Semi-Supervised Learning with Density-Sensitive Manifold Graph

Zheng Wang Yao Zhao Shikui Wei
Institute of Information Science, Beijing Jiaotong University, Beijing 100044, P. R. China
zhengwang100@gmail.com, yzhao@bjtu.edu.cn, shkwei@gmail.com

Abstract

The key problem of Graph-Based Semi-Supervised Learning (GBSSL) methods is how to construct the graph structure under some assumptions. While distance information among graph nodes is investigated well for graph construction, the density information is not given enough attention. In this paper, we propose a novel GBSSL method, named Density-Sensitive Manifold Learning (DSML), which introduces density distribution into graph construction by calculating a new propagation coefficient matrix. The experimental results show that DSML scheme performs better than traditional GBSSL methods. More importantly, the new propagation coefficient matrix can be easily introduced into traditional GBSSL methods to improve their performance, which is also validated in the experiments.

1. Introduction

As a kind of semi-supervised learning methods, *GBSSL* has been attracting more and more attention in machine learning community. The key idea of *GBSSL* is to describe the structure of dataset by constructing a graph in which the nodes indicates the labels of samples in the dataset and the edges reflect the similarity among samples. Generally, these methods are based on the consistency assumption [2] (or cluster assumption [1]), which assume that: (1) nearby points are likely to have the same label; and (2) points on the same structure are likely to have the same label.

Many graph construction methods have been proposed in recent years. For example, a consistency method is proposed in [2] by using the local and global consistency, where the Gaussian function and normalized Laplacian are employed as regularizer to calculate the edge weights of the graph. In [5], Wang etc. proposed the linear neighborhood propagation (*LNP*) method where a similar method to *LLE* [11] is used to construct a parameter stable graph. While these methods have achieved a good performance, most of them focus mainly on pairwise distance measurement among nodes when constructing graph but pay less attention to exploiting the density information. To address this problem, Tang etc. [6] further introduced the local distribution difference information to produce more accurate pairwise distance measurement. In their work, a Gaussian function is explicitly used to model the density differences among nodes. While this method improves the manifold structure, the monotonicity of probability density function is not guaranteed, leading to uncertain result.

To address above problems, we propose a new *GBSSL* method to discover the intrinsic manifold structure hidden in the dataset. The main difference from existing *GBSSL* methods is that this method exploits both density and distance information as well as keeping the monotonicity of probability density function. Another feature of the proposed method is that it can be easily introduced into traditional *GBSSL* methods to enhance their performance.

The rest of the paper is organized as follows: Section 2 introduces necessary notations and analyzes the influences of the probability density. In Section 3 we present our method in detail, and the analysis is given in Section 4. Section 5 presents the experimental results on synthetic and real world data, followed by the conclusions in Section 6.

2. Probability Density in *GBSSL* Methods

2.1. Related *GBSSL* Methods

The basic idea of *GBSSL* is to convert data labeling problem into graph construction problem. Let $X = \{x_1, ...x_l, x_{l+1}, ...x_n\}$ be a set of n data points, where $\{x_i\}_{i=1}^{l}$ are labeled data points and $\{x_u\}_{u=l+1}^{n}$ are unlabeled ones. Given a label set $L = \{1, ..., c\}$, our goal is to predict the labels of the unlabeled points.

Define M as the set of $n \times c$ matrices with nonnegative entries. A matrix $F = [F_1^T, ..., F_n^T] \in M$ corresponds to one classification on the data set X by labeling points using $y_i = \arg\max_{j \le c} F_{ij}^*$. F can be thought as a function that assigns a label for each data point. Initially, we define a $n \times c$ matrix $Y \in M$ with $Y_{ij} = 1$ if x_i is labeled as $y_i = j$ and $Y_{ij} = 0$ otherwise.

