Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Multi-Label Manifold Learning

Peng Hou, Xin Geng*, Min-Ling Zhang
MOE Key Laboratory of Computer Network and Information Integration,
School of Computer Science and Engineering,
Southeast University, Nanjing 210096, China
{hpeng, xgeng, zhangml}@seu.edu.cn

Abstract

This paper gives an attempt to explore the manifold in the label space for multi-label learning. Traditional label space is logical, where no manifold exists. In order to study the label manifold, the label space should be extended to a Euclidean space. However, the label manifold is not explicitly available from the training examples. Fortunately, according to the *smoothness assumption* that *the points close to each other are more likely to share a label*, the local topological structure can be shared between the feature manifold and the label manifold. Based on this, we propose a novel method called ML^2, i.e., *Multi-Label Manifold Learning*, to reconstruct and exploit the label manifold. To our best knowledge, it is one of the first attempts to explore the manifold in the label space in multi-label learning. Extensive experiments show that the performance of multi-label learning can be improved significantly with the label manifold.

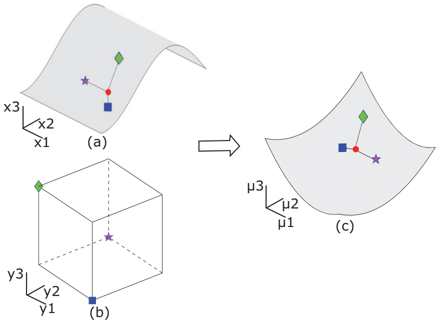


Figure 1: Transformation of the local topological structure from the feature space to the label space. (a) The manifold in the feature space; (b) The traditional logical label space; (c) The manifold in the Euclidean label space.

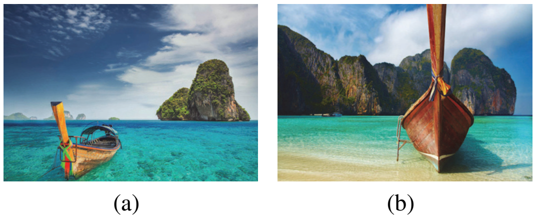
1 Introduction

In multi-label learning, there are multiple labels associated to the same instance simultaneously (Tsoumakas, Katakis, and Vlahavas 2009; Zhang and Zhou 2014). Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional feature space and $\mathcal{Y} = \{y_1, \ldots, y_q\}$ be the label set with q possible labels. Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leq i \leq n\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the feature vector and $\boldsymbol{y}_i \in \{0, 1\}^q$ is the label vector, the task of traditional multi-label learning is to learn a predictor which maps from the space of feature vectors to the space of label vectors. Each element of the label vector $\boldsymbol{y}_i$ is a logical indicator of whether the corresponding label is relevant or irrelevant to the instance $\boldsymbol{x}_i$. During the past decade, multi-label learning has been applied successfully to learn from the data with rich semantics, such as text (Rubin et al. 2012; Yang et al. 2009), image (Cabral et al. 2011; Wang, Huang, and Ding 2009), audio (Lo et al. 2011; Sanden and Zhang 2011), video (Wang et al. 2011), etc.

In this paper, we give an attempt to improve the multi-label learning performance with the manifold in the label space. To our best knowledge, it is one of the first attempts to explore the label manifold in multi-label learning. Traditional label space spanned by the label vector $\boldsymbol{y}_i$ is logical, and the element of $\boldsymbol{y}_i$ can be called *logical label*. In order to study the label manifold, the label space should be extended to a Euclidean space. Each dimension of the space still corresponds to one label in $\mathcal{Y}$, but the value is extended from logical to real. Such label is called *numerical label*, which carries more semantic information and can describe the instance more comprehensively than the *logical label*.

However, the label manifold is not explicitly available from the training examples. To reconstruct the label manifold, the key issue is the topological structure. Fortunately, there is one commonly adopted assumption by many machine learning methods called *smoothness assumption* (Zhu, Lafferty, and Rosenfeld 2005; Chapelle et al. 2006), which says that *the points close to each other are more likely to share a label*. With the extension from the logical label space to the Euclidean label space, we can naturally induce from the smoothness assumption that, the local topological structure can be transferred from the feature space to the label space. One example is shown in Fig. 1. The subfigure (a) shows the manifold in the feature space where the blue square, green rhombus and purple star points are the neighbors of the red point. The subfigure (b) shows the traditional logical label space where the vertex points represent the logical label vectors for the neighbors, respectively. The subfigure (c) shows the manifold in the Euclidean label space

[*] Corresponding author.



(a)

(b)

Figure 2: Two natural scene image examples which are both annotated with the labels *water*, *boat*, *mountain*, and *sky*.

where the local topological structure is transferred from the feature space. The transference is guided by the feature manifold (a) and the logical label vectors (b). Note that although the feature space and the label space share similar local topological structure, the global manifolds in these two spaces are generally different.

Based on the above assumption, we propose an efficient multi-label method called ML^2, i.e., *Muti-Label Manifold Learning*. The feature manifold is represented by a graph and approximated by the overlapped local linear neighborhood patches. The edge weights in each patch can be solved by a least square programming procedure. Then the label manifold is reconstructed with the transferred local topological structure from the feature manifold and the existing logical labels. The reconstruction can be achieved by a quadratic programming process. The sign of the numerical label is used to represent whether the label is relevant or irrelevant to the example. With the label manifold available, the ML^2 can effectively find a mapping from the feature manifold to the label manifold with a regression process.

