Multi-Space Learning for Image Classification Using AdaBoost and Markov Random Fields

W. Zeng[a], X. Chen[b], H. Cheng[c], J. Hua[b]

[a]Department of Eletrical Engineering and Computer Science, The University of Kansas, 1520 West 15th Street, Lawrence, Kansas, USA
[b] Computer Science Department, Wayne State University, 5057 Woodward Ave., 3010, Detroit, MI, USA
[c] University of Electronic Science and Technology of China, Chengdu, China, 611731
wrzeng@ku.edu, xwchen@wayne.edu

Abstract. In various applications (e.g., automatic image tagging), image classification is typically treated as a multi-label learning problem, where each image can belong to multiple classes (labels). In this paper, we describe a novel strategy for multi-label image classification: instead of representing each image in one single feature space, we utilize a set of labeled image blocks (each with a single label) to represent an image in multi-feature spaces. This strategy is different from multi-instance multi-label learning, in which we model the relationship between image blocks and labels explicitly. Furthermore, instead of assigning labels to image blocks, we apply multi-class AdaBoost to learn a probability of a block belonging to a certain label. We then develop a Markov random field-based model to integrate the block information for final multi-label classification. To evaluate the performance, we compare the proposed method to six state-of-art multi-label algorithms on a real world data set collected on the internet. The result shows that our method outperforms other methods in several evaluation indicators, including Hamming loss, ranking-loss, macro-averaging F1, micro-averaging F1 and so on.

Keywords: Image classification, Multi-label learning, Markov random field

1 Introduction

With the rapid development of multimedia applications, the number of images in personal collection, public data sets, and web is growing. It is estimated that every minute around 3000 images are uploaded to Flickr. In 2010, the number of images Flickr hosted had exceeded five billion. The increasingly growing number of images presents significant challenges in organizing and indexing images. In addition to scene analysis [3, 31, 34, 35, 37], image retrieval [4, 19, 26], content-sensitive image filtering [6], and image representation [18], extensive attention has been drawn to automatic management of images. Automatic image tagging, for example, is a process to assign multiple keywords to a digital image. It is typically transformed as a multi-class or multi-label learning problem. In multi-class learning (MCL) [1, 2, 5, 8], an image is assigned with one and only one label from a set of predefined categories, while in multi-label learning (MLL), an image is assigned with one or more labels from a predefined label set. In this paper, we focus on MLL in real-world application.

One of the commonly-used MLL methods is called problem transformation, which transforms a multi-label learning problem into multiple binary learning problems using a strategy called binary relevance (BR) [15, 27]. A BR-based learning model typically constructs a binary classifier for each label using re-grouped data sets. While it is simple to implement, a BR-based method neglects label dependency, which is crucial in image classification. For example, an image labeled with beach may also be labeled with sea, while an image labeled with mountain is unlikely labeled with indoor. More sophisticated algorithms are advanced to address the label dependency [11, 12, 29, 31]. However, in most of the existing methods, an image with multiple labels is represented by one feature vector, while these labels are from different sub-regions in image responsible for different labels. To solve the ambiguity between image regions and labels, multi-instance learning (MIL) methods are developed in MLL methods. In multi-instance learning including multi-instance multi-label learning (MIMLL), an image is transformed into a bag of instances. The bag is positive, if at least one instance in the bag is positive and negative otherwise. MIL [30, 32] attempted to model the relationship between each sub-region in image with an associated label. To extract the sub-region, techniques from image segmentation is applied. However, image segmentation is an open problem in image processing, which will make MIL computationally expensive. Also the accuracy of segmentation interferes the performance of MIL.

In this paper, we propose a multi-space learning (MSL) method using Adaboost and Markov random field to transform MLL tasks into MCL problems. We utilize normalized real-value outputs from one-against-all multi-class AdaBoost to represent the association between a block (instead of a bag or the entire image) and a potential label. The normalized real-valued output will also represent a contribution of a block in an entire image with a label. This step will solve the ambiguity between instances and labels, since the labels in multi-class classification share no intersection in labeling examples. This will solve the ambiguity of labeling an image. Then, we use Markov Random Fields (MRF) models to integrate label sets for an image. Compared to MIMLL, MRF-based integration is a more advanced way to integrate the results from blocks rather than the hard logic provided from MIL. In image classification, that different labels describes different regions is the major reason for labeling ambiguity. In our framework, we follow the characteristics in image classification to transform MLL task into MCL problem. The key contributions of this paper are highlighted as follows:

(1) We propose an algorithm for multi-space learning which explicitly models the relationship between each image block and labels.

