Multi-label Image Classification with A Probabilistic Label Enhancement Model

Xin Li and **Feipeng Zhao** and **Yuhong Guo**
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA
{xinli, feipeng.zhao, yuhong}@temple.edu

Abstract

In this paper, we present a novel probabilistic label enhancement model to tackle multi-label image classification problem. Recognizing multiple objects in images is a challenging problem due to label sparsity, appearance variations of the objects and occlusions. We propose to tackle these difficulties from a novel perspective by constructing auxiliary labels in the output space. Our idea is to exploit label combinations to enrich the label space and improve the label identification capacity in the original label space. In particular, we identify a set of informative label combination pairs by constructing a tree-structured graph in the label space using the maximum spanning tree algorithm, which naturally forms a conditional random field. We then use the produced label pairs as auxiliary new labels to augment the original labels and perform piecewise training under the framework of conditional random fields. In the test phase, max-product message passing is used to perform efficient inference on the tree graph, which integrates the augmented label pair classifiers and the standard individual binary classifiers for multi-label prediction. We evaluate the proposed approach on several image classification datasets. The experimental results demonstrate the superiority of our label enhancement model in terms of both prediction performance and running time comparing to the-state-of-the-art multi-label learning methods.

1 INTRODUCTION

With the development of internet and digital devices, the availability of visual data has been dramatically increasing in recent decades, which provides billions of images and videos. An important task for information retrieval and processing over such image and video data is object-based image annotation, which requires identifying a set of objects presented in each image from a given set of desired object concepts. The image annotation problem for object recognition is an inherent multi-label classification problem, since each image usually contains more than one object of interest. Multi-label classification generalizes the standard multi-class classification by allowing each instance to be simultaneously assigned into multiple label categories. A key challenge for multi-label classification is label sparsity. That is, the multiple labels are supported by the training data at different levels and many rare labels may lack sufficient training supports to be reliably recognized individually. Hence instead of learning binary classifiers independently for each label, many multi-label learning methods have proposed to exploit label correlations or label dependence to improve multi-label classification performance, including second-order strategy methods [7, 11, 15], which model pairwise label correlations, and high-order strategy methods [16, 19, 36], which consider the interactions among subsets of labels.

Moreover, on image classification, multi-label learning also faces the general intra-class variation challenge of standard multi-class classification which is caused by viewpoint and context variations and occlusions, as shown in Figure 1. From the figures, we can see that the appearance of the object *people* can be profoundly different across different images, by co-occurring with different objects and having different occlusion patterns. In such cases, an individual binary classifier may not be reliable for recognizing a target object. But other co-occurred objects can likely provide some useful information. For example, in the image "*people* riding a *bike*", many parts of the *bike* are invisible, but the *people* is easy to recognize and can provide information about the *bike*; in the image "*people* sitting in the *car*", each *people* is severely occluded but the *car* can be detected easily and help the recognition of the *people* if they often co-occur; in the image "*people* riding a *horse*", the objects *people* and *horse* can be helpful to each other as well. Moreover, the recognition of such composite co-occurrence patterns can also help the correct recognition of individual objects, especially for the ones that are occluded

Figure 1: Examples of image occlusion and object co-occurrence patterns in object recognition tasks.

or difficult to be recognized individually.

Motivated by these observations, in this paper, we propose a novel probabilistic label enhancement model that utilizes the label combination patterns to improve multi-label image classification performance. Our assumption is that visual composites can be helpful when single classifier fails. For example, assume the composite "*people riding horses*" often stays in similar poses. A classifier for this visual composite can then capture the co-appearance of the two objects even though both *people* and *horse* classifiers fail to reliably recognize them separately. We propose to construct label combinations, in particular label pairs, as auxiliary new labels to augment the original labels and improve label identification capacity in the original label space. In particular, we identify a set of informative label combination pairs by constructing a tree-structured graph in the label space using the maximum spanning tree algorithm, according to the label co-occurrence information in the training data. This naturally forms a conditional random field framework, under which we perform piecewise training. The label pairs identified by the edges of the tree are used as auxiliary new labels, and a binary classifier is trained for each label in the augmented label space. To integrate the augmented multiple binary classifiers for multi-label prediction in the original label space, we perform exact inference on the tree-structured conditional random field using max-product message passing. We evaluate the proposed approach on a number of multi-label image classification datasets. The experimental results show that the proposed label enhancement model effectively outperforms the related state-of-the-art methods in terms of both prediction performance and running time.

The rest of the paper is organized as follows. In Section 2, we present a brief review over the related work. The proposed approach is presented in Section 3. We report the experimental results in Section 4 and finally conclude the paper in Section 5.

2 RELATED WORK

A considerable amount of research has been devoted to addressing image annotation and multi-label classification problems in the literature. In this section, we will provide a brief review over the most related work to the proposed approach from the perspectives of image annotation, object interaction and multi-label classification.

