

Multi-Label Learning by Image-to-Class Distance for Scene Classification and Image Annotation

Zhengxiang Wang
CeMNet, School of Computer
Engineering
Nanyang Technological
University, Singapore
wang0460@ntu.edu.sg

Yiqun Hu
CeMNet, School of Computer
Engineering
Nanyang Technological
University, Singapore
yiqun.hu@gmail.com

Liang-Tien Chia
CeMNet, School of Computer
Engineering
Nanyang Technological
University, Singapore
aslchia@ntu.edu.sg

ABSTRACT

In multi-label learning, an image containing multiple objects can be assigned to multiple labels, which makes it more challenging than traditional multi-class classification task where an image is assigned to only one label. In this paper, we propose a multi-label learning framework based on Image-to-Class (I2C) distance, which is recently shown useful for image classification. We adjust this I2C distance to cater for the multi-label problem by learning a weight attached to each local feature patch and formulating it into a large margin optimization problem. For each image, we constrain its weighted I2C distance to the relevant class to be much less than its distance to other irrelevant class, by the use of a margin in the optimization problem. Label ranks are generated under this learned I2C distance framework for a query image. Thereafter, we employ the label correlation information to split the label rank for predicting the label(s) of this query image. The proposed method is evaluated in the applications of scene classification and automatic image annotation using both the natural scene dataset and Microsoft Research Cambridge (MSRC) dataset. Experiment results show better performance of our method compared to previous multi-label learning algorithms.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;
I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis

General Terms

Algorithm, Experimentation, Performance

Keywords

Multi-label learning, Image-to-Class distance, Scene classification, Automatic image annotation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIVR '10, July 5-7, Xi'an China

Copyright ©2010 ACM 978-1-4503-0117-6/10/07 ...\$10.00.

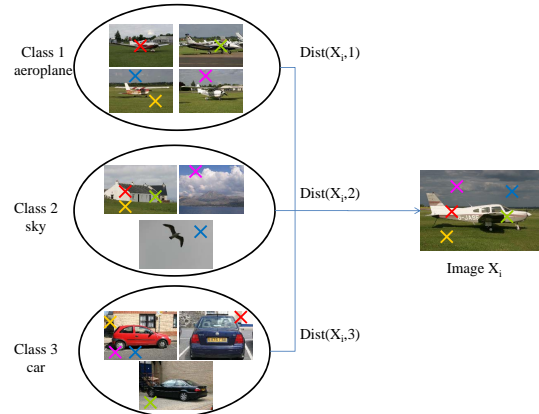


Figure 1: I2C distance. Each cross in the query image X_i denotes a feature patch, and its NN in each class is the cross with the same color. The learned weights of these NN points are multiplied with the Euclidean distance between feature patches and their corresponding NN feature to form the learned I2C distance.

1. INTRODUCTION

Image classification is the task of predicting the class label for a given image from a set of labels. Traditional binary or multi-class classification assigns images to only one label, such methods will perform well only for images with only one object. In multi-label learning problem, images are assumed to have more than one object, thus there is a need to assigned multiple labels for each image. (where each object is assigned to its respective label) In addition, the number of labels assigned for each image is not fixed nor pre-determined. Therefore, the task of multi-label image classification is an extremely challenging problem.

A simple way for multi-label learning is to decompose the problem into multiple independent binary classification tasks. However, the label correlation is not considered in such a method. Whereas, in real-life examples, it has been shown that label co-occurrence is indeed very important for multi-label learning [6, 12, 15, 19]. In this paper, we first propose a label ranking method using weighted Image-to-Class (I2C) distances, thus preventing the decomposition.

