Dynamic Label Propagation for Semi-supervised Multi-class Multi-label Classification

Bo Wang
Stanford Univeristy
wangbo.yunze@gmail.com

Zhuowen Tu
University of California, San Diego
ztu@ucsd.edu

John K. Tsotsos
York University
tsotsos@cse.yorku.ca

Abstract

In graph-based semi-supervised learning approaches, the classification rate is highly dependent on the size of the availabel labeled data, as well as the accuracy of the similarity measures. Here, we propose a semi-supervised multi-class/multi-label classification scheme, dynamic label propagation (DLP), which performs transductive learning through propagation in a dynamic process. Existing semi-supervised classification methods often have difficulty in dealing with multi-class/multi-label problems due to the lack in consideration of label correlation; our algorithm instead emphasizes dynamic metric fusion with label information. Significant improvement over the state-of-the-art methods is observed on benchmark datasets for both multi-class and multi-label tasks.

1. Introduction

In classification, it is often hard to obtain a single fixed distance metric for points in the entire data space. Moreover, nice properties enjoyed by graph-based (built on the distance metric) two-class semi-supervised classification [36] become less obvious in the multi-class classification situations [11], due to the correlations of the multiple labels.

Supervised metric learning methods often learn a Mahalanobis distance by encouraging small distances among points of the same label while maintaining large distances for points of different labels [29, 28]. Graph-based semi-supervised learning frameworks on the other hand utilize a limited amount of labeled data to explore information on a large volume of unlabeled data. Label propagation (LP) [36] specifically assumes that nodes connected by edges of large similarity tend to have the same label through information propagated within the graph. A wide range of applications such as classification, ranking, and retrieval [37] have adopted the label propagation strategy. Another type of semi-supervised learning, co-training [5], utilizes multi-view features to help each other by pulling out unlabeled data to re-train and enhance the classifiers.

The above methods are mainly designed to deal with the binary classification problem. For the multi-class/multi-label case, the label propagation algorithm [36] becomes more problematic, therefore some special care needs to be taken. A common approach to address the multi-class/multi-label learning is to use a one vs. all strategy. The disadvantage of one vs. all approaches is, however, that the correlations among different classes are not fully utilized. As discussed in [35], taking the class correlations into account often leads to a significant performance improvement.

In this paper, we propose a new method, dynamic label propagation (DLP), to simultaneously deal with the multi-class and multi-label problem. Our method incorporates the label correlations and instance similarities into a new way of performing label propagation. Our intuition in DLP is to update the similarity measures dynamically by fusing multi-label/multi-class information, which can be understood in a probabilistic framework. The K nearest neighbor (KNN) matrix is used to preserve the intrinsic structure of the input data. We present comprehensive experimental results illustrating the advantages of the proposed method on multi-class digit categorization, object recognition, and multi-label text classification.

2. Related Work

As discussed in Section (1), a popular strategy toward multi-class/multi-label learning is to divide it into a set of binary classification problems, using techniques such as one-versus-the-rest, one-versus-one, and error-correcting coding[1]. These methods however have certain limitations including: (1) the difficulty to scale up to large data sets, and (2) inability to exploit the coherence and relations among classes due to the use of independent classifiers. Also, they may result in unbalanced classification outputs, especially when the number of classes is large.

A lot of recent attention has been focused on addressing those limitations of semi-supervised multi-class learning. The existing algorithms can be roughly classified into three categories. 1) Density-based: a recent notable ad-

vance in density-based method is a multi-class extension to the TSVM by [30]; however, its high computational cost limits it from being widely adopted. 2) Boosting-based: there are a variety of semi-supervised multi-class extensions to the boosting methods [24, 20]; these methods differ in the loss functions and regularization techniques; the disadvantage of them is the lack of ability to utilize the correlation between labels and input features (especially for the unlabeled data), which, to some extent, jeopardizing the classification accuracy. 3) Graph-based: some recent advances adopt Gaussian Processes [21, 17] or Markov Random Walks [2]. Transduction by Laplacian graph [4, 10] is also shown to be able to solve multi-class semi-supervised problems; although these algorithms make use of the relationship between unlabeled and labeled data, their computational complexity is demanding, e.g. of $\mathcal{O}(n^3)$.

