Simultaneous Nonlinear Label-Instance Embedding for Multi-label Classification

Keigo Kimura(✉), Mineichi Kudo, and Lu Sun

Graduate School of Information Science and Technology, Hokkaido University,
Sapporo 060-0814, Japan
{kkimura,mine,sunlu}@main.hokudai.ac.jp

Abstract. In this paper, unlike previous many linear embedding methods, we propose a non-linear embedding method for multi-label classification. The algorithm embeds both instances and labels into the same space, reflecting label-instance relationship, label-label relationship and instance-instance relationship as faithfully as possible, simultaneously. Such an embedding into two-dimensional space is useful for simultaneous visualization of instances and labels. In addition linear and nonlinear mapping methods of a testing instance are also proposed for multi-label classification. The experiments on thirteen benchmark datasets showed that the proposed algorithm can deal with better small-scale problems, especially in the number of instances, compared with the state-of-the-art algorithms.

Keywords: Multi-label classification · Nonlinear embedding · Visualization

1 Introduction

Multi-Label Classification (MLC), which allows an instance to have more than one label at the same time, has been recently received a surge of interests in a variety of fields and applications [10,15]. The main task of MLC is to learn the relationship between a F-dimensional feature vector $\boldsymbol{x}$ and an L-dimensional binary vector $\boldsymbol{y}$ from N training instances $\{(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)})\}$, and to predict a binary vector $\hat{\boldsymbol{y}} \in \{0,1\}^L$ for a test instance $\boldsymbol{x} \in \mathbb{R}^F$. To simplify the notation, we use a matrix $\mathbf{X} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}]^T \in \mathbb{R}^{N \times F}$ and a matrix $\mathbf{Y} = [\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(N)}]^T \in \{0,1\}^{N \times L}$ for expressing the training set.

A key of learning in MLC is how to utilize dependency between labels [10]. However, an excessive treatment of label dependency causes over-learning and brings larger complexity, sometimes, even intractable. Thus, many algorithms have been proposed to model the label dependency efficiently and effectively. Embedding is one of such methods for MLC. This type of methods utilizes label dependency through dimension reduction. The label dependency is explicitly realized by reducing the dimension of the label space from L to K ($\ll L$). Embedding methods in general learn relationships instances in F-dimensional

A. Robles-Kelly et al. (Eds.): S+SSPR 2016, LNCS 10029, pp. 15–25, 2016.
DOI: 10.1007/978-3-319-49055-7_2

space and latent labels in K-dimensional space, then, linearly transform the relationship to those in F-dimensional and real labels in L-dimensional space [4–6,8,12,16].

In this paper, we propose a novel method of a nonlinear embedding. Usually, either a set of labels or a set of instances is embedded [4–6,8,16], but in our method, both are embedded in the same time. We realize a mapping into a low-dimensional Euclidean space keeping three kinds of relationships between instance-instance, label-label and label-instance as faithfully as possible. In addition, for classification, a linear and a non-linear mappings of a testing instance are realized.

2 The Proposed Embedding

2.1 Objective Function

In contrast to traditional embedding methods, we explicitly embed both labels and instances into the same K-dimensional space ($K < F$) while preserving the relationships among labels and instances.[1] To preserve such relationships, we use a manifold learning method called Laplacian eigen map [1]. It keeps the distance or the degree of similarity between any pair of points or objects even in a low-dimensional space. For example, given similarity measure $\mathbf{W}_{ij}$ between two objects indexed by i and j, we find $\boldsymbol{z}^{(i)}$ and $\boldsymbol{z}^{(j)}$ in $\mathbb{R}^K$ so as to minimize $\sum_{i,j} \mathbf{W}_{ij} \|\boldsymbol{z}^{(i)} - \boldsymbol{z}^{(j)}\|_2^2$ under an appropriate constraint for scaling.