The label manifold brings the following three advantages: 1. It helps to exploit the correlation among the labels via the transference of the topological structure from the feature space according to the smoothness assumption; 2. It extends the traditional logical label to the numerical label, which can describe the instance in greater details and thus brings more possibilities for performance improvement; 3. It helps to make more complicated decisions based on the numerical labels, because the value of the numerical label could be regarded as an indicator of the relative importance of the corresponding label.

In further detail, the difference of the relative importance revealed by the numerical label could be two-fold: 1. *within-instance label variance*, i.e., different labels relevant to the same instance may have different numerical values; 2. *between-instance label variance*, i.e., the same label relevant to different instances may also have different numerical values. An example is shown in Fig. 2. Image (a) and (b) are both annotated with the labels *water*, *boat*, *mountain*, and *sky*. Once extended to the numerical labels, for the *within-instance label variance*, the label *boat* should have larger value than the label *sky* in (b), because the former can describe the image more apparently than the latter. Similarly, for the *between-instance label variance*, the value of the label *boat* in (b) should be larger than the one in (a).

The rest of this paper is organized as follows. First, existing work related to our proposed approach is discussed in Section 2. The details of ML^2 are proposed in Section 3. After that, the results of comparative studies are reported in Section 4. Finally, conclusions are drawn in Section 5.

2 Related Work

Existing multi-label approaches can be roughly grouped into three categories based on the thought of *order of label correlations* (Zhang and Zhou 2014). The simplest ones are the first-order approaches which assume independence among class labels (Boutell et al. 2004; Zhang and Zhou 2007). Then the multi-label classification becomes a series of binary classification problems. On the contrary, second-order approaches consider the correlations between pairs of class labels (Elisseeff and Weston 2001; Frnkranz et al. 2008), and the high-order approaches consider the correlations among label subsets or all the class labels (Tsoumakas, Katakis, and Vlahavas 2011). For all of them, the common modeling strategy is to treat each label in a crisp manner, i.e., being either relevant or irrelevant to an instance. In contrast, ML^2 explores the manifold in the label space and treats the label as numerical. The label manifold contains more semantic information, which is beneficial for the learning process.

There have been some multi-label works which transform the logical label space to the Euclidean label space. For example, (Tai and Lin 2012) tries to reduce the computational effort by seeking the principle correlations between labels, especially for the data sets with large numbers of labels. The bases of the Euclidean space are the combinations of the logical label vectors. Another work (Sun, Ji, and Ye 2011) projects the feature space and the label space to a new space where the correlation between the projections of the two spaces are maximized. In both cases, the dimensionality of the label space is reduced. However, ML^2 differs from them without the dimensionality reduction. Besides, the former cases both produce a new space projected from the original label space, however, ML^2 extends the original logical label space to a Euclidean space where the meaning of each dimension still remains.

Another more related work is *Label Distribution Learning* (LDL) (Geng, Yin, and Zhou 2013), which is a new machine learning paradigm where each instance is annotated by a label distribution. The label distribution covers a certain number of labels, representing the degree to which each label describes the instance. Thus the value of each label is numerical. However, LDL requires the availability of the label distributions in the training set, which is not always satisfiable for the real applications. On the contrary, ML^2 can reconstruct the label manifold automatically from the logical multi-label data.

It is worthy to emphasize the difference between the manifold learning and ML^2. Manifold learning assumes that the data of interest actually lie on an embedded non-linear manifold within the higher-dimensional space. Thus manifold learning is mostly used for the dimensionality reduction and visualization. The three famous local approaches in manifold learning are *Locally Linear Embedding* (LLE) (Roweis and Saul 2000), *Laplacian Eigenmaps* (LE) (Belkin and Niyogi 2003) and *Locality Preserving Projection* (LPP)

(Niyogi 2004). The reconstruction process of the label manifold in ML^2 is similar to LLE. However, the relation between the feature manifold and the label manifold is not embedding or dimensionality reduction. They are in two different spaces that merely share the local topological structure according to the smoothness assumption.

Note that the local topological structure is transferred from the feature space to the label space in ML^2, but it is different from transfer learning (Pan and Yang 2010). Transfer learning deals with two problem spaces of different fields or distributions. The target of the transference mainly refers to the domain knowledge. However, ML^2 transfers the topological structure from the feature space to the label space.

3 The ML^2 Algorithm

As shown in Section 1, the training set of multi-label learning can be expressed as $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \le i \le n\}$. Given any instance $\boldsymbol{x}_i \in \mathbb{R}^d$ and the logical label vector $\boldsymbol{y}_i \in \{+1, -1\}^q$, we use $\boldsymbol{\mu}_i \in \mathbb{R}^q$ to denote the *numerical label vector*. Note that here we use -1 instead of 0 in the logical label vector to represent irrelevant to the example. As many graph based learning methods do, the topological structure can be represented by a graph $\mathcal{G} =< \mathcal{V}, \mathcal{E}, \boldsymbol{W} >$, where $\mathcal{V}$ is the vertex set, $\mathcal{E}$ is the edge set in which each edge e_i^j represents the relationship between the data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and $\boldsymbol{W}$ is the weight matrix with each element W_i^j representing the weight of the edge e_i^j.