Image Annotation There are three major groups of image annotation techniques [13]: (i) Generative models. Some methods in this group use generative topic models such as latent Dirichlet allocation [1], probabilistic latent semantic analysis [24], and hierarchical Dirichlet processes [33]. They model annotated images as samples from a mixture of topics, where each topic is a distribution over image features. Some other methods use mixture models to define a joint distribution over image features and annotation tags [4, 9, 22]. However, generative models perform training by maximizing generative data likelihoods, which are not necessarily optimal for the target prediction performance. (ii) Discriminative models. The methods in this group address image annotation as a classification problem. For example, simple methods in [23, 12] treat labels independently and learn a classifier for each label, while more advanced methods in [30, 3] improve the classification performance by considering the co-occurrences of different labels. (iii) Nearest neighbor based models. For example, the label propagation method in [13] constructs a similarity graph for all images, and propagates the label information via the graph; and the search based method in [10] exploits a regression based kernel metric.

Object Interaction There are a number of works on object interaction that share the same intuition as our proposed work, that is, visual composites can be helpful while single components fail. The work in [18] learns object interactions by modeling the prepositions and adjectives that relate nouns. The work in [34] models the co-occurrence of objects and human poses in human-object interaction activities. In [8], the interactions between objects are modeled implicitly in the context of predicting sentences for images. [28] introduces a complex visual composite concept called visual phrase for object detection, which treats each phrase as a new label. Though this work shares similarity with our proposed work in exploiting visual composites, there are significant differences between it and our work. First, its visual phrases are not automatically discovered but predefined. By contrast, the label combinations in our work are

constructed automatically. Second, it addresses very different problems from ours. It tackles object detection tasks while we address multi-label image annotation problems; its goal is to find a bounding box where the visual composite occurs, while our goal is to predict the category labels of an image.

Multi-label Classification The most straightforward multi-label classification method is binary relevance [2], which trains a binary classifier for each label. The obvious flaw of such method is the complete ignorance of label correlations. Hence, numerous methods that encode label correlations have been proposed. One group is the ranking based methods [7, 11, 32], which rank the relevant labels higher than irrelevant ones and capture label correlations implicitly in the loss function. This technique however relies on a good distance metric and a fine-tuned threshold in determining the number of relevant labels. The method in [15] hence further eliminates this drawback by developing a novel calibrated separation ranking loss function. Another group is the graph-based methods, which implicitly incorporate label correlations into label propagation algorithms as either part of the graph weights [20, 5] or additional constraints [30, 35]. There are also a set of probabilistic graph-based methods [6, 14, 17, 27]. The method in [14] uses directed graphs over the label variables to capture label dependence under a probabilistic conditional dependency network model. [17] further improves this model by learning sparse conditional dependency graphs. [6, 27] integrate multiple classifiers in a chain graph to capture label correlations. These methods share similarity with our proposed approach in capturing label correlations by integrating probabilistic classifiers on graphs over labels. However, [6, 27] are limited to chain graphs and they apply greedy heuristics to search for the best label vector on each test instance; [14, 17] use cyclic directed graphs and their test phases involve approximate inference. By contrast, our method can exploit any automatically generated acyclic tree graphs, not necessarily chains, while using a max-product message passing algorithm to perform efficient exact inference.

3 PROPOSED MODEL

3.1 Preliminaries

Multi-label Classification systems can be described as below. Given the input feature space $\mathcal{X} \in \mathbb{R}^d$ and the output label space $\mathcal{Y} = \{0, 1\}^L$, a mapping function $\mathbf{h}$: $\mathcal{X} \rightarrow \mathcal{Y}$ can be used to predict the corresponding label vector $\mathbf{y} \in \mathcal{Y}$ for each input data instance $\mathbf{x} \in \mathcal{X}$. Multi-label learning focuses on identifying a good mapping function $\mathbf{h}$ from the training data. The most straightforward method to learn such a mapping function is the binary relevance method, which assumes labels are independently generated and learns one binary classifier h_i for each label. The output of $\mathbf{h}$ is a L-length binary vector

$$\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_L(\mathbf{x})] .$$

The binary classifier h_i can be trained by minimizing different loss functions, such as log-loss and hinge loss.

Conditional Random Fields (CRFs) are undirected graphical models that model the conditional distribution of the output labels given an input vector based on undirected graphs. An undirected graph $G = (V, E)$ in the label space is formed by a set of vertices V, each of which represents a label variable, and a set of undirected edges E, where each edge consists of a pair of vertices $(s, t) \in E$ and represents the dependence relationship between the label variables. The joint probability of a configuration of the label variables in a CRF can be given by

$$P(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \psi_C(\mathbf{y}_c, \mathbf{x})$$

where $Z(\mathbf{x})$ is a partition function that ensures a valid conditional distribution given the input data instance $\mathbf{x}$, $\mathcal{C}$ is the set of cliques of the graph, and ψ_c is the potential function for clique c, which maps the clique label configuration $\mathbf{y}_c$ and the input data instance $\mathbf{x}$ into a positive scalar value. The standard training procedure of conditional random fields typically involves first-order gradient descent or second-order Newton methods, which require performing inference over each training instance in each iterative parameter update step. For general graphs, performing exact inference in CRFs is intractable, and approximate inference algorithms are usually used instead. Moreover, performing inference in each parameter update step can make the training process computationally expensive, especially for large and densely connected graphs and large training sets.

