Multi-label Feature Transform for Image Classifications

Hua Wang, Heng Huang, and Chris Ding

Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, USA
huawang2007@mavs.uta.edu, heng@uta.edu, chqding@uta.edu

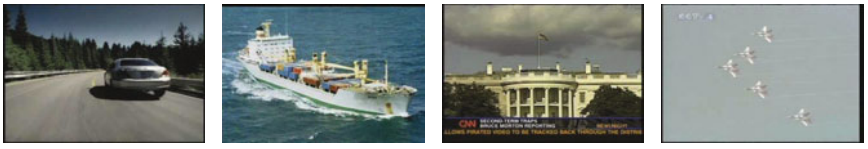
Abstract. Image and video annotations are challenging but important tasks to understand digital multimedia contents in computer vision, which by nature is a *multi-label multi-class* classification problem because every image is usually associated with more than one semantic keyword. As a result, label assignments are no longer confined to class membership indications as in traditional single-label multi-class classification, which also convey important characteristic information to assess object similarity from knowledge perspective. Therefore, besides implicitly making use of label assignments to formulate label correlations as in many existing multi-label classification algorithms, we propose a novel Multi-Label Feature Transform (MLFT) approach to also explicitly use them as part of data features. Through two transformations on attributes and label assignments respectively, MLFT approach uses kernel to implicitly construct a *label-augmented feature vector* to integrate attributes and labels of a data set in a balanced manner, such that the data discriminability is enhanced because of taking advantage of the information from both data and label perspectives. Promising experimental results on four standard multi-label data sets from image annotation and other applications demonstrate the effectiveness of our approach.

Keywords: Multi-label classification, Feature transformation, Image annotation.

1 Introduction

Automatically annotating image and video is a key task to understand digital multimedia contents for browsing, searching, and navigation. In the real world, an image or a video clip is usually attached with several different semantic keywords, *e.g.*, all the images in Fig. 1 are annotated with more than one keyword. This poses so-called *multi-label multi-class* classification problems, which refer to problems where each object can be assigned to multiple classes. Multi-label problems are more general than traditional *single-label* (multi-class) problems, in which each object is assigned to exactly one class. Driven by its broad applications in diverse domains, such as image/video annotation, gene function annotation, and text categorization, *etc.*, multi-label classification is receiving increasing attentions in recent years.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 793–806, 2010.



(a) car, road. (b) sea, ship. (c) building, sky. (d) sky, plane.

Fig. 1. Sample images from TRECVID 2005 dataset. Each image is annotated with multiple semantic keywords.

An important difference between single-label classification and multi-label classification is that the classes in single-label classification are assumed to be mutually exclusive while those in multi-label classification are normally interdependent from one another. For example, "sea" and "ship" tend to appear in a same image, while "fire" typically does not appear together with "ice". Thus, many multi-label classification algorithms have been developed to make use of label correlations to improve the overall multi-label classification performance. Two ways are popularly used to employ label correlations: incorporating label correlations into the existing label propagation learning algorithms, as either part of graph weight [10,2] or an additional constraint [17,15]; or utilizing label correlations to seek a more discriminative subspace [16,18,8]. Besides, there also exist many other methods based on different mechanisms, such as matrix factorization[12], maximizing label entropy [19], Bayesian model [6], and so on.

Our perspectives and motivations. One common aspect of many existing multi-label classification algorithms is that they all attempt to leverage correlations among classes, which are typically computed from label assignments on training data. Although this paradigm of implicitly using label assignments has shown their strength, it would be also favorable to explicitly use them as part of data attributes for classification [5]. In multi-label classification, multiple labels may be associated with a single object, hence the number of common labels shared by two different objects is no longer restricted to be one. Label assignments thus turn out a similarity measurement among data objects.

Therefore, in this work, we propose a novel Multi-Label Feature Transform (MLFT) approach to use label assignments as both part of data attributes (explicit usage) and label correlations (implicit usage) for enhanced multi-label classification performance. We first transform the original data attributes (via proposed Multi-label Kernel Laplacian Embedding (MLKLE) method) and corresponding label assignments (via proposed Correlative Kernel Transform (CKT) method) from their native spaces to two new (sub)spaces with similar dimensionality. Then the transformed feature vectors from these two types of data are integrated together to form a new *label augmented feature vector* in the spanned hyperspace. Because the dimensionalities of the two transformed feature vectors are close, the vector concatenation thereby space spanning is balanced. Most importantly, the label-augmented feature vector not only preserves original attributes information, but also captures label correlations through label

assignments implicitly and explicitly. As a result, data points in the spanned feature space are more discriminable, by which succeeding classification can be conducted more effectively. As an additional advantage, MLFT averts possible difficulties to directly employ state-of-the-art single-label classification methods to solve multi-label problems, such that their powerful classification capabilities can be utilized on multi-label data.