However, there are much fewer attempts to tackle semi-supervised multi-label problem, despite there being a rich body of literature about supervised multi-label learning. One popular method is label ranking [8], which learns a ranking function of category labels from the labeled instances and classifying each unlabeled instance by thresholding the scores of the learned ranking functions. Although being easy to scale up, label ranking fails to exploit the correlations among data categories. Recently, category correlations are given more attention in multi-label learning. A maximum entropy method is employed to model the correlations among categories in [35]. [18] studies a hierarchical structure to handle the correlation information. In [12], a correlated label propagation framework is developed for multi-label learning that explicitly fuses the information of different classes. However, these methods are only for supervised learning, and how to make use of label correlation among unlabeled instances is still unclear. [16] uses constrained non-negative matrix factorization to propagate the label information by enforcing the examples with similar input patterns to share similar sets of class labels. Another semi-supervised multi-label learning technique [7] develops a regularization with two energy terms about smoothness of input instances and label information by solving a Sylvester Equation. A similar algorithm [33] solves the multi-label problem with an optimization framework with an regularization of Laplacian matrix.

Different from these semi-supervised multi-label methods, the proposed method explicitly merges the input data and label correlations. Moreover, by doing projection on the fused manifolds, DLP further takes advantage of the correlations among labeling information of unlabeled data. Our work also differs significantly from a very recent algorithm [13], which emphasizes the learning of fusion parameters for unlabeled data; the focus here is however the dynamic update of the similarity functions from both data and label information. In addition, our method is a unifying framework for both multi-class and multi-label classification.

The current literature addressing combined multi-class and multi-label problem is still limited. The reason is twofold. First, the multi-label problem considers the label correlations, but it may lead to a loss in the discrimination power of the multi-class classifiers. On the other hand, the prediction function learned in the multi-class problem often fails to solve the multiple overlaps of different labels in the multi-label problem. The proposed dynamic label propagation method (DLP) aims to solve semi-supervised multi-class and multi-label problem simultaneously by combining the discriminative graph similarities and the label correlations in a dynamic way, while preserving the intrinsic structure of input data. These two steps can well balance the difference in the multi-class and multi-label problems.

3. Label Propagation

First, a brief introduction of the well-known label propagation algorithm is provided in this section. We are given a finite weighted graph $G = (V, E, W)$, consisting of a set of vertices V based on a data set $X = \{x_i, i = 1, \ldots, n\}$, a set of edges E of $V \times V$, and a nonnegative symmetric weight function $W : E \to [0, 1]$. If $W(i, j) > 0$, we say that there is an edge between x_i and x_j. We interpret the weight $W(i, j)$ as a similarity measure between the vertices x_i and x_j. If ρ is a distance metric defined on the graph, then the similarities matrix can be constructed as follows:

$$W(i,j) = h(\frac{\rho(x_i, x_j)^2}{\mu\sigma^2}), \tag{1}$$

for some function h with exponential decay at infinity. A common choice is $h(x) = exp(-x)$. Note that μ and σ are hyper-parameters. σ is learned by the mean distance to K-nearest neighborhoods [31].

A natural transition matrix on V can be defined by normalizing the weight matrix as:

$$P(i,j) = \frac{W(i,j)}{\sum_{k \in V} W(i,k)}, \tag{2}$$

so that $\sum_{j \in V} P(i, j) = 1$. Note that P is asymmetric after the normalization.

Denote the dataset as $X = \{X_l \bigcup X_u\}$, where X_l represents the labeled data and X_u represents the unlabeled data. One important step in label propagation (LP) is clamping, i.e., the labels of labeled data must be reset after each iteration. For the two-class LP, we refer readers to [36]; for the multi-class problem, $1-$of$-C$ coding representation is often used, so the label matrix is $Y = [Y^{(l)}; Y^{(u)}] \in \mathbb{R}^{n \times C}$, where n is the number of data points, C is the number of classes, $Y^{(l)}$ is the label matrix for labeled data, and $Y^{(u)}$ is the label matrix for unlabeled data. We let $Y^{(l)}(i, k)$ be 1 if x_i is labeled as class k, and 0 otherwise. During each iteration, two steps are performed: 1) Labels are propagated

$Y_t = P * Y_{t-1}$. 2) Labels of labeled data X_l are reset. The main algorithm of label propagation is summarized in Fig.(1).

1. Construct a probabilistic transition matrix P by (2).
2. Let $Y_0 = [Y_0^l; \mathbf{0}]$.
3. Performing the following steps for T steps:
   3.a $Y_{t+1} = P * Y_t$,
   3.b $Y_{t+1}^{(l)} = Y_0^l$.
4. Output Y_T

Figure 1. Algorithm of Label Propagation (LP).

4. Dynamic Label Propagation

4.1. Local Similarity

Given a dataset X and its corresponding graph $G = (V, E, W)$, we construct a KNN graph $\mathcal{G} = (V, \mathcal{E}, \mathcal{W})$: the vertices of $\mathcal{G}$ are the same as in G, and weighted edges are those nearby ones only. In other words, those similarities between non-neighboring points (in terms of the pairwise similarity values) are set to zero. Essentially we make the assumption that local similarities (high values) are more reliable than far-away ones; and accordingly local similarities can be propagated to non-local points through a diffusion process on the graph. This is a mild assumption widely adopted by other manifold learning algorithms [23, 19].

