feature transformation $F_x(x_i)$ in training time. The reason why we ally $F_e(y_i^{(t)})$ instead of $F_e(y_i^{(p)})$ is that the regressor will be trained with the former. Moreover, it is worthwhile to mention that the feature transformation function F_x can also be used as an end-to-end regressor. The resulting regression loss is formulated as:

$$L_{regress}(F_e, F_x) = \sum_{x_i, y_i} \Delta(F_x(x_i), F_e(y_i^{(t)}))^2 \quad (5)$$

Until now, we have introduced the complete training structure of FaCLE. Once the training of FaCLE is accomplished, we either easily apply F_x as the regressor r or train a new one with input (X, Z), where $Z = F_e(Y^{(t)})$. With a coming test instance $\hat{x}$, we first compute its prediction $\hat{z} = r(\hat{x})$, then find its nearest neighbor $\hat{z}_{nbr}$ in embedded candidate set $P^{(e)} = F_e(P^{(p)})$, and finally decode it as the corresponding label vector $\hat{y} \in P$. Because of the intuition that the distances in $P^{(e)}$ are monotonic to real costs, $\hat{y}$ should be a reasonable prediction of $\hat{x}$.

To the best of our knowledge, we are the first to apply Siamese network to cost-sensitive label embedding, and the first to design a feature-aware CSMLC method, which are the main contributions of this work. The pseudo code of FaCLE is summarized in Algorithm 1 and Algorithm 2.

Algorithm 1: Training of FaCLE

Input: X, Y, δ, c, α, m
Output: r, P, $P^{(e)}$
Decide P as the distinct training label vectors
Let $D = (X, Y)$
Sample $S = \{(x_i, y_i, y_j)\}$ from D and Y
Compute $C = \{\delta(c(y_i, y_j))\}$ of S
Initialize F_e, F_x with m
Train F_e and F_x to minimize (2), (4), (5) with S, C, α, and AB-trick
Compute embedded vector set $Z = F_e(Y^{(t)})$
Apply F_x as regressor r or train a new one with (X, Z)
Compute embedded candidate set $P^{(e)} = F_e(P^{(p)})$

Algorithm 2: Predicting of FaCLE

Input: x, r, P, $P^{(e)}$
Output: $\hat{y}$
Compute $\hat{z} = r(x)$
Find the nearest neighbor $\hat{z}_{nbr} \in P^{(e)}$ of $\hat{z}$
Make prediction $\hat{y}$ as the corresponding $y \in P$ of $\hat{z}_{nbr}$

IV. Experiments

To evaluate the performance of FaCLE, we conduct experiments on the following real-world data sets: *birds*, *emotions*, *medical*, *CAL500*, *scene*, *yeast*, *enron*, *tmc2007* [19]. The detailed statistics of each data set are listed in Table I. We consider four evaluation criteria frequently used in CSMLC problems: $Hamming\ loss(y, \hat{y}) = \frac{1}{k} \sum_{i=1}^{k} [\![y[i] \neq \hat{y}[i]]\!]$, $Accuracy\ loss(y, \hat{y}) = 1 - \frac{||y \cap \hat{y}||_1}{||y \cup \hat{y}||_1}$, $F1\ loss(y, \hat{y}) = 1 - \frac{2||y \cap \hat{y}||_1}{||y||_1 + ||\hat{y}||_1}$, and $Rank\ loss(y, \hat{y}) = \frac{1}{R(y)} \sum_{(i,j): y[i] > y[j]} ([\![\hat{y}[i] < \hat{y}[j]]\!] + \frac{1}{2}[\![\hat{y}[i] = \hat{y}[j]]\!])$ where $R(y) = |\{(i, j)|y[i] > y[j]\}|$. Please notice that $Hamming\ loss$, $Accuracy\ loss$, and $F1\ loss$ are symmetric, while $Rank\ loss$ is asymmetric.

All the experiment results are reported as the average of 20 independent runs if not specifically acknowledged. In each run, the data sets are randomly split to 50%, 25%, and 25% for training, validation, and testing correspondingly. The best parameters of each method are selected by using the validation

data set	k	d	N	#distinct labels
birds	19	260	645	133
emotions	6	72	593	27
medical	45	1449	978	94
CAL500	174	68	502	502
scene	6	294	2407	15
yeast	14	103	2417	198
enron	53	1001	1702	753
tmc2007	33	500	28596	1172

TABLE I
STATISTICS OF THE DATA SETS WHERE k IS THE DIMENSION OF LABEL VECTORS, d IS THE DIMENSION OF FEATURE VECTORS, AND N IS THE NUMBER OF INSTANCES

part and then used for testing.