Although originated from simple vector concatenation, the proposed MLFT does not need to explicitly construct label-augmented feature vector due to introducing kernel. Moreover, using a more discriminative kernel, label-augmented feature vectors are mapped to a more linearly separable high-dimensional space such that classifications can be carried out much easier.

2 Multi-label Feature Transform

For a classification task with n data points and K classes, each data point $\mathbf{x}_i \in \mathbb{R}^p$ is associated with a subset of class labels represented by a binary vector $\mathbf{y}_i \in \{0,1\}^K$ such that $\mathbf{y}_i(k) = 1$ if $\mathbf{x}_i$ belongs to the kth class, and 0 otherwise. Meanwhile, we also have the pairwise similarities $W \in \mathbb{R}^{n \times n}$ among the n data points with W_{ij} indicating how closely $\mathbf{x}_i$ and $\mathbf{x}_j$ are related. W may be computed from attributes data or directly obtained from experimental observations. Suppose the number of labeled data points is $l(< n)$, our goal is to predict labels $\{\mathbf{y}_i\}_{i=l+1}^n$ for the unlabeled data points $\{\mathbf{x}_i\}_{i=l+1}^n$. We write $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$.

2.1 Outlines of MLFT Approach

In multi-label classification, each data point may be assigned to multiple classes at the same time, *i.e.*, $\sum_k \mathbf{y}_i(k) \geq 1$. Thus, the overlap of labels assigned to two data points computed by $\mathbf{y}_i^T \mathbf{y}_j$ is an integer ranged from 0 to K, which induces an affinity relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. This is different from the single-label case, where the number of overlapped labels of two data points is restricted to be either 0 or 1. Namely, $\mathbf{y}_i$ not only indicates the class membership for a data point, but also contains important attribute information to assess the similarity between data points. Therefore, we propose MLFT approach to construct a new label-augmented feature vector $\mathbf{z}_i \in \mathbb{R}^r$ to integrate the characteristic information conveyed by both $\mathbf{x}_i$ and $\mathbf{y}_i$, such that $\mathbf{z}_i$ are more separable to achieve enhanced classification performance.

A naive construction of $\mathbf{z}_i$ can be the simple concatenation of $\mathbf{x}_i$ and $\mathbf{y}_i$, which, however, suffers from two critical problems. Firstly, the original features $\mathbf{x}_i$ are often compromised by noise during the generation process, and thereby might not be very discriminable. Secondly, and more importantly, in many data sets from real computer vision applications, such as image annotation, the number of features of a data set is usually much greater than that of classes, *i.e.*, $p \gg K$. That is, directly concatenating $\mathbf{x}_i$ and $\mathbf{y}_i$ causes unbalance problem. Therefore, we need to transform $\mathbf{x}_i$ and $\mathbf{y}_i$ into two (sub)spaces with close dimensionalities, and also eliminate the irrelevant features.

Let $U \in \mathbb{R}^{p\times(K-1)}$ be a linear transformation for $\mathbf{x}_i$ (*i.e.*, the dimensionality of feature space is reduced to the number of class minus 1, $K-1 \ll p$), the input feature vector $\mathbf{q}_i^x \in \mathbb{R}^{K-1}$ is computed by $\mathbf{q}_i^x = U^T\mathbf{x}_i$. Similarly, we denote $\mathbf{p}_i^y \in \mathbb{R}^K$ as the input label vector, which is transformed from $\mathbf{y}_i$ and computed by a kernel function $\mathbf{p}_i^y = \phi(\mathbf{y}_i)$. Thus, the dimensionality of $\mathbf{z}_i$ is $r = 2K-1$. U is computed by Multi-label Kernel Laplacian Embedding (MLKLE) as detailed in Section 3, and $\phi(\cdot)$ is obtained by Correlative Kernel Transform (CKT) as introduced Section 4. The label-augmented feature vector $\mathbf{z}_i$ is thus constructed as follows:

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{q}_i^x \\ \mathbf{p}_i^y \end{bmatrix} \quad . \tag{1}$$

2.2 Implicit Construction of Label-Augmented Feature Vector

Once the label-augmented feature vector $\mathbf{z}_i$ is computed, any traditional single-label classification method can be used to carry out classification. In this work, we use support vector machine (SVM) because of its elegant theoretical foundation and powerful classification capability. A special benefit of using SVM with kernel is that the construction of the label-augmented feature vector $\mathbf{z}_i$ can be conveniently interpreted from kernel perspective. To be more specific, let $\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j)$ be a radial basis function (RBF) kernel on $\mathbf{z}_i$:

$$\begin{aligned} \mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j) &= \exp\left(-\gamma\|\mathbf{z}_i - \mathbf{z}_j\|^2\right) = \exp\left(-\gamma\|\mathbf{q}_i^x - \mathbf{q}_j^x\|^2\right)\exp\left(-\gamma\|\mathbf{p}_i^y - \mathbf{p}_j^y\|^2\right) \\ &= \mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y), \end{aligned} \tag{2}$$

where $\mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)$ and $\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y)$ are the kernels with respect to input feature vectors and input label vectors respectively. Therefore, the construction of $\mathbf{z}_i$ in Eq. (1) indeed can be seen as a multiplicative kernel, which comprises information from both data attributes and label assignments. Similarly, $\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j)$ can be seen as an additive kernel when linear kernel is used, and the same for other popular kernels used in SVM.

