Label Embedding Based on Multi-Scale Locality Preservation

Cheng-Lun Peng[1,2], **An Tao**[3], **Xin Geng**[1,2,*]
[1] MOE Key Laboratory of Computer Network and Information Integration, China
[2] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[3] School of Information Science and Engineering, Southeast University, Nanjing 210096, China
{chenglunpeng, taoan, xgeng}@seu.edu.cn

Abstract

Label Distribution Learning (LDL) fits the situations well that focus on the overall distribution of the whole series of labels. The numerical labels of LDL satisfy the integrity probability constraint. Due to LDL's special label domain, existing label embedding algorithms that focus on embedding of binary labels are thus unfit for LDL. This paper proposes a specially designed approach MSLP that achieves label embedding for LDL by **M**ulti-**S**cale **L**ocality **P**reserving (MSLP). Specifically, MSLP takes the locality information of data in both the label space and the feature space into account with different locality granularity. By assuming an explicit mapping from the features to the embedded labels, MSLP does not need an additional learning process after completing embedding. Besides, MSLP is insensitive to the existing of data points violating the smoothness assumption, which is usually caused by noises. Experimental results demonstrate the effectiveness of MSLP in preserving the locality structure of label distributions in the embedding space and show its superiority over the state-of-the-art baseline methods.

1 Introduction

Learning with label ambiguity, i.e., one instance sometimes can not be fully described by only one label, is a hot topic in recent machine learning research as more and more real applications encounter that. Compared to the traditional Single-Label Learning (SLL), Multi-Label Learning (MLL) allows an instance to be assigned with multiple labels simultaneously so as to tackle the problem of label ambiguity to some extent [Zhang and Zhou, 2014]. However, MLL merely considers whether the label is positive (assigned with '1') or negative (assigned with '0'), but fails to point out the concrete relevant degree of the positive labels. Also, MLL can not deal with some applications where the overall distribution of the whole series of labels matters. Recently, a general learning paradigm named Label Distribution Learning (LDL) has been proposed to deal with such cases [Geng, 2016]. Unlike SLL and MLL, LDL shows more flexibility in tackling label ambiguity because it annotates each instance with a label distribution rather than a binary label vector. Fig. 1 shows an example of LDL in the filed of natural scene annotation. This natural scene image can be integrally described by all these listed labels with different relevant degrees. LDL shares some similarities with multi-output regression [Borchani *et al.*, 2015] and multi-ordinal regression [Zeng *et al.*, 2017] for they all annotate instances with numerical labels. To formulize LDL, let $S = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$ be a dataset with N samples, where $\boldsymbol{x}_i \in \mathbb{X} = \mathbb{R}^M$ and the corresponding $\boldsymbol{y}_i \in \mathbb{Y} = \mathbb{R}^L$. The description degree $y_i^c \in [0, 1]$ represents how much the cth label is relevant to the instance $\boldsymbol{x}_i$. Meanwhile, all the description degrees of an instance sum up to 1, i.e. $\sum_{c=1}^{L} y_i^c = 1$, which means that each instance can be fully described by using the whole series of labels. Given $\boldsymbol{X}_{N*M} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^t$ and $\boldsymbol{Y}_{N*L} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_N]^t$, the aim of LDL is to learn the function $g : \mathbb{X} \to \mathbb{Y}$ from the training dataset and use it to predict the label distributions for the unseen instances.

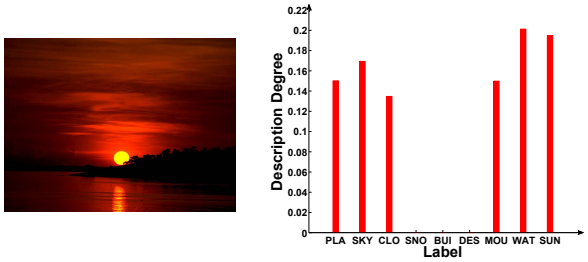


Figure 1: The label distribution of a natural scene image. The labels 'PLA', 'SKY', 'CLO', 'SNO', 'BUI', 'DES', 'MOU', 'WAT' and 'SUN' represent 'Plant', 'Sky', 'Cloud', 'Snow', 'Building', 'Desert', 'Mountain', 'Water' and 'Sun' respectively. Description Degree represents the relevant degree of each label with the image

Many LDL algorithms have already been proposed. Observing that the faces at the close ages own quite similar look, Geng et al. [2010] propose IIS-LDL to estimate the facial age, which is the first proposal of the LDL algorithm. Due to the lack of label distribution in the original data, they generate the age distribution manually through the Gaussian distribution. The improved algorithm BFGS-LDL and the neural network based approach CPNN for LDL are also proposed later [Geng *et al.*, 2013]. Then, Geng formulizes

*Corresponding author

LDL as a general learning paradigm and puts forward four more algorithms [2016]: AA-BP, AA-KNN, PT-SVM and PT-Bayes. Some initial efforts are also made to realize LDL by deep learning and ensemble learning. For example, Gao et al. [2017] develop a DNN based model called Deep LDL. Shen et al. [2017] propose LDL Forests based on differentiable decision trees. In addition, some well designed algorithms, such as LDSVR [Geng and Hou,], view LDL as a regression problem. Besides, LDL has also been applied to many real-world applications successfully. For example, Zhou et al. [2015] utilize the emotion distribution to recognize facial expressions. Ren and Geng [] use the distribution of ratings to portray the human sense toward facial beauty. LDL has also been studied in the field of natural scene annotation [Geng and Luo, 2014], crowd counting [Zhang *et al.*, 2015], video parsing [Geng and Ling, 2017], etc.