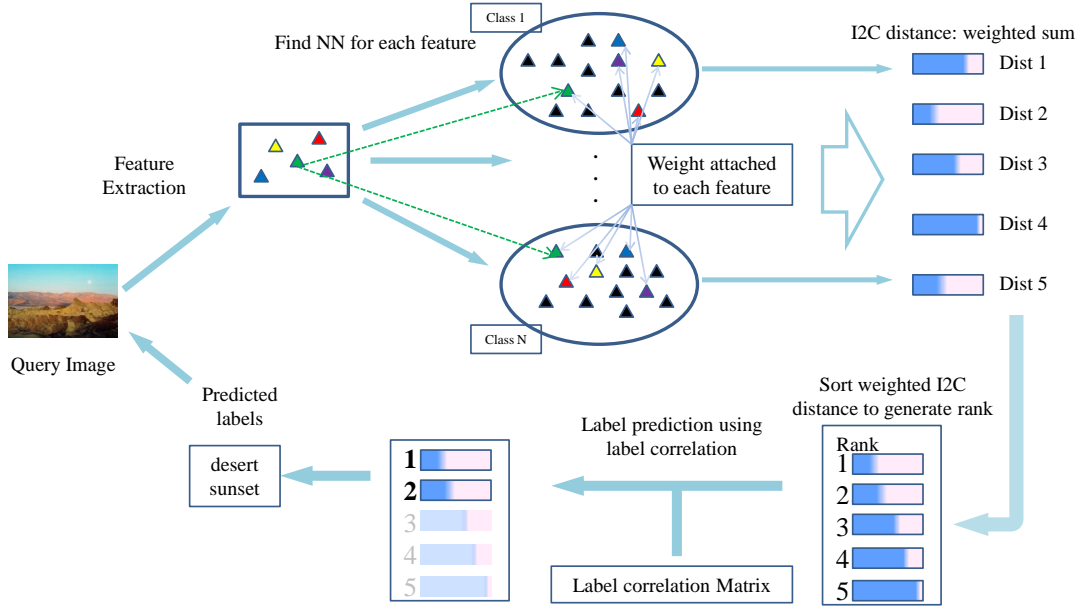


Figure 2: The procedure of classifying a query image. Local features of this image are denoted as triangles with different colors. The NN of these features in each class are denoted with the same color. The weighted I2C distances to all classes are denoted as different length of blue bars for expressing the relative size.

We also make use of label correlation for label prediction given the label rank. These two contributions make our method quite effective for classifying multi-label images.

The I2C distance is firstly introduced in [2] for classifying multi-class single label images and achieves excellent performance for such multi-class classification problems. Specifically, an image is represented by a set of local features using some descriptor. Then the I2C distance from the query image to a certain class is formulated as the sum of Euclidean distance from each feature in the query image to its nearest neighbor (NN) in that class. The prediction of this query image is simply by selecting the class label with shortest I2C distance.

In this paper, we extend the use of this I2C distance for multi-label classification. The main advantage of I2C distance is that its similarity measure is between Image-to-Class. Image-to-Image (I2I) distance may not be a good measurement for multi-label learning since in [14], they have shown that directly using I2I distance for measuring the similarity cannot truly reflect the real class label semantic similarity. For example, two images containing one common object may also have other visually incompatible objects, which make these two images less similar and thus can not reflect the label similarity for that common object. Previously, [2] has showed that the I2C distance is better than I2I distance for multi-class learning. In this paper, we will show that I2C distance, coupled with label correlation, will be a better distance measure in multi-label learning.

However, the I2C distance proposed in [2] is designed for multi-class learning. In order to use it for multi-label learning where each training image contains multiple labels, a learning procedure is required to incorporate label information of training images for learning a model. In this paper,

we adopt the idea of learning the weight attached to each feature used in [5, 17] for our multi-label learning. We build a large margin framework to learn these weights with the constraint that for each training image, its weighted I2C distance to its relevant class should be less than its distance to the other classes. As all weights are learned simultaneously in this learning framework, the learned weight is able to reflect the relative importance of its attached feature. Figure 1 shows an example of the I2C distance in multi-label learning. The query image X_i belongs to the first two classes but is irrelevant to the third class. Feature patches located for different objects of the query image (“sky” and “aeroplane” here as example) is able to find the corresponding correct NN in their relevant class but is less likely to achieve that in the irrelevant class. The goal of learning weight is to further enhance the I2C distance of relevant class from other irrelevant class by emphasizing these correct NN feature.

The learned I2C distance only provides a label rank for a query image. For label prediction, a splitting point is needed to separate the label rank for selecting the relevant labels. We incorporate label correlation information to select top-ranked labels for label prediction. The whole procedure for classifying a query image is shown in Figure 2. This query image is represented by a set of local features extracted from patches at its keypoints. Then these features find their NN in each class and the I2C distance is formulated as the weighted sum of the Euclidean distances between features and their NN, where weights are learned through our optimization framework to associate with these NN features. By sorting these weighted I2C distances in ascending order, a label rank is generated, which is then used for label prediction with its label correlation information.

We apply our method for scene classification and auto-

matic image annotation, and we use the natural scene dataset and Microsoft Research Cambridge (MSRC) image dataset for evaluation respectively. Experiment results show promising performance for our method when compared to previous multi-label learning algorithms.