(2) We derive a MRF-based model to integrate the results from every block in an image. Instead of predefining values for parameters, the values of parameters in MRF and integration thresholds are obtained via training images.

The rest of the paper is organized as follows. In Section 2, we will describe the proposed method with multi-space learning and MRF-based integration. In Section 3, we will discuss our data set including all the image descriptors we have used and the final statistics. This is followed by experiment results of performance comparison. Finally, we will conclude and discuss about our future work in Section 5.

2 Methodology

The system overview is showed in Figure 1. In this framework, we convert training images and testing images into multi-space representations with overlapped blocks of fixed size. We utilize a set of single-labeled training blocks with the same fixed size to train a one-against-all multi-class AdaBoost classifier. The classifier is used to calculate the real-valued outputs related to the association between block and labels for training images and testing images. A Markov random field model is used to integrate these real-valued outputs. Via thresholds estimation from the integration results of training images, testing images are predicted label by label. We will discuss feature extraction in Section 3. In the

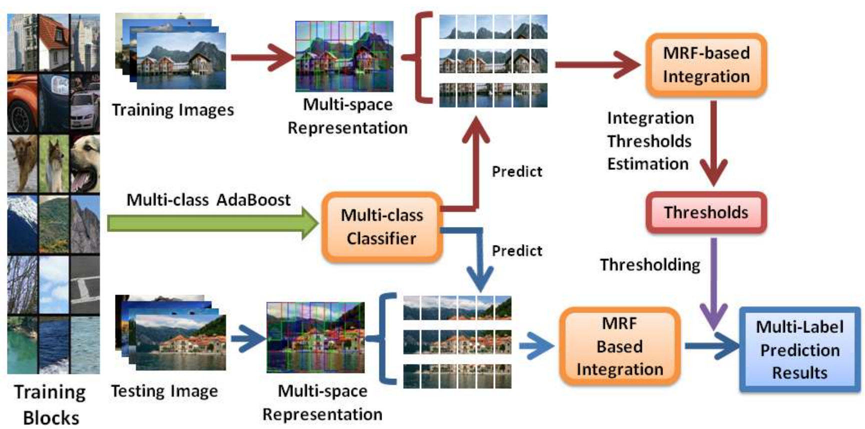

Fig. 1. The Framework of Our Algorithm

framework, we use a multi-space representation to extract blocks from every image. The blocks are fixed-sized and overlapped. From training images, we can create a set of training blocks. The set of training blocks are denoted as $\mathbf{B_{trn}}$, which contains image blocks labeled with one and only one label from a finite label set of L with q semantic labels and a newly introduced label called *background* to filter out non-object blocks. The set of training images $\mathbf{I_{trn}}$ is labeled with the same label set L and $\mathbf{I_{tst}}$ denotes the testing images. Therefore, for labeled training blocks, we have $q+1$ categories, i. e.,

$$\mathbf{B_{trn}} = \{(\mathbf{b}_1, y_{b_1}), \ldots, (\mathbf{b}_N, y_{b_N})\}, \forall \mathbf{b_i}, \mathbf{y_{b_i}} \in L^*,$$

where $L^* = \{L, \mathit{background}\}$. For training and testing images, we have the following representations as $\mathbf{I_{trn}} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}, \forall i, Y_i \subseteq L$, $\mathbf{I_{tst}} = \{T_1, \ldots, T_m\}$. In multi-space representation, we have the following definitions. Let $X_i = \{\mathbf{b}^{(i)}_{(1,1)}, \ldots, \mathbf{b}^{(i)}_{(j,k)}, \ldots, \mathbf{b}^{(i)}_{(r_i,c_i)}\}$, where $\mathbf{b}^{(i)}_{(j,k)}$ denotes the image block in j-th row and k-th column in the i-th training image, and r_i and c_i denote the row and column numbers of blocks contained in X_i. For testing images, we have the same representation as $T_i = \{\mathbf{t}^{(i)}_{(1,1)}, \ldots, \mathbf{t}^{(i)}_{(j,k)}, \ldots, \mathbf{t}^{(i)}_{(r_i,c_i)}\}$, where $\mathbf{t}^{(i)}_{(j,k)}$ denotes the image block in j-th row and k-th column in the i-th testing image, and r_i and c_i denote the row and column numbers of blocks contained in T_i. It should be noted that the extracted blocks $\mathbf{B_{trn}}$ is not necessarily contained in the union set of multi-space representations of training images, which is $\mathbf{B_{trn}} \nsubseteq \bigcup_i X_i$.