978-1-4244-5900-1/10/$26.00 ©2010 IEEE

As mentioned above, graph construction is the key step in *GBSSL* methods. In previous methods [2,3], the Gaussian function is employed to calculate the edge weights among nodes(i.e., the distances among points):

$$e_{ij} = \exp(-\left\|x_i - x_j\right\|^2 / (2\sigma^2)) \quad (1)$$

Then, the propagation coefficient from x_j to x_i is defined as:

$$s_{ij} = e_{ij} / \sqrt{d_i d_j} \quad (2)$$

where, d_i is computed as

$$d_i = \sum_{j=1}^{n} e_{ij} \quad (3)$$

Note that $s_{ij} \geq 0$, $\sum_{j=1}^{n} s_{ij} \neq 1$ in above equations.

2.2. Connecting with the probability density

Before exploiting the density information, we must model it first. We assume that a sequence $\{\rho_i\}_{i=1}^{n}$ stands for the probability density of each data point. Among the non-parametric density estimate methods, Parzen window [9] is frequently used for density estimation, by which the probability density can be estimated for a fixed point x_i by

$$\rho_i = \frac{1}{nh}\sum_{j=1}^{n} k(\frac{x_i - x_j}{h}) \quad (4)$$

where h is a smoothing parameter called the *bandwidth* and $k(x)$ is a kernel function that satisfies both $k(x) > 0$ and $\int k(x)dx = 1$. If we adopted the Gaussian Kernel

$$k(\frac{x_i - x_j}{h}) = \frac{1}{(2\pi)^{m/2} h^m} \exp(-\frac{\left\|x_i - x_j\right\|^2}{2h^2}) \quad (5)$$

we can then induce the following result from above five equations.

$$d_i = (2\pi)^{m/2} h^{m+1} n \rho_i \quad (6)$$

Since $(2\pi)^{m/2} h^{m+1} n$ is constant, d_i can be treated as a measure of density for sample x_i. If we further introduce Eq.(6) into Eq.(2), we can rewrite the propagation coefficient as

$$s_{ij} = \frac{e_{ij}}{\sqrt{d_i d_j}} = \frac{e_{ij}}{d_i}(\frac{\sqrt{d_i}}{\sqrt{d_j}}) = \frac{e_{ij}}{d_i}(\frac{d_j}{d_i})^{-\frac{1}{2}} = \frac{e_{ij}}{d_i}(\frac{\rho_j}{\rho_i})^{-\frac{1}{2}} \quad (7)$$

Assume that e_{ij} and d_i are fixed in Eq.(7). Then, for a fixed point x_i, the propagation coefficient from its nearby point x_j strictly decreases accompanied with the increase of the density of x_j.

As indicated in Eq.(7), the propagation coefficient in Eq. (2) also implicitly investigated the influence of density distribution. That is, the traditional *GBSSL* methods in [2, 3] also implicitly use the density information while constructing graph. However, there are two drawbacks for this coefficient definition. First, $\sum_{j=1}^{n} s_{ij} \neq 1$, which means that s_{ij} cannot be seen as a transition probability. Second, the density function in Eq.(2) is fixed to $g(\rho_j) = (\rho_j / \rho_i)^{-1/2}$, which maybe not the optimal function since the optimization objectives vary with different real-world applications. To meet the requirement of video concept detection application, the *Structure Sensitive Manifold Ranking* (*SSMR*) method [6] adopted a Gaussian function to measure the density similarity between x_i and x_j:

$$g_{ij} = \exp(-(\rho_i - \rho_j)^2 / 2\sigma_\rho^2)$$

however, since this function is not a monotonic decreasing function of ρ_j, it is possible to lead to an uncertain results for other application scenario.

Therefore, it is necessary to find out a new function which can model the density distribution as well as keep monotonicity. In the next subsection, we will propose a new probability density function and discuss how to use it to construct graph.

3. The algorithm-*DSML*

As mentioned in Section 1, the graph will reveal the data manifold more accurate if the *consistency assumption* is embedded into it. Considering Eq. (7), we assume that: for a fixed point x_i and its neighbor x_j, the transition probability from x_j to x_i not only decreases with respect to the increment of their distance in the feature space, but also decreases with the increment of the density of x_j. Instead of considering pairwise relationships as Eq. (1) in traditional *GBSSL* methods, we propose to use the neighborhood transition probability among points to construct a directed graph. So our algorithm can be achieved by two steps:

Step 1: Construct the transition probability matrix S: First, we propose to use the probability density information to construct the affinity matrix G, and g_{ij} is the propagation coefficient from x_j to x_i. We will use $N(x_i)$ to denote the set composed of the k nearest neighbors of x_i.