According to the smoothness assumption, the topological structure of the feature space can be transferred to the numerical label space local by local. In order to keep the locality, we need to use the local neighborhood information of each point to construct $\mathcal{G}$. For computational convenience, we assume that each data point can be optimally reconstructed using a linear combination of its neighbors (Roweis and Saul 2000; Wang and Zhang 2008). Then the approximation of the feature manifold is to induce the minimization of

$$\mathcal{E}(\boldsymbol{W}) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \sum_{j \ne i} W_i^j \boldsymbol{x}_j\|^2, \tag{1}$$

where $W_i^j = 0$ unless $\boldsymbol{x}_j$ is one of $\boldsymbol{x}_i$'s K-nearest neighbors. Note that under most conditions $W_i^j \ne W_j^i$. Further for translation invariance, we constrain $\mathbf{1}^{\mathrm{T}} \boldsymbol{W}_i = 1$, where $\boldsymbol{W}_i = [W_i^1, \ldots, W_i^n]^{\mathrm{T}}$, and $\mathbf{1}$ is the vector of all ones.

Then the approximation can be solved by the following n standard least square programming problems

$$\begin{aligned} \min_{\boldsymbol{W}_i} \quad & \boldsymbol{W}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{W}_i \\ \text{s.t.} \quad & \mathbf{1}^{\mathrm{T}} \boldsymbol{W}_i = 1, \end{aligned} \tag{2}$$

where $\boldsymbol{G}_i$ is the local Gram matrix at point $\boldsymbol{x}_i$ with $G_i^{jk} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} (\boldsymbol{x}_i - \boldsymbol{x}_k)$.

With the transferred topological structure, the reconstruction of the label manifold can infer to the minimization of

$$\Phi(\boldsymbol{\mu}) = \sum_{i=1}^{n} \|\boldsymbol{\mu}_i - \sum_{j \ne i} W_i^j \boldsymbol{\mu}_j\|^2. \tag{3}$$

Note that we are now minimizing with respect to the numerical label vector $\boldsymbol{\mu}$ rather than $\boldsymbol{W}$.

Besides, we add a constraint that makes the sign of the numerical label represent whether the corresponding label is relevant or irrelevant to the example that

$$\forall 1 \le i \le n, 1 \le l \le q \qquad y_i^l \mu_i^l \ge \lambda, \tag{4}$$

where $\lambda > 0$. The optimization for (3) with constraint (4) is a constrained quadratic programming process, and it can be solved efficiently.

There are three advantages for the constraint (4): 1. It is convenient to judge whether a label is relevant or irrelevant to the example by the sign of it; 2. It guarantees that the relevant numerical labels are larger than the irrelevant ones; 3. The minimum of the relevant numerical labels will be equal to λ or the maximum of the irrelevant numerical labels will be equal to $-\lambda$. This makes the scale of the reconstructed numerical labels on the control.

The reconstructed numerical labels are real and the problem can not be treated as a classification but rather a regression problem. In the multi-label case, it is actually a multi-output regression problem. There have been some efficient algorithms proposed such as multi-output support vector regression (MSVR) (Prez-Cruz et al. 2002; Tuia et al. 2011; Chung et al. 2014), k-nearest neighbor regression (KNNR) (Burba, Ferraty, and Vieu 2009) and structured output-associative regression (SOAR) (Bo and Sminchisescu 2009). Here we propose a regressor based on the MSVR.

Similar to the MSVR, we generalize the 1-D SVR to solve the multi-dimensional case. In addition, our regressor not only concerns the distance between the predicted and the real values, but also the sign consistency of them. It leads to the minimization of

$$L(\boldsymbol{\Theta}, \boldsymbol{b}) = \frac{1}{2} \sum_{j=1}^{q} \|\boldsymbol{\theta}^j\|^2 + C_1 \sum_{i=1}^{n} L_1(r_i) + C_2 \sum_{i=1}^{n} \sum_{j=1}^{q} L_2(t_i^j), \tag{5}$$

where $r_i = \|\boldsymbol{e}_i\| = \sqrt{\boldsymbol{e}_i^{\mathrm{T}} \boldsymbol{e}_i}$, $\boldsymbol{e}_i = \boldsymbol{\mu}_i - \varphi(\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{\Theta} - \boldsymbol{b}$, $t_i^j = y_i^j (\varphi(\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{\theta}^j + b^j)$, $\boldsymbol{\Theta} = [\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^q]$, $\boldsymbol{b} = [b^1, ..., b^q]$, and $\varphi(\boldsymbol{x})$ is a nonlinear transformation of $\boldsymbol{x}$ to a higher-dimensional feature space $\mathbb{R}^{\mathcal{H}}$.

To consider all dimensions into a unique restriction and yield a single support vector for all dimensions, the L_1 loss is set as

$$L_1(r) = \begin{cases} 0 & r < \varepsilon \\ r^2 - 2r\varepsilon + \varepsilon^2 & r \ge \varepsilon. \end{cases} \tag{6}$$

This will create an insensitive zone determined by ε around the estimate, i.e., the loss of r less than ε will be ignored.

To make the signs of the numerical label and the logical label same as much as possible, the L_2 loss is set as

$$L_2(t) = -t\sigma(-t) = \begin{cases} 0, & t > 0 \\ -t, & t \le 0 \end{cases}, \tag{7}$$

where $\sigma(t)$ is an activation function where the value will be equal to 0 if t is negative, otherwise the value will be equal to 1. The meaning of Eq. (7) is that if the signs of the predicted numerical label and the logical label are different, there will be some positive loss, otherwise the loss will be zero.

Table 1: Characteristics of the bench mark multi-label data sets.