Therefore, instead of explicitly transforming $\mathbf{y}_i$, we focus on devising a discriminative kernel as described in Section 4. By introducing kernel, explicit construction of label-augmented feature vector $\mathbf{z}_i$ in Eq. (1) is no longer needed. Instead, we may use a more delicate kernel to incorporate additional useful information for better classification.

We outline the classification procedures by the proposed MLFT approach in Table 1 and will describe the details of each step in the rest of this paper. As can be seen, we concentrate on the data preparation phase, because it is the most essential part to boost classification performance.

3 Multi-label Kernel Laplacian Embedding

As analyzed in Section 2.1, two problems, indiscriminability of $\mathbf{x}_i$ and unbalanced cardinalities of attributes space and label space, prevent us from using direct concatenation of $\mathbf{x}_i$ and $\mathbf{y}_i$. Therefore, we first solve these two difficulties for $\mathbf{x}_i$

Table 1. Classification using the proposed MLFT approach

Data preparation:
(a) Initialize unlabeled data points $\{\mathbf{x}_i\}_{i=l+1}^n$ to get the initial labels $\{\hat{\mathbf{y}}_i\}_{i=l+1}^n$. (Section 5)
(b) Compute the linear transformation U using proposed MLKLE method from $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^l$ and $\{(\mathbf{x}_i, \hat{\mathbf{y}}_i)\}_{i=l+1}^n$. Project all the data points including those unlabeled into the embedding subspace to obtain $\{\mathbf{q}_i^x\}_{i=1}^n$ as in Eq. (5). (Section 3)
(c) Compute kernel matrix $\mathcal{K}_z$ on label-augmented feature vectors $\{\mathbf{z}_i\}_{i=1}^n$ as in Eq. (15) by $\{\mathbf{q}_i^x\}_{i=1}^n$ and implicit transform of $\{\mathbf{y}_i\}_{i=1}^l$ using proposed CKT method. (Section 4)
Training:
Training K SVM classifiers, one for each class.
Testing:
Classify $\{\mathbf{z}_i\}_{i=l+1}^n$ using the trained classifiers to obtain the predicted labels $\{\mathbf{y}_i\}_{i=l+1}^n$, one class at a time.

and propose a Multi-label Kernel Laplacian Embedding (MLKLE) method to reduce the dimensionality of $\mathbf{x}_i$ and seek its intrinsic structure with irrelevant patterns pruned. This produces a transformation $U \in \mathbb{R}^{p\times(K-1)}$ by maximizing the following optimization objective:

$$J = \mathbf{tr}\left(\frac{U^T X \mathcal{K} X^T U}{U^T X (D-W) X^T U}\right), \tag{3}$$

where $\mathcal{K}$ is a $n \times n$ kernel matrix, and $D = \mathrm{diag}\,(d_1, \ldots, d_n)$, $d_i = \sum_j W_{ij}$. Thus, $L = D - W$ is the graph Laplacian [3]. $\mathcal{K}$ and W are defined later in Eq. (11) and Eq. (13) respectively. The solution to this problem is well established in mathematics by resolving the following generalized eigenvalue problem:

$$X \mathcal{K} X^T \mathbf{u}_k = \lambda_k X (D-W) X^T \mathbf{u}_k, \tag{4}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$ are the result eigenvalues and $\mathbf{u}_k$ are the corresponding eigenvectors. Hence, $U = [\mathbf{u}_1, \ldots, \mathbf{u}_{K-1}]$.

Using U, $\mathbf{x}_i$ can be projected into the embedding space by

$$\mathbf{q}_i^x = U^T \mathbf{x}_i \ . \tag{5}$$

Because many modern data sets in real life come in from multiple data sources, we often have both *attributes* information about data objects (vector data X) and various *pairwise similarities* between data objects (graph data W) for a same data set at the same time. Since these two types of data both convey valuable information for class membership inference, many existing embedding algorithms designed for only one type of input data are usually insufficient, especially the correlations between these two types of data are not exploited. In contrast, the true power of the proposed MLKLE in Eq. (3) lies in that it integrates both *attributes* and *pairwise relationships* in an integral optimization objective, such

that the discriminability of projected data in the shared low-dimensional subspace is doubly reinforced due to taking advantage of the information from the both data sources. In the rest of this section, we derive the optimization objective in Eq. (3).