Define the propagation coefficient from x_j to x_i as:

$$g_{ij} = 1 / (1 + e^{-r(\rho_i - \rho_j)}) \quad (8)$$

where $x_j \in N(x_i)$ and $r > 0$ is the parameter that ensures this is a monotonic decreasing function with respect to ρ_j. Obviously, the lower probability density of x_j, the larger g_{ij} will be. The new similarity of x_j to x_i is defined as:

$$w_{ij} = e_{ij} \cdot g_{ij} = \exp(-\left\|x_i - x_j\right\|^2 / 2\sigma^2) / (1 + e^{-r(\rho_i - \rho_j)}) \quad (9)$$

while $i \neq j$ and $w_{ii} = 0$. The first term in the right side of Eq. (9) shows that for each point x_i with its neighbor x_j, the propagation coefficient from x_j to x_i decreases with the increment of their distance in the feature space. And the second one shows that the propagation coefficient from x_j to x_i decreases with the increment of ρ_j.

Then W is normalized as: $S = D^{-1}W = D^{-1}(E \bullet G)$ and $\bullet$ represents the compoint-wise *Hadamard product* [10]. E is the normal similarity matrix with entries e_{ij} defined in Eq. (1) and G is the density similarity matrix with entries defined in Eq. (8). D is a diagonal matrix with its (i,i)-element equal to the sum of the (i,i)-th row of W.

Intuitively, S is the final transition probability matrix.

Step 2: Regularization framework: Motivated by the *consistency assumption*, we infer the cost function associated with F is defined to be

$$Q(F)=\sum_{i=1}^{n}\sum_{x_j\in N(x_i)}s_{ij}(F_i-F_j)^2+\mu\sum_{i=1}^{n}(F_i-Y_i)^2 \qquad (10)$$

where $\mu>0$ is the regularization parameter.
Then the prediction function is $F^*=\arg\min_F Q(F)$. The first term of the right side of the cost function describes the total variation of the data labels with respect to the neighborhood structures, also called *smooth-term*. The second term is the *fitting constraint*, which means a good classifying function should not change too much from the initial label assignment.

Derivative of $Q(F)$ with respect to $F=(F_1,F_2,\cdots,F_n)^T$ and represent it with matrix form:

$$\frac{\partial Q(F)}{\partial F}=[(I-S)+(I-S)^T]F+2\mu(Y-F) \qquad (11)$$

Under certain conditions, we have $(I-S)F\approx LF$, where L is the *Laplacian-Beltrami* operator defined on the data manifold, and F is the function defined on this manifold [5]. Therefore

$$[(I-S)+(I-S)^T]F\approx 2LF\approx 2[(I-S)]F$$

Then we can get the approximating solution of minimizing the cost function by set Eq. (11) to zero

$$F^*=(1-a)(1-aS)^{-1}Y \qquad (12)$$

where $a=1/(1+\mu)$.
For classification, Eq. (12) is clearly equivalent to

$$F^*=(1-aS)^{-1}Y \qquad (13)$$

Now we can compute F^* directly, and our algorithm is summarized in Table 1.

Table 1. *Density-Sensitive Manifold Learning* (*DSML*)

1. Calculate the density of each data point and construct the density similarity matrix G by Eq. (8).
2. Construct the similarity matrix E using distance between points in the feature space by Eq. (1).
3. Construct the matrix $S=D^{-1}(E\bullet G)$ in which D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of $E\bullet G$.
4. Calculate $F^*=(1-aS)^{-1}Y$, and output the labels of each data object x_i by $y_i=\arg\max_{j\le c}F_{ij}^*$.