Data set	$\lvert\mathcal{S}\rvert$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$F(\mathcal{S})$	$LCard(\mathcal{S})$	$LDen(\mathcal{S})$	$DL(\mathcal{S})$	$PDL(\mathcal{S})$	Domain
cal500	502	68	174	numeric	26.044	0.150	502	1.000	audio
llog	1460	1004	75	nominal	1.180	0.016	286	0.196	text
enron	1702	1001	53	nominal	3.378	0.064	753	0.442	text
image	2000	294	5	numeric	1.236	0.247	20	0.010	images
scene	2407	294	6	numeric	1.074	0.179	15	0.006	images
yeast	2417	103	14	numeric	4.237	0.303	198	0.082	biology
slashdot	3782	1079	22	nominal	1.181	0.054	156	0.041	text
corel5k	5000	499	374	nominal	3.522	0.009	3175	0.635	images
rcv1-s1	6000	944	101	numeric	2.880	0.029	1028	0.171	text
rcv1-s2	6000	944	101	numeric	2.634	0.026	954	0.159	text
bibtex	7395	1836	159	nominal	2.402	0.015	2856	0.386	text
corel16k-s1	13766	500	153	nominal	2.859	0.019	4803	0.349	images
corel16k-s2	13761	500	164	nominal	2.882	0.018	4868	0.354	images
tmc2007	28696	981	22	nominal	2.158	0.098	1341	0.047	text

To minimize $L(\boldsymbol{\Theta}, \boldsymbol{b})$, we use an iterative quasi-Newton method called Iterative Re-Weighted Least Square (IRWLS) (Prez-Cruz et al. 2000). Firstly, $L_1(\boldsymbol{\Theta}, \boldsymbol{b})$ is approximated by its first order Taylor expansion at the solution of the current k-th iteration, denoted by $\boldsymbol{\Theta}^{(k)}$ and $\boldsymbol{b}^{(k)}$:

$$L_1'(r_i) = L_1(r_i^{(k)}) + \frac{dL_1(r)}{dr}\Bigg|_{r_i^{(k)}} \frac{(\boldsymbol{e}_i^{(k)})^{\mathrm{T}}}{r_i^{(k)}} \left(\boldsymbol{e}_i - \boldsymbol{e}_i^{(k)}\right), \tag{8}$$

where $\boldsymbol{e}_i^{(k)}$ and $r_i^{(k)}$ are calculated from $\boldsymbol{\Theta}^{(k)}$ and $\boldsymbol{b}^{(k)}$. Then a quadratic approximation is further constructed as

$$\begin{aligned} L_1''(r_i) =& L_1(r_i^{(k)}) + \frac{dL_1(r)}{dr}\Bigg|_{r_i^{(k)}} \frac{r_i^2 - (r_i^{(k)})^2}{2r_i^{(k)}} \\ =& \frac{1}{2} a_i r_i^2 + \tau, \end{aligned} \tag{9}$$

where

$$a_i = \frac{1}{r_i^{(k)}} \frac{dL_1(r)}{dr}\Bigg|_{r_i^{(k)}} = \begin{cases} 0 & r_i^{(k)} < \varepsilon, \\ \frac{2\left(r_i^{(k)} - \varepsilon\right)}{r_i^{(k)}} & r_i^{(k)} \geq \varepsilon, \end{cases} \tag{10}$$

and τ is a constant term that does not depend on either $\boldsymbol{\Theta}^{(k)}$ or $\boldsymbol{b}^{(k)}$. Combining Eq. (5), (7) and (9) can get

$$L''(\boldsymbol{\Theta}, \boldsymbol{b}) = \frac{1}{2}\sum_{j=1}^{c} \|\boldsymbol{\theta}^j\|^2 + \frac{1}{2} C_1 \sum_{i=1}^{n} a_i r_i^2 - C_2 \sum_{i=1}^{n}\sum_{j=1}^{q} t_i^j \sigma(-t_i^j) + \tau. \tag{11}$$

It is a piecewise quadratic problem whose optimum can be integrated as solving a system of linear equations for $j = 1, \ldots, q$:

$$\begin{bmatrix} C_1 \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{D}_a \boldsymbol{\Phi} + \boldsymbol{I} & C_1 \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{a} \\ C_1 \boldsymbol{a}^{\mathrm{T}} \boldsymbol{\Phi} & C_1 \boldsymbol{1}^{\mathrm{T}} \boldsymbol{a} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}^j \\ b^j \end{bmatrix} = \begin{bmatrix} C_1 \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{D}_a \boldsymbol{\mu}^j + C_2 \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{D}_j \boldsymbol{y}^j \\ C_1 \boldsymbol{a}^{\mathrm{T}} \boldsymbol{\mu}^j + C_2 (\boldsymbol{\sigma}^j)^{\mathrm{T}} \boldsymbol{y}^j \end{bmatrix}, \tag{12}$$

where $\boldsymbol{\Phi} = [\varphi(\boldsymbol{x}_1), ..., \varphi(\boldsymbol{x}_n)]^{\mathrm{T}}$, $\boldsymbol{a} = [a_1, ..., a_n]^{\mathrm{T}}$, $(\boldsymbol{D}_a)_i^k = a_i \delta_i^k$ (δ_i^k is the Kronecker's delta function), $(\boldsymbol{D}_j)_i^k = \sigma(-t_i^j)\delta_i^k$, $\boldsymbol{\sigma}^j = [\sigma(-t_1^j), \ldots, \sigma(-t_n^j)]^{\mathrm{T}}$, $\boldsymbol{y}^j = [y_1^j, \ldots, y_n^j]^{\mathrm{T}}$. Then, the direction of the optimal solution of Eq. (12) is used as the descending direction for the optimization of $L(\boldsymbol{\Theta}, \boldsymbol{b})$, and the solution for the next iteration ($\boldsymbol{\Theta}^{(k+1)}$ and $\boldsymbol{b}^{(k+1)}$) is obtained via a line search algorithm along this direction.