Using K nearest neighbor (KNN) to measure local affinity, we construct $\mathcal{G}$ with associated similarity matrix:

$$\mathcal{W}(i,j) = \begin{cases} W(i,j) & \text{if } x_j \in KNN(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Then the corresponding KNN matrix becomes:

$$\mathcal{P}(i,j) = \frac{\mathcal{W}(i,j)}{\sum_{x_k \in KNN(x_i)} \mathcal{W}(i,k)}. \tag{4}$$

Note that P carries the full pair-wise similarity information among the data whereas $\mathcal{P}$ only encodes the similarity to nearby data points. However, $\mathcal{P}$ incorporates the robust structural information about the input data space. For clarity, we call P the status matrix and $\mathcal{P}$ the corresponding KNN matrix.

4.2. Label Fusion on Diffusion Space

One disadvantage of label propagation is that it does not work well on multi-class/multi-label classification problem due to a lack of interplay among labels within different classes. In this paper, we propose a dynamic version of label propagation that aims to improve the effectiveness on multi-class/multi-label classification. Our main idea is to have an improved transition matrix by fusing information of both data features and data labels in each iteration.

Given the kernel P_t, where t denotes the number of iterations, we can define the diffusion distance [14] at time t as:

$$D_t(i,j) = \| P_t(i,:) - P_t(j,:) \| . \tag{5}$$

The diffusion process maps the data space into an n-dimensional space $\mathfrak{R}_t^n$ in which each data point is represented by its transition probability to the other data points. It is reasonable to assume that for each data $\mathbf{x}_t \in \mathfrak{R}_t^n$, we have $p(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t|\mu_t, P_t)$, where μ_t is unknown. Note that the label matrix Y_t contains information about class labels, and the correlation of these labels $K_Y = Y_t Y_t^T$ can be viewed as the similarity between data points in the label space $\mathfrak{Q}_t^n$, and data points in this label space $\mathfrak{Q}_t^n$ have the probability $p(\mathbf{y}_t) = \mathcal{N}(\mathbf{y}_t|0, K_t)$.

We divide our method into two steps:

1) Kernel Fusion.

The first part of dynamic label propagation is the fusion of the status matrix P_t and the label kernel $K_Y = Y_t Y_t^T$. A weight α is assigned to the label kernel K_Y. The fused kernel is then

$$F_t = (P_t + \alpha Y_t Y_t^T). \tag{6}$$

This operation corresponds to an addition operator in the diffusion spaces:

$$\mathbf{z}_t = \mathbf{x}_t + \sqrt{\alpha}\mathbf{y}_t. \tag{7}$$

We can then verify that

$$p(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|\mu_t, P_t + \alpha Y_t Y_t^T) = \mathcal{N}(\mathbf{z}_t|\mu_t, F_t). \tag{8}$$

This simple fusion technique considers the correlation among the instance label vectors. The underlying assumption is that two instances with high correlated label vectors tend to have high similarity in the input data space. The correlation between label vectors can represent the label dependency among instances, especially for the multi-label/multi-class problem. The advantage of fusing transition kernel and the label correlation is two-fold: On one hand, two instances with high correlated label vectors are likely to have high similarity in input data space, this fusion process therefore enhances the fitness of the kernel matrix for the input manifold. On the other hand, the resulting kernel matrix leads to better label information through next round of label propagation. In this way, we build up a dynamic interaction process between the feature space and label space. However, since the label information is dynamically updated during the propagation process, the resulting label information after the initial several rounds no longer improves the transition matrix, sometimes even makes it worse. To deal with this problem, we design a novel fusion-operator based on the local neighbours as follows .

2) Kernel Diffusion.

Assume P_0 is the initial status matrix of the input data calculated using (1) and (2), and $\mathcal{P} = KNN(P_0)$ by (3)

and (4); We employ this linear operator $\mathcal{P}$ to do the projection

$$\mathbf{x}_{t+1} = \mathcal{P}\mathbf{z}_t + \lambda_t \varepsilon, \tag{9}$$

where ε is white noise, i.e. $p(\varepsilon) = \mathcal{N}(\varepsilon|0,1)$. Note that $\mathcal{P}$ is a sparse version of P_0 and only local neighbor information in the space is kept in the operator $\mathcal{P}$:

$$\begin{aligned}\mathbf{x}_{t+1}(i) &= \sum_{j\in KNN(i)} P_0(i,j)\mathbf{z}_t(j) + \lambda_t\varepsilon \\ &= \sum_{j\in KNN(i)} P_0(i,j)(\mathbf{x}_t(j) + \alpha\mathbf{y}_t(j)) + \lambda_t\varepsilon\end{aligned}$$