Label Embedding (LE) has become a popular research topic in SLL and MLL. LE usually embeds the original labels into a new space and then conducts the learning from the features to the embedded labels. Finally, a specific label decoder recovers the predicted embeddings to the original label space. LE owns the advantage in addressing the problematical label space as well as capturing the high-order label correlations. The labels in LDL may encounter problems such as the redundancy and the noise. Also, the effective exploitation of the label correlations is considered to be crucial for the success for LDL. However, almost all existing LE approaches are proposed for SLL and MLL [Hsu *et al.*, 2009; Tai and Lin, 2012; Chen and Lin, 2012; Lin *et al.*, 2014; Bhatia *et al.*, 2015; Yeh *et al.*, 2017; Zhang and Schneider, 2012; Ferng and Lin, 2013], i.e., they focus on the embedding of binary labels whose values are whether positive or negative. The corresponding decoders also aim at recovering the binary labels from the encoded ones. These algorithms thus can not be applied to LDL directly for the special label domain of LDL. Moreover, many leading LE methods assume that the training label matrix is sparse or low-rank, for there are usually less positive labels than vast negative labels. However, such assumption is violated in LDL case. In order to realize LE in LDL, at least two main issues should be tackled properly: 1) How to exploit the information of label distributions efficiently. Compared to binary labels, the numerical labels usually contain more abundant information that is beneficial for subsequent prediction. 2) How to design a decoder that restricts the recovered label vector to satisfy the constraints of the label distribution.

This paper takes an initial attempt to combine LE with LDL and proposes a specially designed LE approach named MSLP. Inspired by the typical manifold learning method Laplacian Eigenmaps [Belkin and Niyogi, 2002], MSLP tries to conduct locality-preserving label embedding. That is, MSLP aims to search the embedded label vectors $\boldsymbol{y'} \in \mathbb{Y}' = \mathbb{R}^l$ that maintain the neighborhood relationship among label distributions of data points. After obtaining the embedded labels, most LE methods need to choose an extra model to learn the mapping from the features to the embedded labels. By assuming an explicit mapping $\boldsymbol{y'} = \boldsymbol{x}^T\boldsymbol{V}$ when conducting embedding, MSLP omits that additional learning process and becomes feature-aware. In order to maintain the label neighborhood structure better through $\boldsymbol{x}^T\boldsymbol{V}$, MSLP restricts that the neighbors of one point found in the label space should be only determined within its neighbors in the feature space. The benefit of this restriction will be explained in Section 2. Combining the locality information of data points in both the spaces of label and feature with different locality granularity, MSLP achieves the Multi-Scale Locality Preserving label embedding for LDL. For real-world data, some data points may violate the smoothness assumption [Chapelle *et al.*, 2006], which is caused by the noises that arise unavoidably during the collection and the preparation of data. MSLP is designed to be insensitive to the existing of such data points so as to alleviate this problem.

The rest of the paper is organized as follows. The proposed algorithm MSLP is derived and discussed in Section 2. Then, the experimental results are reported in Section 3. Finally, the conclusion is drawn in Section 4.

2 The Proposed Method

Unlike SLL and MLL, labels in LDL are numerical, which enables MSLP to exploit the label structure hidden in the label space. Inspired by Laplacian Eigenmaps [Belkin and Niyogi, 2002], MSLP also aims to perform locality preserving embedding. Focusing on the label space, it tries to maintain the neighborhood structure of label distributions in the embedding space. Such embedded label vectors can be obtained through minimizing the following objective:

$$\begin{aligned} \min_{\boldsymbol{Y'}} \ & \frac{1}{2}\sum_{i,j} \parallel \boldsymbol{y'_i} - \boldsymbol{y'_j} \parallel^2 \boldsymbol{W}^+_{\boldsymbol{y},ij} \\ s.t. \ & \boldsymbol{Y'}^T\boldsymbol{D}^+\boldsymbol{Y'} = \boldsymbol{I} \end{aligned}, \tag{1}$$

where $\boldsymbol{W}^+_{\boldsymbol{y}}$ is an $N * N$ weight matrix incorporating neighborhood information, $\boldsymbol{D}^+$ is an $N*N$ diagonal matrix whose entries $\boldsymbol{D}^+_{ii} = \sum_j \boldsymbol{W}^+_{\boldsymbol{y},ij}$, $\boldsymbol{Y'}_{N*l} = [\boldsymbol{y'_1}, ..., \boldsymbol{y'_N}]^t$ is the embedded label matrix, and $\boldsymbol{I}$ is a $l * l$ unit matrix. The goal of the constraint above is to remove an arbitrary scaling factor of the embedded labels while $\boldsymbol{D}^+$ provides an natural measurement on the data points [Belkin and Niyogi, 2002]. Here, we adopt the form of heat kernel as the weight matrix, i.e.,

$$\boldsymbol{W}^+_{\boldsymbol{y},ij} = \begin{cases} exp(-\dfrac{dis(\boldsymbol{y}_i, \boldsymbol{y}_j)}{\sigma}), i \in Nei_{\boldsymbol{y}}(j) \ or \ j \in Nei_{\boldsymbol{y}}(i) \\ 0, \qquad\qquad otherwise \end{cases}, \tag{2}$$

where $\sigma > 0$ is a scalar parameter, and $dis()$ is an optional function measuring the distance between probability distributions. Here, we first define the set $Nei_{\boldsymbol{y}}$ for a specific data point $\boldsymbol{p}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)$ in the training set $\boldsymbol{S}$ as follow:

$$Nei_{\boldsymbol{y}}(i) = \psi_{\boldsymbol{y}}(\boldsymbol{p}_i, \ k^+, \ \{\boldsymbol{p}_j \mid \boldsymbol{p}_j \neq \boldsymbol{p}_i \wedge \boldsymbol{p}_j \in S\}). \tag{3}$$

The function $\psi_{\boldsymbol{y}}$ with variables above returns the k^+ nearest neighbors of the point $\boldsymbol{p}_i$ in the label space among the point set $\{\boldsymbol{p}_j \mid \boldsymbol{p}_j \neq \boldsymbol{p}_i \wedge \boldsymbol{p}_j \in S\}$.