3.2 Probabilistic Label Enhancement Model

The straightforward method for multi-label image classification casts the problem as a set of independent binary classification problems, one for each object label, and trains one binary classifier for each label using the one-vs-all scheme. Training binary classifiers is computationally efficient and the one-vs-all training scheme can scale linearly with the increasing of the label set. However, as we discussed before, such binary classifiers can fail to accurately recognize the individual objects in an image, due to label sparsity, intra-class variations, and occlusions. In this work, we propose to enhance these standard binary classifiers by exploiting the label combination patterns which typically present as co-occurred object composites in images of the training data, as shown in Figure 1. We first identify the informative label combination pairs by learning a tree-structured undirected graph in the label space, which forms the structure of a conditional random field.

Then we use the label combination pairs as augmenting new labels and formulate the learning process as a piecewise training procedure under the framework of conditional random fields. Finally we apply the trained probabilistic model to predict labels for test images using a max-product exact inference algorithm.

3.2.1 Learning Tree-Structured Graph

Given the labeled training images $D=\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$, where each label vector $\mathbf{y}^{(i)}$ contains $\{0,1\}$ values with length L, corresponding to the L label classes, we aim to identify the useful object composites by finding the informative label co-occurrence patterns. Though in principle we can consider any label combination patterns, for computational simplicity we focus on second-order patterns, i.e., label pairs. We take all possible label pairs as candidates by constructing a fully connected graph over the L label variables. Then we measure the combination strength of each label pair as the weight of the corresponding edge using an appropriate criterion. One standard criterion is the empirical *mutual information* measure, which is popularly used to measure the dependence strength of two variables and can be computed from the training data. For example, the empirical mutual information between label variables Y_i and Y_j can be computed as

$$MI(Y_i; Y_j) = \sum_{y_i, y_j \in \{0,1\}} \hat{P}(y_i, y_j) \log \left(\frac{\hat{P}(y_i, y_j)}{\hat{P}(y_i)\hat{P}(x_j)} \right)$$

with the empirical probabilities computed from the training data. However, this measure treats the co-presence of the two labels and the co-missing of them equivalently, while we want to find the label composites that have significant co-presence patterns. Hence we propose a simple new measure, *normalized co-occurrence*, to use. For two label variables Y_i and Y_j, the normalized co-occurrence measure is defined as

$$NC(Y_i; Y_j) = \frac{count(Y_i, Y_j)}{\min\left(count(Y_i), count(Y_j)\right)}$$

where $count(Y_i, Y_j)$ is the number of co-occurrence of the two labels in the training data, such that

$$count(Y_i, Y_j) = \sum_{\ell=1}^{n} I[\mathbf{y}_i^{(\ell)} = 1, \mathbf{y}_j^{(\ell)} = 1] \qquad (1)$$

and $I[\cdot]$ denotes an indicator function. Similarly, $count(Y_i)$ and $count(Y_j)$ are the numbers of occurrences of single labels in the training data. By normalizing the co-occurrence counts of the two labels with the minimum of their individual occurrence counts, the measure emphasizes the relative relatedness of the two objects and favors the less frequently appeared objects. For example, assume there are 15 images containing *people* and *dogs*, and 10 images containing *people* and *cars*, while there are totally 100 people images, 80 dog images and 20 car images. The composite of *people* and *cars* can be more important to capture than the composite of *people* and *dogs*, towards the goal of assisting the objects with sparse supports in the training data. Our proposed measure encodes this principle.

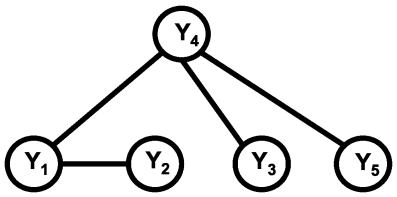


Figure 2: An example of the constructed tree-structured graph over labels.

Given the proposed normalized co-occurrence criterion, we can compute the weights for all edges between the label variables. Then we use a maximum spanning tree algorithm to select $(L-1)$ edges according to the computed weights to form a tree-structured graph. In our implementation, we used Prim's algorithm [26] to produce the maximum spanning tree. Figure 2 demonstrates an example of the constructed tree graph over the label variables. The label pair connected by each edge on the constructed tree graph will be used as a constructed new label to augment the original labels. For example, for the tree graph in Figure 2, since there is an edge between the node Y_1 and the node Y_2, we will consider a constructed new label $Y_{1\sim2}$, which has binary values $\{0, 1\}$. The label value for $Y_{1\sim2}$ in each instance can be produced based on that $Y_{1\sim2} = 1$ is equivalent to $Y_1 = 1 \wedge Y_2 = 1$. Then for each instance $\mathbf{x}^{(i)}$, we set $\mathbf{y}_{1\sim2}^{(i)} = 1$ if and only if the instance has been assigned both label Y_1 and label Y_2 such that $\mathbf{y}_1^{(i)} = 1$ and $\mathbf{y}_2^{(i)} = 1$. Otherwise, we have $\mathbf{y}_{1\sim2}^{(i)} = 0$. Thus each constructed new label can be treated as a new prediction class from the prediction perspective. The reason that we produce tree graphs instead of densely connected cyclic graphs is that tree graphs have acyclic structures and permit efficient exact inference in the test phase to integrate the augmented label classifiers with the binary classifiers in the original label space.