With this linear operation, we have:

$$p(\mathbf{x}_{t+1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\mathcal{P}\mathbf{z}_t, \lambda_t I). \tag{10}$$

The marginal distribution of $\mathbf{x}_{t+1}$ is

$$\begin{aligned}p(\mathbf{x}_{t+1}) &= \int \mathcal{N}(\mathbf{z}_t|\mu_t, F_t)\mathcal{N}(\mathbf{x}_{t+1}|\mathcal{P}\mathbf{z}_t, \lambda_t I)d\mathbf{z}_t \\ &= \mathcal{N}(\mathbf{x}_{t+1}|\mathcal{P}\mu_t, \mathcal{P}F_t(\mathcal{P})^T + \lambda_t I).\end{aligned} \tag{11}$$

The above equation implies that, the essence of dynamic label propagation is to do linear operations on diffusion space iteratively. Note that $\mathbf{x}_{t+1}$ is a point in the diffusion space. Instead of performing linear projection in the original data space, we do projection in the diffusion space. The advantages of projection onto the diffusion space are two-fold: 1) we avoid the need to perform computational expensive sampling procedures in the input space; 2) The resulting variance matrix again is a good diffusion kernel for label propagation.

The intuition behind this projection lies in the fact that simple fusion of label correlation in Eqn. (6) would result in a degeneration at the first round when the learned label information of unlabeled data is not accurate enough to infer the similarities in the input space. Hence, inspired by [25], we need to re-emphasize the intrinsic structure between all the input data by the KNN matrix. From (13), we can see that, the diffusion process propagates the similarities through the KNN matrix. In this way, we can adjust the fused kernel matrix to maintain part of the information of the initial structure.

The direct reflection of this projection on diffusion space is that , at each iteration, we construct the transition matrix for next iteration to be:

$$P_{t+1} = \mathcal{P}(P_t + \alpha Y_t Y_t^T)\mathcal{P}^T + \lambda_t I. \tag{12}$$

Thus, we have

$$\begin{aligned}P_{t+1}(i,j) &= \sum_{k\in KNN(i)}\sum_{l\in KNN(j)} P_0(i,k)P_0(j,l)(P_t(k,l) \\ &+ \alpha < Y_t(k,:), Y_t(l,:) >) + \lambda_t\delta_{ij}.\end{aligned} \tag{13}$$

where $< \mathbf{x}_1, \mathbf{x}_2 >$ denotes the inner product of two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, and $\delta_{ij} = 1$ if $i = j$, 0 otherwise. From Eqn.(13), we see that only information between dominant neighbours are propagated into the transition matrix of next iteration. An important observation is that if data i and j have common dominant neighbours in both similarity metrics, it is highly possible that they belong to the same class.

We summarize the details of dynamic label propagation in Fig.(2).

1. Construct a probabilistic transition matrix P_0 by (2).
2. Let $Y_0 = [Y_0^l; \mathbf{0}]$.
3. Calculate the KNN matrix $\mathcal{P}$ of P_0,
4. Performing the following steps for a desired T steps:
   - 4.a $Y_{t+1} = P_t * Y_t$,
   - 4.b $Y_{t+1}^{(l)} = Y_0^l$,
   - 4.c $P_{t+1} = \mathcal{P}(P_t + \alpha Y_t Y_t^T)\mathcal{P}^T + \lambda_t I$.
5. Output Y_T.

Figure 2. Algorithm of Dynamic Label Propagation (DLP).

4.3. Analysis

4.3.1 Convergence Analysis

It is difficult to give a formal theoretical proof of the convergence of DLP. However, empirical experience shows DLP converges much faster than LP (see Fig.(3) and Fig.(5)). Usually, LP needs $1,000 - 5,000$ iterations to converge, while DLP only needs $10-50$ iterations. This is because the diffusion process projects the fused manifold into a KNN structure where only local similarities are preserved. The learned labels can improve the similarity between input instances quickly.

A loose theoretical proof of convergence can be constructed based on the spectral analysis of the diffusion projection $\mathcal{P}$. Since $\mathcal{P}$ is a KNN matrix of P_0, it is easy to see that the spectral radius of $\mathcal{P}$ is less than 1. We have

$$Y_t \propto Y^{(\infty)} + [(\mathcal{P})^t(P_0 + \alpha Y_0 Y_0^T)(\mathcal{P}^T)^t]P_0 Y_0 + o(t) \tag{14}$$

where $o(t)$ is an infinitesimal as t approaches infinity, and $Y^{(\infty)} \in \mathbb{R}^{n\times C}$ is a constant label matrix. We observe that since the spectral radius of $\mathcal{P}$ is less than 1, we have $\lim_{t\to\infty}\mathcal{P}^t \to \mathbf{0}$. Hence, the final label is $\lim_{t\to\infty} Y_t = Y^{(\infty)}$, although we do not have a closed form for $Y^{(\infty)}$ at present.