3.1 PCA Laplacian Embedding

For attribute data X, Principle Component Analysis (PCA) [9] maximizes the data variance in the embedding space to retain the most information. Let $Q^T = [\mathbf{q}_1^x, \ldots, \mathbf{q}_n^x] \in \mathbb{R}^{(K-1)\times n}$ be the projected feature matrix, PCA aims to find the projection by maximizing

$$J_{\text{PCA}} = \mathbf{tr}\left(Q^T X^T X Q\right) \quad . \tag{6}$$

For pairwise relationships W, Laplacian embedding preserves the same relationships and maximizes the smoothness with respect to the intrinsic manifold of the data set by minimizing [7]

$$J_{\text{Lap}} = \mathbf{tr}\left(Q^T (D - W) Q\right) \quad . \tag{7}$$

Combining Eq. (6) and Eq. (7), we can construct an additive objective as:

$$J = \alpha J_{\text{Lap}} - (1 - \alpha) J_{\text{PCA}}, \tag{8}$$

where $0 < \alpha < 1$ is a tradeoff parameter. In practice, however, optimal α is hard to choose. Thus, instead of using a trace *difference* as in Eq. (8), we formulate the objective as a trace *quotient* so that α is removed:

$$J_{\text{PCA-Lap}} = \mathbf{tr}\left(\frac{Q^T X^T X Q}{Q^T (D - W) Q}\right) \quad . \tag{9}$$

We call Eq. (9) as PCA Laplacian embedding, which integrates both attributes and pairwise relationship in a same embedding space.

3.2 Kernel Laplacian Linear Embedding (KLE)

PCA Laplacian embedding in Eq. (9) is a purely unsupervised embedding, while label information, though useful, is not leveraged. We notice that $X^T X$ in Eq. (9) is a linear kernel, which could be replaced by a more discriminative kernel $\mathcal{K}$. Besides the supervision information from training data, label correlations can also be incorporated via $\mathcal{K}$, therefore we defer the definition of $\mathcal{K}$ now and will give its detailed implementation later by Eq. (11) in Section 3.3. Replacing $X^T X$ by $\mathcal{K}$, we have a new objective as:

$$\max_{Q} \mathbf{tr}\left(\frac{Q^T \mathcal{K} Q}{Q^T (D - W) Q}\right) \quad . \tag{10}$$

Using linear embedding $Q^T = U^T X$, the optimization objective in Eq. (3) is derived. We call it as Kernel Laplacian Embedding (KLE).

3.3 Label Correlation Enhanced Kernel Laplacian Embedding

Because label correlations are important information contained exclusively in multi-label data and useful for multi-label classification, it would be beneficial to take advantage them in KLE. Equipped with the kernel matrix $\mathcal{K}$ and graph Laplacian $L = D - W$ in Eq. (3), we can easily incorporate label correlations into the optimization objective. We first denote $C \in \mathbb{R}^{K \times K}$ as the label correlation matrix, which will be defined later by Eq. (17) in Section 6. C is a symmetric matrix and its entry C_{kl} captures the correlation between the kth and lth classes.

Correlation Enhanced Kernel. Instead of using the simplest linear kernel XX^T in Eq. (9), we may use a multiplicative kernel to carry more information:

$$\mathcal{K}_{ij} = \mathcal{K}_x(\mathbf{x}_i, \mathbf{x}_j)\mathcal{K}_y(\mathbf{y}_i, \mathbf{y}_j), \tag{11}$$

where $\mathcal{K}_x$ is a kernel with respect to data attributes X, and $\mathcal{K}_y$ is a kernel with respect to label assignments Y.

Same as most existing related works, $\mathcal{K}_x$ is constructed using the Gaussian kernel $\mathcal{K}_x(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma\right)$, where σ is fine tuned by 5-fold cross validation in our evaluations.

$\mathcal{K}_y$ encodes label correlations via assessing the label similarities between data points. The simplest way to measure the label overlap of two data points computes $\mathbf{y}_i^T \mathbf{y}_j$. The bigger the overlap is, the more similar the two data points are. The problem of this straightforward similarity is that it treats all the classes independent and can not exploit the correlations among them. In particular, it will give zero similarity whenever two data points do not share any labels. However, data points with no common label can still be strongly related if their attached labels are highly correlated. Therefore, instead of computing the label similarity by the dot product, we compute it by $\mathbf{y}_i^T \, C \, \mathbf{y}_j$ with normalization:

$$\mathcal{K}_y(\mathbf{y}_i, \mathbf{y}_j) = \frac{\mathbf{y}_i^T \, C \, \mathbf{y}_j}{\|\mathbf{y}_i\| \, \|\mathbf{y}_j\|} \ . \tag{12}$$

We call $\mathcal{K}$ as the *correlation enhanced kernel*. In order to compute $\mathcal{K}_y$, we need to initialize the unlabeled data point first, which will be detailed later in Section 5.