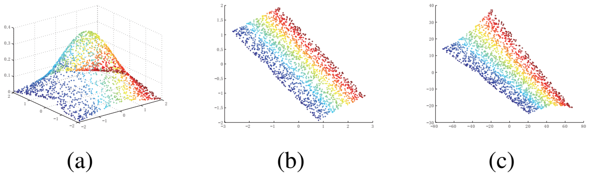


Figure 3: Experimental result on the toy data. (a) The toy data points in the feature space; (b) The real numerical label points; (c) The reconstructed numerical label points.

According to the representor's theorem (Smola and Schlkopf 1998), under fairly general conditions, a learning problem can be expressed as a linear combination of the training examples in the feature space, i.e., $\boldsymbol{\theta}^j = \sum_i \varphi(\boldsymbol{x}_i)\boldsymbol{\beta}^j = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\beta}^j$. If we replace this expression into Eq. (12), it will generate the inner product $< \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) >$, then the kernel trick can be applied. After that the line search algorithm can be expressed in terms of $\boldsymbol{\beta}^j$ and b^j.

4 Experiments

4.1 Experiment Configuration

Data Sets For comprehensive performance evaluation, we collect one toy data set and fourteen real data sets for experimental studies. The toy data is 3-dimensional, where the third dimension is calculated as the Gaussian distribution of the first two dimensions. The mean of the Gaussian distribution is 0, and the variance on each dimension is 1. The label space is 2-dimensional, where the numerical label vector is calculated by $\boldsymbol{\mu} = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{x}$, and

$$\boldsymbol{M} = \begin{bmatrix} 0.48 & 0.21 \\ 0.87 & -0.76 \\ -0.12 & 0.3 \end{bmatrix}. \tag{13}$$

Fig. 3(a) shows the toy data points in the feature space and (b) shows the real numerical label points. The two figures

Table 2: Predictive performance of each comparing algorithm (mean ± std. deviation) on the regular-scale data sets.

Comparing	Hamming loss ↓						
algorithm	CAL500	llog	enron	image	scene	yeast	slashdot
ML^2	0.138±0.002	0.021±0.001	**0.051±0.001**	**0.156±0.004**	**0.076±0.003**	**0.196±0.003**	**0.043±0.001**
BR	**0.137±0.002**	**0.017±0.001**	0.060±0.001	0.185±0.004	0.111±0.003	0.201±0.003	0.049±0.001
CLR	**0.137±0.002**	0.018±0.001	0.055±0.001	0.186±0.005	0.112±0.003	0.201±0.003	0.050±0.001
ECC	0.182±0.005	0.025±0.001	0.056±0.001	0.218±0.027	0.096±0.003	0.207±0.003	0.056±0.001
RAKEL	0.138±0.002	**0.017±0.001**	0.058±0.001	0.173±0.004	0.096±0.004	0.202±0.003	0.048±0.001
Comparing	One-error ↓						
algorithm	CAL500	llog	enron	image	scene	yeast	slashdot
ML^2	0.141±0.016	**0.683±0.018**	**0.258±0.090**	**0.272±0.009**	**0.194±0.008**	**0.228±0.009**	**0.382±0.009**
BR	0.362±0.039	0.858±0.009	0.498±0.012	0.406±0.012	0.348±0.007	0.256±0.008	0.501±0.007
CLR	**0.121±0.016**	0.756±0.008	0.279±0.010	0.328±0.017	0.255±0.009	**0.228±0.007**	0.436±0.005
ECC	0.137±0.021	0.720±0.012	0.293±0.008	0.408±0.069	0.247±0.010	0.244±0.009	0.418±0.009
RAKEL	0.286±0.039	0.838±0.014	0.412±0.016	0.312±0.010	0.247±0.009	0.251±0.008	0.453±0.005
Comparing	Coverage ↓						
algorithm	CAL500	llog	enron	image	scene	yeast	slashdot
ML^2	0.780±0.008	0.162±0.008	0.256±0.017	**0.168±0.007**	**0.067±0.003**	**0.454±0.004**	0.112±0.003
BR	0.972±0.001	0.468±0.010	0.595±0.010	0.280±0.008	0.158±0.004	0.641±0.005	0.238±0.005
CLR	**0.751±0.008**	**0.155±0.010**	**0.229±0.006**	0.190±0.007	0.083±0.003	0.462±0.005	**0.109±0.003**
ECC	0.806±0.016	0.309±0.014	0.349±0.014	0.229±0.034	0.084±0.002	0.464±0.005	0.130±0.004
RAKEL	0.971±0.001	0.459±0.011	0.523±0.008	0.209±0.009	0.104±0.003	0.558±0.006	0.212±0.005
Comparing	Ranking loss ↓						
algorithm	CAL500	llog	enron	image	scene	yeast	slashdot
ML^2	0.188±0.002	0.158±0.005	0.090±0.012	**0.143±0.007**	**0.064±0.003**	**0.168±0.003**	0.095±0.003
BR	0.518±0.008	0.421±0.008	0.308±0.007	0.285±0.009	0.171±0.005	0.315±0.005	0.216±0.005
CLR	**0.181±0.002**	**0.121±0.007**	**0.079±0.002**	0.171±0.008	0.083±0.004	0.172±0.004	**0.094±0.003**
ECC	0.204±0.008	0.367±0.011	0.133±0.004	0.224±0.043	0.085±0.003	0.186±0.003	0.131±0.005
RAKEL	0.444±0.005	0.412±0.010	0.241±0.005	0.196±0.008	0.107±0.003	0.245±0.004	0.190±0.005
Comparing	Average precision ↑						
algorithm	CAL500	llog	enron	image	scene	yeast	slashdot
ML^2	**0.501±0.003**	**0.405±0.013**	**0.681±0.053**	**0.824±0.006**	**0.885±0.004**	**0.765±0.005**	**0.711±0.005**
BR	0.275±0.006	0.178±0.009	0.449±0.011	0.709±0.008	0.771±0.005	0.672±0.005	0.572±0.005
CLR	0.499±0.005	0.377±0.008	0.675±0.005	0.789±0.009	0.850±0.006	0.758±0.005	0.674±0.003
ECC	0.482±0.008	0.316±0.009	0.651±0.006	0.739±0.043	0.853±0.005	0.752±0.006	0.680±0.006
RAKEL	0.353±0.006	0.197±0.013	0.539±0.006	0.788±0.006	0.843±0.005	0.720±0.005	0.617±0.004
Comparing	Macro-averaging AUC ↑						
algorithm	CAL500	llog	enron	image	scene	yeast	slashdot
ML^2	**0.558±0.006**	**0.705±0.017**	**0.714±0.021**	**0.861±0.005**	**0.948±0.004**	**0.702±0.007**	**0.870±0.006**
BR	0.500±0.001	0.517±0.002	0.579±0.007	0.705±0.007	0.801±0.003	0.565±0.003	0.656±0.009
CLR	0.533±0.007	0.676±0.014	0.698±0.013	0.816±0.007	0.917±0.004	0.645±0.007	0.833±0.016
ECC	0.507±0.005	0.544±0.004	0.646±0.008	0.807±0.030	0.931±0.004	0.646±0.003	0.767±0.010
RAKEL	0.547±0.007	0.520±0.002	0.596±0.007	0.803±0.005	0.884±0.004	0.614±0.003	0.687±0.011