4.3.2 Time Complexity

Traditional Label Propagation algorithm has a complexity of $\mathcal{O}(n^2)$, however, since our DLP only diffuses the similarities on KNN structures, DLP shares the same scale of time complexity. For the step of kernel fusion, we only perform the addition of two matrices, so the time cost is $\mathcal{O}(n^2)$.

For the step of diffusion in Eqn.(12), we decompose it as in Eqn.(13), from which we observe that only local neighbours are used to propagate the similarities. An easy way to speed up the diffusion process is, first we keep a record of the KNN matrix and then every time we perform the diffusion process, we extract the fixed local structure from the KNN structure and only perform multiplication K times for each pair of data points. Therefore we can update the transition kernel in (12) in time Kn^2+Kn. To summarize, the overall time complexity of DLP is $\mathcal{O}(Kn^2)$, where $K \ll n$.

4.3.3 Parameter Analysis

There are several parameters to tune in DLP. How to choose the number of neighbors in the KNN matrix $\mathcal{P}$ remains an open problem. A small K leads to insufficient structural information in $\mathcal{P}$; a large K value results in an increase in the time complexity and loss in the sparsity in $\mathcal{P}$. There is a trade-off between accuracy and complexity. In all our experiments, we choose K from $\{10, 20, 30, 40, 50\}$ by 10-fold cross-validation. Another two important parameters in DLP are α and λ. α is the weight of label correlation, while λ represents the importance of regularization. Fortunately, DLP is not sensitive to these two parameters. So we fix $\alpha = 0.05$ and $\lambda = 0.1$ in all experiments (see an empirical illustration in Fig.(6)).

4.4. A Toy Data

We first test our dynamic label propagation on a toy data set. It consists of five circles (i.e., 5 classes) (see Fig.(3)(A)). This is a challenging dataset since it contains multiple classes and only one in each class is labeled. We test the effect of the two steps in the dynamic label propagation. We construct the KNN matrix same in [26]. First, we omit the first step that fuses the label correlation with the kernel matrix. The other steps are all the same. The result is shown in Fig.(3).(B). Second, we do the first step to fuse label correlations but omit the second step of kernel diffusion. The result is shown in Fig.(3).(C). Comparing these two results, we see that, each step is important to the final result of DLP. Without the label correlation, DLP fails to capture the dependence between different classes; without the kernel diffusion process, the DLP goes wild because the label correlation in the beginning provides a poor guidance for the kernel matrix. In addition, we show the classification results of DLP and LP in Fig.(3)(D)(E). It is observed that our method only needs a few iterations to converge while LP gets a reasonable result after thousands of iterations.

5. Experiments

5.1. Semi-supervised Multi-class Learning

We compare our DLP with several popular semi-supervised learning methods: 1) Label Propagation (LP) ; 2) A variant of LP on KNN structure(LP+KNN) [22]; 3) Local and Global Consistency (LGC) [34]; 5) Transductive SVM (TSVM) 6) LapRLS [3]. Note that for LP and LGC, we use one-vs-the-rest methods to deal with multi-class problems; for TSVM and LapRLS, they have their own multi-class extensions.

5.1.1 Benchmarks

We test our method on the benchmarks in [6]. An extensive review of the performance of existing algorithms are also availabel in [6]. All the datasets have 12 splits each of which has 100 labeled and 1,400 unlabeled instances. To show the effect of fusing label correlation, we especially set $\alpha = 0$ in our method and denote this special method as DLP_0. The comparisons are shown in Tab.(1). We can see that, our method is still capable of performing binary classification but it is especially suitable for the multi-class classification problem, such as in the dataset COIL. Another important observation is that, although we set $\alpha = 0.05$ for DLP, it does not indicate that the label correlation is of little importance. The only reason for small value of α lies in difference of the numerical scale of label correlation and transition probability. We don't show the results of DLP_0 in the subsequent experiments.

5.1.2 Digit Classification

Table 2. Comparison of error rate on the MNIST dataset.

labeled	LGC	TSVM	LapRLS	LP	LP+KNN	DLP
1%	3.96	4.87	2.92	8.57	4.27	2.01
5%	2.14	2.18	1.54	5.82	2.48	0.90

In this section, we test our method on the popular digit dataset: MNIST[1]. It consists of 60,000 training and 10,000 test images of ten handwritten digits (0 to 9), with 28×28 pixels. In our first experiment, we randomly extract 1% (600) training images, together with 10,000 test images. Our second experiment consists of 5% (3000) training images, together with 10,000 test images. The average error rates of test samples are reported in Tab(2).