Correlation Enhanced Pairwise Similarities. Many existing embedding algorithms construct the pairwise similarity matrix (W) only from data attributes (X), while label correlations are overlooked, which, however, is useful in multi-label classification. Therefore, we propose a *Correlation Enhanced Pairwise Similarity* scheme to make use of them as follows:

$$W = W_{\mathrm{X}} + \beta W_{\mathrm{L}}, \tag{13}$$

where W_{X} can be constructed from attribute data or obtained directly from experimental observations, and W_{L} is the label similarity matrix. β is a parameter determining how much the pairwise relationship should be biased by label similarities, and empirically selected as $\beta = \sum_{i,j,i \neq j} W_{\mathrm{X}}(i,j) / \sum_{i,j,i \neq j} W_{\mathrm{L}}(i,j)$.

As $\mathcal{K}_x$ and $\mathcal{K}_y$ defined above readily assess the similarities between data points from data and knowledge perspectives respectively, we define:

$$W_{\mathrm{X}}(i,j) = \begin{cases} \mathcal{K}_x(i,j) & i \neq j, \\ 0 & i = j; \end{cases} \qquad W_{\mathrm{L}}(i,j) = \begin{cases} \mathcal{K}_y(i,j) & i \neq j, \\ 0 & i = j \ . \end{cases} \tag{14}$$

Finally, when $\mathcal{K}$ and W are defined as in Eq. (11) and Eq. (13), we call Eq. (3) as the proposed Multi-Label Kernel Laplacian Embedding (MLKLE) method, which is summarized in Table 2.

Table 2. Algorithm of MLKLE

Input:
$X \in \mathbb{R}^{p \times n}$: centralized feature matrix
$Y \in \mathbb{R}^{K \times n}$: label matrix

Steps:
1 Construct $\mathcal{K}$ as in Eq. (11) and W as in Eq. (13).
2 Compute $X\mathcal{K}X^T$ and $X(D-W)X^T$.
3 Resolve the generalized eigenvalue problem as in Eq. (4). Construct the projection matrix U by the eigenvectors corresponding to the $(K-1)$ leading eigenvalues.

Output:
$U \in \mathbb{R}^{p \times (K-1)}$: the projection matrix to project data from original feature space $\mathbb{R}^p$ to the embedding space $\mathbb{R}^{(K-1)}$.

4 Correlative Kernel Transformation

Because the construction of label-augmented feature vector $\mathbf{z}_i$ can be seen as a multiplicative kernel $\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j)$ computed individually from $\mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)$ and $\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y)$ when using SVM for classification, we do not need to explicitly define the kernel function $\phi(\cdot)$ any longer. Therefore, instead of using the simple vector concatenation as in Eq. (1), we may devise a more discriminative kernel, which also provides another opportunity to incorporate more useful information for improved classification accuracy, such as the label correlations of a multi-label data set. Thus, in this work, we use a multiplicative kernel as:

$$\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j) = \mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y), \tag{15}$$

where $\mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x) = \mathbf{q}_i^{x\,T}\mathbf{q}_j^x$ is a linear kernel on transformed input feature vectors and let

$$\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y) = \mathcal{K}_y \ . \tag{16}$$

We call the implicit transformation from $\mathbf{y}_i$ to $\mathbf{p}_i^y$ using $\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y)$ defined in Eq. (16) as Correlative Kernel Transformation (CKT).

5 Initialization of Unlabeled Data

The ultimate goal of our algorithm is to predict labels $\mathbf{y}$ for unlabeled data points $\mathbf{x}_i$. Using our classification framework, we first need to initialize them to compute label-augmented feature vector $\mathbf{z}_i$ or the hybrid kernel $\mathcal{K}_z$. We can use any classification method to get the initialized labels $\hat{\mathbf{y}}_i$ for unlabeled data points. Although the initializations are not completely correct, a big portion of them are (assumed to be) correctly predicted. Our classification framework will self-consistently amend the incorrect labels. In this work, we use K-nearest neighbor (KNN) method for initialization because of its simplicity and clear intuition ($K = 1$ is used in this work and we abbreviate it as 1NN).

Another important point of the initialization step lies in that it provides an opportunity to make use of existing multi-label classification algorithms, *i.e.*, through the initialization step, the proposed MLFT approach can naturally benefit from the advantages of previous related works.

6 Motivation and Formulation of Label Correlations

Label correlations play a significant role in multi-label classification tasks, which are routinely utilized in most, if not all, existing multi-label classification algorithms as a primary mechanism to improve the overall classification performance. In this work, we also attempt to use them from the following three perspectives: attribute kernel as in Eq. (11), pairwise similarity as in Eq. (13) and label augmentation as in Eq. (15).