demonstrate the rationality of the smoothness assumption, where the points close to each other in the feature space (a) are also close in the label space (b).

Table 1 summarizes detailed characteristics of the real data sets, which are roughly organized in ascending order of the number of examples $|\mathcal{S}|$, with seven of them being regular-scale, i.e., $|\mathcal{S}| < 5,000$ and seven of them being large-scale, i.e., $|\mathcal{S}| \geq 5,000$. As shown in Table 1, the fourteen data sets cover a broad range of cases with diversified multi-label properties and thus serve as a solid basis for thorough comparative studies.

Comparing Algorithms In this paper, we choose to compare the performance of ML^2 against four well-established multi-label learning algorithms: Binary Relevance (BR) (Boutell et al. 2004), Calibrated Label Ranking (CLR) (Frnkranz et al. 2008), Ensemble of Classifier Chains (ECC) (Read et al. 2011) and RAndom k-labelsets (RAKEL) (Tsoumakas, Katakis, and Vlahavas 2011), which learn from multi-label data based on various correlation orders among labels.

The number of neighbors K for ML^2 is set to $q+1$, because it is necessary that K is larger than q to generate a q-dimensional space using K vectors. The parameters λ, C_1 and C_2 are set to 1, 1 and 10, respectively. The ensemble size for RAKEL is set to $2q$ with $k=3$.

Evaluation Metrics We use six evaluation metrics widely-used in multi-label learning in this paper, i.e., *Hamming loss*, *One-error*, *Coverage*, *Ranking loss*, *Average precision* and *AUC* (Zhang and Zhou 2014). Note that for all the six multi-label metrics, their values vary between [0,1]. Furthermore, for average precision and AUC, the *larger* the values the better the performance; While for the other four

Table 3: Predictive performance of each comparing algorithm (mean ± std. deviation) on the large-scale data sets.