Now, we consider to embed both instances and labels at once. Let $\boldsymbol{g}^{(i)} \in \mathbb{R}^K$ be the low-dimensional representation of ith instance $\boldsymbol{x}^{(i)}$ on the embedding space and $\boldsymbol{h}^{(l)} \in \mathbb{R}^K$ be the representation of lth label on the same space as well. In this embedding, we consider three types of relationships: instance-label, instance-instance and label-label relationships. In this work, we quantify the above relationships by focusing on their localities. In more detail, we realize a mapping to preserve the following three kinds of properties in the training set:

1. Instance-Label (IL) relationship: Explicit relationship given by $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ $(i = 1, \ldots, N)$ should be kept in the embedding as closeness between $\boldsymbol{g}^{(i)}$ and $\boldsymbol{h}^{(l_i)}$ where l_i is one label of value one in $\boldsymbol{y}^{(i)}$
2. Label-Label (LL) relationship: Frequently co-occurred label pairs should be placed more closely in the embedded space $\mathbb{R}^K$.
3. Instance-Instance (II) relationship: Instances close in $\mathbb{R}^F$ should be placed closely even in $\mathbb{R}^K$.

Let us denote them by $\mathbf{W}^{(IL)} \in \mathbb{R}^{N \times L}$, $\mathbf{W}^{(LL)} \in \mathbb{R}^{L \times L}$ and $\mathbf{W}^{(II)} \in \mathbb{R}^{N \times N}$, respectively. Then our objective function of $\{\boldsymbol{g}^{(i)}, \boldsymbol{h}^{(l)}\}$ become, with α, β (>0),

[1] Note that labels do not have their representations explicitly before embedding.

$$\begin{aligned} O &= 2O_{IL} + \alpha O_{II} + \beta O_{LL} \\ &= 2\sum_{i,l} \mathbf{W}_{il}^{(IL)} \|\boldsymbol{g}^{(i)} - \boldsymbol{h}^{(l)}\|_2^2 + \alpha \sum_{ij} \mathbf{W}_{i,j}^{(II)} \|\boldsymbol{g}^{(i)} - \boldsymbol{g}^{(j)}\|_2^2 \\ &\qquad + \beta \sum_{l,m} \mathbf{W}_{l,m}^{(LL)} \|\boldsymbol{h}^{(l)} - \boldsymbol{h}^{(m)}\|_2^2 \\ &= \sum_{s,t} \mathbf{W}_{st} \|\boldsymbol{e}^{(s)} - \boldsymbol{e}^{(t)}\|_2^2 \quad (s,t = 1,2,\ldots,(N+L)), \end{aligned} \tag{1}$$

where $\boldsymbol{e}^{(s)} = \boldsymbol{g}^{(s)}$ or $\boldsymbol{h}^{(s)}$, and $\mathbf{W}_{st} = \mathbf{W}_{st}^{(IL)}$, $\mathbf{W}_{st}^{(II)}$ or $\mathbf{W}_{st}^{(LL)}$ depending on the values of s and t. As their matrix representation, let us use $\mathbf{G} = [\boldsymbol{g}^{(1)},\ldots,\boldsymbol{g}^{(N)}]^T \in \mathbb{R}^{N\times K}$ and $\mathbf{H} = [\boldsymbol{h}^{(1)},\ldots,\boldsymbol{h}^{(L)}]^T \in \mathbb{R}^{L\times K}$. Then using

$$\mathbf{W} = \begin{pmatrix} \alpha\mathbf{W}^{(II)} & \mathbf{W}^{(IL)} \\ \underbrace{\mathbf{W}^{(IL)T}}_{N} & \underbrace{\beta\mathbf{W}^{(LL)}}_{L} \end{pmatrix} \text{ and } \mathbf{E} = \underbrace{\begin{pmatrix} \mathbf{G} \\ \mathbf{H} \end{pmatrix}}_{K} \begin{matrix} \}N \\ \}L \end{matrix},$$

our objective function is rewritten as

$$O = \sum_{s,t} \mathbf{W}_{st} \|\boldsymbol{e}^{(s)} - \boldsymbol{e}^{(t)}\|_2^2 = 2\mathrm{Tr}(\mathbf{E}^T\mathbf{L}\mathbf{E}), \quad \text{s.t. } \mathbf{E}^T\mathbf{D}\mathbf{E} = \mathbf{I} \tag{2}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D}$ is a diagonal matrix with elements $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ [1]. The constraint $\mathbf{E}^T\mathbf{D}\mathbf{E} = \mathbf{I}$ is imposed to remove an arbitrary scaling factor in the embedding. This formulation is that of the Laplacian eigen map. Next, let us explain how to determine the similarity matrix $\mathbf{W}$.