4. Connection to other *GBSSL* methods

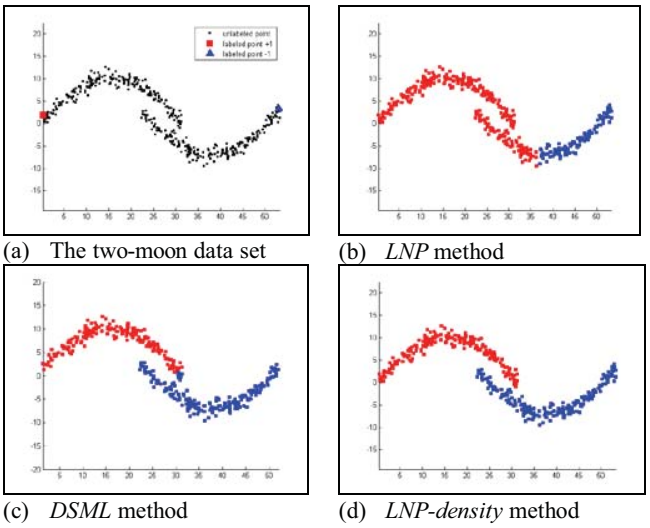
The basic idea of many traditional *GBSSL* methods is to construct a graph and get a smooth function. It is natural to consider improving an existing classifier by reconstructing the graph with the proposed density similarity matrix. In other words, we use the transition probability matrix given by other classifiers as the input of our algorithm. And table 2 shows the basic procedure of this *Density smooth method.*

Table 2. *Density smooth method*

1. Construct the (transition probability) matrix W using other *GBSSL* methods, such as *LNP* or *Consistency method*.
2. Construct the density similarity matrix G by Eq. (8), and then construct the matrix $S=D^{-1}(W\bullet G)$ in which D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of $W\bullet G$.
3. Calculate $F^*=(1-aS)^{-1}Y$, and output the labels of each data object x_i by $y_i=\arg\max_{j\le c}F_{ij}^*$.

5. Experiments

In this section, we test the performance of our method with both the toy-data and real-world data sets.



(a) The two-moon data set (b) *LNP* method (c) *DSML* method (d) *LNP-density* method

Figure 1. (a) A toy two-moon data set with the lower moon comes closer to the upper moon; (b) (c) (d) show the classification results by *LNP*, *DSML*, *LNP smooth with density* respectively.

5.1. Toy problem

To give an intuitive illustration, we adopt a two-moon example in which the points in the upper and lower moons are ambiguous. As shown in Fig. (1), the points in the bottom right of the upper moon and the bottom left of the lower moon are almost interweaved together. There are only one labeled point and 199 unlabeled points in each class. The classification results of *LNP*, *DSML* and *LNP-density* method are shown in Fig. (1).

As shown above, the *DSML* and its variant method are capable of learning the intrinsic structure of the data and thus specially surprised well for the ambiguous data set.

5.2. The real-world data sets

In this section, experiments have been performed on two real-world data sets: the USPS handwritten 16×16 digit recognition dataset and the COIL_100 image database [7]. In the USPS dataset, the images of digits 1, 2, 3 and 4 are used in our experiments as the four classes, and there are 1269, 929, 824 and 852 examples in each class, with a total of 3874. In the COIL_100 dataset, we adopted the top 50 objects with a total number of 3600 images (72 gray-scale images per object).

Here we compare the following algorithms: (1) *Consistency* method [2]; (2) *LNP* [5]; (3) *SSMR* [6]; (4) our *DSML* method; (5) *Consistency-density*: *Consistency* method smooth with the density (described in Table 2); (6) *LNP-density*: *LNP* smooth with density.

The neighborhood size in all these methods was set to 5, and the length scale of the Gaussian function in these methods had been tuned to achieve the best performance. In the *DSML* and its variant method, for a fixed point x_i together with its neighbors $x_j \in N(x_i)$, the parameter r in Eq. (8) was set as: $r = 1/|\max_{x_j \in N(x_i)}(\rho_i - \rho_j)|$. And the probability density was estimated by the distance-weighted Parzen window. The recognition accuracies averaged over 100 independent times are summarized in Figure 2, from which we can clearly see the advantage of our method.