As mentioned above, most LE approaches need an additional learning process from the features to the embedded labels. Avoiding that, MSLP assumes $\boldsymbol{y'} = \boldsymbol{V}^T\boldsymbol{x}$, where $\boldsymbol{V}_{D*l}$ is the regression matrix to be learned. By substituting $\boldsymbol{y'}$ for

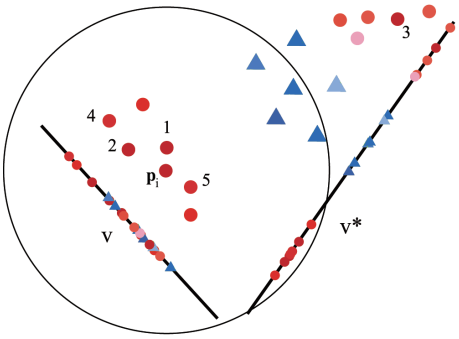


Figure 2: Different embedding results led by different definitions of $Nei_{\boldsymbol{y}}$. Consider a 2-dimensional feature space, a 3-dimensional label space and a 1-dimensional target embedded label space. The label distributions are displayed as data points' RGB values. The more similar color data points have, the closer label distribution they own. With $k^+ = 5$, the signs '1'-'5' denote five points within $Nei_{\boldsymbol{y}}(i)$ computed through Eq. (3). Compared with other data pairs, the solution of minimizing the objective (4) will overemphasize the data pair $\boldsymbol{p}_i$ and its 3rd nearest neighbor, for it owns much larger feature distance. This will lead to an overall bad mapping $\boldsymbol{v}$. Using the new definition of $Nei_{\boldsymbol{y}}$ in Eq. (5) with $k^+ = 5$ and $\alpha = 2$, the original 3rd neighbor will not be included in $Nei_{\boldsymbol{y}}(i)$ because it does not exist in $\boldsymbol{p}_i$'s $Nei_{\boldsymbol{x}}$ set. Then, the better mapping $\boldsymbol{v}^*$ can be obtained.

$\boldsymbol{V}^T\boldsymbol{x}$ into Eq. (1) and introducing a regulation term to avoid overfitting, the objective becomes

$$\min_{\boldsymbol{V}} \ \frac{1}{2}\sum_{ij} \parallel \boldsymbol{V}^T\boldsymbol{x}_i - \boldsymbol{V}^T\boldsymbol{x}_j \parallel^2 \boldsymbol{W}^+_{\boldsymbol{y},ij} + \lambda \parallel \boldsymbol{V} \parallel_F^2 \quad , \quad (4)$$
$$s.t. \ \boldsymbol{V}^T\boldsymbol{X}^T\boldsymbol{D}^+\boldsymbol{X}\boldsymbol{V} = \boldsymbol{I}$$

where λ is a regulation parameter and $\parallel \cdot \parallel_F$ means Frobenious norm. For an unseen instance $\boldsymbol{x}_u$, we first compute its corresponding embedded label vector $\widehat{\boldsymbol{y}}'_u = \boldsymbol{V}^T\boldsymbol{x}_u$. Then, the *knn*-based decoder recovers the predicted label distribution $\widehat{\boldsymbol{y}}_u$ by averaging the label distributions of k nearest neighbors of $\widehat{\boldsymbol{y}}'_u$ among the embedded label matrix $\boldsymbol{Y}'$.

$Nei_{\boldsymbol{y}}$ and $\boldsymbol{W}^+_{\boldsymbol{y}}$ are yet determined by the distance of label distributions between data points but ignore their distance in the feature space. Viewing regression matrix $\boldsymbol{V}$ as the projection directions for the feature, we can see from the objective (4) that the solution for $\boldsymbol{V}$ will tend to be dominated by the large feature distances of data pairs where $\boldsymbol{W}^+_{\boldsymbol{y},ij} \neq 0$. As an example showed in Fig. 2, this may lead to the overlap of points with dissimilar label distributions and the scatter of points with similar label distributions. The bad resulting embeddings will result in an overall low prediction accuracy, because the decoder of MSLP is knn-based in the embedded label space.

This problem happens either when some outliers are within some data points' $Nei_{\boldsymbol{y}}$ set, or when data points are essentially distributed in different feature areas but share quite similar label distribution. To overcome it, we introduce another parameter $\alpha \geq 1$ and restrict that the k^+ nearest neighbors of one data point in the label space should be found *within* its αk^+ nearest neighbors in the feature space. That is, utilizing different locality granularity in the label space and the feature

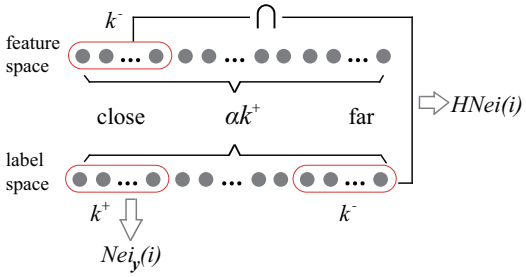


Figure 3: The computation process of sets $Nei_{\boldsymbol{y}}$ and $HNei$ for $\boldsymbol{p}_i$. Both sets are computed among the αk^+ points in $Nei_{\boldsymbol{x}}(i)$. These αk^+ points are listed with ascending order of feature distance in the 1st line and of the label distance in the 2nd line.

space, the locality information of data points in both spaces are integrated. The set $Nei_{\boldsymbol{y}}$ is now redefined as follow:

$$\begin{aligned} Nei_{\boldsymbol{y}}(i) &= \psi_{\boldsymbol{y}}(\boldsymbol{p}_i, \ k^+, \ Nei_{\boldsymbol{x}}(i)) \\ Nei_{\boldsymbol{x}}(i) &= \psi_{\boldsymbol{x}}(\boldsymbol{p}_i, \ \alpha k^+, \ \{\boldsymbol{p}_j \mid \boldsymbol{p}_j \neq \boldsymbol{p}_i \wedge \boldsymbol{p}_j \in S\}) \end{aligned} . \quad (5)$$

The function $\psi_{\boldsymbol{x}}$ is similar to $\psi_{\boldsymbol{y}}$ except that it finds neighbors in the feature space. Now each point will focus on data points with similar label distribution in its neighboring feature space, as the circle for $\boldsymbol{p}_i$ shows in Fig. 2. Therefore, the label embedding is achieved by the multi-scale locality preserving.