Instance-Label Relationship: For the instance-label relationship $\mathbf{W}^{(IL)}$, we use $\mathbf{W}^{(IL)} = \mathbf{Y}$. In this case, $\mathbf{W}^{(IL)}$ has elements of zero or one. The corresponding objective function of Instance-Label relationship becomes:

$$O_{IL} = \min_{\boldsymbol{g}^{(i)},\boldsymbol{h}^{(l)}} \sum_{\substack{i=1,\ldots,N \\ l=1,\ldots,L}} \mathbf{W}_{il}^{(IL)} \|\boldsymbol{g}^{(i)} - \boldsymbol{h}^{(l)}\|_2^2,$$

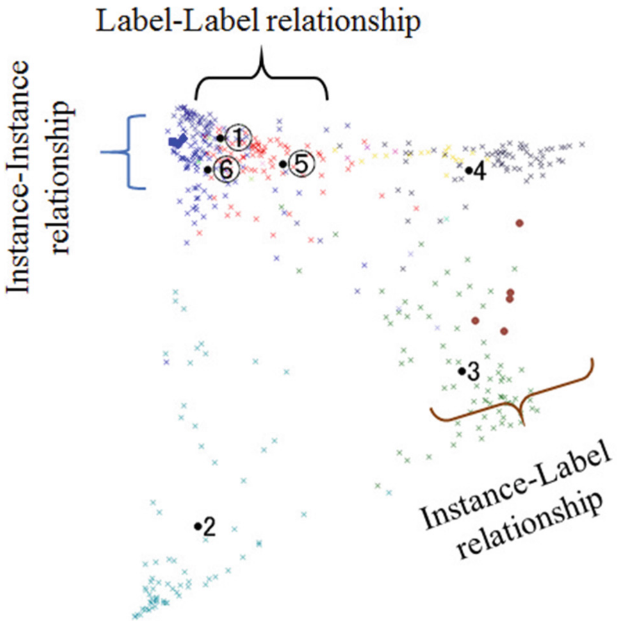
where $\mathbf{W}_{il}^{(IL)} = \mathbf{Y}_{il} \in \{0,1\}$.

Instance-Instance Relationship: We use the symmetric k-nearest neighbor relation in $\mathbb{R}^F$ for constructing $\mathbf{W}^{(II)}$ as seen in [3]. Thus, our second objective function becomes

$$O_{II} = \min_{\boldsymbol{g}^{(i)},\boldsymbol{g}^{(j)}} \sum_{i,j=1,\ldots,N} \mathbf{W}_{ij}^{(II)} \|\boldsymbol{g}^{(i)} - \boldsymbol{g}^{(j)}\|_2^2,$$

where

$$\mathbf{W}_{ij}^{(II)} = \begin{cases} 1 & (i \in \mathcal{N}_k(\boldsymbol{x}^{(j)}) \vee j \in \mathcal{N}_k(\boldsymbol{x}^{(i)})), \\ 0 & (\text{otherwise}), \end{cases}$$



Colors	Label Subset	Freq.
×	{6}	405
×	{5}	405
×	{5,6}	1
×	{4}	327
×	{4,6}	6
×	{4,5}	75
×	{3}	360
×	{3,5}	13
●	{3,4}	23
×	{3,4,5}	1
×	{2}	364
×	{1}	369
×	{1,6}	19
×	{1,5}	38
×	{1,4}	1

Fig. 1. The result of the proposed embedding in *Scene* dataset. Only 20 % of instances are displayed. The numbers indicate the labels $1, \ldots, 6$, and small crosses show the instances. (Color figure online)

where $\mathcal{N}_k(\boldsymbol{x}^{(i)})$ denotes the index set of k nearest neighbors of the ith instance. It is worth noting that we can construct $\mathbf{W}^{(II)}$ on the basis of the similarity between $\boldsymbol{y}^{(i)}$ and $\boldsymbol{y}^{(j)}$ as seen in [3] instead of that between $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ above.