Training image is blocked via multi-space representation. The blocks in training images are fixed, once the image is given. However, training blocks in $\mathbf{B_{trn}}$ are extracted in training images at random positions where an object locates. Figure 1 shows an multi-space representation, the rectangle size is 75*100 pixels. The overlapping along the x axis is 25 pixels, and 40 pixels along the y axis. The blocks are extracted with sequence. Thus, it efficiently records the content and spatial information of an individual image. Features are extracted upon every block in the image. It should be noted that given the size of the image, the number and the location of blocks can be calculated.

In our framework, we train a multi-class classifier mapping every block in $\mathbf{B_{trn}}$ to a label in L^*. In the experiment, we use multi-class AdaBoost [38] with one-against-all strategy and one dimensional decision stumps as weak learners denoted as $h^i_l(\mathbf{b}), i = 1, \ldots, M; l \in L^*$, where i is the index of different weak learners, l is the index of different labels, and M is the iteration number of AdaBoost. Accordingly, The weight for its corresponding weak learner is denoted as $\alpha^i_l, i = 1, \ldots, M; l \in L^*$ The AdaBoost we used follows Algorithm 1 in [38]. The only difference is we record normalized real-valued outputs instead of direct labels to block $\mathbf{b}$. In the step of multi-class AdaBoost, for an assigned label l, all the image blocks in $\mathbf{B_{trn}}$ labeled with l are considered as positive examples; while remaining image blocks in $\mathbf{B_{trn}}$ are considered as negative examples. To describe the normalized real-valued outputs, we firstly introduce a boolean expression operator as $[\![\pi]\!]$ for a boolean statement π. If π is true, $[\![\pi]\!] = 1$; otherwise, $[\![\pi]\!] = 0$. Then the normalized real-valued output $f_l(\mathbf{b})$ for an image block $\mathbf{b}$ in $\mathbf{I_{trn}}$ and $\mathbf{I_{tst}}$ given an assigned label l is as follows:

$$f_l(\mathbf{b}) = \frac{\sum_{i=1}^{M} \alpha^i_l \cdot [\![h^i_l(\mathbf{b}) = l]\!]}{\sum_{k \in L^*} \sum_{i'=1}^{M} \alpha^{i'}_l \cdot [\![h^{i'}_l(\mathbf{b}) = l]\!]}. \tag{1}$$

All these normalized outputs viewed as block-label association for image blocks in $\mathbf{I_{trn}}$ are kept in parameter estimation for MRF-based integration, which will be discussed in the next part.

Algorithm 1. Estimation of 2-node Potentials

For $l = 1$ to $|L|$:
  Initialize $n_l = 0$;
  Initialize $J_{H,l}$ and $J_{V,l}$ with 10*10 zero metrices.

  For $X_i \in \mathbf{I_{trn}}$ and $l \in Y_i$
    $n_l \leftarrow n_l + 1$;
    Initialize C_{H,l,X_i} and C_{V,l,X_i} with two 10*10 zeros matrices;

    For $(\mathbf{b}_i^{(j,k)}, \mathbf{b}_i^{(j,k+1)}) \in X_i$ and $k + 1 \leqslant c_i$
      $x = Q(f_l(\mathbf{b}_{(j,k)}^{(i)}))$;
      $y = Q(f_l(\mathbf{b}_{(j,k+1)}^{(i)}))$;
      $C_{H,l,X_i}(x, y) \leftarrow C_{H,l,X_i}(x, y) + 1$.
    End
    $J_{H,l,X_i}(x, y) = \frac{C_{H,l,X_i}}{r_i \cdot (c_i - 1)}$;

    For $(\mathbf{b}_i^{(j,k)}, \mathbf{b}_i^{(j+1,k)}) \in X_i$ and $j + 1 \leqslant r_i$
      $x = Q(f_l(\mathbf{b}_{(j,k)}^{(i)}))$;
      $y = Q(f_l(\mathbf{b}_{(j+1,k)}^{(i)}))$;
      $C_{V,l,X_i}(x, y) \leftarrow C_{V,l,X_i}(x, y) + 1$.
    End
    $J_{V,l,X_i}(x, y) = \frac{C_{V,l,X_i}}{(r_i - 1) \cdot c_i}$;

    $J_{H,l} \leftarrow J_{H,l} + J_{H,l,X_i}$;
    $J_{V,l} \leftarrow J_{V,l} + J_{V,l,X_i}$;
  End
  $JH, l \leftarrow \frac{J_{H,l}}{n_l}$;
  $JV, l \leftarrow \frac{J_{V,l}}{n_l}$.
End

Outputs: $J_{H,l}$ and $J_{V,l}$.