According to the smoothness assumption [Chapelle *et al.*, 2006], the neighboring data points in the feature space are more likely to share the similar labels. But for real-world data, the noise is usually an unavoidable problem, affecting the collection and the preparation of data. To reduce negative influences caused by that, MSLP tries to keep each data point far away from its hetero-neighbors after embedding. Here, we define the hetero-neighbors of one data point $\boldsymbol{p}_i$ as points which keep very close to it in the feature space, but keep far away from it in the label space. Specifically, the hetero-neighbor set $HNei$ of $\boldsymbol{p}_i$ is defined as

$$HNei(i) = \psi_{\boldsymbol{x}}(\boldsymbol{p}_i, k^-, Nei_{\boldsymbol{x}}(i)) \cap \psi_{\boldsymbol{y}}(\boldsymbol{p}_i, -k^-, Nei_{\boldsymbol{x}}(i)), \quad (6)$$

where k^- $(0 \leq k^- \leq \alpha k^+ - k^+)$ controls the maximum possible number of hetero-neighbors we may find for $\boldsymbol{p}_i$. And $\psi_{\boldsymbol{y}}$ with $-k^-$ means finding k^- *farthest* points from $\boldsymbol{p}_i$ among the given set of points. Note that there may be no hetero-neighbor around $\boldsymbol{p}_i$ which is reflected by an empty intersection $HNei(i)$. Fig. 3 illustrates the computation process of sets $Nei_{\boldsymbol{y}}$ and $HNei$ for $\boldsymbol{p}_i$. We denote $\boldsymbol{W}^-$ to show the existing of hetero-neighbor pairs, i.e.,

$$\boldsymbol{W}^-_{ij} = \begin{cases} 1, & i \in HNei(j) \ or \ j \in HNei(i) \\ 0, & otherwise \end{cases} . \quad (7)$$

Then, the final objective becomes:

$$\begin{aligned} \min_{\boldsymbol{V}} \ & \frac{\beta}{2}\sum_{ij} \parallel \boldsymbol{V}^T\boldsymbol{x}_i - \boldsymbol{V}^T\boldsymbol{x}_j \parallel^2 \boldsymbol{W}^+_{\boldsymbol{y},ij} - \\ & \frac{(1-\beta)}{2}\sum_{ij} \parallel \boldsymbol{V}^T\boldsymbol{x}_i - \boldsymbol{V}^T\boldsymbol{x}_j \parallel^2 \boldsymbol{W}^-_{ij} + \lambda \parallel \boldsymbol{V} \parallel_F^2, \quad (8) \\ & s.t. \ \boldsymbol{V}^T\boldsymbol{X}^T\boldsymbol{D}^+\boldsymbol{X}\boldsymbol{V} = \boldsymbol{I} \end{aligned}$$

Algorithm 1 MSLP

Input: Training dataset $S = \{\boldsymbol{X}, \boldsymbol{Y}\}$, testing instance $\boldsymbol{x}_u$, embedding dimensionality l, distance function $dis()$, parameters $\{k^+, k^-, \alpha, \beta, \lambda\}$ for the encoder and k for the *knn*-based decoder.

Output: Predicted label distribution $\widehat{\boldsymbol{y}}_u$.

1: Compute the sets $Nei_{\boldsymbol{y}}$ and $HNei$ for each point in S.
2: Compute corresponding $\boldsymbol{W}^+$ and $\boldsymbol{W}^-$.
3: Obtain the optimal $\boldsymbol{V}$ via eigen-decomposition of (10).
4: Compute the embedded label vector $\widehat{\boldsymbol{y}}'_u = \boldsymbol{V}^T\boldsymbol{x}_u$.
5: Find k nearest neighbors of $\widehat{\boldsymbol{y}}'_u$ within $\boldsymbol{XV}$.
6: Decode $\widehat{\boldsymbol{y}}_u$ from $\widehat{\boldsymbol{y}}'_u$ by averaging the label distributions of these k nearest points found above.

where $\beta \in [0, 1]$ balances the importance of the first two terms of Eq. (8).

The first term in (8) can be reduced to

$$\begin{aligned}
&\frac{\beta}{2}\sum_{ij} tr[\boldsymbol{V}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{V}]\boldsymbol{W}^+_{\boldsymbol{y},ij} \\
&= \beta\sum_i tr[\boldsymbol{V}^T\boldsymbol{x}_i\boldsymbol{D}^+_{\boldsymbol{ii}}\boldsymbol{x}_i^T\boldsymbol{V}] - \beta\sum_{ij} tr[\boldsymbol{V}^T\boldsymbol{x}_i\boldsymbol{W}^+_{\boldsymbol{y},ij}\boldsymbol{x}_j^T\boldsymbol{V}] \\
&= \beta\, tr[\boldsymbol{V}^T(\sum_i \boldsymbol{x}_i\boldsymbol{D}^+_{\boldsymbol{ii}}\boldsymbol{x}_i^T - \sum_{ij}\boldsymbol{x}_i\boldsymbol{W}^+_{\boldsymbol{y},ij}\boldsymbol{x}_j^T)\boldsymbol{V}] \\
&= \beta\, tr[\boldsymbol{V}^T\boldsymbol{X}^T(\boldsymbol{D}^+ - \boldsymbol{W}^+)\boldsymbol{XV}]
\end{aligned}$$

Through similar computing, the second term in (8) becomes $(1-\beta)tr[\boldsymbol{V}^T\boldsymbol{X}^T(\boldsymbol{D}^- - \boldsymbol{W}^-)\boldsymbol{XV}]$, where $\boldsymbol{D}^-_{ii} = \sum_j \boldsymbol{W}^-_{ij}$.