Label-Label Relationship: We construct $\mathbf{W}^{(LL)}$ in such a way that $\mathbf{W}^{(LL)}_{lm}$ takes a large positive value when labels l and m co-occur frequently in $\mathbf{Y}$, otherwise a small positive value. We also use the symmetric k-nearest neighbor relation in the frequency. The corresponding third objective function becomes

$$O_{LL} = \min_{\boldsymbol{h}^{(l)},\boldsymbol{h}^{(m)}} \sum_{l,m=1,\ldots L} \mathbf{W}^{(LL)}_{lm} \|\boldsymbol{h}^{(l)} - \boldsymbol{h}^{(m)}\|_2^2,$$

where

$$\mathbf{W}^{(LL)}_{lm} = \begin{cases} 1 & \text{(if } l \text{ is one of top-}k \text{ co-occurrence labels of } m \text{ and vice versa),} \\ 0 & \text{(otherwise).} \end{cases}$$

Note that $\mathbf{W}^{(LL)}$ is symmetric as well as $\mathbf{W}^{(II)}$. The symmetricity of those guarantees the existence of a solution in (2).

The solution of (2) is obtained by solving the following generalized eigen problem:

$$\mathbf{LE} = \lambda \mathbf{DE}. \tag{3}$$

Algorithm 1. MLLEM (Training)

```
Input: Label matrix $\mathbf{Y}$,Feature matrix $\mathbf{X}$, The number of dimension $K$, wighting parameters $\alpha$ and $\beta$;
Output: $K$-dimensional representation of labels $\mathbf{H}$ and instances $\mathbf{G}$;
Construct $\mathbf{W}^{(II)}$ for instances (Section 2);
Construct $\mathbf{W}^{(LL)}$ for labels (Section 2);
$\mathbf{W}^{(IL)} = \mathbf{Y}$;
$\mathbf{W} = \begin{bmatrix} \alpha\mathbf{W}^{(II)} & \mathbf{W}^{(IL)} \\ \mathbf{W}^{(IL)T} & \beta\mathbf{W}^{(LL)} \end{bmatrix}$;
Solve the generalized eigen problem $\mathbf{LE} = \lambda\mathbf{DE}$ where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ and obtain the bottom $K$ eigen vectors excluding an eigen vector with zero eigen value;
$\begin{bmatrix} \mathbf{G} \\ \mathbf{H} \end{bmatrix} = \mathbf{E}$;
```

Hence, the optimal solution $\mathbf{E}$ of the objective function is the bottom K eigenvectors excluding an eigenvector with zero eigenvalue [1].

An example of this embedding is shown in Fig. 1. This is the result of mapping for *Scene* dataset [11] where $N = 2407$, $F = 294$, $L = 6$ and $K = 2$. In Fig. 1, we can see that the instance-label, instance-instance and label-label relations are fairly preserved. First, for instance-label relationship, four instances that share a label subset $\{3, 4\}$ (large brown dots) are mapped between labels 3 and 4. Second, for label-label relationship, highly co-occurred labels 1, 5 and 6 are closely mapped (highlighted by a circle). Finally, for instance-instance relationship, an instance and its k nearest neighbors ($k = 2$) in the original F-dimensional space (a blue square and 2 blue diamonds) are closely placed.

2.2 Embedding Test Instances

For assigning labels for a testing instance, we need to embed it into the same low-dimensional space constructed from the training instances with multiple labels. Unfortunately above embedding is not functionally realized, we do not have an explicit way of mapping. Therefore, we propose two different ways of a linear mapping and a nonlinear mapping.