Our method outperforms the existing semi-supervised learning techniques on multi-class digit recognition. Note that DLP achieves a significant improvement over the LP algorithm. Also, as to TSVM and LapRLS, they have much heavier computational burdens ($\mathcal{O}(n^3)$) than that of DLP.

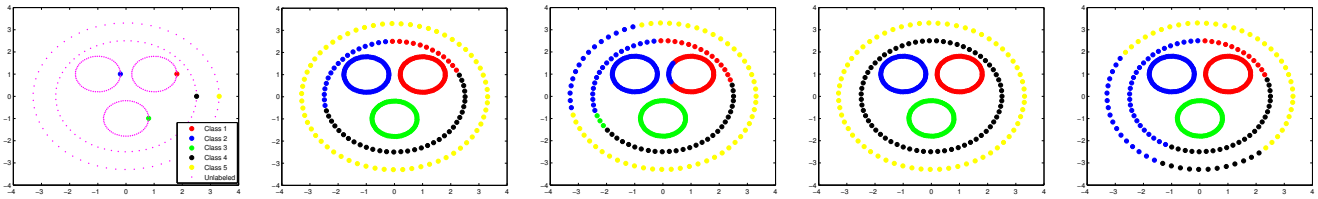
5.1.3 Caltech 101

We also test our algorithm on the well-known Caltech-101 dataset [9] which consists of 101 classes and a collection of background images. We selected 12 classes (including animals, faces, buildings, etc.) from Caltech-101, which

[1] http://yann.lecun.com/exdb/mnist/

Table 1. A quantitative comparison of error rate on the benchmark datasets.

Methods/Dataset	digit1	USPS	BCI	g241c	COIL	gc241n	text
LGC	4.80	8.39	34.21	28.54	10.72	27.78	23.90
TSVM	6.15	9.77	**33.25**	**18.46**	25.80	22.42	24.52
LapRLS	1.81	4.31	27.89	23.45	11.92	24.77	23.32
LP	4.15	7.35	46.22	30.05	11.03	28.11	25.71
LP+KNN	4.01	7.46	40.35	29.49	10.71	27.46	24.07
DLP_0	3.65	6.53	35.87	25.53	6.314	25.21	23.78
DLP	**1.64**	**3.00**	33.48	21.86	**3.57**	**21.82**	**22.84**



(A) (B) (C) (D) (E)

Figure 3. (A) is the toy data with only one labeled data (the colored dots) for each class. (B) is the classification result without using label correlations. (C) is the classification result without using diffusion process. (D) is the result of DLP with only 20 iterations. (E) is the result of LP with 5000 iterations.

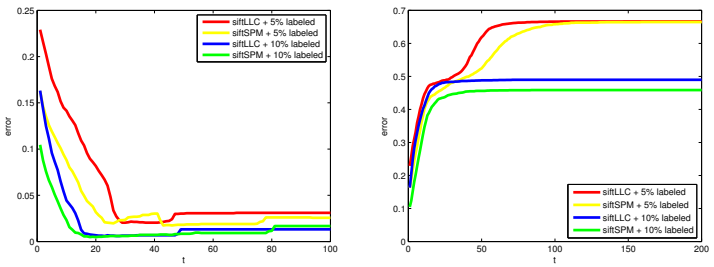
Table 3. A quantitative comparison of error rate on the Caltech 101 dataset.

Experiments/Methods	LGC	TSVM	LapRLS	LP	LP+KNN	DLP
siftLLC+5% labeled	13.48%	9.82%	7.39%	22.91%	14.22%	2.04%
siftSPM+5% labeled	10.24%	8.79%	7.33%	16.36%	12.38%	1.74%
siftLLC+10% labeled	9.47%	7.50%	5.35%	16.33%	8.96%	0.60%
siftSPM+10% labeled	7.43%	5.42%	4.20%	10.46%	7.38%	0.48%

contains totally 2,788 images. These classes are chosen due to the relatively large number of availabel images within the category. The number of images per category varies from 41 to 800, most of which are medium resolution, i.e. about 300×200 pixels. Fig.4 shows some samples of the subset.

Figure 4. Sample images chosen from Caltech 101.