As the number of shared data points belonging to two classes measures how closely they are related, we use the cosine similarity to quantify label correlations. Let $Y = [\mathbf{y}_{(1)}, \ldots, \mathbf{y}_{(K)}]$, $\mathbf{y}_{(k)} \in \mathbb{R}^n (1 \leq k \leq K)$ is an n-vector, which is a class-wise label indicator vector for the kth class. Note that $\mathbf{y}_{(k)}$ is different from the data-point-wise label indicator vectors $\mathbf{y}_i$, which is a K-vector. We define the label correlation matrix, $C \in \mathbb{R}^{K \times K}$, to characterize label correlations as following:

$$C(k,l) = \cos(\mathbf{y}_{(k)}, \mathbf{y}_{(l)}) = \frac{\langle \mathbf{y}_{(k)}, \mathbf{y}_{(l)} \rangle}{\|\mathbf{y}_{(k)}\| \, \|\mathbf{y}_{(l)}\|} \ . \tag{17}$$

Using the TRECVID 2005 data set[1] with LSCOM-Lite annotation scheme [13], the label correlations defined in Eq. (17) is illustrated in Fig. 2. The high correlation value between "person" and "face" depicted in Fig. 2 shows that they are highly correlated, which perfectly agree with the common sense in real life due to the simplest fact that everybody has a face. Similar observations can be also found for "outdoor" and "sky", "waterscape-waterfront" and "boat-ship", "studio" and "TV-computer scree", "road" and "car", *etc.*, which concretely confirm the correctness of the formulation of label correlations defined in Eq. (17) at semantic level.

[1] `http://www-nlpir.nist.gov/projects/trecvid/`

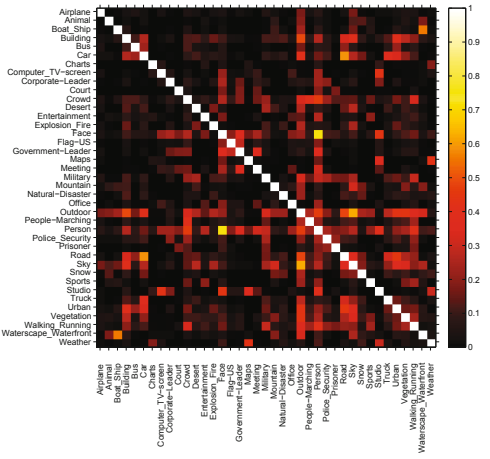


Fig. 2. Pairwise label correlations of 39 keywords in LSCOM-Lite on TRECVID 2005.

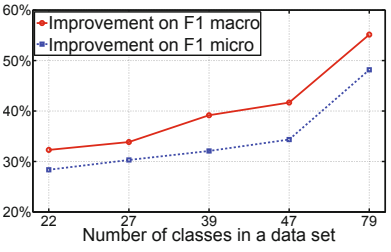


Fig. 3. Improvements (%) on F1 macro/micro average by MLFT algorithm over SVM using original features grows with the number of classes of multi-label data sets.

7 Experimental Evaluations

We evaluate the proposed MLFT algorithm using the following three standard multi-label image data sets.

TRECVID 2005 data set contains 137 broadcast news videos, which are segmented into 61901 sub-shots and labeled with 39 concepts according to LSCOM-Lite annotations [13]. We randomly sample the data set such that each concept (label) has at least 100 video key frames. We use 384-dimensional block-wise (64) color moments (mean and variable of each color band) as features.

Mediamill data set [14] includes 43907 sub-shots labeled with 101 classes (lexicon concepts), where each image are characterized by a 120-dimensional vector. Eliminating the classes containing less than 1000 data points, we have 27 classes in experiments. We randomly pick up 2609 sub-shots such that each class has at least 100 data points.

Corel natural scene data set [1] contains 2407 images represented by a 294-dimensional vector, which are labeled with 6 semantic concepts (labels).

Besides image annotation, we also extend our evaluation of the proposed algorithm to one more application in bioinformatics and use the following broadly used data set.

Yeast data set [4] is formed by micro-array expression data and phylogenetic profiles with 2417 genes. Each gene is expressed as a 107-dimensional vector, which is associated with at most 190 biological functions (labels) simultaneously. Filtering out the minor classes with small number of labeled genes, we end up with 14 labels.

Obviously, the number of features of every data set is much greater than the corresponding number of labels.

7.1 Evaluation Metrics for Multi-label Classification

The conventional classification performance metrics in statistical learning, *precision* and *F1 score*, are used to evaluate the proposed algorithms. For every class, the precision $p^{(k)}$ and F1 score $F_1^{(k)}$ for the kth class are computed following the standard definition for a binary classification problem. To address the multi-label scenario, as recommended in [11], macro average and micro average are used to assess the overall performance across multiple labels.

7.2 Multi-label Classification Performance

We use standard 5-fold cross validation to evaluate the classification performance of the proposed MLFT algorithm, and compare the experimental results with the most recent multi-label classification methods. We choose two label propagation based approaches: (1) Multi-Label Gaussian harmonic Function (MLGF) [17] method and (2) Semi-supervised learning by Sylvester Equation (SMSE) [2]; and two dimensionality reduction based approaches: (3) multi-label dimensionality reduction via Dependence Maximization (MDDM) [18] method and (4) Multi-Label Least Square (MLLS) [8] method. We use LIBSVM[2] to implement SVM throughout this paper, including the final classification step in the proposed MLFT approach. For MLGF and SMSE methods, we follow the detailed algorithms described in [17,2]. For MDDM, we use KNN for classification after dimensionality reduction, where 1NN classifiers are run one class at a time. We tried different 1NN, 3NN, 5NN, and the results are similar. Due to the limited space, we only report 1NN results. For MLLS, we use the codes posted on the authors' website [8].