In the linear mapping, we simulate the nonlinear mapping from $\mathbf{X}$ to $\mathbf{G}$ (the former part of $\mathbf{E}$) by a linear mapping $\mathbf{V}$ so as to $\mathbf{G} \simeq \hat{\mathbf{G}} = \mathbf{XV}$. We use *Ridge regression* to find such a $\mathbf{V}$:

$$\min_{\mathbf{V}} \|\mathbf{XV} - \mathbf{G}\|_2^2 + \lambda\|\mathbf{V}\|_2^2.$$

where λ is a parameter. A test instance $\boldsymbol{x}$ is mapped to $\boldsymbol{g}$ such as $\boldsymbol{g} = \boldsymbol{x}^T\mathbf{V}$.

Algorithm 2. MLLEM (Testing)

```
Input: Test instance $\boldsymbol{x}$, Feature matrix $\mathbf{X}$, $K$-dimensional representation of labels $\mathbf{H}$ and instances $\mathbf{G}$;
Output: Predicted multi-label (score) $\hat{\boldsymbol{y}} \in \mathbb{R}^L$;
{Linear Embedding}
Minimize $\|\mathbf{XV} - \mathbf{G}\|_2^2 + \lambda\|\mathbf{V}\|_2^2$ in $\mathbf{V}$;
Embed test instance by $\boldsymbol{g} = \boldsymbol{x}^T\mathbf{V}$;
{Nonlinear Embedding}
Find k-nn $\mathcal{N}_{k(t)}$ of test instance $\boldsymbol{x}$ in training set $\mathbf{X}$;
$\boldsymbol{g} = \underset{\boldsymbol{g}'}{\mathrm{argmin}} \frac{1}{k} \sum_{i \in \mathcal{N}_k(t)} \|\boldsymbol{g}' - \boldsymbol{g}^{(i)}\|_2^2$;
{After Linear Embedding or Nonlinear Embedding}
Calculate the score for each label $\hat{\boldsymbol{y}}_i = \|\boldsymbol{g} - \boldsymbol{h}^{(i)}\|_2^2$;
```

In the nonlinear mapping, we use again the k-nearest neighbor relation to the testing instance $\boldsymbol{x}$. We map $\boldsymbol{x}$ into $\boldsymbol{g}$ by the average point of its k-nearest neighbors in the training instances.

$$\boldsymbol{g} = \underset{\boldsymbol{g}'}{\mathrm{argmin}} \frac{1}{K} \sum_{i \in \mathcal{N}_k(\boldsymbol{x})} \|\boldsymbol{g}' - \boldsymbol{g}^{(i)}\|_2^2 = \frac{1}{K} \sum_{i \in \mathcal{N}_k(\boldsymbol{x})} \boldsymbol{g}^{(i)}.$$

Since the objective function (2) is solved by Laplacian Eigen Map [1], we name the proposed method **M**ulti-**L**abel classification using **L**aplacian **E**igen **M**ap (shortly, **MLLEM**). The combined pseudo-code of **MLLEM-L** (for linear mapping of a testing instance) and **MLLEM-NL** (for nonlinear mapping of a test instance) is described in Algorithms 1 and 2.

2.3 Computational Complexity

The training procedure of the proposed algorithm (Algorithm 1) can be divided into two parts. The first part constructs k-nn graphs for both labels and instances (Step 3 and Step 4), in $O(NL^2)$ for labels and in $O(FN^2)$ for instances, respectively. The second part solves the generalized eigen problem (Step 6). This part takes $O((N + L)^3)$. However, it is known that this complexity can be largely reduced when the matrix $\mathbf{W}$ is sparse and only a small number K of eigen vectors are necessary [9]. Therefore, the complexity of the proposed algorithm can be estimated as $O(NL^2 + FN^2)$. This complexity is the same to those of almost all embedding methods including the compared methods on the experiments.

In the testing phase, the linear embedding needs $O(F^2N)$ for the ridge regression. In contrast, nonlinear embedding needs only $O(FN)$ for each test instance that is faster than linear embedding.