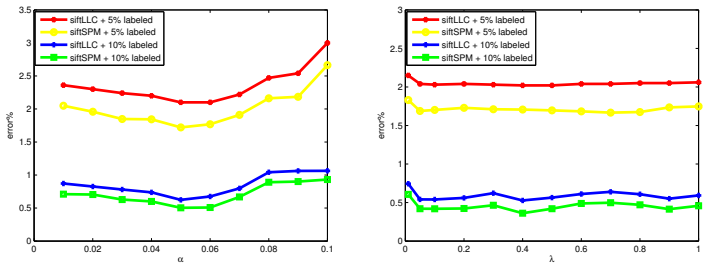
We use two kinds of variants of SIFT feature: SIFT with locality-constrained linear coding (siftLLC) [27] and SIFT with Spatial Pyramid Matching (siftSPM) [15]. The SIFT features are both extracted from 16×16 pixel patches on a grid with step size of 8 pixels. The codebooks are obtained by standard K-means clustering with the codebook size $2,048$. The distance between two images is obtained by the χ^2 distance between two feature vectors. Two experiments are conducted: 1)Only 5% of the samples are labeled, and the remaining samples are tested. 2) 10% samples are labeled, and the rest are tested. We reported the results of error rate in Tab(3).



(A) (B)

Figure 5. (A)Error rate versus the iteration numbers for DLP. (B)Error rate versus the iteration numbers for LP.

We also show the dynamics of label propagation and the proposed methods. We report the error rate of each iteration of DLP and LP (see Fig.(5)). We can see that, as iterations go on, DLP decreases the error rate while on the other hand, LP worsens. This is obviously a big disadvantage of LP for multi-class classification. The $1 - of - C$ coding sometimes makes the LP unable to discriminate the multiple class labels. However, our method does not suffer from this problem because DLP iteratively update the transition matrix based on local similarity and label information. In



(A) (B)

Figure 6. (A)Error rate versus the parameter α for DLP. (B)Error rate versus the parameter λ for DLP.

addition, sensitivity test of the two parameters α and λ are conducted. For the sensitivity of α, we fix $\lambda = 0.1$ and vary α in the range of $[0.01, 0.1]$. For λ, we fix $\alpha = 0.05$ and vary λ in the range of $[0.01, 1]$. The errors of recognition are reported in Fig.(6). We can see that, our proposed DLP is insensitive to α and λ.

5.2. Semi-supervised Multi-label Classification

In this section, we test our method on the task of semi-supervised multi-label classification. We use the data from [7]: a subset of $RCV1 - v2$ text data which includes the information of topics, regions and industries for each document. We first randomly pick 3000 documents, then choose words with more than 5 occurrences and topics with more than 40 positive assignments. We compare our methods with five existing baseline algorithms in semi-supervised multi-label classification. The first one is a *Semi-supervised Multi-label learning method by solving Sylvester Equation* (SMSE) [7]. Here we use the first version mentioned in [7] which needs only one parameter to tune. The second one is based on *Constrained Non-negative Matrix Factorization* (CNMF) [16], which assumes that two instances tend to have large overlap in their assigned labels if they share high similarity in their input patterns. The third one is *Multi-label Informed Latent Semantic Indexing* (MISL) [32], which maps the input features into a new feature space which captures the structure of both input data and label dependency, and then uses SVM on the projected space. The fourth one is the a recent method TRAM, i.e., a transductive multi-label classification algorithm via label set propagation [13], which estimates the label sets of the unlabeled instances by utilizing the information from both unlabeled instances and unlabeled data. The last one is *Support Vector Machine* (SVM), in which a linear SVM classifier is built for each category independently. We evaluate the performance using a common evaluation metric like in [12]: *F1 Micro* which can be seen as the weighted average of F1 scores over all the categories.

Fig.(7) shows the performance measured by *F1 Micro* of six algorithms: DLP, SMSE, CNMF, MLSI, TRAM, SVM at different ranks when the number of training data is 500 or 2000. Note that a 10-fold experiment using the same training/test split of the data set is performed and all the parameters used in the five algorithms are tuned by grid search. We can see that our dynamic label propagation can properly capture the inner structure of label correlation and improve the classification accuracy. When the number of predicted labels for each instance increases, our method can still provide good performance.

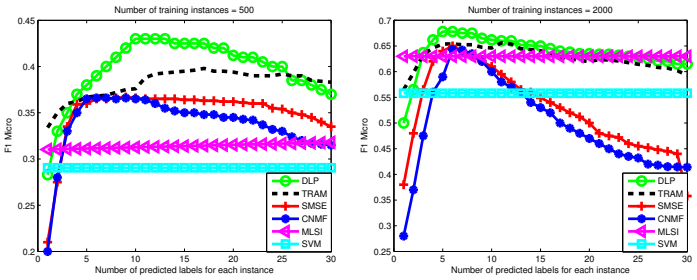


Figure 7. *F1 Micro* when the number of training samples is 500 or 2000. Higher values indicate better performance.