Then, by simple algebraic operations, the final objective becomes

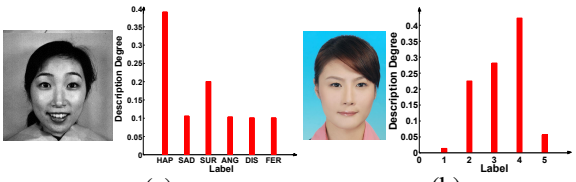
$$\begin{aligned}
&\min_{\boldsymbol{V}}\ tr[\boldsymbol{V}^T(\boldsymbol{X}^T(\beta\boldsymbol{M}^+ - (1-\beta)\boldsymbol{M}^-)\boldsymbol{X} + \lambda\boldsymbol{I})\boldsymbol{V}] \\
&s.t.\ \ \boldsymbol{V}^T\boldsymbol{X}^T\boldsymbol{D}^+\boldsymbol{XV} = \boldsymbol{I}
\end{aligned}, \quad (9)$$

where $\boldsymbol{M}^+ = \boldsymbol{D}^+ - \boldsymbol{W}^+$ and $\boldsymbol{M}^- = \boldsymbol{D}^- - \boldsymbol{W}^-$. Applying the Lagrangian method, the problem can be transfered into a general eigen-decomposition problem, i.e.

$$(\boldsymbol{X}^T\boldsymbol{M}\boldsymbol{X} + \lambda\boldsymbol{I})\boldsymbol{v} = \eta(\boldsymbol{X}^T\boldsymbol{D}^+\boldsymbol{X})\boldsymbol{v}, \quad (10)$$

where $\boldsymbol{M} = \beta\boldsymbol{M}^+ - (1-\beta)\boldsymbol{M}^-$ and the closed-form solution of optimal $\boldsymbol{V}$ consists of the first l normalized eigenvectors corresponding to the top l smallest eigenvalues of above. The procedures of MSLP are summarized in Algorithm 1.

One thing deserves to be mentioned is the relationship between MSLP and Feature Embedding (FE) methods. FE mainly focuses on maintaining the characteristic of the feature space (unsupervised) or finding a new feature space easier for subsequent learning (supervised). Instead, MSLP concentrates on the label space essentially, although utilizes $\boldsymbol{V}^T\boldsymbol{x}$ to approximate $\boldsymbol{y}'$ via an explicit linear assumption. On the other hand, MSLP and many FE methods all compute a transformation matrix for the feature to map data points into a subspace. Under this perspective, MSLP can be viewed to transfer the information of *label neighborhood structure* to guide the process of feature embedding. In the following section, we will also do experiments to show the different performances of MSLP and some popular FE methods.



(a) (b)

Figure 4: (a) An image in s-JAFFE and its emotion distribution. The labels 'HAP', 'SAD', 'SUR', 'ANG', 'DIS' and 'FER' represent 'Happiness', 'Sadness', 'Surprise', 'Anger', 'Disgust' and 'Fear' respectively. (b) An image in SCUT-FBP and its beauty distribution.

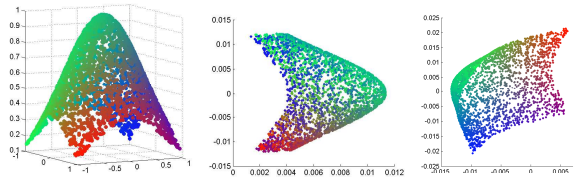
3 Experiment

To validate the effectiveness of MSLP, we visualize the embedding results of MSLP and also conduct extensive experiments on five real-world datasets among different fields.

3.1 Experiment Configuration

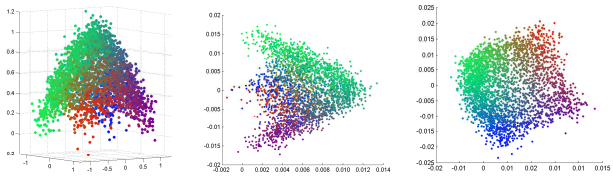
Real-world Datasets The LDL dataset Natural_Scene (NS) [2014] is utilized. NS includes 2,000 natural scene images and each image is represented by 294 features and a 9-dimensional label distribution. Fig. 1 has showed one image of NS. Two widely used facial expression datasets s-JAFFE and s-BU_3DFE are used here, for they have been extended to the standard LDL datasets by Zhou et al. [2015]. There are 213 facial images in s-JAFFE and 2500 in s-BU_3DFE. Each image is described by 6 basic emotions, as an example image showed in Fig. 4 (a). The 243 features are extracted for each image by the same method used in [Zhou *et al.*, 2015]. In detail, images are manually cropped to make eyes at the same positions and resized to 110*140 pixels. Then, the features are extracted by LBP. We also collect two facial beauty datasets SCUT-FBP and Multi-Modality Beauty (M^2B) with the information of label distributions [Ren and Geng,]. The beauty distribution of each image consists of five rates '1'-'5'. And a higher rate indicates a more intensive sense about beauty. Fig. 4 (b) displays a typical image of that. Following the same way as Ren and Geng [] did, we extract 300 features for 1500 images in SCUT-FBP by LBP, HOG and Gabor filter, and 250 features for 1240 images in M^2B by LBP, Gabor filter and Color moment.

Comparing Algorithms We compare MSLP with eight popular LDL methods: IIS-LDL, CPNN, BFGS-LDL, LDSVR, AA-BP, AA-KNN, PT-SVM and PT-Bayes. For MSLP, the squared Euclidean distance is chosen as $dis()$, and the σ is set to the average of squared Euclidean distances among all pairs whose $\boldsymbol{W}^+_{\boldsymbol{y},ij} \neq 0$. k^+, α is chosen from {5, 10} and $k^- = k^+$ simply, β is selected from {0, 0.1, 0.5}, λ is chosen from {0, 0.01, 0.1}, and k for decoding is chosen from {5, 10, 15}. The number of hidden-layer neurons for CPNN and AA-BP is set to 50, and k for AA-KNN is selected from {5, 10, 15}. BFGS-LDL and IIS-LDL follow the advised settings in [Geng, 2016]. The rbf kernel is used for LDSVR and PT-SVM with its width set to the average Euclidean distance among training instances. Also, aiming to show the superiority of MSLP versus FE, four typical FE methods CCA [Hardoon *et al.*, 2003], LPP [He and Niyogi, 2004], NPE [He *et al.*, 2005] and PCA [Jolliffe, 1986] are



(a) Original (b) LPP (c) MSLP

Figure 5: Embedding results on the toy dataset.