Our method fully utilizes the normalized outputs from the multi-class AdaBoost classifier to build up Markov random field models for MLL. We derive a MRF model for information integration as follows. For an assigned label $l \in L$, our goal is to maximize the likelihood defined as $P(X_i|l)$, which is proportional to a Gibbs distribution as follows [14], $P(X_i|l) \propto e^{-U(X_i|l)}$, where $U(X_i|l)$ is called energy function. The energy function takes the following form [14],

$$U(X_i|l) = \sum_{\mathbf{b}_{(j,k)}^{(i)}} (V_l^1(\mathbf{b}_{(j,k)}^{(i)}) + \sum_{\mathbf{b}_{(j',k')}^{(i)} \in N(\mathbf{b}_{(j,k)}^{(i)})} V_l^2(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)})). \tag{2}$$

Note that $V_l^1(\mathbf{b}_{(j,k)}^{(i)})$ and $V_l^2(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)})$ are potentials for one block and two adjacent blocks in MRFs horizontally or vertically, given a label l. We introduce the following definition, $V_l^1(\mathbf{b}_{(j,k)}^{(i)}) = -f_l(\mathbf{b}_{(j,k)}^{(i)})$ where $f_l(\mathbf{b}_{(j,k)}^{(i)})$ represents the contribution of each block belonging to a certain label. When the contribution is increasing, the energy function is decreasing. Thus, we introduce the format of one-block potential as in Formula (2).

To formulate the potentials of two blocks $V_l^2(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)})$, we firstly quantize the normalized real-valued outputs from multi-class AdaBoost with function Q, where $\mathbf{b}$ denotes a image block.

$$\begin{aligned} &As \;\; 0 \leqslant f_l(\mathbf{b}) \leqslant 1, \\ &\qquad Q(f_l(\mathbf{b})) = p, \; if \;\; \frac{p-1}{10} \leqslant f_l\mathbf{b} < \frac{p}{10}, \\ &\qquad Q(f_l(\mathbf{b})) = 10, \; if \;\; f_l\mathbf{b} = 10. \end{aligned} \tag{3}$$

After quantization, given a label l, we count the different combinations of the ten levels horizontally and vertically upon a training image. Thus, we get two count matrices denoted as C_{H,l,X_i} and C_{V,l,X_i}. The two matrices are normalized by the numbers of the two-adjacent blocks horizontally and vertically. J_{H,l,X_i} and J_{V,l,X_i} denote the normalized count matrices. Finally, the averages of J_{H,l,X_i} and J_{V,l,X_i} over all positive images labeled with l is created as $J_{H,l}$ and $J_{V,l}$ called joint contribution matrices of two adjacent blocks horizontally and vertically.

After parameter estimation for potentials of two blocks, $J_{H,l}$ and $J_{V,l}$ are outputs as a codebook to extend $V_l^2(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)})$ defined as follows, $x = Q(f_l(\mathbf{b}_{(j,k)}^{(i)}))$, $y = Q(f_l(\mathbf{b}_{(j',k')}^{(i)}))$, and if the two blocks are located in horizontal direction,

$$V_l^2(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)}) = -\frac{\lambda}{|N(\mathbf{b}_{(j,k)}^{(i)})|} J_{H,l}(x,y);$$

otherwise,

$$V_l^2(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)}) = -\frac{\lambda}{|N(\mathbf{b}_{(j,k)}^{(i)})|} J_{V,l}(x,y), \tag{4}$$