For all the methods in our experiments, if not mention specifically, we use random forest implemented in scikit-learn [20] as the regressor, with the maximum depth of trees selected from {5, 10, ..., 35}. For FaCLE, F_e is constructed of 2 fully connected layers with 2 times the input label dimension neurons for each layer, and F_x is constructed of 3 fully connected layers with 10% dropout and 5 times the input feature dimension neurons for each layer. ReLU is deployed as the activation function and the mini-batch size is 10. The learning rate is selected in a range [0.00001, 0.01] and α is fixed as 0.03. We set the sampling number $|S|$ to be $\frac{1}{4}|P|^2$, and the distance function Δ to be L^2-norm. Square root function is chosen as the monotonic function δ according to the suggestion of [11]. Additionally, we name the feature-unaware version of FaCLE, which has only the deep cost-sensitive label embedding part, as DCLE.

A. Comparing with Cost-sensitive Label Embedding Methods

We first compare DCLE, FaCLE with CLEMS [11], which is introduced in section 3. The embedding dimension m is appointed to be equivalent to k, the dimension of label vectors. The parameters of CLEMS are selected in the same range within its original paper. The results are illustrated in the Table II.

From the table we can find that DCLE is comparable to CLEMS within the first 7 small data sets, which illustrates the efficient use of only $\frac{1}{4}|P|^2$ samples in DCLE comparing to the size of dissimilarity matrix $|P|^2$ in CLEMS. Moreover, DCLE performs obviously better than CLEMS on the data set tmc2007, which is much larger than others, supporting the effectiveness of our deep cost-sensitive label embedding method. In addition, according to the results of DCLE and FaCLE, feature-awareness plays a positive role in half of the results (16 of 32), and we can find some data sets not suitable for feature-aware methods. That depends on the difficulty of extracting useful feature information from the data set.

Hamming Loss			
	CLEMS	DCLE	FaCLE
birds	$\mathbf{0.044 \pm 0.001}$	0.047 ± 0.001	0.046 ± 0.001
emotions	0.193 ± 0.003	$\mathbf{0.184 \pm 0.003}$	0.193 ± 0.003
medical	0.024 ± 0.002	0.016 ± 0.001	$\mathbf{0.012 \pm 0.002}$
CAL500	$\mathbf{0.159 \pm 0.001}$	0.162 ± 0.001	$\mathbf{0.159 \pm 0.001}$
scene	$\mathbf{0.092 \pm 0.003}$	0.097 ± 0.003	0.096 ± 0.002
yeast	0.193 ± 0.002	$\mathbf{0.191 \pm 0.001}$	0.194 ± 0.001
enron	0.042 ± 0.002	0.051 ± 0.000	$\mathbf{0.026 \pm 0.005}$
tmc2007	0.052 ± 0.002	$\mathbf{0.043 \pm 0.000}$	0.055 ± 0.000
Accuracy Loss			
	CLEMS	DCLE	FaCLE
birds	0.391 ± 0.008	0.375 ± 0.006	$\mathbf{0.347 \pm 0.020}$
emotions	$\mathbf{0.411 \pm 0.007}$	0.421 ± 0.006	0.420 ± 0.006
medical	0.344 ± 0.013	$\mathbf{0.241 \pm 0.030}$	0.278 ± 0.030
CAL500	$\mathbf{0.729 \pm 0.002}$	0.736 ± 0.002	0.731 ± 0.003
scene	0.268 ± 0.005	$\mathbf{0.260 \pm 0.003}$	0.272 ± 0.005
yeast	0.436 ± 0.002	0.440 ± 0.003	$\mathbf{0.435 \pm 0.003}$
enron	0.424 ± 0.012	0.535 ± 0.004	$\mathbf{0.321 \pm 0.052}$
tmc2007	0.381 ± 0.013	$\mathbf{0.274 \pm 0.013}$	0.352 ± 0.009
Rank Loss			
	CLEMS	DCLE	FaCLE
birds	$\mathbf{0.152 \pm 0.005}$	0.201 ± 0.005	0.205 ± 0.004
emotions	$\mathbf{0.203 \pm 0.004}$	0.222 ± 0.007	0.238 ± 0.008
medical	$\mathbf{0.114 \pm 0.010}$	$\mathbf{0.114 \pm 0.008}$	0.137 ± 0.009
CAL500	$\mathbf{0.327 \pm 0.001}$	0.333 ± 0.002	0.349 ± 0.006
scene	$\mathbf{0.132 \pm 0.002}$	0.144 ± 0.005	0.148 ± 0.006
yeast	$\mathbf{0.217 \pm 0.002}$	0.228 ± 0.001	0.233 ± 0.002
enron	$\mathbf{0.132 \pm 0.013}$	0.182 ± 0.001	0.173 ± 0.005
tmc2007	0.124 ± 0.009	$\mathbf{0.109 \pm 0.001}$	0.124 ± 0.001
F1 Loss			
	CLEMS	DCLE	FaCLE
birds	$\mathbf{0.325 \pm 0.007}$	0.333 ± 0.007	0.329 ± 0.007
emotions	$\mathbf{0.314 \pm 0.004}$	0.325 ± 0.005	0.323 ± 0.003
medical	0.321 ± 0.014	$\mathbf{0.298 \pm 0.017}$	0.312 ± 0.015
CAL500	$\mathbf{0.580 \pm 0.003}$	0.592 ± 0.002	0.583 ± 0.002
scene	0.252 ± 0.003	0.248 ± 0.005	$\mathbf{0.247 \pm 0.005}$
yeast	$\mathbf{0.335 \pm 0.003}$	0.336 ± 0.002	0.337 ± 0.002
enron	0.359 ± 0.008	0.412 ± 0.005	$\mathbf{0.236 \pm 0.039}$
tmc2007	0.306 ± 0.017	$\mathbf{0.222 \pm 0.001}$	0.274 ± 0.001