(a) Original (b) LPP (c) MSLP

Figure 6: Embedding results on the noisy toy dataset.

compared, and AA-KNN is used as their subsequent predictor. The number of neighbors for LPP and NPE is selected from {5, 10, 15} and the width of heat kernel for LPP is also set to the average squared Euclidean distances among its neighbor pairs. Moreover, the compared FE methods are allowed to be extended to their kernel version with the rbf kernel, which gives them full chances to beat MSLP. To be fair, the ratio u of the embedding dimensionality over the original feature dimensionality for MSLP and FE methods ranges over $\{10\%, 20\%, ..., 100\%\}$, and the best one is adopted. For all datasets and algorithms, the 10-fold cross validation is conducted and the average performance is recorded.

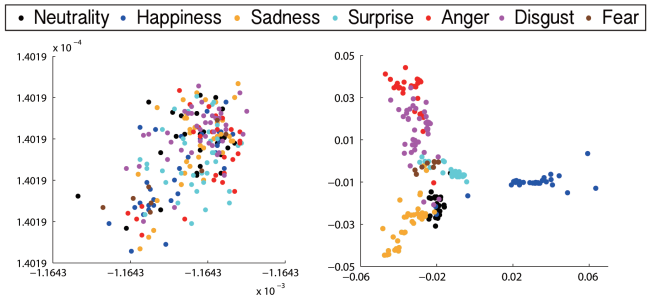
Evaluation Metrics As Geng suggests [2016], we select four distance metrics (Chebyshev, Clark, Kullback-Leibler and Canberra) and two similarity metrics (Cosine and Intersection), which are widely-used in LDL, to evaluate the performance of different approaches.

3.2 Visualization

The proposed MSLP is motivated by the Laplacian Eigenmaps, and they share some similar spirits in some way. To visually show the embedding effect of MSLP and its difference from FE, we compare MSLP with LPP — the linear version of the Laplacian Eigenmaps. As being a popular manifold learning method, LPP aims to find the embedded features that optimally preserves the locality of data points in the original feature space. We generate a toy LDL dataset with 3 features and 3 labels in the similar way to [Geng, 2016]. The label distribution $\boldsymbol{y} = \{y^1, y^2, y^3\}$ of the corresponding $\boldsymbol{x} = \{x^1, x^2, x^3\}$ is created as follows:

$$\begin{aligned} t^j &= x^j + 0.5(x^j)^2 + 0.2(x^j)^3 + 1, \quad j = 1, 2, 3, \\ \phi^1 &= (\boldsymbol{w}_1^T \boldsymbol{t})^2, \\ \phi^2 &= (\boldsymbol{w}_2^T \boldsymbol{t} + 0.01\phi^1)^2, \\ \phi^3 &= (\boldsymbol{w}_3^T \boldsymbol{t} + 0.01\phi^2)^2, \\ y^j &= \frac{\phi^j}{\phi^1 + \phi^2 + \phi^3}, \quad j = 1, 2, 3, \end{aligned} \tag{11}$$

where $\boldsymbol{w}_1 = [4, 2, 1]^T$, $\boldsymbol{w}_2 = [1, 2, 4]^T$, and $\boldsymbol{w}_3 = [1, 4, 2]^T$.



(a) LPP (b) MSLP

Figure 7: Embedding results on the s-JAFFE.

We generate 3000 data points to form the dataset. The first two components of $\boldsymbol{x}$ are randomly sampled within $[-1, 1]$ and the third one $x^3 = exp[-(x^1)^2 - (x^2)^2]$. We transfer these points into a 2-dimensional embedding space with different methods and utilize RGB colors to show their label distributions. In addition, a noisy toy dataset is also formed by adding the Gaussian noise with a zero mean and a variance of 0.1 to the features. The results are displayed in Fig 5 $\sim$ 6. As these figures show, MSLP maintains the structure of label neighborhood well within the local feature space, while LPP ignores the label locality and results in a mess in color areas.

Using s-JAFFE as an example, we also conduct the visualization for real-world datasets. We use black color to describe facial images with neutral expression (i.e., the emotion distribution is close to the uniform distribution) as these images have been labeled by Lyons et al. [1998]. Besides, other six different colors are used to display images according to the highest description degree of six different basic emotions. Fig. 7 shows the results. Again, MSLP gathers points sharing similar emotion distributions and forms some evident color areas. In contrast, LPP leads to the chaos of colors. Moreover, the distribution of emotion areas in Fig. 7 (b) is also consistent to the human intuition. For example, the points of 'Neutrality' are closer to the points of 'Sadness' than 'Happiness', for it is usually much easier for a neutral face to be judged as expressing negative other than positive emotions in the real world. Also, points of 'Anger' and 'Disgust' are near, as well as the points of 'Surprise' and 'Fear', for these emotions usually appear simultaneously.

3.3 Experimental Results

The quantitative experimental results are presented in Table 1 $\sim$ 3. For each evaluation metric, '$\downarrow$' indicates "the smaller the better", while '$\uparrow$' indicates "the larger the better". The pairwise t-tests at 90% significance level is conducted and the best performance of each column is highlighted in boldface. Across all metrics, MSLP ranks 1st in 93.3% cases, which reveals the superiority of MSLP against many state-of-the-art well-established LDL algorithms. Note that FE methods compared here are even allowed to be extended to their kernel version, ensuring to make the embedding nonlinear. But even so, MSLP still achieves better performance, which proves the effectiveness of exploiting multi-scale locality. It can be seen that no approach can hold the 1st place over all six metrics in