where $N(\mathbf{b}_{(j,k)}^{(i)})$ denotes the neighbors of block $\mathbf{b}_{(j,k)}^{(i)}$ and λ denotes a parameter to rate the different contribution on one-block potential and two-block potentials. The integration results are calculated to predict label sets via thresholds to testing images. For the convenience of numerical calculation, we derive the following formula for normalized integration by block number n_i. As the number of blocks will lead bias to the integration results $Intg_l X_i = \frac{ln(e^{-U(X_i|l)})}{n_i} = \frac{-U(X_i|l)}{n_i}$. This formula is expanded with the following forms:$Intg_l^1(X_i) = \sum_{\mathbf{b}_{(j,k)}^{(i)} \in X_i} f_l(\mathbf{b}_{(j,k)}^{(i)})$, $Intg_l^2(X_i) = \sum_{\mathbf{b}_{(j,k)}^{(i)} \in X_i} \sum_{\mathbf{b}_{(j',k')}^{(i)} \in N(\mathbf{b}_{(j,k)}^{(i)})} \frac{\lambda \cdot J_l(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)})}{|N(\mathbf{b}_{(j,k)}^{(i)})|}$, and $Intg_l(X_i) = \frac{Intg_l^1(X_i) + Intg_l^2(X_i)}{n_i}$, where $J_l(\mathbf{b}_{(j,k)}^{(i)}, \mathbf{b}_{(j',k')}^{(i)})$ denotes joint contribution obtained by Algorithm 1. Whether to use horizontal or vertical direction depends on the locations of $\mathbf{b}^{(i)}_{(j,k)}$ and $\mathbf{b}^{(i)}_{(j',k')}$. With the integration results got from $\mathbf{I_{trn}}$, the threshold for predicting assigned label l is estimated in maximizing F1 measurement in $\mathbf{I_{trn}}$. An image will be predicted as a positive image for l, when the integration result is above the threshold. Otherwise, the image is predicted as a negative image for l. Integration results are used to predict label sets for testing images via thresholding. Since we want to evaluate the performance from ranking-based criteria in multi-label classification, we use the integration result subtracted by threshold given a label l. The thresholds are considered as zero-baselines for prediction.

3 Database

We collect 4100 images from the internet and label them with *building*, *car*, *dog*, *human*, *mountain*, and *water*, according to their contents. The resolution of all the images is controlled under 800*600.

In feature extraction, we use 13 different descriptors to represent the images. They focus on different characteristics on the images, such as color, texture, edges, contour and frequency information. They also showed different advantages to describe local details or global features in images. Table 2 describes the feature sets we have used. In the proposed algorithm, we extract features upon every fixed size block. The dimensionality of features on a block is 2684. In experiment comparison, six MLL algorithms are used. Features for entire images are extracted with the same 13 feature sets. The dimensionality for an entire image is 2629.

Table 1. Sample Numbers per Label Set

Label	Train/Test	Label	Train/Test
`b`	250/125	`b+h`	167/83
`c`	250/125	`c+h`	167/83
`d`	250/125	`d+h`	167/83
`h`	250/125	`m+w`	167/83
`m`	250/125	`b+c+h`	133/67
`w`	250/125	`b+m+w`	133/67
`b+c`	167/83	`d+h+w`	133/67

Among 4100 images, 2734 images are selected randomly for training and the remaining 1366 images are used for testing (2/3 for training and 1/3 for testing). In our algorithm, the training blocks is normalized according to sample mean and variance of every dimension. These sample means and variances are recorded to normalize the image blocks in both training and test images. The similar normalization strategy is used to normalize the data set used in multi-label classification for comparison experiment.

Label-Cardinality and Label-Density [27] are used commonly in multi-label data,$Label - Cardinality = \frac{\sum_{i=1}^{n}|Y_i|}{n}$, $Label - Density = \frac{\sum_{i=1}^{n}|Y_i|}{n \cdot q}$, where n denotes the sample size of training set, and q denotes the predefined label set size. For MLL problem in our database, Label-Cardinality in our database is 1.5973, and Label-Density is 0.2662. In Table 1, we use b, c, d, h, m, and w

Table 2. Feature Set Description

Features	**Description**	**Parameter(s) Value**
`Block-wise color moment` [24]	Mean, standard deviation and skewness of HSV	——
`RGB histogram` [23]	64-bin normalized histogram of RGB	*bin-num* = 64
`HSV histogram` [23]	64-bin normalized histogram of HSV	*bin-num* = 64
`color correlogram` [10]	Co-occurrence of pixels with a given distance and color level	*dist*=1 or 3 *color-level*=64
`edge distribution histogram` [13]	Global, local and semi-label edge distribution with five filters	*global-bin*=1 *local-bin*=25 *horizon-bin*=5 *vertical-bin*=5 *center-bin*=1
`Gabor wavelet transformation` [17]	Gabor wavelet transformation for texture	$U_h = 0.4$ $U_l = 0.05$ $K = 6$ $S = 4$
`LBP` [[20], [36]]	Local descriptor of binary patterns	Default parameter values
`LPQ` [21]	Local descriptor of phase quantization	Default parameter values
`moment invariants` [7]	Shape descriptor	——
`Tamura texture feature` [25]	Global descriptor of coarseness, contrast, and directionality	——
`Haralick Texture Feature` [9]	Please refer to [9]	——
`SIFT`[16]	Scale-invariant feature transform	*numspatialbins*=4 *numorientbins*=12
`GFD` [28]	Generic Fourier descriptor	*max-rad*=4 *max-ang*=15

for short to represents *building*, *car*, *dog*, *human*, *mountain*, and *water*. From the training images, we generate 3000 sample blocks for training for every label including background. These image blocks are used to train the AdaBoost model.