Details of this paper are as follows: Section 2 reviews recent multi-label studies on scene classification and automatic image annotation. We formulate our large margin optimization problem for learning weight in Section 3, and describe the label prediction method incorporating label correlation in Section 4. The performance of our method to the applications of scene classification and image annotation are evaluated in Section 5. Finally, we conclude this paper in Section 6.

2. RELATED WORK

Multi-label learning is an emerging and promising research topic in recent years. In multi-label learning, each instance is associated with multiple class labels simultaneously, so the traditional multi-class is its special case. The multi-label learning is originally developed for text classification. For example, Schapire and Singer [11] proposed a BoosTexter method extended from AdaBoost to classify multi-label text data. The success of multi-label learning for text data inspired researchers to apply it for dealing with image data. Boutell *et al.* [3] first presented a framework to classify scene images by multi-label learning, which is also investigated by [20, 21]. Subsequently, Zhou *et al.* [23] developed a multi-instance multi-label (MIML) learning framework and applied it for scene classification. They decomposed MIML learning into multi-instance learning and multi-label learning tasks and thus proposed MIML-Boost and MIML-SVM algorithms respectively. To prevent decomposition when losing useful information, they then proposed a maximum margin method [22] for MIML learning and achieved improvement on scene classification.

Image annotation has recently been an active research topic in the computer vision community due to its great impact on image retrieval and indexing via keywords. Since an image, in real life, will contain more than one keywords, many recent studies attempted to use multi-label learning algorithms, to deal with the task of image annotation, by treating each keyword as an independent class label. For example, Wang *et al.* [16] proposed a weighted KNN multi-label classification (KMLC) method for the image annotation problem. Li *et al.* [8] used contextual image decomposition for image annotation, which simultaneously maximizes inter-label difference and minimizes intra-label difference of the target label representations. In [14], sparse coding is involved for multi-label learning, which propagate label information from training images to query images. Lu *et al.* [10] also proposed a label propagation method by incorporating the context between visual keywords into the similarity between images. Label correlation information is utilized in [15] by integrating the correlated Green’s function to annotate images into a graph. In this paper, we also make use of the label correlation to predict the label of query images.

3. LARGE MARGIN OPTIMIZATION FOR LEARNING WEIGHT

In this section, we propose a large margin optimization framework for learning the weight attached to each fea-

ture patch, so that the I2C distance is reconstructed as the weighted sum between feature patches and their NN. This local feature is extracted from the patch at its key-point represented by local feature descriptor. Let $F_i = \{f_{i1}, f_{i2}, \dots, f_{im_i}\}$ denote features belonging to image X_i , where m_i represent the number of features in X_i and each feature is denoted as $f_{ij} \in R^d, \forall j \in \{1, \dots, m_i\}$. The feature set of each class c is composed of features from images that belong to this class. Since in multi-label learning each image may belong to multiple classes, the feature set of different classes may have overlap. Our objective is to learn the weight attached to local features in these feature sets. Although images may belong to multiple classes, we decide to learn one weight for each feature patch. In most cases, each local feature patch contains pixels from one object. Therefore assigning each local feature to only one class label is acceptable. Besides, learning one weight per feature will maintain consistency in the large margin framework. Therefore, we can concatenate all weights to a global weight vector denoted as W . Then the weighted I2C distance between image X_i to class c is formulated as:

$$W \cdot Dist(X_i, c) = \sum_{j=1}^{m_i} W_{ij}^c \cdot \|f_{ij} - f_{ij}^c\|^2 \quad (1)$$

Here f_{ij}^c represent the NN point that f_{ij} find in the feature set of class c and W_{ij}^c is its attached weight, which is the component in the global weight vector W learned through our optimization framework. We adopt the idea of large margin and formulate the constraint by keeping the weighted I2C distance from each image to its relevant class less than to other class with a margin. Let Y_i and \bar{Y}_i denote the set of class labels that image X_i belongs or not-belongs respectively, then the constraint can be formulated as follows:

$$W \cdot (Dist(X_i, n) - Dist(X_i, p)) \geq 1 - \xi_{ipn} \quad (2) \\ \forall p \in Y_i, \forall n \in \bar{Y}_i$$