Datasets	s-JAFFE						s-BU_3DFE					
Metrics	Cheb ↓	Cla ↓	Can ↓	KL ↓	Cos ↑	Inter ↑	Cheb ↓	Cla ↓	Can ↓	KL ↓	Cos ↑	Inter ↑
CPNN	0.1112	0.3995	0.8351	0.0642	0.9389	0.8567	0.1342	0.4052	0.8778	0.0792	0.9227	0.8426
LDSVR	0.1157	0.4050	0.8470	0.0675	0.9363	0.8555	0.1260	0.3834	0.8261	0.0726	0.9289	0.8518
BFGS-LDL	0.1042	0.4143	0.8528	0.0674	0.9399	0.8582	**0.1043**	0.3488	0.7267	**0.0542**	**0.9470**	0.8711
IIS-LDL	0.1194	0.4246	0.8855	0.0727	0.9315	0.8491	0.1346	0.4142	0.8978	0.0821	0.9200	0.8392
AA-BP	0.1290	0.4919	1.0110	0.1041	0.9128	0.8289	0.1321	0.4386	0.9255	0.0880	0.9185	0.8359
AA-KNN	0.0992	0.3523	0.7269	0.0544	0.9477	0.8408	0.1244	0.3871	0.8177	0.0734	0.9280	0.8522
PT-SVM	0.1232	0.4447	0.9329	0.0787	0.9260	0.8467	0.1415	0.4273	0.9224	0.0910	0.9125	0.8342
PT-Bayes	0.1204	0.4291	0.9002	0.0738	0.9304	0.8467	0.1389	0.4128	0.9028	0.0851	0.9177	0.8387
LPP	0.1024	0.3701	0.7614	0.0557	0.9468	0.8695	0.1229	0.3876	0.8041	0.0734	0.9278	0.8534
NPE	**0.0950**	**0.3376**	**0.6924**	0.0491	0.9529	**0.8807**	0.1231	0.3914	0.8108	0.0741	0.9270	0.8520
PCA	0.0987	0.3469	0.7143	0.0528	0.9493	0.8767	0.1277	0.4012	0.8305	0.0806	0.9212	0.8483
CCA	0.1174	0.4123	0.8505	0.0725	0.9319	0.8530	0.1148	0.3676	0.7646	0.0652	0.9357	0.8616
MSLP	**0.0919**	**0.3307**	**0.6748**	**0.0451**	**0.9566**	**0.8842**	**0.1035**	**0.3305**	**0.6808**	**0.0520**	**0.9480**	**0.8762**

Table 1: Experimental results on facial expression recognition.

Datasets	SCUT-FBP						Multi-Modality Beauty (M^2B)					
Metrics	Cheb ↓	Cla ↓	Can ↓	KL ↓	Cos ↑	Inter ↑	Cheb ↓	Cla ↓	Can ↓	KL ↓	Cos ↑	Inter ↑
CPNN	0.4436	1.5943	3.2116	1.1801	0.5759	0.4918	0.3726	1.3012	2.6129	0.5649	0.7082	0.5584
LDSVR	0.2694	1.4280	2.7575	0.5684	0.8341	0.7121	0.3611	1.2704	2.5457	**0.5363**	**0.7235**	0.5781
BFGS-LDL	0.3517	1.5321	2.9985	0.8572	0.6905	0.5557	0.3720	**1.2180**	**2.4166**	0.6898	0.6759	0.5698
IIS-LDL	0.6493	1.8615	3.9359	3.2270	0.3593	0.2985	0.3790	1.3136	2.6368	0.5864	0.6970	0.5522
AA-BP	0.2538	1.4002	2.5981	0.4157	0.8372	0.6948	0.3781	1.3142	2.6356	0.5851	0.6992	0.5529
AA-KNN	0.2148	1.2761	2.2953	0.3602	0.8691	0.7435	0.3754	1.2204	2.4190	0.6884	0.6711	0.5600
PT-SVM	0.4184	1.5736	3.1111	1.1354	0.5784	0.5080	0.4139	1.3576	2.7366	0.8057	0.5990	0.5004
PT-Bayes	0.3836	1.5376	3.0516	1.1328	0.6779	0.5156	0.6905	2.0668	4.4951	11.832	0.4474	0.3044
LPP	0.2202	1.3219	2.4098	0.3256	0.8667	0.7358	0.3675	1.2557	2.5167	0.5893	0.6987	0.5662
NPE	0.2133	1.3057	2.3630	**0.2854**	0.8784	0.7446	0.3645	1.2688	2.5295	0.5643	0.7095	0.5695
PCA	0.2144	1.3082	2.3701	0.3144	0.8717	0.7435	0.3688	1.2433	2.4858	0.6151	0.6902	0.5666
CCA	0.2141	1.2938	2.3404	**0.2927**	0.8724	0.7473	**0.3559**	1.2382	2.4690	0.5476	0.7192	**0.5811**
MSLP	**0.2046**	**1.2602**	**2.2477**	**0.2813**	**0.8823**	**0.7608**	**0.3549**	**1.2095**	**2.4058**	0.5684	0.7127	**0.5844**

Table 2: Experimental results on facial beauty sense.

Datasets	Natural_Scene (NS)					
Metrics	Cheb ↓	Cla ↓	Can ↓	KL ↓	Cos ↑	Inter ↑
CPNN	0.3136	2.4720	6.8613	0.9022	0.6847	0.4908
LDSVR	0.4082	2.3884	6.7650	1.1158	0.6372	0.5093
BFGS-LDL	0.3342	2.3956	6.5829	0.9310	0.6979	0.5416
IIS-LDL	0.3569	2.4737	6.8221	0.9437	0.6649	0.4618
AA-BP	0.3387	2.4593	6.7762	0.8925	0.6898	0.4907
AA-KNN	0.3055	2.2548	5.8358	**0.7919**	0.7309	0.5567
PT-SVM	0.4282	2.5696	7.2756	1.5533	0.4583	0.3497
PT-Bayes	0.4047	2.5206	7.1398	2.2363	0.5601	0.3305
LPP	0.3113	2.2959	6.0166	0.8307	0.7178	0.5390
NPE	0.3021	2.2324	5.7596	**0.7910**	0.7359	0.5619
PCA	0.3048	2.2512	5.8199	**0.7913**	0.7310	0.5577
CCA	0.3188	2.3139	6.1598	0.8361	0.7243	0.5319
MSLP	**0.2927**	**2.2198**	**5.7214**	**0.7849**	**0.7406**	**0.5696**

Table 3: Experimental results on natural scene annotation.

the field of facial beauty sense. Unlike the task of expression recognition and scene annotation, the sense about beauty is a quite personal feeling and shows much ambiguities, thus it is a harder task for a method to perform well in all metrics.