6. Conclusion

In this paper, we have proposed a novel classification method named dynamic label propagation (DLP), which improves the discriminative power in multi-class/multi-label problems in the framework of semi-supervised learning. Our method explores the effect of labeled information and local structure in improving the transition matrix in semi-supervised learning. The significant performance improvement on toy data and some popular natural object images has demonstrated the effectiveness of DLP for multi-class/ multi-label classification. Our future work will focus on providing deeper theoretical understanding of the approach.

7. Acknowledgement

Funding for this research was gratefully received from the Natural Sciences and Engineering Research Council of Canada and the Canada Research Chairs Program. This work is also partly supported by NSF IIS-1216528 (IIS-1360566) and NSF CAREER award IIS-0844566 (IIS-1360568).

References

[1] E. Allwlein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifier. *Journal of Machine Learning Research*, (1):113 –141, 2000.

[2] A. Azran. The rendezvous algorithm: multiclass semi-supervised lenaring with markov random walks. Proc. of ICML, pages 49–56, 2007.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. On Manifold Regularization. In *Proc. of AISTAT*, 2005.

[4] M. B. Blaschko, C. H. Lampert, and A. Gretton. Semi-supervised laplacian regularization of kernel canonical correlation analysis. In *Proceedings of ECMLKDD*, pages 133–145, 2008.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of COLT*, 1998.

[6] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[7] G. Chen, Y. Song, and F. Wang. Semi-supervised multi-label learning by solving a sylvester equation. *Proc of SDM*, 6(10):28–30, 2008.

[8] K. Crammer and Y. Singer. A new family of online algirithms for category ranking. Proc. of SIGIR, 2002.

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:594–611, April 2006.

[10] A. Goldberg, X. J. Zhu, B. Recht, J. Xu, and R. Nowak. Transduction with matrix completion: Three birds with one stone. In *Advances in Neural Information Processing Systems 23*, pages 757–765, 2010.

[11] S. Har-peled, D. Roth, and D. Zimak. Constraint classification for multiclass classification and ranking. In *Proc.of NIPS*, pages 785–792. MIT Press, 2003.

[12] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *Proc. of CVPR*, pages 1719–1726, 2006.

[13] X. Kong, M. K. Ng, and Z.-H. Zhou. Transductive multi-label learning via label set propagation. *TKDE*, 99, 2011.

[14] S. Lafon and A. B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. on PAMI*, 28(9):1393–1403, 2006.

[15] S. Lazebnik and C. Schmid. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of CVPR*, 2006.

[16] Y. Liu, R. Jin, and L. Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proc. of AAAI*, pages 421–426, 2006.

[17] S. Rogers and M. Girolami. Multi-class semi-supervised learning with the e-truncted multinomial probit gaussian process. *Journal of Machine Learning Research*, (1):17 –32, 2007.

[18] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. On maximum margin hierarchical multi-label classification. Proc. of NIPS Workshop on Learning With Structured Outputs, 2004.

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

[20] A. Saffari, C. Leistner, and H. Bischof. Regularized multi-class semi-supervised boosting. In *CVPR*, 2009.

[21] Y. Song, C. Zhang, and J. Lee. Graph based multi-class semi-supervised leanring using gaussian process. In *In IAPR workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 450–458, 2006.

[22] A. Subramanya, S. Petrov, and F. Pereira. Efficient graph-based semi-supervised learning of structured tagging models. In *Proc. of EMNLP*, pages 167–176, 2010.

[23] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2322, 2000.

[24] H. Valizadegan, R. Jin, and A. K. Jain. Semi-supervised boosting for multi-class classification. In *Proc. of ECML PKDD*, pages 522–537, 2008.

[25] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu. Unsupervised metric fusion by cross diffusion. In *CVPR*, pages 2997–3004, 2012.

[26] B. Wang and Z. Tu. Affinity learning via self-diffusion for image segmentation and clustering. In *CVPR*, pages 2312–2319, 2012.

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. of CVPR*, 2010.

[28] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of NIPS*, pages 1473–1480, 2006.

[29] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *Proc. of NIPS*, pages 505–512, 2002.

[30] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *Proc. of AAAI*, 2005.

[31] X. Yang, X. Bai, L. Latecki, and Z. Tu. Improving shape retrieval by learning graph transduction. In *Proc. of ECCV*, 2008.

[32] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *Proc. of SIGIR*, pages 258–265, 2005.

[33] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, and X.-S. Hua. Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation*, 20(2):97 – 103, 2009.

[34] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Proc. of NIPS*, pages 321–328. MIT Press, 2004.

[35] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *Proc. of SIGIR*, pages 274–281, 2005.

[36] X. Zhu. *Semi-supervised Learning with Graphs*. Doctoral thesis, Department of Computer Science, Carnegie Mellon University, 2005.

[37] X. Zhu. Semi-supervised learning literature survey. *Computer Science TR 1530, University of Wisconsin-Madison*, 2008.