4 Experiment

In this section, we evaluate the proposed multi-space learning on the database. We compare our algorithm with six state-of-art MLL algorithms, namely AdaBoost.MH [22] which combines MCL with label ranking, back-propagation (BP) for MLL (BP-MLL) [33] which modifies the error term of traditional BP, instance differentiation (INSDIF) [35] which converts MLL into MIML upon differences between an image from different label centroids, binary relevance SVM with linear kernel (LBRSVM), multi-label kNN (ML-KNN) [34] which combines MAP principle with kNN, SVM with low-dimensional shared space named as MLLS [11].

In the experiment, we assign 100 as maximum number of iterations for BP-MLL, AdaBoost.MH, and also multi-class AdaBoost. Other parameters are obtained via 3-fold cross-validation. The criteria of optimization in cross-validation is F1-measure. We use three different aspects of criteria for evaluation, namely, example-based, ranking-based and label-based criteria, including hamming loss, one-error, coverage, ranking loss, average precision, together with micro-averaging and macro-averaging recall, precision and F1.

Let H be a learned classifier, f denote the real-valued function associated with H, and $T = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_m\}$ be the testing data set. Y_i is the true label for $\mathbf{t}_i$. The definitions for example-based and ranking-based criteria are listed as follows:

$$hloss(H) = \frac{\sum_{i=1}^{m} |H(\mathbf{t}_i) \Delta Y_i|}{mq}, \quad \mathbf{1} - err(f) = \frac{[\![argmax_{y \in L} f_y(\mathbf{t}_i) \notin Y_i]\!]}{m}, \tag{5}$$

$$cov(f) = \frac{\sum_{i=1}^{m} max_{y \in Y_i} rank_f(\mathbf{t}_i, Y_i)}{m} - 1, \tag{6}$$

$$rloss(f) = \frac{\sum_{i=1}^{m} |S_i|}{m}, \quad S_i = \{(y_1, y_2) | f_{y_i}(\mathbf{t}_i) \leqslant f_{y_2}(\mathbf{t}_i), (y_1, y_2) \in Y_i \times \bar{Y}_i\}, \tag{7}$$

$$avgprec(f) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{|S_i'|}{rank_f(\mathbf{t}_i, y)}, \tag{8}$$

$$S_i' = \{y' \in Y_i | rank_f(\mathbf{t}_i, y') \leqslant rank_f(\mathbf{t}_i, y)\}. \tag{9}$$

TP_l, FN_l, and FP_l denote true positive rate, false negative rate, and false positive rate for label l. Thus, micro-averaging and macro-averaging recall, precision, and F1 are defined in the following way.

$$micro\text{-}rec = \frac{\sum_{l=1}^{q} TP_l}{\sum_{l=1}^{q} (TP_l + FN_l)}, \quad micro\text{-}prec = \frac{\sum_{l=1}^{q} TP_l}{\sum_{l=1}^{q} (TP_l + FP_l)}, \tag{10}$$

$$macro\text{-}rec = \frac{1}{q}\sum_{i=1}^{q}\frac{TP_l}{TP_l + FN_l}, \quad macro\text{-}prec = \frac{1}{q}\sum_{i=1}^{q}\frac{TP_l}{TP_l + FP_l}, \tag{11}$$

$$micro\text{-}F1 = 2\cdot\frac{micro\text{-}rec \times micro\text{-}prec}{micro\text{-}rec + micro\text{-}prec}, \quad macro\text{-}F1 = 2\cdot\frac{macro\text{-}rec \times macro\text{-}prec}{macro\text{-}rec + macro\text{-}prec}. \tag{12}$$

Table 3 and Table 4 show the comparison results. (-) means the smaller the value is, the better the performance is; while (+) means the opposite. As can be seen, the proposed MSL algorithm outperforms the other six algorithms in several important criteria, including hamming loss, one-error, coverage, ranking loss, average precision, micro-averaging F1 and macro-averaging F1.

For multi-label classification, not only the prediction with higher accuracy is important, but also the ranking on the association between examples and labels is vital. As other than thresholding, label ranking is another popular integration strategy for prediction on multi-label data. Label-based criteria are borrowed from the field of information retrieval, which reflect classifier performance excluding the imbalance factor from learning domain. The label-based criteria are recall and precision. F1 measure shows the balance between recall and precision. F1 is highly related to two factors. The first one is the absolute value of either recall or precision. The second one is the difference between recall and precision. High-value of F1 means the values of precision and recall are both high. Multi-label classification utilizes F1 as a crucial evaluation criterion via different averaging strategies. Among them, micro-averaging and macro-averaging are two common ones. The former describes the performance based on equal power of every example, while the latter focuses on the equal power of every label to generate F1 measure.