3.2.2 Piecewise Training of CRFs

The tree-structured graph constructed in the label space actually forms a standard CRF model that permits label vector prediction from the input data, where we treat each node and each edge as separate cliques. Based on our motivation of capturing object composite concept to help the multi-label prediction in the original label space, we propose to perform piecewise training for the tree-structured CRF by learning the potential functions for each node clique and each edge clique separately. That is, we train a set of L binary classifiers independently from the data, one for each

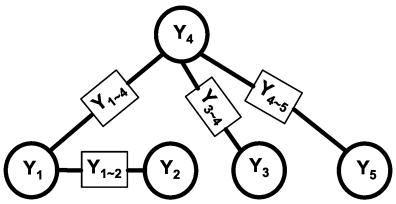


Figure 3: The factor graph constructed from the tree graph in Figure 2. The circle nodes are variable nodes and the rectangle nodes are factor nodes.

label in the original label space, as the potential functions for the node cliques, and train a set of $(L-1)$ binary classifiers independently from the data for the constructed new labels as the potential functions for the corresponding edge cliques. Piecewise training can effectively avoid the repeated inference required for each step of parameter updates in the standard CRF training procedure and make the learning process efficient and scalable. It has been shown in [29] that piecewise training of a CRF can be justified as minimizing a family of upper bounds on the log partition function of the data log-likelihood.

To have the outputs of potential functions compatible to each other, we propose to use binary probabilistic classifiers, in particular binary logistic regression classifiers, for training. Each binary logistic regression classifier can be trained efficiently by using second-order Newton methods to minimize the regularized log-likelihood. For the k-th classifier, this is to minimize

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + e^{-\hat{\mathbf{y}}_k^{(i)} \mathbf{w}^\top \mathbf{x}^{(i)}}\right) + \frac{\beta}{2}\mathbf{w}^\top \mathbf{w} \qquad (2)$$

where β is a trade-off parameter, and $\hat{\mathbf{y}}_k^{(i)}$ is simply the translation of $\mathbf{y}_k^{(i)}$ from values $\{1, 0\}$ to $\{1, -1\}$.

3.2.3 Inference with Max-product Algorithm

Given the trained tree-structured CRF model, the multi-label prediction on a test instance can be performed using the max-product inference algorithm [21]. The max-product algorithm conducts label decoding through message passing which operates in factor graphs. Given the trained pairwise CRF model, we then first transfer it into a factor graph by simply keeping all variable nodes and adding a factor node for each edge clique. For example, the factor graph constructed for the tree-structured CRF in Figure 2 is given in Figure 3, where each variable node is represented as a circle and each factor node is represented as a rectangle. For a given test instance $\mathbf{x}$, the potentials of the two types of nodes in the factor graph can be computed using the probabilistic binary classifiers produced in the training phase, such as $\psi(\mathbf{y}_i) = P(\mathbf{y}_i|\mathbf{x})$ and $\psi(\mathbf{y}_i, \mathbf{y}_j) = \psi(\mathbf{y}_{i\sim j}) = P(\mathbf{y}_{i\sim j}|\mathbf{x})$.

The decoding process on the test instance $\mathbf{x}$ aims to find the maximum a posteriori (MAP) label assignment by solving

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \; P(\mathbf{y}|\mathbf{x}) \qquad (3)$$

The max-product algorithm performs this decoding on the factor graph using the following message passing. First, we randomly select a variable node r as the root of the tree, and pass messages from leaves until they reach the root. There are two types of messages: node-to-factor messages and factor-to-node messages. The message from the node i to the factor a (e.g., $a = i \sim j$) can be computed as

$$\mu_{i\to a}(\mathbf{y}_i) = \psi(\mathbf{y}_i) \prod_{c\in N(i)\backslash a} \mu_{c\to i}(\mathbf{y}_i) \qquad (4)$$

where $N(i)\backslash a$ represents all the neighboring factor nodes of node i excluding factor node a. The message from the factor $a = i \sim j$ to the node j can be computed as

$$\mu_{a\to j}(\mathbf{y}_j) = \max_{\mathbf{y}_i} \left(\psi_a(\mathbf{y}_i, \mathbf{y}_j)\mu_{i\to a}(\mathbf{y}_i)\right) \qquad (5)$$

Back pointers are kept for each value that achieves the maximum at a max operation. At the root, we multiply all incoming messages to obtain the maximum probability and the MAP configuration of the root node $\mathbf{y}_r^*$

$$P^* = \max_{\mathbf{y}_r} \left(\psi(\mathbf{y}_r) \prod_{a\in N(r)} \mu_{a\to r}(\mathbf{y}_r)\right) \qquad (6)$$

$$\mathbf{y}_r^* = \arg\max_{\mathbf{y}_r} \left(\psi(\mathbf{y}_r) \prod_{a\in N(r)} \mu_{a\to r}(\mathbf{y}_r)\right) \qquad (7)$$

We then back trace the pointers and find the complete values $\mathbf{y}^*$ that lead to P^*. For tree graphs, this max-product algorithm provides an exact inference solution. But for an arbitrary graph with loops, it can only provide an approximate solution.