Also, across all five datasets, MSLP reaches the best performance with the embedding ratio $u = 10\%$ consistently, while the best ratios of compared FE methods vary from 10% to 90% in different datasets. Observing the datasets collected here, the possible reason of such phenomenon is that MSLP essentially performs embedding for the label, which owns much lower dimensionality compared to that of the feature. Thus, MSLP usually needs less embedding dimensions than FE methods to maintain the beneficial information or property of the original label space in the embedding space.

4 Conclusion

This paper proposes a novel LE approach named MSLP for LDL. Unlike most previous works, MSLP deals with the numerical labels and integrates the neighborhood structure of points in both the spaces of label and feature. MSLP is also insensitive to the presence of hetero-neighbors. Experimental results reveal the effectiveness of MSLP in gathering points with similar label distributions in the embedding space, i.e., the label locality is well preserved through MSLP. Although proposed for LDL, it is also interesting to note that MSLP can be directly shifted to some other learning paradigms (e.g., multi-output regression) which own numerical labels. In the future, we will explore if there exist better ways to utilize the structure information described by the label distributions.

Acknowledgments

This research was supported by the National Key Research & Development Plan of China (No. 2017YFB1002801), the National Science Foundation of China (61622203), the Jiangsu Natural Science Funds for Distinguished Young Scholar (BK20140022), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Collaborative Innovation Center of Wireless Communications Technology.

References

[Belkin and Niyogi, 2002] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, Vancouver, Canada, 2002.

[Bhatia *et al.*, 2015] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *NIPS*, pages 730–738, Montréal, Canada, 2015.

[Borchani *et al.*, 2015] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga. A survey on multi-output regression. *WIREs Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[Chapelle *et al.*, 2006] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, London, U.K., 2006.

[Chen and Lin, 2012] Y.-N. Chen and H.-T. Lin. Feature-aware label space dimension reduction for multi-label classification. In *NIPS*, pages 1529–1537, Lake Tahoe, NV, 2012.

[Ferng and Lin, 2013] C.-S. Ferng and Hsuan-T. Lin. Multi-label classification using error-correcting codes of hard or soft bits. *TNNLS*, 24(11):1888–1900, 2013.

[Gao *et al.*, 2017] B.-B. Gao, C. Xing, C.-W. Xie, J.-X. Wu, and X. Geng. Deep label distribution learning with label ambiguity. *TIP*, 26(6):2825–2838, 2017.

[Geng and Hou,] X. Geng and P. Hou. Pre-release prediction of crowd opinion on movies by label distribution learning. In *AAAI*, pages 3511–3517, Buenos Aires, Argentina.

[Geng and Ling, 2017] X. Geng and M.G. Ling. Soft video parsing by label distribution learning. In *AAAI*, pages 1331–1337, San Francisco, CA, 2017.

[Geng and Luo, 2014] X. Geng and L.-R. Luo. Multilabel ranking with inconsistent rankers. In *CVPR*, pages 3742–3747, Columbus, OH, 2014.

[Geng *et al.*, 2010] X. Geng, K. Smith-Miles, and Z.-H. Zhou. Facial age estimation by learning from label distributions. In *AAAI*, Atlanta, GA, 2010.

[Geng *et al.*, 2013] X. Geng, C. Yin, and Z.-H. Zhou. Facial age estimation by learning from label distributions. *TPAMI*, 35(10):2401–2412, 2013.

[Geng, 2016] X. Geng. Label distribution learning. *TKDE*, 28(7):1734–1748, 2016.

[Hardoon *et al.*, 2003] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neurocomputing*, 16(12):2639–2664, 2003.

[He and Niyogi, 2004] X.-F. He and P. Niyogi. Locality preserving projections. In *NIPS*, pages 153–160, Vancouver, Canada, 2004.

[He *et al.*, 2005] X.-F. He, D. Cai, S.-C. Yan, and H.-J. Zhang. Neighborhood preserving embedding. In *ICCV*, pages 1208–1213, Beijing, China, 2005.

[Hsu *et al.*, 2009] D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, pages 772–780, Lake Tahoe, NV, 2009.

[Jolliffe, 1986] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[Lin *et al.*, 2014] Z.-J. Lin, G-.G. Ding, M.-Q. Hu, and J.-M. Wang. Multi-label classification via feature-aware implicit label space encoding. In *ICML*, pages 325–333, Beijing, China, 2014.

[Lyons *et al.*, 1998] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In *FGR*, pages 200–205, Nara, Japan, 1998.

[Ren and Geng,] Y. Ren and X. Geng. Sense beauty by label distribution learning. In *IJCAI*, Melbourne, Australia.

[Shen *et al.*, 2017] W. Shen, K. Zhao, Y.-L. Guo, and A. Yuille. Label distribution learning forests. In *NIPS*, Long Beach, CA, 2017.

[Tai and Lin, 2012] F. Tai and H.-T. Lin. Multilabel classification with principal label space transformation. *Neurocomputing*, 24(9):2508–2542, 2012.

[Yeh *et al.*, 2017] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang. Learning deep latent space for multi-label classification. In *AAAI*, pages 2838–2844, San Francisco, CA, 2017.

[Zeng *et al.*, 2017] J. Zeng, Y. Liu, B. Leng, Z. Xiong, and Y. M. Cheung. Dimensionality reduction in multiple ordinal regression. *TNNLS*, (99):1–14, 2017.

[Zhang and Schneider, 2012] Y. Zhang and J. Schneider. Maximum margin output coding. In *ICML*, pages 1575–1582, Edinburgh, UK, 2012.

[Zhang and Zhou, 2014] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *TKDE*, 26(8):1819–1837, 2014.

[Zhang *et al.*, 2015] Z.-X. Zhang, M. Wang, and X. Geng. Crowd counting in public video surveillance by label distribution learning. *Neurocomputing*, pages 151–163, 2015.

[Zhou *et al.*, 2015] Y. Zhou, H. Xue, and X. Geng. Emotion distribution recognition from facial expressions. In *ACMM*, pages 1247–1250, Brisbane, Australia, 2015.