Since this constraint cannot be completely satisfied, we use a slack variables ξ to relax it. The whole large margin optimization framework is similar to that used in SVM. However, the difference between our problem and SVM is that, in our framework the default value for each weight component should be 1 rather than 0 since each feature is of equal importance in the original NBNN [2]. So we introduced a prior weight vector W_0 to keep the learned weight vector W close to it, where all the components in W_0 are 1. Our large margin optimization problem is then represented as:

$$\arg \min_{W, \xi} \quad \frac{1}{2} \|W - W_0\|^2 + C \sum_{i,p,n} \xi_{ipn} \quad (3) \\ s.t. \forall i : \quad p \in Y_i, n \in \bar{Y}_i \\ W \cdot (Dist(X_i, n) - Dist(X_i, p)) \geq 1 - \xi_{ipn} \\ \xi_{ipn} > 0 \\ \forall k : \quad W(k) > 0$$

We rewrite this formula by denoting the difference between two I2C distances ($Dist(X_i, n) - Dist(X_i, p)$) as Q_{ipn} , which is a vector of equal length with weight vector W . The Euclidean distance between each feature and its NN ($\|f_{ij} - f_{ij}^c\|^2$ in (1)) for formulating the I2C distance comprises the component of this vector and is put in the location of the NN feature f_{ij}^c in this vector. Then the triplet con-

straint is represented as:

$$W \cdot Q_{ipn} \geq 1 - \xi_{ipn} \quad (4)$$

Where

$$Q_{ipn} = (Dist(X_i, n) - Dist(X_i, p)) \quad (5)$$

The formulation of this optimization problem is similar to SVM. To solve this problem, we use the method introduced in [5, 17] for learning the weight. Specifically, the primer form (3) of this problem can be reformulated into its dual form:

$$\begin{aligned} \arg \min_{\alpha, \mu} \quad & -\frac{1}{2} \left\| \sum_{i,p,n} \alpha_{ipn} \cdot Q_{ipn} + \mu \right\|^2 + \sum_{ipn} \alpha_{ipn} \quad (6) \\ & - \left[\sum_{ipn} \alpha_{ijk} \cdot Q_{ipn} + \mu \right] \cdot W_0 \\ \text{s.t.} \quad & \forall i, p, n : 0 < \alpha_{ipn} < C \\ & \forall j : \mu(j) > 0 \end{aligned}$$

We solve this dual problem by iteratively updating the dual variable α and μ alternatively. Each time we update those α variables that violate the KKT conditions by taking the derivative of (6) and then update μ for ensuring the positiveness of weight vector W in each iteration. The updating formula of α and μ is given as:

$$\begin{aligned} \alpha_{ipn} &= \left\{ \left[1 - \sum_{\{i',p',n'\} \neq \{i,p,n\}} \alpha_{i'p'n'} \langle Q_{i'p'n'} \cdot Q_{ipn} \rangle - \right. \right. \\ & \quad \left. \left. \langle (\mu + W_0) \cdot Q_{ipn} \rangle \right] / \| Q_{ipn} \|^2 \right]_{[0,C]} \\ \mu &= \max \left\{ 0, - \sum_{i,p,n} \alpha_{ipn} \cdot Q_{ipn} - W_0 \right\} \quad (7) \end{aligned}$$

and KKT conditions are:

$$\begin{aligned} \alpha_{ipn} = 0 &\Rightarrow W \cdot Q_{ipn} \geq 1 \\ 0 < \alpha_{ipn} < C &\Rightarrow W \cdot Q_{ipn} = 1 \\ \alpha_{ipn} = C &\Rightarrow W \cdot Q_{ipn} \leq 1 \quad (8) \end{aligned}$$

The operation of $[f(x)]_{[0,C]}$ is to clip the value of $f(x)$ to the region $[0, C]$. Since the optimization problem is convex, the objective function is guaranteed to reach the global minimum after iterative updating. Then the weight vector W in the primer form can be form by:

$$W = \sum_{i,p,n} \alpha_{i,p,n} \cdot Q_{i,p,n} + \mu + W_0 \quad (9)$$

This training procedure generates a weighted I2C distance for a query image to each class. Then a label rank can be identified by sorting the weighted I2C distance to all classes. Since the number of labels for the query image is unknown, we need to predict its label given this label rank. In this paper, we employ label correlation information for the label prediction task.