TABLE II
PERFORMANCE COMPARISON OF COST-SENSITIVE LABEL EMBEDDING METHODS IN DIFFERENT EVALUATION CRITERIA (MEAN ± STANDARD ERROR)

B. Comparing with Feature-aware Label Embedding Methods

We further compare FaCLE with other existing feature-aware label embedding based methods. Canonical Correlated Autoencoder (C2AE) [15] is another state-of-the-art feature-aware label embedding method. C2AE derives the embedded vectors by jointly optimizing the embedding loss of an auto-encoder and the regression loss of a deep neural network regressor in a way of canonical correlation analysis. A label-correlation sensitive loss function is also proposed to better recover the prediction to the original label space. Furthermore, C2AE can be easily extended to address the missing label problems. In this experiment, the detailed architecture of C2AE is set to be identical to that in [15].

The embedding dimension m is set to be $0.5*k$, and parameters are all selected within the same range referring to the original settings in [15]. Because C2AE is designed for using deep neural network as the regressor, we demonstrate the results of FaCLE directly using feature transformation function F_x as the regressor. Also for avoiding confusion, We denote FaCLE in this experiment as FaCLE_0.5_NN. The results are listed in the Table III

We can find FaCLE_0.5_NN performs mostly better than C2AE, which illustrates the effectiveness of cost-sensitivity and supports our intuition to design a general cost-sensitive label embedding method. Additionally, we can find that on some data sets FaCLE_0.5_NN performs better than FaCLE. And we especially claim that our model has the flexibility to choose proper regressors for better performance according to each targeted data set.

V. CONCLUSION

We propose Feature-aware Cost-sensitive Label Embedding (FaCLE) for multi-label classification problems. By exploiting Siamese network, FaCLE successfully learns a cost-sensitive label embedding where the cost information is kept as the distance measure. With Additional-Bit trick, the asymmetric cost can be also handled by FaCLE. We further design a

Hamming Loss		
	C2AE	FaCLE_0.5_NN
birds	0.320 ± 0.017	$\mathbf{0.045 \pm 0.002}$
emotions	0.627 ± 0.010	$\mathbf{0.241 \pm 0.007}$
medical	0.324 ± 0.005	$\mathbf{0.021 \pm 0.001}$
CAL500	0.288 ± 0.002	$\mathbf{0.158 \pm 0.000}$
scene	0.136 ± 0.002	$\mathbf{0.082 \pm 0.002}$
yeast	0.221 ± 0.002	$\mathbf{0.195 \pm 0.002}$
enron	0.074 ± 0.001	$\mathbf{0.056 \pm 0.000}$
tmc2007	0.052 ± 0.000	$\mathbf{0.043 \pm 0.000}$
Accuracy Loss		
	C2AE	FaCLE_0.5_NN
birds	0.923 ± 0.002	$\mathbf{0.304 \pm 0.039}$
emotions	0.683 ± 0.002	$\mathbf{0.532 \pm 0.011}$
medical	0.930 ± 0.001	$\mathbf{0.221 \pm 0.040}$
CAL500	$\mathbf{0.714 \pm 0.001}$	0.745 ± 0.002
scene	0.525 ± 0.004	$\mathbf{0.269 \pm 0.007}$
yeast	0.484 ± 0.002	$\mathbf{0.451 \pm 0.002}$
enron	0.644 ± 0.003	$\mathbf{0.583 \pm 0.001}$
tmc2007	0.350 ± 0.001	$\mathbf{0.287 \pm 0.000}$
Rank Loss		
	C2AE	FaCLE_0.5_NN
birds	$\mathbf{0.211 \pm 0.006}$	0.225 ± 0.005
emotions	0.474 ± 0.005	$\mathbf{0.356 \pm 0.015}$
medical	0.225 ± 0.002	$\mathbf{0.197 \pm 0.006}$
CAL500	$\mathbf{0.260 \pm 0.000}$	0.386 ± 0.002
scene	0.256 ± 0.002	$\mathbf{0.244 \pm 0.010}$
yeast	0.243 ± 0.001	$\mathbf{0.235 \pm 0.003}$
enron	$\mathbf{0.191 \pm 0.002}$	0.201 ± 0.000
tmc2007	0.118 ± 0.000	$\mathbf{0.095 \pm 0.000}$
F1 Loss		
	C2AE	FaCLE_0.5_NN
birds	0.875 ± 0.003	$\mathbf{0.370 \pm 0.006}$
emotions	0.531 ± 0.002	$\mathbf{0.396 \pm 0.006}$
medical	0.871 ± 0.001	$\mathbf{0.241 \pm 0.031}$
CAL500	$\mathbf{0.559 \pm 0.001}$	0.606 ± 0.001
scene	0.491 ± 0.003	$\mathbf{0.245 \pm 0.006}$
yeast	0.368 ± 0.002	$\mathbf{0.341 \pm 0.002}$
enron	0.509 ± 0.003	$\mathbf{0.331 \pm 0.048}$
tmc2007	0.270 ± 0.001	$\mathbf{0.204 \pm 0.001}$