3 Related Work

Label embedding methods for MLC are employed to utilize label-dependency via the low-rank structure of an embedding space. Recently, several methods based on traditional factorizations [4,6,8] and based on regressions with various loss functions [12,13] have been proposed. Canonical Correlation Analysis based method [16] is also one of them. This method conceptually embeds both instance and labels at the same time like the proposed **MLLEM** does. However, it conducts only one-side embedding in the actual classification process. This is because the linear regression after embedding includes the other-side embedding. Although all methods utilizes low-rank structure and succeeded to improve classification accuracy, they are limited to linear transformation.[2] In contrast to these methods, our **MLLEM** utilizes label dependency in a nonlinear way so that it is more flexible for mapping. On the other hand, we have to be careful for overfitting when we use nonlinear mappings. In **MLLEM**, the nonlinear mappings rely only on the similarity measures $\mathbf{W}^{(IL)}$, $\mathbf{W}^{(II)}$ and $\mathbf{W}^{(LL)}$. Therefore, overfitting is limited to some extent.

Bhatia *et al.* proposed linear embedding method for instances [3]. In their embedding, only instance locality on the label space is considered and ML-KNN [14] is conducted on the low-dimensional space. In the sense of using locality, the proposed **MLLEM** is close to theirs, but the proposed **MLLEM** is different from their approach in the sense that label-instance relationship, label-label relationship and instance-instance relationship are all taken into consideration at the same time.

4 Experiments

4.1 Setting

We conducted experiments on thirteen benchmark datasets [11] (Table 1). Each dataset was separated into 67 % of training instances and 33 % of test instances at random. On large datasets (*i.e. delicious*, *bookmarks* and *mediamill*), we sampled randomly 5000 instances (4000 samples for training and 1000 samples for testing) according to [6].

Since all embedding methods return scores of labels, not a label subset, we used Area Under ROC-Curve (AUC) and top-k precision to evaluate the results [13].[3] AUC is a popular criteria to evaluate the ranking of all labels. We used ROC-curve between true-positive rate and false-positive rate for AUC and Top-1 precision [3].[4]

We compared the following three state-of-the-art embedding methods to ours:

1. Low-rank Empirical risk minimization for MLC (**LEML**) [13]
2. Feature-aware Implicit Encoding(**FaIE**) [6]

[2] Several methods can utilize kernel regressions instead a liner regression, however, after regression, they linearly transform the latent labels into the original labels. This means that the way of utilizing label dependency is still limited to be linear.

[3] All embedding methods use a threshold to obtain a label subset.

[4] We only show the result of top-1 precision since the ordering was not changed in top-3 or top-5.

Table 1. Dataset used in the experiment. F_{nnz} and L_{nnz} are the average of number non-zero features and labels among instances in the corresponding set, respectively.

Dataset	F	L	Training set			Test set		
			N	F_{nnz}	L_{nnz}	N	F_{nnz}	L_{nnz}
CAL500	68	174	335	67.84	26.09	167	68	25.94
enron	1001	53	387	79.17	3.37	192	93.30	3.33
emotions	72	6	396	71.79	1.88	197	71.71	1.83
birds	258	19	431	158.00	1.10	214	158.40	1.10
genbase	1186	27	442	2.52	1.20	220	2.55	1.34
medical	1449	45	653	13.04	1.24	325	14.10	1.25
scene	294	6	1606	290.54	1.07	801	290.78	1.07
yeast	103	14	1613	103.00	4.21	804	102.99	4.27
corel5k	499	374	3335	8.28	3.53	1665	8.23	3.49
bibtex	1836	159	4933	69.13	2.41	2462	67.71	2.38
delicious	500	983	4000	18.08	19.02	1000	18.71	19.00
bookmarks	2150	208	4000	125.27	2.02	1000	126.17	2.03
mediamill	120	101	4000	120.00	4.38	1000	120.00	4.36

3. Sparse Local Embedding for Extreme Multi-label Classification (**SLEEC**) [3]
4. Proposal with linear embedding (**MLLEM-L**) and nonlinear embedding (**MLLEM-NL**)