4. LABEL PREDICTION

Label correlation is widely used for multi-label learning [6, 12, 15, 19] and shown to be very effective. The idea of exploring the correlation between pairwise labels comes from the fact that some labels are usually co-assigned to the same image while some other labels are unlikely to appear in one image. For example, an image contains “plane” is likely to also contain “sky”, while it is much less likely to contain

“car” (See Figure 1). Since the learned I2C distance only solves the problem of label ranking but do not provide the exact labels in prediction, we make use of label correlation to split the label rank into positive and negative part and thus predict the labels for query image. By using the label correlation generated from training images to split the label rank for query image prediction, we are able to ensure the consistency of label correlation between training and test images.

We construct the label correlation from training set using the method in [12], which counts the combination of frequency for pairwise labels. A label correlation matrix M is built with its element M_{ij} representing the correlation coefficient between label l_i and l_j . The value of M_{ij} is defined in [12] as follows:

$$M_{ij} = \frac{AD - BC}{\sqrt{(A+B)(C+D)(A+C)(B+D)}} \quad (10)$$

where A,B,C,D is the percentage of frequency count in training set for the combinations of the co-occurrence as follows:

$$\begin{aligned} A &= l_i \wedge l_j \\ B &= l_i \wedge \neg l_j \\ C &= \neg l_i \wedge l_j \\ D &= \neg l_i \wedge \neg l_j \end{aligned}$$

This label correlation formulation is a specialized version of the Pearson product moment correlation coefficient for categorical variables with two values and seems to be better than the standard co-occurrence measure. Then we use this label correlation matrix to split the label rank into positive part and negative part, where labels in the positive part is the predicted label for a query image. The splitting point is initially set between the top rank and the second rank, so that the top label is always in the positive part to ensure at least one label is assigned to the query image. Then we shift the splitting point along the label rank by deciding whether the current label should be put into the positive part. This is done by checking whether there is any label “close” enough to the current label. A threshold θ is required to define the closeness of two labels so that if M_{ij} is larger than θ , label l_i and l_j is recognized as close to each other. If no labels in the positive part is “close” to the current label, the shifting is finished and the splitting point is fixed to separate the positive part and negative part. Note that the order of labels in the rank is determined by weighted I2C distance and keeps unchanged, while label correlation only determines the position of splitting point for label prediction, since the effort to put relevant labels at top of the rank is mainly achieved by label ranking.

The quality of label prediction is measured by *Hamming Loss*, which checks the difference between predicted labels and true labels using the XOR operation. Let P_i and Y_i denote the predicted label set and true labels for a query image X_i . The *Hamming Loss* of X_i is defined by:

$$HL(X_i) = \frac{P_i \Delta Y_i}{N_c} \quad (11)$$

where Δ denotes the XOR operation and N_c is the number of total labels. We use *Hamming Loss* averaging over training set to determine the best θ value for predicting query images. We also use it for measuring the performance of our label prediction method in experiment.

5. EXPERIMENT

We evaluate the performance of our method on two real-world applications: Scene classification and image annotation. For feature generation, we extract keypoints by dense sampling with a fixed grid of 8 pixels for each image. Then we use SIFT [9] feature to describe 40×40 patches around keypoints, so that each image is represented by a set of these local features. The only parameter C in formula (3) is fixed to be 1 throughout the experiment.

We also incorporate the spatial information by splitting each image into several spatial portions, and restrict each feature patch to only discover its NN feature in the same spatial portion during the I2C NN search. This idea is inspired by the spatial pyramid match method [7], while we only use a single spatial level instead of the whole multi-layers pyramid combination. In this way, we are able to enhance the efficiency and reduce the memory required for NN search, as the searching target is reduced from the whole image to its corresponding spatial portion. Meanwhile, the performance can also be improved since spatial information helps in finding the correct NN feature. We name our proposed weight learning method as **ML-LI2C**, short for Multi-Label learnt Image-to-Class distance. To validate the effectiveness of our optimization framework for learning the weight, we compare it with the unweighted I2C distance named as ML-I2C.

5.1 Natural scene classification

The natural scene dataset is first used in [23], which consists of 2000 images with 5 class labels: *desert*, *mountains*, *sea*, *sunset* and *trees*. About 22% of images contain more than one label and each image is associated with 1.24 labels on average. We split the whole dataset into 10 equal groups and perform a 10-fold cross validation experiment same as that in [20, 21] for comparing. Specifically, each time one group is selected for test and the rest are used for training, and the result is averaged over 10 runs.