Table 3. Performance on Hamming loss, One-error, Coverage, Ranking loss, and Average precision

	Hamming Loss	**One-Error**	**Coverage**	**Ranking Loss**	**Average Precision**
	(-)	(-)	(-)	(-)	(+)
`AdaBoost.MH`	21.18	33.82	1.49	16.46	75.98
`BP-MLL`	27.45	23.72	1.68	23.26	73.48
`INSDIF`	17.18	24.31	1.23	11.61	83.48
`LBRSVM`	18.95	26.13	1.31	12.95	81.78
`ML-KNN`	16.34	24.45	1.25	12.04	83.77
`MLLS`	24.91	22.41	1.23	11.38	84.48
`MSL`	**13.68**	**10.83**	**1.04**	**7.16**	**91.03**

In addition to the evaluation showed above in Table 3 and Table 4, we change the threshold values for prediction to draw precision-recall curves in Figure 2 and Figure 3. The micro-averaging and macro-averaging precision-recall curves basically show how sensitive a classifier would be interfered by threshold change. In summary, the larger the area under precision-recall curve (AUPRC) is, the

Table 4. Performance on micro-averaging and macro-averaging recall, precision and F1

	macro-rec (+)	**macro-prec** (+)	**macro-F1** (+)	**micro-rec** (+)	**micro-prec** (+)	**macro-F1** (+)
`AdaBoost.MH`	57.01	75.18	64.84	60.61	60.16	60.38
`BP-MLL`	77.63	48.51	59.71	78.52	49.04	60.37
`INSDIF`	60.65	**75.93**	67.43	62.21	69.96	65.86
`LBRSVM`	62.84	73.56	67.78	64.32	64.46	64.39
`ML-KNN`	61.67	72.95	66.84	62.91	72.19	67.22
`MLLS`	**90.41**	53.01	66.82	**89.61**	51.87	65.71
`MSL`	77.01	72.26	**74.56**	77.65	**72.81**	**75.15**

better the classifier is. The better here means more robust with the threshold change. Overall, MSL method yields superior performance compared to the other six MLL algorithms.

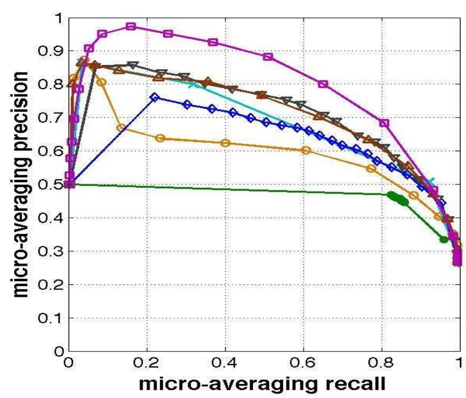

Fig. 2. micro-averaging precision-recall curve

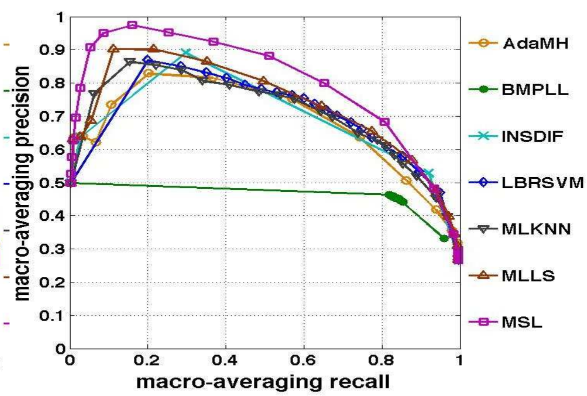

Fig. 3. macro-averaging precision-recall curve

5 Conclusion

In this paper, we present an algorithm using training blocks extracted in training images and image multi-space representations to generate a multi-space learning method, which utilizes a multi-class AdaBoost to train a multi-class classifier. In this sense, we try to transform image classification from a multi-label learning problem to a multi-class learning problem. In addition to that, rather than using a predefined logic to integrate results from regions in multi-instance learning, we derive a Markov random field model to integrate the normalized real-valued outputs from AdaBoost. MRF-based multi-space learning maintains the content and spatial information in images. Hence, MRF-based integration is a more advanced method to integrate results from different regions. Our algorithm is experimentally evaluated through a multi-label image database and proven highly effective for image classification.