We also run SVM directly using the original features of data sets, and report the classification results as baseline. The classification is conducted one class at a time, where every class is treated as a binary classification problem.

As mentioned in Section 2, instead of using MLKLE and CKF to obtain the balanced input vector $\mathbf{q}_i^x$ and $\mathbf{p}_i^y$ to construct label-augmented feature vector $\mathbf{z}_i$ using Eq. (1), we can also simply concatenate the original feature vectors $\mathbf{x}_i$ and label vector $\mathbf{y}_i$, we call this method as Naive Multi-label Feature Transform (NMLFT) method and report its results. In order to alleviate the unbalanced problem, we replace $\mathbf{p}_i^y$ by $\mu\mathbf{y}_i$, and empirically select $\mu = \sqrt{\frac{\sum_{i,j,i\neq j}\|\mathbf{y}_i-\mathbf{y}_j\|^2}{\sum_{i,j,i\neq j}\|\mathbf{x}_i-\mathbf{x}_j\|^2}}$ as SVM computes the decision hyperplane using Euclidean distance.

Table 3 presents the classification performance comparisons of the seven compared methods by 5-fold cross validation on the four multi-label data sets, which show that the proposed MLFT constantly outperforms the other methods. This demonstrates that the mapping of training data points from the original feature space into the transformed label augmented feature space through the proposed MLFT algorithm is generalizable to the test data, and the classification performance is hence improved for multi-label classification tasks. In addition, the

[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Table 3. Performance evaluation of the seven compared methods on the four multi-label data sets by 5-fold cross validation

Datasets	Evaluation metrics		Compared methods						
			SVM	MLGF	SMSE	MDDM	MLLS	NMLFT	MLFT
TRECVID 2005 (39 classes)	Macro avg.	Precision	0.269	0.108	0.107	0.366	0.272	0.243	0.421
		F1 score	0.236	0.151	0.150	0.370	0.275	0.330	0.398
	Micro avg.	Precision	0.252	0.107	0.107	0.352	0.279	0.339	0.420
		F1 score	0.371	0.167	0.165	0.491	0.375	0.483	0.527
Mediamill (27 classes)	Macro avg.	Precision	0.301	0.204	0.205	0.385	0.307	0.376	0.395
		F1 score	0.302	0.206	0.213	0.389	0.314	0.380	0.431
	Micro avg.	Precision	0.297	0.201	0.199	0.382	0.304	0.369	0.388
		F1 score	0.459	0.304	0.301	0.541	0.470	0.518	0.572
Yeast (14 classes)	Macro avg.	Precision	0.657	0.564	0.670	0.794	0.689	0.747	0.824
		F1 score	0.114	0.107	0.109	0.147	0.129	0.132	0.154
	Micro avg.	Precision	0.826	0.652	0.675	0.854	0.827	0.828	0.885
		F1 score	0.139	0.124	0.128	0.160	0.147	0.149	0.168
Corel natural scene (6 classes)	Macro avg.	Precision	0.582	0.341	0.362	0.661	0.588	0.642	0.691
		F1 score	0.542	0.410	0.415	0.667	0.545	0.651	0.687
	Micro avg.	Precision	0.591	0.404	0.410	0.669	0.593	0.650	0.693
		F1 score	0.581	0.421	0.431	0.675	0.585	0.662	0.690

experimental results also show that MLFT is always superior to NMLFT, which testifies that the balanced label augmentation using the transformed input vectors produced by the proposed MLKLE and CKT is indispensable for an effective feature transformation.

In addition, Fig. 4 shows the class-wise classification performance measured by precision on TRECVID 2005 data set (as an example, because we can not show the results on all the data sets due space limit). The results show that, besides the overall performance as listed in Table 3, the proposed MLFT approach consistently outperform the other approaches in most of the individual functional classes, which again confirms the effectiveness of the proposed algorithms.

7.3 Label Enhancement in Multi-label Classification

A more careful analysis shows that the performance improvements of MLFT algorithm (measured by macro/micro average of F1 scores) over those of SVM using original features grow with the number of classes in a data set. The results are summarized in Fig. 3. Because F1 score is a balanced measurement over precision and recall via the harmonic mean, it is more representative for classification performance assessment. Therefore, we tentatively conclude that the benefit of using label augmentation is proportional to the number of classes in a multi-label data set. When the number of labels in a data set is larger, more label correlations are included in $\mathbf{z}_i$ to amend incorrect predictions in initialization. The extreme case is in single-label classification or even binary classification,