For measuring the quality of multi-label scene classification, we use five evaluation metrics commonly used in previous works for multi-label learning [11, 20, 21, 23]: *Hamming loss*, *One-Error*, *Coverage*, *Ranking Loss* and *Average Precision*. Note only under the *Average Precision* metric, larger values indicate better results, while for the other 4 metrics, it should be the smaller the better. We briefly describe the measurement of these five evaluation metrics:

- *Hamming loss*: It measures the quality of predicted label by checking the difference between predicted labels and true labels using the XOR operation. Formula has been given in Section 4.
- *One-Error*: This metric only evaluates whether the top label in the rank list is not in the true label set. Let $l_1(X_i)$ denotes the top label ranked for image X_i and Y_i denotes the true label set, for a test set with N_x images, the average one-error is:

$$OneError = \frac{1}{N_x} \sum_i [l_1(X_i) \notin Y_i]$$

Here the $[f(x)]$ operation gives 1 when $f(x)$ is true and 0 otherwise.

- *Coverage*: This metric measures how far away the rank list is from “covering” all labels in the true label set.

Let $rank(l, X_i)$ denotes the position of label l in the rank list of X_i , then Coverage is given as:

$$Coverage = \frac{1}{N_x} \sum_i \max_{l \in Y_i} rank(l, X_i) - 1$$

- *Ranking Loss*: This metric counts the fraction of label pairs between true labels and irrelevant labels that are in the reverse order in the rank list:

$$RankingLoss = \frac{1}{N_x} \sum_i \frac{F(X_i)}{|Y_i| \cdot |\bar{Y}_i|}$$

Where $F(X_i) = |\{(l, l') | rank(l, X_i) > rank(l', X_i), l \in Y_i, l' \in \bar{Y}_i\}|$ is the number of reverse ordered label pairs in the rank list for image X_i .

- *Average Precision*: This measure is similar to Ranking Loss for measuring the quality of the whole rank list. It evaluates the average fraction of true labels ranked above a particular true label $l \in Y_i$:

$$Aver.Prec. = \frac{1}{N_x} \sum_i \frac{1}{|Y_i|} \sum_{l \in Y_i} \frac{G(X_i)}{rank(l, X_i)}$$

Where $G(X_i) = |\{l' \in Y_i | rank(l', X_i) \leq rank(l, X_i)\}|$ is the number of true labels ranked above label l .

We first compare our ML-LI2C with weight learnt to ML-I2C. As shown in Table 1, ML-LI2C gets significantly better result over all evaluation metrics, which validate the effectiveness of our learned weight. Then we compare our method with state-of-the-art results of previous multi-label methods, including InsDif [21], ML-KNN [20], BoosTexter [11], Rank-SVM [1] as well as two MIML algorithms MIML-Boost and MIML-SVM in [23] since these methods have reported the results on this dataset.

From the results summarized in Table 1, we can see that for every evaluation metrics at each column, our method of ML-LI2C again performs the best. Since the evaluation on *One-Error*, *Coverage*, *Ranking Loss* and *Average Precision* do not require the classifier to output the exact predicted labels, the performance improved over these four metrics should be contributed by our optimization framework for learning the effective weight and the use of I2C distance. More specifically, *One-Error* only measures the correctness of the top label, and we find our method can reduce about 1/3 prediction error for the top label compared to other methods. On the other side, *Coverage* only reflects the ranking position of the last true label in the rank list. Since the average labels per image is 1.24, our method requires about 0.38 more labels to cover all the true labels in the rank list, while other methods need 0.6-0.7 labels, nearly a one-fold increase in the number of labels compared to our method. The other two evaluation metrics *Ranking Loss* and *Average Precision* measure the whole quality for the rank list, and our significant improvement over the others methods. In Table 1 reflects the excellent quality of rank list generated by our method.

Meanwhile, *Hamming loss* is used for measuring the quality of predicted labels, and under this evaluation metric our ML-LI2C is also able to achieve significantly better performance compared to previous methods. Even without the learned weight, the unweighted I2C distance can also predict labels relatively well, as can be seen by comparing the *Hamming loss* of MI-I2C to previous methods.