It is interesting to discover that when there are only few labeled data in the data set, our *DSML* method is still very effective, while *SSMR* gets a low accuracy. We think this is because the density similarity function in *SSMR* isn't consistent with the *consistency assumption*. We also find that the *density smooth method* can improve other *GBSSL* methods not too much but very stable. We think the reason lies in twofold. First, the density of the dataset varies substantially across different clusters, the classification results may be affected when using non-normalized smoothness regularizer [4]. Second, the image data of different classes resides in different sub manifolds, and the density difference measure function (defined in Eq. (8)) can effectively reveal these manifold structures.

6. Conclusions

In this paper, we have proposed a novel *GBSSL* algorithm named *Density-Sensitive Manifold Learning,* in which both pairwise distances among nodes and density distribution of nodes are taken into account for graph construction. The main difference from existing methods lies in that a new density function is given for modeling density distribution as well as keeping the monotonicity of propagation. In addition, the proposed method can be easily transferred to traditional *GBSSL* methods to improve their performance. The experimental results show that the classification performance is remarkably improved after introducing proposed method.

7. References

[1] X. Zhu, Semi-Supervised Learning Literature Survey. Technical Report 1530, 2006.

[2] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Sch¨olkopf. Learning with Local and Global Consistency. *NIPS*, 2003.

[3] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. *ICML*, 2003.

[4] F. Wang and C. Zhang, Label Propagation through Linear Neighborhoods. *IEEE Trans. Knowledge and Data Engineering*, 2008.

[5] F. Wang and C. Zhang, Label Propagation through Linear Neighborhoods. *ICML*, 2006.

[6] J. Tang, X.-S. Hua, G.-J. Qi, M.Wang, T. Mei, and X.Wu, Structure-sensitive manifold ranking for video concept detection. *ACM Multimedia*, 2007.

[7] S. A. Nene, S. K. Nayar and H. Murase, Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, 1996.

[8] Bousquet, O., Chapelle, O., & Hein, M. (2004). Measure based regularization. *NIPS*, 2003.

[9] R. Duda, D. Stork, and P. Hart. Pattern Classification. JOHN WILEY, 2nd edition, 2000.

[10] R. A. Horn and C. R. Johnson. Matrix Analysis. Cambridge University Press (Reprint Edition), 1999.

[11] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000.

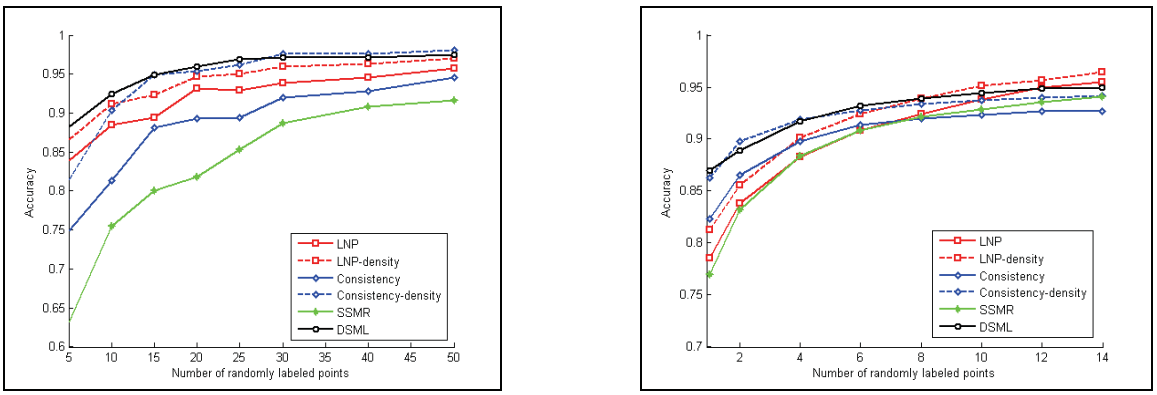


Figure 2. Left panel: the accuracy of digit recognition with USPS dataset, and abscissa represents the number of randomly labeled data in the training set (we guarantee that there is at least one labeled point for each class). Right panel: the accuracy of object recognition with COIL_100 dataset, and the abscissa represents the number of randomly labeled object images per subject.