4 EXPERIMENTS

To evaluate the proposed approach, we conducted experiments on several standard multi-label image classification datasets, comparing to a few state-of-the-art multi-label learning methods and baselines. We report our empirical results in this section.

4.1 Datasets

We used the following three image datasets in our experiments: *Pascal VOC 2007 (Pascal07)*, *Corel5K*, and *SUN 2012*. *Pascal07* is one of the most famous image datasets for classification and detection and it contains 20 different object classes. *Corel5K* is a standard image set for multi-label classification with $5,000$ instances and 260 classes. To have a fair comparison with a few comparison methods,

some of which are too slow to deal with many classes, we selected two subsets with 50 most frequent labels and 100 most frequent labels respectively to use. *SUN 2012* [31] is a recently released large-scale image set for object detection. Similarly, we used two subsets with 50 labels and 100 labels respectively. The properties of the datasets used in our experiment are briefly summarized in Table 1, where cardinality denotes the average number of labels assigned to one image. In these datasets, each image is represented as a 512-dimension *GIST* [25] feature vector.

Table 1: Summary information of the datasets.

Dataset	#images	#labels	cardinality
Pascal07	4168	20	2.26
Corel5K(s50)	4999	50	2.32
Corel5K(s100)	4999	100	2.89
SUN12(s50)	5000	50	8.98
SUN12(s100)	5000	100	11.18

4.2 Experimental Results

On each of five datasets, we compared the proposed approach to the following state-of-the-art multi-label classification methods and baseline method:

- *Ensembled Probabilistic Classifier Chain (EPCC)*. This probabilistic multi-label learning method is developed in [6] and it integrates base classifiers in a chain structure in the label space.
- *Maximum Margin Output Coding (MMOC)*. This is a multi-label learning method developed in [36], which performs classification on a simultaneously learned lower-dimensional label space within a maximum margin framework.
- *Logistic Regression (LR)*. This is a baseline method that trains a set of independent binary logistic regression classifiers, one for each label, to perform multi-label classification.

For our proposed approach, there is one regularization trade-off parameter β to set for the logistic regression classifiers. We found that logistic regression classifiers are not very sensitive to this parameter. In our experiments, we set β as a very small value around 0.0002. For the comparison methods *EPCC* and *MMOC*, we used the code packages released on the internet[1]. These packages contain parameter selection procedures and settings.

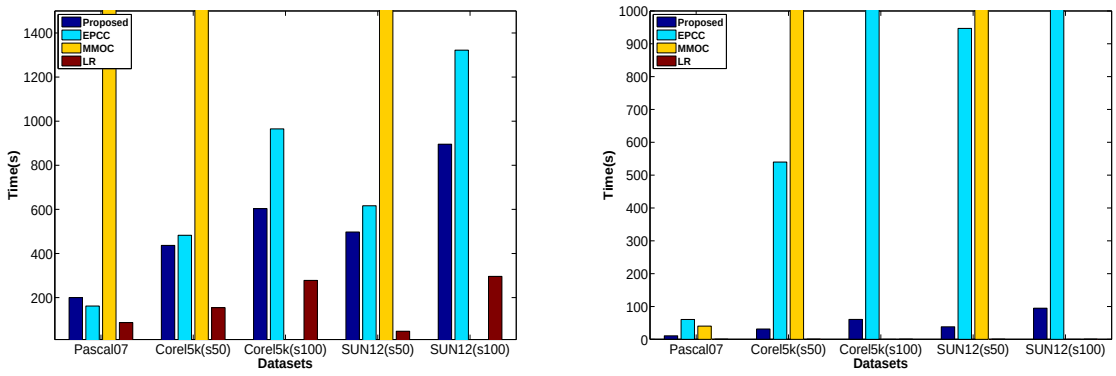
On each dataset, we performed a 5-fold cross validation to compare all the methods. To evaluate the multi-label classification results from different perspective, we used five standard criteria: macro-F1, micro-F1, hamming loss, precision and recall. The average results and standard deviations in terms of the five criteria for all the four methods are reported in Table 2. We can see that our proposed method outperforms all the other comparison methods across all five datasets and in terms of all the four measure criteria: macro-F1, micro-F1, Precision and Recall. In terms of hamming loss, the proposed approach produced the best results on four out of the five datasets. Moreover, the proposed approach significantly outperforms the baseline *LR* method across all different settings. This clearly shows that the augmenting new labels in our model are very effective in assisting identifying the individual labels, and it is very beneficial to exploit the label co-occurrence patterns. By comparing the results on the *Corel5k(s50)* dataset and the *Corel5k(s100)* dataset, we can see that with the increasing of the label set size, the performance of all methods in terms of the four measures, macro-F1, micro-F1, precision and recall, has the general trend of decreasing. In terms of the hamming loss, however, the results of all approaches are even better on *Corel5k(s100)* than on *Corel5k(s50)*. This seems very strange. But if we check the two datasets, we can see that though *Corel5k(s100)* contains 50 more labels than *Corel5k(s50)*, the difference between their label cardinality values is very small. This indicates that the labels are even more sparse in *Corel5k(s100)* than in *Corel5k(s50)*. By producing similar number of positive labels, the performance of each approach will automatically get better in terms of hamming loss, with the increasing of the label set size. This result suggests that hamming loss is not an appropriate criterion for multi-label classification when the label cardinality is small while the number of label classes is large. Similar results are observed across *SUN12(s50)* and *SUN12(s100)* as well. Another observation over the table is that *EPCC* produced the second best results in most cases. The *EPCC* method greatly outperforms the baseline *LR* almost on all the datasets and in terms of all criteria, except that on *SUN12(s100)* in terms of macro-F1 and on *SUN12(s50)* in terms of precision, where it produces similar results with *LR*. The *MMOC* method is much more time-consuming than other methods. On the two datasets with 100 labels, it fails to yield any result within reasonable period of running time. It has inferior performance comparing to the proposed approach and *EPCC* in most cases.