6 Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 0066126.

References

1. O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR'08*, pages 1–8, June 2008.
2. A. Bosch, A. Zisserman, and X. M. noz. Image classification using random forests and ferns. In *ICCV'07*, pages 1–8, October 2007.
3. M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classfication. *Pattern Recognition*, 37:1757–1771, September 2004.
4. G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 29(3):394–410, March 2007.
5. X. Chen, X. Zeng, and D. van Alphen. Multi-class feature selection for texture classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27:1685–1691, October 2006.
6. T. Deselaers, L. Pimenidis, and H. Ney. Bag-of-visual-words models for adult image classification and filtering. In *ICPR'08*, pages 1–4, December 2008.
7. S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft identification by moment invariants. *IEEE trans. on Computers*, 26:39–46, January 1977.
8. Y. Fu and T. S. Huang. Image classification using correlation tensor analysis. *IEEE trans. on Image Processing*, 17:226–234, February 2008.
9. R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE trans. on Systems, Man and Cybernetics*, 3:610–621, November 1973.
10. J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR'97*, pages 762–768, June 1997.
11. S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *KDD'08*, pages 381–389, August 2008.
12. F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *CVPR'06*, pages 1719–1726, October 2006.
13. D. kwon Park, Y. S. Jeon, and C. S. Won. Efficient use of local edge histogram descriptor. In *Proceedings of the 2000 ACM workshops on Multimedia*, pages 51–54, December 2000.
14. S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag New York, Inc., Secaucus, New Jersey, 2001.
15. X. Lin and X. Chen. Mr. knn: Soft relevance for multi-label classification. In *CIKM'10*, pages 349–358, October 2010.
16. D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV'99*, pages 1150–1157, September 1999.
17. B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18:837–842, August 1996.
18. E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV'06*, pages 490–503, May 2006.

19. J. F. Nunes, P. M. Moreira, J. M. R. S. T. E. Nowak, F. Jurie, and B. Triggs. Shape based image retrieval and classification. In *Iberian Conference on Information Systems and Technologies*, pages 1–6, August 2010.
20. T. Ojala, M. Pietikäinen, and T. Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In *ICAPR'01*, pages 397–406, March 2001.
21. V. Ojansivu, E. Rahtu, and J. Heikkilä. Rotation invariant local phase quantization for blur insensitive texture analysis. In *ICPR'08*, pages 1–4, December 2008.
22. R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, May 2000.
23. L. G. Shapiro and G. Stockman. *Computer Vision*. Prentice Hall, Inc., Upper Saddle River, New Jersey, 2001.
24. M. Stricker and M. Orengo. Similarity of color images. In *Proceedings of Storage and Retrieval for Image and Video Databases*, pages 381–392, February 1995.
25. H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Trans. on Systems, Man and Cybernetics*, 8:460–473, June 1978.
26. B. Tomasik, P. Thiha, and D. Turnbull. Tagging products using image classification. In *SIGIR'09*, pages 792–793, July 2009.
27. G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, 2010.
28. A. Vijay and mahua Bhattacharya. Content-based medical image retrieval using the generic fourier descriptor with brightness. In *ICMV'09*, pages 330–332, December 2009.
29. H. Wang, H. Huang, and C. Ding. Image annotation using multi-label correlated green's function. In *ICCV'09*, pages 2029–2034, October 2009.
30. C. Yang, M. Dong, and J. Hua. Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In *CVPR'06*, pages 2057–2063, June 2006.
31. Z. Younes, F. Abdallah, and T. Denceux. Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies. In *European Signal Processing Conference*, August 2008.
32. Z. Zha, X. Hua, T. Mei, J. Wang, G. Qi, and Z. Wang. Joint multi-label multi-instance learning for image classification. In *CVPR'08*, pages 1–8, June 2008.
33. M. Zhang and Z. Zhou. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Trans. on Knowledge and Data Engineering*, 18:1338–1351, 2006.
34. M. Zhang and Z. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40:2038–2048, July 2007.
35. M. Zhang and Z. Zhou. Multi-label learning by instance differentiation. In *AAAI'07*, pages 669–674, July 2007.
36. G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):915–928, June 2007.
37. Z. Zhou and M. Zhang. Multi-instance multi-label learning with application to scene classification. In *NIPS'06*, pages 1609–1616, December 2006.
38. J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class adaboost. *Statistics and Its Interface*, 2:349–460, 2009.