The proposed **MLLEM** has five parameters, the number of nearest neighbors $k^{(I)}$ for instances, $k^{(L)}$ for labels, weighting parameters α for $\mathbf{W}^{(II)}$ and β for $\mathbf{W}^{(LL)}$ and the dimension K of the embedding space. On all datasets, we used $K = 20$. All the other parameters were tuned by five-cross validation on training dataset. The code is available at the authors' web site.[5] For **CPLST** and **FaIE**, we set their numbers of dimension for labels space to the 80 % of their numbers of labels following the setting in [3]. The other parameters were tuned as well. We used the implementations provided by the authors [13][6], [6][7]. For **SLEEC**, we set the number of dimension as $K = 100$ following the their setting [3]. We tuned best the number of k-nn and the number of neighborhoods for ML-KNN after embedding. The other parameters such as smoothing parameter in their regression is set to the default setting they used. We used the implementations provided by the authors too.

[5] https://dl.dropboxusercontent.com/u/97469461/MLLEM.zip.

[6] http://www.cs.utexas.edu/~rofuyu/exp-codes/leml-icml14-exp.zip.

[7] The code is available at the authors' site (https://sites.google.com/site/linzijia72/).

4.2 Results

Table 2 shows the averaged AUC and Top-1 precision on all thirteen benchmark datasets (the larger, the better). **SLEEC** was the best in AUC. The proposed **MLLEM** follows. In Top-1 **MLLEM** was the best, especially in relatively small-scale datasets. This difference is possibly explained from the difference between objective functions of **MLLEM**. **MLLEM** ignores the distance between two objects which do not have a local similarity relation to each other. Therefore, occasionally, such two objects are placed closely in the embedding space in spite that they are not similar. This affects the result measured by AUC which takes overall ranking into evaluation. On the other hand, on top-k labels, such an indicated bad effect problem seldom occurs.

Table 2. Results

Dataset	Averaged-AUC					Top-1 precision				
	LEML	FaIE	SLEEC	MLLEM		LEML	FaIE	SLEEC	MLLEM	
				L	NL				L	NL
CAL500	.7592	.7738	.8187	.8206	**.8211**	.7365	.7605	.8503	**.8623**	.8563
enron	.7929	.7710	.8748	.8857	**.8902**	.6458	.5260	.5677	.6146	**.7292**
emotions	.8073	.8106	.8362	**.8372**	.7565	7107	.7107	.7107	**.7563**	.6599
birds	.7604	.7240	.7394	**.8265**	.7240	.2850	.3271	.2617	**.3645**	.2196
genbase	.9944	.9950	.9950	**.9985**	.9979	1.000	1.000	1.000	1.000	1.000
medical	.9620	.9502	**.9736**	.9697	.9342	.8246	.8245	.8246	**.8646**	.7446
scene	.8852	.8683	**.9134**	.9114	.9074	.6667	.6442	.7378	.7466	**.7528**
yeast	.6318	.8188	**.8299**	.8288	.8247	.6457	.7550	.7836	**.7886**	.7724
corel5k	.7805	.7862	**.8906**	.8700	.8717	.3495	.3610	**.3928**	.3538	.2505
bibtex	.8895	.8868	**.9480**	.9066	.7563	**.6409**	.6186	.6255	.5902	.3298
delicious	.8097	.8768	**.8921**	.8463	.8416	.5940	.6274	**.6830**	.6291	.6253
bookmarks	.8000	.7551	**.8662**	.8023	.7700	.3370	.3040	**.4110**	.3020	.2950
mediamill	.9451	.9425	**.9475**	.9286	.9283	.8380	**.8440**	.8210	.8010	.7801

Table 3. Training time ($K = 20$).

Dataset	LEML	FaIE	SLEEC	MLLEM
CAL500	2.62	0.36	0.48	0.61
bibtex	3.62	5.51	9.44	12.72
enron	3.53	0.39	0.49	0.43
genbase	1.00	0.32	0.36	0.66
medical	2.78	0.38	1.21	0.96
corel5k	6.24	1.46	5.55	11.62
delicious	3.53	2.91	10.54	19.15
bookmarks	7.19	5.56	7.86	8.40
mediamill	8.14	3.92	8.69	7.66

MLLEM was superior to **SLEEC** when the number N of instances is relatively small even in AUC. This is probably because **SLEEC** considers only locality of instances (instance-instance relationship). When the number of available instances is limited, instance-instance relationship is not enough to capture the relationship between features and labels. From this viewpoint, **MLLEM** is the best choice for small- to medium-sample size problem (Table 3).