Running time To compare the empirical efficiency of the approaches, we have also recorded the training time and testing time of each approach on a 64-bit machine with 16GB memory and quad core intel i7 processors. The results of average running time are reported in Figure 4. We can see the baseline *LR* is the most efficient method in terms of both training and testing time, since it only needs to train a set of binary classifiers and perform classification independently for each label. Among the remaining three methods, our proposed approach is significantly more ef-

[1] https://github.com/multi-label-classification/PCC; http://www.cs.cmu.edu/ yizhang1/files/ICML2012_Code.zip

Table 2: The average results and standard deviations of all the comparison methods on the five datasets in terms of different evaluation criteria. On each dataset, the best result in each criterion across different methods is shown in bold font. '-' denotes the fact that the method fails to run on the corresponding dataset due to the large label size.

Measure	Methods	Datasets				
		Pascal07	Corel5k(s50)	Corel5k(s100)	SUN12(s50)	SUN12(s100)
Macro-F1	Proposed	**0.268±0.005**	**0.276±0.003**	**0.167±0.004**	**0.377±0.004**	**0.315±0.002**
	EPCC	0.252±0.005	0.245±0.006	0.158±0.002	0.355±0.002	0.210±0.003
	MMOC	0.220±0.003	0.218±0.002	-	0.318±0.002	-
	LR	0.247±0.006	0.201±0.003	0.135±0.002	0.323±0.002	0.215±0.002
Micro-F1	Proposed	**0.579±0.004**	**0.362±0.005**	**0.333±0.003**	**0.581±0.003**	**0.514±0.002**
	EPCC	0.567±0.003	0.351±0.002	0.327±0.002	0.563±0.001	0.507±0.002
	MMOC	0.543±0.006	0.238±0.003	-	0.514±0.002	-
	LR	0.481±0.007	0.222±0.003	0.197±0.003	0.486±0.003	0.386±0.003
Hamming Loss	Proposed	**0.057±0.003**	0.062±0.002	**0.023±0.002**	**0.146±0.003**	**0.089±0.004**
	EPCC	0.094±0.001	0.077±0.001	0.047±0.000	0.166±0.001	0.110±0.000
	MMOC	0.089±0.002	**0.057±0.001**	-	0.154±0.001	-
	LR	0.121±0.002	0.079±0.001	0.049±0.000	0.170±0.001	0.138±0.001
Precision	Proposed	**0.697±0.013**	**0.343±0.003**	**0.310±0.005**	**0.656±0.003**	**0.541±0.003**
	EPCC	0.649±0.004	0.311±0.002	0.300±0.003	0.544±0.001	0.517±0.002
	MMOC	0.689±0.011	0.225±0.004	-	0.614±0.003	-
	LR	0.518±0.010	0.198±0.003	0.185±0.004	0.548±0.004	0.403±0.003
Recall	Proposed	**0.570±0.010**	**0.458±0.014**	**0.418±0.008**	**0.641±0.002**	**0.565±0.005**
	EPCC	0.557±0.003	0.453±0.005	0.404±0.004	0.610±0.003	0.523±0.003
	MMOC	0.482±0.004	0.184±0.005	-	0.464±0.003	-
	LR	0.507±0.006	0.235±0.004	0.206±0.003	0.456±0.002	0.396±0.002



(a) Training (b) Testing

Figure 4: Training and testing time (seconds) for all methods. Note on Corel5k(s100) and SUN12(s100), the yellow bar (for MMOC) is missing due to that MMOC fails to handle these datasets.

ficient than the other two methods in terms of the testing time. For training time, the proposed approach is similar to *EPCC* on the dataset *Pascal07* which has small label set, and is more efficient than *EPCC* on the other larger scale datasets. *MMOC* is the most inefficient one among all the four methods. It even fails to produce any results on the two datasets with 100 labels.

Illustration of the Results To have an illustrative understanding about the image annotation problem and the prediction results, in Table 3 we presented the predicted labels on four testing images from the *SUN12(s50)* dataset by the four methods. The true positive labels are shown in bold font. We can see that our proposed approach is in general more accurate than the other methods, though *EPCC* has

Table 3: The predicted labels on four test images of SUN12(s50) by the comparison methods. The true positive labels are shown in bold font.