Table 1: Performance evaluation on natural scene dataset, ↓ means the smaller the better and ↑ means the larger the better

Method	Evaluation Metric				
	<i>Hamming Loss</i> ↓	<i>One-Error</i> ↓	<i>Coverage</i> ↓	<i>Ranking Loss</i> ↓	<i>Average Precision</i> ↑
ML-LI2C	0.129±0.011	0.190±0.024	0.624±0.060	0.091±0.011	0.881±0.014
ML-I2C	0.159±0.019	0.311±0.047	0.883±0.089	0.156±0.024	0.804±0.027
InsDif [21]	0.152±0.016	0.259±0.030	0.834±0.091	0.140±0.018	0.830±0.019
ML-KNN [20]	0.169±0.016	0.300±0.046	0.939±0.100	0.168±0.024	0.803±0.027
BoosTexter [11]	0.179±0.015	0.311±0.041	0.939±0.092	0.168±0.020	0.798±0.024
Rank-SVM [1]	0.253±0.055	0.491±0.135	1.382±0.381	0.278±0.096	0.682±0.092
MIML-Boost [23]	0.189±0.009	0.351±0.039	0.989±0.103	0.181±0.026	0.777±0.025
MIML-SVM [23]	0.180±0.017	0.327±0.033	1.022±0.085	0.187±0.018	0.783±0.020

Table 2: Performance evaluation on MSRC dataset, ↓ means the smaller the better and ↑ means the larger the better

Method	Evaluation Metric				
	<i>Hamming Loss</i> ↓	<i>One-Error</i> ↓	<i>Coverage</i> ↓	<i>Ranking Loss</i> ↓	<i>Average Precision</i> ↑
ML-LI2C	0.071±0.002	0.200±0.011	3.254±0.224	0.066±0.007	0.809±0.014
ML-I2C	0.099±0.004	0.330±0.027	5.246±0.122	0.145±0.004	0.655±0.010









				
ML-LI2C	grass, tree, sky	grass, tree, cow, sheep	building, road, car, tree, sky, grass	sky, building, road, grass, tree, aeroplane
ML-I2C	grass	grass	building, road	building, sky
Ground Truth	grass, tree, sky	grass, tree, cow	building, road, car	sky, building, road, grass, aeroplane
				
ML-LI2C	body, face	road, building, bicycle	sky, grass, tree, building, road	grass, sky, tree, building, road
ML-I2C	building, road	building, road	grass	grass
Ground Truth	body, face, building	road, building, bicycle	sky, tree, building, road	sky, tree, building, body

Figure 3: Annotation result of ML-LI2C compared to ML-I2C and ground truth of MSRC dataset. Wrongly annotated keywords are in italic font.

5.2 Image annotation on MSRC dataset

For the application of image annotation task, we use MSRC dataset for evaluation, which is a new dataset but widely used in recent years [8, 10, 13, 15, 18]. This dataset consists of 591 images assigned to 21 predefined labels (As suggested by MSRC, the label “horse” and “mountain” are ignored due to a limited number of labeled images). Around 80% images are assigned to more than 1 label with about 3 labels per image on average. We randomly split the dataset into 391 training images and 200 test images, with additional constraint that each label contains at least 10 images in the

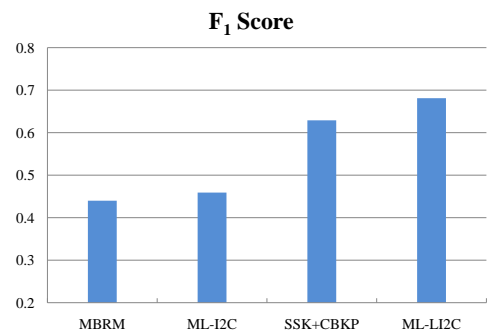


Figure 4: F_1 score for measuring the performance of image annotation.

training set. Results are averaged over 5 runs with different training and test set.

We first report our result of ML-LI2C and the unweighted ML-I2C on the previous five evaluation metrics. As shown in Table 2, the weighted I2C distance once again greatly improves the performance over the unweighted one in all metrics. Although there are more labels in this dataset compared to scene dataset, results on *Hamming Loss*, *Ranking Loss* and *Average Precision* is similar to that in scene dataset. As each image is annotated with more labels on average, the value of *Coverage* is also increased. But compared to ML-I2C, ML-LI2C only requires about half of rank list for covering all the true labels. For *One-Error*, we notice this error increased a little compared to that in scene dataset. Nevertheless, the better result on *Hamming Loss* indicates that true labels are still at the front of the rank list and is able to be predicted by our method. Some examples anno-

tated by our method are present in Figure 3. Compared to the annotation results of unweighted ML-I2C that only predict very limited inaccurate keywords, ML-LI2C with weight learnt is able to provide better results covering more accurate keywords. ML-LI2C predicts more labels than ML-I2C in most images, but as shown in Table 2, the *Hamming Loss* of ML-LI2C is still lower than ML-I2C.