5 Discussion

Since the proposed **MLLEM** uses Laplacian eigen map for the nonlinear embedding, there are several ways to increase the scalability of **MLLEM** such as an incremental method [2], Nyström approximation or column sampling [9] and efficient k-NN constructors [7]. Note that the framework used in **MLLEM** is very general. It comes from the freedom of choice of matrix $\mathbf{W}$ (*e.g.*, using a heat kernel). It is also able to handle categorical features. This generalization is not shared with **SLEEC** [3].

6 Conclusion

In this paper, we have proposed an embedding based approach for multi-label classification. The proposed algorithm takes into consideration three relationships: label-instance relationship, label-label relationship and instance-instance relationship, and realized a nonlinear mapping. All these three relationships are preserved in the embedded low-dimensional space as the closeness between instances and individual labels. We have shown that the algorithm is useful to visualize instances and labels at the same time, which helps us to understand a given multi-label problem, especially, how strongly those labels are related to each other. Linear and nonlinear mapping have been also proposed for classification. On experiments, the proposed algorithm outperformed the other state-of-the-art methods in small-scale datasets in sample number.

Acknowledgment. We would like to thank Dr. Kush Bhatia for providing the code of SLEEC and large-scale datasets. This work was partially supported by JSPS KAKENHI Grant Number 14J01495 and 15H02719.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
2. Bengio, Y., Paiement, J.F., Vincent, P., Delalleau, O., Le Roux, N., Ouimet, M.: Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. Adv. Neural Inf. Process. Syst. **16**, 177–184 (2004)
3. Bhatia, K., Jain, H., Kar, P., Varma, M., Jain, P.: Sparse local embeddings for extreme multi-label classification. Adv. Neural Inf. Process. Syst. **28**, 730–738 (2015)

4. Chen, Y.N., Lin, H.T.: Feature-aware label space dimension reduction for multi-label classification. In: Advances in Neural Information Processing Systems, pp. 1529–1537 (2012)
5. Hsu, D., Kakade, S., Langford, J., Zhang, T.: Multi-label prediction via compressed sensing. Adv. Neural Inf. Process. Syst. **22**, 772–780 (2009)
6. Lin, Z., Ding, G., Hu, M., Wang, J.: Multi-label classification via feature-aware implicit label space encoding. In: Proceedings of the 31st International Conference on Machine Learning, pp. 325–333 (2014)
7. Liu, T., Moore, A.W., Yang, K., Gray, A.G.: An investigation of practical approximate nearest neighbor algorithms. In: Advances in Neural Information Processing Systems, pp. 825–832 (2004)
8. Tai, F., Lin, H.T.: Multilabel classification with principal label space transformation. Neural Comput. **24**(9), 2508–2542 (2012)
9. Talwalkar, A., Kumar, S., Rowley, H.: Large-scale manifold learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
10. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. Department of Informatics, Aristotle University of Thessaloniki, Greece (2006)
11. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: a Java library for multi-label learning. J. Mach. Learn. Res. **12**, 2411–2414 (2011)
12. Weston, J., Bengio, S., Usunier, N.: Wsabie: scaling up to large vocabulary image annotation. IJCAI **11**, 2764–2770 (2011)
13. Yu, H.f., Jain, P., Kar, P., Dhillon, I.: Large-scale multi-label learning with missing labels. In: Proceedings of the 31st International Conference on Machine Learning, pp. 593–601 (2014)
14. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)
15. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014)
16. Zhang, Y., Schneider, J.G.: Multi-label output codes using canonical correlation analysis. In: International Conference on Artificial Intelligence and Statistics, pp. 873–882 (2011)