Methods				
Proposed	**wall, floor, ceiling, chair, door, cabinet, table, vase**, bottle, window	**floor, wall, ceiling, door, table, person, box, books**, chair	**wall, door, road, car, sky**, trees, person, mountain	**door, sky, trees, grass**
EPCC	**wall, floor, ceiling, chair, ceiling lamp, table, vase, flowers**, window, plant	**wall, floor, ceiling, chair, door, person**, ceiling lamp, window, cabinet	**sky**, window, door, plant, building, tree, grass	**sky, tree**, wall, floor, window, ceiling, chair, door, table, plant
MMOC	**wall, floor, window, ceiling, chair, table**, curtain, sofa, window	**wall, floor, ceiling, person**, window, ceiling lamp	**sky, car**, ceiling, grass, plant, building, tree, streetlight	**sky, tree**, wall, window, plants
LR	**wall, floor, ceiling, chair, table**, bottle, window, curtain, rug, sofa	**wall, floor, ceiling, person**	**wall, sky, road, car**, plant, building, tree, grass, streetlight	**sky, tree, grass**, plant, wall

good precision result on the first image as well.

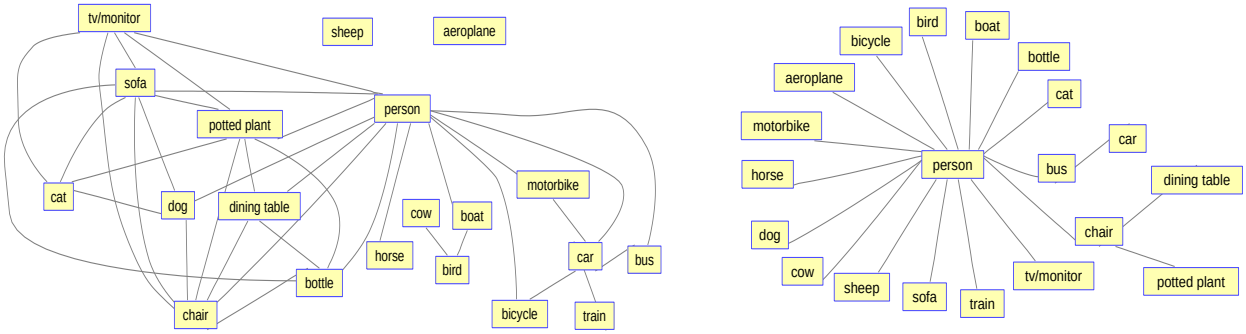
All these results suggest that by capturing the object combination patterns in newly created labels, the proposed probabilistic label enhancement model provides an effective and efficient framework for multi-label image classification.

4.3 Experiments with Dense Graphs

Our proposed approach constructs a tree-graph to identify the informative label combination patterns. In order to produce the tree structure, the maximum spanning tree algorithm needs to ignore the edges with larger (normalized co-occurrence) weights to avoid cycles. To investigate whether this is problematic, we tried an alternative version of the proposed approach by constructing a densely connected graph for label combinations, instead of restricting to singly connected trees. Specifically, we produce the dense graph by simply keeping a proportion of the existing edges (there is no edge between label pairs that never co-occur) with largest weights. In the experiments, we kept the top 30% of the edges.

We compared the two variants of the proposed model across the five datasets. In particular, Figure 5 shows the examples of the dense graph and the tree graph constructed on the Pascal07 dataset. The tree graph only has 19 edges, while the dense graph has 37 edges. We can see that the two graphs have many common edges but also capture some very different label combination patterns. There are some interesting pairs missing in the tree graph. For example, *sofa* and *tv monitor* can be observed in most sitting rooms, but their combination pair is not kept in the tree graph. On the other hand, the tree graph captured many important co-occurrence patterns with *much less* number of edges. For example, the tree graph captures the frequent co-occurrences between *person* and many other objects. By looking at the images in Pascal07, we can find many images containing *person* in different classes, which explains why the tree graph is *person*-centric. Moreover, even with much more edges, there are two isolated nodes in the dense graph, while none of nodes can be isolated in the tree graph.

The classification results of these two variants are reported in Table 4, in terms of macro-F1, micro-F1 and hamming loss. We can see that though the tree graph sacrificed edges with larger weights to maintain a singly connected tree structure, its performance is similar or even slightly better than the performance of the dense graph in most cases. In terms of the three measures, the dense graph only outperforms the tree graph on *Corel5k(s50)* and *SUN12(s50)* in terms of macro-F1, and on *Pascal07* in terms of hamming loss. With cyclic dense graphs, the max-product algorithm in the test phase can only produce approximate inference results. This can contribute to the inferior performance of the dense graph in many cases. Moreover, with more edges kept in the graph, there will be more auxiliary classifiers to train in the training phase, this makes the dense graph variant to have larger training time. All these information suggests that the tree graph is a desirable structure.



(a) dense graph

(b) tree graph

Figure 5: The densely connected graph and tree graph constructed on Pascal07.