Comparing	Hamming loss ↓						
algorithm	corel5k	rcv1-s1	rcv1-s2	bibtex	corel16k-s1	corel16k-s2	tmc2007
ML^2	**0.010±0.001**	**0.026±0.001**	**0.023±0.001**	**0.013±0.001**	0.021±0.001	0.019±0.001	**0.061±0.001**
BR	0.012±0.001	0.031±0.001	0.028±0.001	0.015±0.001	0.020±0.001	0.019±0.001	0.071±0.001
CLR	0.011±0.001	0.029±0.001	0.025±0.001	0.014±0.001	**0.019±0.001**	**0.018±0.001**	0.068±0.001
ECC	0.015±0.001	0.030±0.001	0.024±0.001	0.017±0.001	0.030±0.001	**0.018±0.001**	0.066±0.011
RAKEL	0.012±0.001	0.031±0.001	0.027±0.001	0.015±0.001	0.020±0.001	0.019±0.001	0.068±0.001
Comparing	One-error ↓						
algorithm	corel5k	rcv1-s1	rcv1-s2	bibtex	corel16k-s1	corel16k-s2	tmc2007
ML^2	**0.647±0.007**	**0.409±0.005**	**0.410±0.009**	**0.365±0.004**	**0.647±0.004**	**0.643±0.005**	**0.225±0.004**
BR	0.849±0.008	0.602±0.011	0.522±0.009	0.559±0.004	0.920±0.006	0.920±0.005	0.339±0.003
CLR	0.721±0.007	0.421±0.005	0.418±0.004	0.401±0.004	0.702±0.005	0.697±0.005	0.242±0.003
ECC	0.699±0.006	0.427±0.008	0.427±0.008	0.404±0.003	0.706±0.006	0.712±0.005	0.232±0.003
RAKEL	0.819±0.010	0.548±0.014	0.472±0.007	0.506±0.005	0.886±0.007	0.897±0.006	0.253±0.003
Comparing	Coverage ↓						
algorithm	corel5k	rcv1-s1	rcv1-s2	bibtex	corel16k-s1	corel16k-s2	tmc2007
ML^2	0.372±0.006	0.109±0.002	0.111±0.003	0.128±0.003	0.322±0.003	0.312±0.002	**0.126±0.001**
BR	0.898±0.003	0.448±0.005	0.383±0.006	0.461±0.006	0.673±0.002	0.671±0.001	0.380±0.003
CLR	**0.267±0.004**	**0.102±0.002**	**0.106±0.003**	**0.118±0.003**	**0.281±0.002**	**0.267±0.002**	**0.126±0.001**
ECC	0.562±0.007	0.187±0.003	0.206±0.007	0.327±0.008	0.446±0.003	0.436±0.002	0.173±0.002
RAKEL	0.886±0.004	0.414±0.004	0.353±0.006	0.443±0.006	0.667±0.002	0.666±0.001	0.279±0.003
Comparing	Ranking loss ↓						
algorithm	corel5k	rcv1-s1	rcv1-s2	bibtex	corel16k-s1	corel16k-s2	tmc2007
ML^2	0.163±0.003	0.043±0.001	0.045±0.002	0.067±0.002	0.167±0.001	0.160±0.001	**0.048±0.001**
BR	0.655±0.004	0.279±0.004	0.251±0.004	0.303±0.004	0.422±0.001	0.424±0.001	0.216±0.003
CLR	**0.114±0.002**	**0.040±0.001**	**0.042±0.001**	**0.065±0.002**	**0.146±0.001**	**0.139±0.001**	0.050±0.001
ECC	0.292±0.003	0.079±0.002	0.096±0.004	0.192±0.003	0.233±0.002	0.229±0.001	0.074±0.001
RAKEL	0.627±0.004	0.243±0.004	0.216±0.004	0.286±0.003	0.414±0.002	0.418±0.001	0.139±0.002
Comparing	Average precision ↑						
algorithm	corel5k	rcv1-s1	rcv1-s2	bibtex	corel16k-s1	corel16k-s2	tmc2007
ML^2	**0.297±0.002**	0.627±0.003	**0.643±0.005**	**0.596±0.004**	**0.332±0.002**	**0.327±0.003**	**0.813±0.002**
BR	0.101±0.003	0.383±0.007	0.434±0.005	0.363±0.004	0.085±0.002	0.078±0.002	0.643±0.002
CLR	0.274±0.002	**0.628±0.003**	0.641±0.003	0.564±0.004	0.306±0.003	0.303±0.002	0.798±0.002
ECC	0.264±0.003	0.606±0.004	0.616±0.005	0.515±0.004	0.282±0.003	0.276±0.003	0.787±0.002
RAKEL	0.122±0.004	0.436±0.006	0.487±0.005	0.399±0.004	0.103±0.003	0.092±0.003	0.735±0.002
Comparing	Macro-averaging AUC ↑						
algorithm	corel5k	rcv1-s1	rcv1-s2	bibtex	corel16k-s1	corel16k-s2	tmc2007
ML^2	0.667±0.007	**0.914±0.008**	**0.913±0.005**	**0.911±0.002**	0.692±0.003	0.699±0.003	**0.926±0.001**
BR	0.518±0.001	0.609±0.003	0.599±0.004	0.624±0.002	0.516±0.001	0.519±0.001	0.724±0.002
CLR	**0.678±0.005**	0.898±0.005	0.884±0.003	0.908±0.002	**0.723±0.003**	**0.739±0.003**	0.902±0.001
ECC	0.568±0.003	0.777±0.005	0.763±0.005	0.763±0.003	0.627±0.004	0.633±0.002	0.880±0.002
RAKEL	0.521±0.001	0.637±0.004	0.627±0.004	0.641±0.002	0.523±0.001	0.525±0.001	0.796±0.002

metrics, the *smaller* the values the better the performance. These metrics serve as good indicators for comprehensive comparative studies as they evaluate the performance of the learned models from various aspects.

4.2 Experimental Results

Fig. 3(c) shows the numerical labels reconstructed by ML^2 using the topological structure transferred from the feature space in (a) and the signs of the real labels in (b) (note that only the signs of the label vectors in (b) are used in ML^2 to simulate the real multi-label data). We can see from (b) and (c) that ML^2 indeed effectively recover the label manifold.

Table 2 and 3 report the detailed experimental results of each comparing algorithm on the regular-scale and large-scale data sets respectively. On each data set, 50% examples are randomly sampled without replacement to form the training set, and the rest 50% examples are used to form the test set. The sampling process is repeated for ten times and the average predictive performance across ten training/testing trials are recorded. For each evaluation metric, ↓ indicates the smaller the better while ↑ indicates the larger the better. Furthermore, the best performance among the five comparing algorithms is shown in boldface.