In addition, we compare our method to related methods like MBRM [4] and SSK [10] by F_1 score, since this evaluation metric is commonly used in automatic image annotation studies and has been used by others in this dataset. It measures the quality of retrieved relevant images with each keyword, which is treated as class label here. Given each keyword, the number of images annotated with this label in the ground truth is denoted as N_t , the number of images annotated with this label by the classifier is denoted as N_r , and the number of images correctly annotated with this label by the classifier is denoted as N_c . Then F_1 score is defined as:

$$\begin{aligned} Precision &= \frac{N_c}{N_r} \\ Recall &= \frac{N_c}{N_t} \\ F_1 &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \end{aligned}$$

Our methods are compared with MBRM [4] and SSK combined with CBKP [10]. Same as [10], the top 3 keywords in the rank list is taken as automatic annotation of a test image. All the results are summarized in Figure 4. Without learning the weight, ML-I2C is a little better than MBRM but lower than both ML-LI2C and SSK+CBKP. However, our ML-LI2C with weight learnt perform better than SSK+CBKP and has achieved state-of-the-art performance.

6. CONCLUSION

In this paper, we have used I2C distance for multi-label learning problem. First, we established a label rank by sorting the I2C distance to each predefined class labels. Second, we made use of the label correlation information for deciding the splitting position in the label rank list, so as to predict labels associated with a given query image. Third, to cater I2C distance in the multi-label problem, we have formulated a large margin optimization problem to learn a weight attached to each local feature patch. The effectiveness of these three contributions were validated through two image datasets for scene classification and image annotation applications. Compared to previous related works, our method achieved better performance on various different evaluation metrics. For future work, we would want to continue exploring learning the I2C distance and try to incorporate label correlation information within the learning framework.

7. REFERENCES

- [1] J. W. A. Elisseeff. A kernel methods for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, pages 681–687, Cambridge, MA, 2002. MIT Press.
- [2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2008.
- [3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1751–1771, 2004.
- [4] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1002–1009, 2004.
- [5] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of IEEE International Conference on Computer Vision*, October 2007.
- [6] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1719–1726, 2006.
- [7] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.
- [8] T. Li, T. Mei, S. Yan, I.-S. Kweon, and C. Lee. Contextual decomposition of multi-label images. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2270–2277, June 2009.
- [9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [10] Z. Lu, H. H. Ip, and Q. He. Context-based multi-label image annotation. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, July 2009.
- [11] R. E. Schapire and Y. Singer. BoosTexter: a boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.
- [12] G. Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, and I. Vlahavas. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *ECML PKDD Workshop on Learning From Multi-Label Data*, pages 101–116, September 2009.
- [13] S. Vijayanarasimhan and K. Grauman. What’s it going to cost you?: predicting effort vs. informativeness for multi-label image annotations. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2262–2269, June 2009.
- [14] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1643–1650, June 2009.
- [15] H. Wang, H. Huang, and C. Ding. Image annotation using multi-label correlated green’s function. In *Proceedings of IEEE International Conference on Computer Vision*, 2009.
- [16] M. Wang, X. Zhou, and T.-S. Chua. Automatic image annotation via local multi-label classification. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 17–26, July 2008.

- [17] Z. Wang, Y. Hu, and L.-T. Chia. Learning instance-to-class distance for human action recognition. In *IEEE International Conference on Image Processing*, pages 3545–3548, November 2009.
- [18] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang. Joint multi-label multi-instance learning for image classification. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2008.
- [19] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, and X.-S. Hua. Graph-based semi-supervised learning with multi-label. In *IEEE International Conference on Multimedia and Expo*, pages 1321–1324, 2008.
- [20] M.-L. Zhang and Z.-H. Zhou. Ml-knn: a lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.
- [21] M.-L. Zhang and Z.-H. Zhou. Multi-label learning by instance differentiation. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 669–674, 2007.
- [22] M.-L. Zhang and Z.-H. Zhou. M3MIML: a maximum margin method for multi-instance multi-label learning. In *Proceedings of IEEE International Conference on Data Mining*, pages 688–697, December 2008.
- [23] Z.-H. Zhou and M.-L. Zhang. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems 19*, pages 1634–1641, Cambridge, MA, 2007. MIT Press.