Table 4: The average comparison results between the proposed approach (with tree graph) and its alternative version with a dense graph.

Measure	Methods	Datasets				
		Pascal07	Corel5k(s50)	Corel5k(s100)	SUN12(s50)	SUN12(s100)
Macro-F1	Tree	**0.268±0.005**	0.276±0.003	**0.167±0.004**	0.377±0.004	**0.315±0.002**
	Dense	0.250±0.005	**0.278±0.005**	0.162±0.003	**0.420±0.005**	0.298±0.002
Micro-F1	Tree	**0.579±0.004**	**0.362±0.005**	**0.333±0.003**	**0.581±0.003**	**0.514±0.002**
	Dense	0.568±0.002	0.360±0.004	0.330±0.003	0.579±0.003	0.504±0.003
Hamming Loss	Tree	0.057±0.003	**0.062±0.002**	**0.023±0.002**	**0.146±0.003**	**0.089±0.004**
	Dense	**0.052±0.001**	0.078±0.001	0.032±0.001	0.172±0.001	0.120±0.001

5 CONCLUSION

In this paper, we presented a novel probabilistic label enhancement model for multi-label image classification. The idea is to use informative label combination pairs (i.e., the object composite concepts in images) to augment the original labels which can be difficult to predict individually due to label sparsity, intra-class variations and occlusions, aiming to enhance the overall multi-label prediction performance. We formulated our model under the conditional random field framework by first constructing a tree graph in the label space based on the label co-occurrence patterns in the training data, and then performing efficient piecewise training. The learning process of the proposed model only requires training a set of independent binary classifiers, while its tree structure permits efficient and exact max-product inference in the test phase. Our experiments on several image classification datasets showed the proposed approach has superior performance in terms of both prediction quality and empirical computational complexity, comparing to the state-of-the-art comparison methods.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3:1107–1135, 2003.

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

[3] X. Cai, F. Nie, W. Cai, and H. Huang. New graph structured sparsity model for multi-label image annotations. In *Proceedings of ICCV*, 2013.

[4] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE TPAMI*, 29(3), 2007.

[5] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of SDM*, 2008.

[6] K. Dembczynski, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probability classifier chains. In *Proceedings of ICML*, 2010.

[7] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Proceedings of NIPS*, 2002.

[8] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *Proceedings of ECCV*, 2010.

[9] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of CVPR*, 2004.

[10] Z. Feng, R. Jin, and A. Jain. Large-scale image annotation by efficient and robust kernel metric learning. In *Proceedings of ICCV*, 2013.

[11] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Mach. Learn.*, 73(2):133–153, 2008.

[12] D. Grangier and S. Bengio. A discriminative kernel-based approach to rank images from text queries. *IEEE TPAMI*, 30(8):1371–1384, 2008.

[13] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceedings of CVPR*, 2009.

[14] Y. Guo and S. Gu. Multi-label classification using conditional dependency networks. In *Proceedings of IJCAI*, 2011.

[15] Y. Guo and D. Schuurmans. Adaptive large margin training for multilabel classification. In *Proceedings of AAAI*, 2011.

[16] Y. Guo and D. Schuurmans. Multi-label classification with output kernels. In *Proceedings of ECML*, 2013.

[17] Y. Guo and W. Xue. Probablistic mult-label classification with sparse feature learning. In *Proceedings of IJCAI*, 2013.

[18] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proceedings of ECCV*, 2008.

[19] S. Ji, L. Sun, R. Jin, and J. Ye. Multi-label multiple kernel learning. In *Proceedings of NIPS*, 2008.

[20] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *Proceedings of CVPR*, 2006.

[21] F. Kschischang, B. Frey, and H.A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47:498–519, 2001.

[22] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Proceedings in NIPS*, 2003.

[23] T. Mensink, J. Verbeek, and G. Csurka. Learning structured prediction models for interactive image labeling. In *Proceedings of CVPR*, 2011.

[24] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: Constraining the latent space. In *Proceedings of MM*, 2004.

[25] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[26] R.C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401, 1957.

[27] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, 5782:254–269, 2009.

[28] M. Sadeghi and A. Farhadi. Recognition using visual phrases. In *Proceedings of CVPR*, 2011.

[29] C. Sutton and A. McCallum. Piecewise training of undirected models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.

[30] H. Wang, H. Huang, and C. Ding. Image annotation using multi-label correlated green's function. In *Proceedings of CVPR*, 2009.

[31] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of CVPR*, 2010.

[32] M. Xu, Y. Li, and Z. Zhou. Multi-label learning with pro loss. In *Proceedings of AAAI*, 2013.

[33] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical dirichlet process model. In *Proceedings of MDM*, 2008.

[34] B. Yao and Fei-Fei Li. Modeling mutual context of object and human pose in human-object interaction activities. In *Proceedings of CVPR*, 2010.

[35] Z. Zha, T. Mei, J. Wang, Z. Wang, and X. Hua. Graph-based semi-supervised learning with multiple labels. *J. Vis. Comun. Image Represent.*, 20(2):97–103, February 2009.

[36] Y. Zhang and J. Schneider. Maximum margin output coding. In *Proceedings of ICML*, 2012.