From the result table we can see, on the regular-scale data sets (Table 2), across all the evaluation metrics, ML^2 ranks $1st$ in 73.8% cases and ranks $2nd$ in 21.4% cases, and on the large-scale data sets (Table 3), across all the evaluation metrics, ML^2 ranks $1st$ in 57.1% cases and ranks $2nd$ in 40.5% cases. Thus ML^2 achieves competitive performance against the well-established multi-label learning algorithms across extensive benchmark data sets and diverse evaluation metrics, which validate the effectiveness of the label manifold for multi-label learning.

5 Conclusion

This paper explores the manifold in the label space for multi-label learning. Because the label manifold is not explicitly available from the training examples, we propose a novel method called ML^2 to reconstruct and exploit the label manifold based on the smoothness assumption. Extensive comparative studies clearly validate the advantage of ML^2 against the state-of-the-art multi-label learning approaches. In the future, we will explore if there exists better ways to estimate and make use of the label manifold for multi-label learning.

Acknowledgements

This work was partly supported by the National Science Foundation of China (61273300, 61232007, 61175049, 61222309), Jiangsu Natural Science Funds for Distinguished Young Scholar (BK20140022), and the MOE Program for New Century Excellent Talents in University (NCET-13-0130).

References

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15(6):1373–1396.

Bo, L., and Sminchisescu, C. 2009. Structured output-associative regression. In *Computer Vision and Pattern Recognition*, 2403–2410. IEEE.

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

Burba, F.; Ferraty, F.; and Vieu, P. 2009. k-Nearest Neighbour method in functional nonparametric regression. *Journal of Nonparametric Statistics* 21(4):453–469.

Cabral, R. S.; Torre, F.; Costeira, J. P.; and Bernardino, A. 2011. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, 190–198.

Chapelle, O.; Schlkopf, B.; Zien, A.; and others. 2006. Semi-supervised learning.

Chung, W.-H.; Kim, J.-H.; Lee, H.; and Kim, E. 2014. General Dimensional Multiple-Output Support Vector Regressions and Their Multiple Kernel Learning.

Elisseeff, A., and Weston, J. 2001. A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, 681–687.

Frnkranz, J.; Hllermeier, E.; Menca, E. L.; and Brinker, K. 2008. Multilabel classification via calibrated label ranking. *Machine learning* 73(2):133–153.

Geng, X.; Yin, C.; and Zhou, Z.-H. 2013. Facial age estimation by learning from label distributions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(10):2401–2412.

Lo, H.-Y.; Wang, J.-C.; Wang, H.-M.; and Lin, S.-D. 2011. Cost-sensitive multi-label learning for audio tag annotation and retrieval. *Multimedia, IEEE Transactions on* 13(3):518–529.

Niyogi, X. 2004. Locality preserving projections. In *Neural information processing systems*, volume 16, 153. MIT.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on* 22(10):1345–1359.

Prez-Cruz, F.; Vzquez, A. N.; Alarcn-Diana, P. L.; and Arts-Rodrguez, A. 2000. An IRWLS procedure for SVR. In *Signal Processing Conference, 2000 10th European*, 1–4. IEEE.

Prez-Cruz, F.; Camps-Valls, G.; Soria-Olivas, E.; Prez-Ruixo, J. J.; Figueiras-Vidal, A. R.; and Arts-Rodrguez, A. 2002. Multi-dimensional function approximation and regression estimation. In *Artificial Neural NetworksICANN 2002*. Springer. 757–762.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine learning* 85(3):333–359.

Roweis, S. T., and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326.

Rubin, T. N.; Chambers, A.; Smyth, P.; and Steyvers, M. 2012. Statistical topic models for multi-label document classification. *Machine learning* 88(1-2):157–208.

Sanden, C., and Zhang, J. Z. 2011. Enhancing multi-label music genre classification through ensemble techniques. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 705–714. ACM.

Smola, A. J., and Schlkopf, B. 1998. *Learning with kernels*. Citeseer.

Sun, L.; Ji, S.; and Ye, J. 2011. Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33(1):194–200.

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

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2009. Mining Multi-label Data. In Maimon, O., and Rokach, L., eds., *Data Mining and Knowledge Discovery Handbook*. Springer US. 667–685.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2011. Random k-labelsets for multilabel classification. *Knowledge and Data Engineering, IEEE Transactions on* 23(7):1079–1089.

Tuia, D.; Verrelst, J.; Alonso, L.; Prez-Cruz, F.; and Camps-Valls, G. 2011. Multioutput support vector regression for remote sensing biophysical parameter estimation. *Geoscience and Remote Sensing Letters, IEEE* 8(4):804–808.

Wang, F., and Zhang, C. 2008. Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on* 20(1):55–67.

Wang, J.; Zhao, Y.; Wu, X.; and Hua, X.-S. 2011. A transductive multi-label learning approach for video concept detection. *Pattern Recognition* 44(10):2274–2286.

Wang, H.; Huang, H.; and Ding, C. 2009. Image annotation using multi-label correlated green's function. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2029–2034. IEEE.

Yang, B.; Sun, J.-T.; Wang, T.; and Chen, Z. 2009. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 917–926. ACM.

Zhang, M.-L., and Zhou, Z.-H. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* 40(7):2038–2048.

Zhang, M.-L., and Zhou, Z.-H. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26(8):1819–1837.

Zhu, X.; Lafferty, J.; and Rosenfeld, R. 2005. *Semi-supervised learning with graphs*. Carnegie Mellon University, Language Technologies Institute, School of Computer Science.