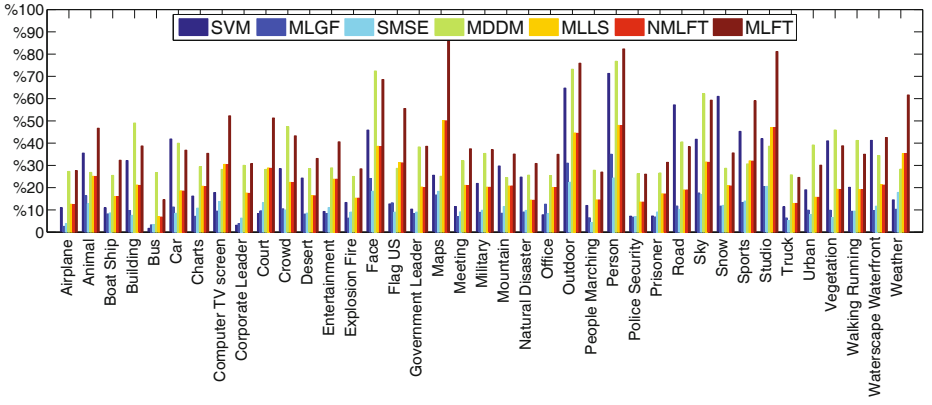


Fig. 4. Class-wise precisions of compared approaches on TRECVID 2005 data set

MLFT algorithm usually does not benefit from the augmentation by the label vector $\mathbf{y}_i$ because everything depends solely on exactly one label prediction. On the other hand, as shown in Fig. 3, as long as the number of classes of a data set is not small, MLFT algorithm always exhibits satisfying classification performance.

8 Conclusions

In this work, we first revealed that label assignments in multi-label classification not only indicate class membership of data points, but also convey very important characteristic information to assess similarity among data points from *knowledge* perspective. We then proposed a novel Multi-Label Feature Transform (MLFT) approach to use label assignments explicitly as part of data attributes and implicitly to formulate label correlations. Through two transformations on data attributes (via MLKLE) and label assignments (via CKT) respectively, the transformed input feature vector and input label vector have similar dimensionality, such that they can be integrated to form the label-augmented feature vector in a balanced manner. Data discriminability in the spanned space is thereby enhanced by taking advantage of the information coming from both data and knowledge perspectives. Moreover, although the proposed MLFT approach is originated from simple vector concatenation, by introducing kernel utility we may avert the explicit construction of label-augmented feature vector and thereby use a discriminative kernel to utilize data in a more flexible and effective manner. Extensively empirical evaluations are conducted on five standard multi-label data sets. Promising experimental results have demonstrated the effectiveness of our approach.

Acknowledgments. This research is supported by NSF-CCF 0830780, NSF-CCF 0939187, NSF-CCF 0917274, NSF-DMS 0915228, NSF-CNS 0923494.

References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)
2. Chen, G., Song, Y., Wang, F., Zhang, C.: Semi-supervised Multi-label Learning by Solving a Sylvester Equation. In: Jonker, W., Petković, M. (eds.) SDM 2008. LNCS, vol. 5159, Springer, Heidelberg (2008)
3. Chung, F.: Spectral graph theory. Amer. Mathematical Society, Providence (1997)
4. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Proc of NIPS (2001)
5. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
6. Griffiths, T., Ghahramani, Z.: Infinite latent feature models and the Indian buffet process. In: Proc. of NIPS (2006)
7. Hall, K.: An r-dimensional quadratic placement algorithm. Management Science, 219–229 (1970)
8. Ji, S., Tang, L., Yu, S., Ye, J.: Extracting shared subspace for multi-label classification. In: Proc of SIGKDD (2008)
9. Jolliffe, I.: Principal component analysis. Springer, Heidelberg (2002)
10. Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: Proc of CVPR, pp. 1719–1726 (2006)
11. Lewis, D., Yang, Y., Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
12. Liu, Y., Jin, R., Yang, L.: Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: Proc. of AAAI, p. 421 (2006)
13. Naphade, M., Kennedy, L., Kender, J., Chang, S., Smith, J., Over, P., Hauptmann, A.: LSCOM-lite: A light scale concept ontology for multimedia understanding for TRECVID 2005. Tech. rep., Technical report, IBM Research Tech. Report, RC23612, W0505-104 (2005)
14. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Proc. of ACM Multimedia (2006)
15. Wang, H., Huang, H., Ding, C.: Image Annotation Using Multi-label Correlated Greens Function. In: Proc. of ICCV, pp. 2029–2034 (2009)
16. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proc of SIGIR (2005)
17. Zha, Z., Mei, T., Wang, J., Wang, Z., Hua, X.: Graph-based semi-supervised learning with multi-label. In: Proc. of IEEE ICME, pp. 1321–1324 (2008)
18. Zhang, Y., Zhou, Z.: Multi-Label Dimensionality Reduction via Dependence Maximization. In: Proc of AAAI, pp. 1503–1505 (2008)
19. Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labelled classification using maximum entropy method. In: Proc. of SIGIR, pp. 274–281 (2005)