TABLE III
PERFORMANCE COMPARISON OF FEATURE-AWARE LABEL EMBEDDING METHODS IN DIFFERENT EVALUATION CRITERIA (MEAN $\pm$ STANDARD ERROR)

feature-aware component to make FaCLE jointly optimize the embedding loss and the regression loss instead of separately. With the embedding, FaCLE decodes the predictions to the nearest neighbors within a pre-decided candidate set. The experiment results show that FaCLE achieves decent performance by efficiently using a small quantity of sampling against other cost-sensitive label embedding method, and the feature-awareness further improves the performance on some data sets. Moreover, we also demonstrate that FaCLE is superior to other feature-aware label embedding methods, which supports the effectiveness of cost-sensitivity. As far as we know, FaCLE is the first cost-sensitive label embedding method to utilize deep learning structure, and the first feature-aware CSMLC method.

VI. ACKNOWLEDGMENT

The work arises from the Masters thesis of the first author [21]. We thank Profs. Yu-Chiang Frank Wang, Yun-Nung Chen, the anonymous reviewers and the members of the NTU Computational Learning Lab for valuable suggestions. This work is partially supported by the Ministry of Science and Technology of Taiwan under number MOST 103-2221-E- 002-149-MY3.

REFERENCES

[1] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang, "Multi-label sparse coding for automatic image annotation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1643–1650.

[2] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2285–2294.

[3] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Machine learning*, vol. 88, no. 1, pp. 157–208, 2012.

[4] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, 2006.

[5] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012.

[6] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Advances in neural information processing systems*, 2015, pp. 730–738.

[7] Y.-N. Chen and H.-T. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Advances in Neural Information Processing Systems*, 2012, pp. 1529–1537.

[8] X. Li and Y. Guo, "Multi-label classification with feature-aware non-linear label space transformation." in *IJCAI*, 2015, pp. 3635–3642.

[9] Z. Lin, G. Ding, J. Han, and L. Shao, "End-to-end feature-aware label space encoding for multilabel classification with many classes," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[10] C.-L. Li and H.-T. Lin, "Condensed filter tree for cost-sensitive multi-label classification," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 423–431.

[11] K.-H. Huang and H.-T. Lin, "Cost-sensitive label embedding for multi-label classification," *Machine Learning*, vol. 106, no. 9-10, pp. 1725–1746, 2017.

[12] W. Bi and J. Kwok, "Efficient multi-label classification with many labels," in *International Conference on Machine Learning*, 2013, pp. 405–413.

[13] H.-F. Yu, P. Jain, P. Kar, and I. Dhillon, "Large-scale multi-label learning with missing labels," in *International Conference on Machine Learning*, 2014, pp. 593–601.

[14] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[15] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang, "Learning deep latent space for multi-label classification." in *AAAI*, 2017, pp. 2838–2844.

[16] W. Cheng, E. Hüllermeier, and K. J. Dembczynski, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 279–286.

[17] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.

[18] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[19] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

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

[21] H.-C. Chiu, "Multi-label classification with feature-aware cost-sensitive label embedding," *Master Thesis, National Taiwan University*, pp. 1–26, 2017.