Graph-Based Semi-Supervised Learning as a Generative Model

Jingrui He
Carnegie Mellon University
Machine Learning Department
5000 Forbes Avenue, Pittsburgh 15213
jingruih@cs.cmu.edu

Abstract

This paper proposes and develops a new graph-based semi-supervised learning method. Different from previous graph-based methods that are based on discriminative models, our method is essentially a generative model in that the class conditional probabilities are estimated by graph propagation and the class priors are estimated by linear regression. Experimental results on various datasets show that the proposed method is superior to existing graph-based semi-supervised learning methods, especially when the labeled subset alone proves insufficient to estimate meaningful class priors.

1 Introduction

In many real world classification tasks, the number of labeled instances is very few due to the prohibitive cost of manually labeling every single data point, while the number of unlabeled data can be very large since they are easy to obtain. Traditional classification algorithms, known as supervised learning, only make use of the labeled data, therefore prove insufficient in these situations. To address this problem, semi-supervised learning has been developed, which makes use of unlabeled data to boost the performance of supervised learning. In particular, graph-based semi-supervised learning algorithms have proved to be effective in many applications, such as hand-written digit classification [Zhu *et al*., 2003; Zhu *et al*., 2005], medical image segmentation [Grady and Funka-Lea, 2004], word sense disambiguation [Niu, Ji and Tan, 2005], image retrieval [He *et al*., 2004], etc.

Compared with other semi-supervised learning methods, such as TSVM [Joachims, 1999], which finds the hyperplane that separates both the labeled and unlabeled data with the maximum margin, graph-based semi-supervised learning methods make better use of the data distribution revealed by unlabeled data. In graph-based semi-supervised learning, a weighted graph is first constructed in which both the labeled and unlabeled data are represented as vertices. Then many of these methods can be viewed as estimating a function on the graph [Zhu, 2005]. Based on the assumption that nearby points in the feature space are likely to have the same label, the function is defined to be locally smooth and consistent with the labeled data. Finally, the classification labels are obtained by comparing the function value and a pre-specified threshold. For example, in the Gaussian random fields and harmonic function method, the learning problem is formulated in terms of a Gaussian random field on the graph, and the mean of the field serves as the function [Zhu *et al*., 2003]. Another example is the local and global consistency method, in which the function at each point is iteratively determined by both the information propagated from its neighbors and its initial label [Zhou *et al*., 2004]. Yet another example is the graph mincut method whose function corresponds to partitioning the graph in a way that roughly minimizes the number of similar pairs of examples that are given different labels [Blum and Chawla, 2001]. In the mincut method, the function can only take binary values.

Up till now, graph-based semi-supervised learning methods are generally approached from the discriminative perspective [Zhu, 2005] in that the function on the graph corresponds to posterior probabilities in one way or another. In the discriminative setting, however, the use of unlabeled data does not necessarily guarantee better decision boundaries. In addition, there is no clear explanation why the function on the graph should correspond to posterior probabilities from statistics point of view.

In this paper, we propose a new graph-based semi-supervised learning method from the generative model perspective. Specifically, the class conditional probabilities and the class priors are estimated from the weighted graph. The potential advantages involve several aspects: first, it can be theoretically justified that in the ideal cases where the two classes are separable, the output functions in terms of certain eigenvectors of the graph Laplacian converge to the class conditional probabilities as the number of training data goes to infinity. In non-ideal cases, our functions still provide a good estimate of the class conditional probabilities. Finally, the estimated class priors make use of both the labeled and unlabeled data, which compensate for the lack of label information in many practical situations. Experimental results show that our approach leads to better performance than other existing graph-based methods on a variety of datasets. Hence we can claim both stronger theoretical justification and better empirical results.

Compared with previous theoretical work on graph-based semi-supervised learning, such as [Hein *et al*., 2007], in

which the authors determined the pointwise limit of three different graph Laplacians used in the literature as the sample size increases and the neighborhood size approaches zero, and [Niyogi, 2008], in which the author exposed the natural structure of a class of problems on which manifold regularization methods are helpful, the major theoretical contribution of this paper is to relate the eigenvectors of the graph Laplacian to the class conditional probabilities. As far as we know, this is the first attempt in this line of research.

The rest of the paper is organized as follows. In Section 2 and Section 3, we introduce how to estimate the class conditional probabilities and the class priors respectively. Section 4 deals with the out-of-sample problem, followed by an outline of the algorithm in Section 5. Then the experimental results are shown in Section 6. Finally, we give conclusion and hint on future work in Section 7.

2 Estimating Class Conditional Probabilities

2.1 Notation

In a binary classification problem, suppose that we are given a set of n training examples: $x_1,\ldots x_n \in \mathbb{R}^d$. The first n_l examples are labeled, including n_{l1} positive ($y_i = 1, i = 1,\ldots,n_{l1}$) and $n_{l0} = n_l - n_{l1}$ negative ($y_i = 0, i = n_{l1}+1,\ldots,n_l$) examples. The remaining $n_u = n - n_l$ examples are unlabeled. Our goal is to predict the class labels of these n_u points by computing the posterior probability $P(y_i|x_i)$.

By Bayes rule, we have

$$P(y_i \mid x_i) = \frac{P(x_i \mid y_i) \times P(y_i)}{\sum_{y_i'=0,1} P(x_i \mid y_i') \times P(y_i')}, \tag{1}$$

y_i is predicted to be 1 iff $P(y_i = 1 \mid x_i) \geq 0.5$. In our generative model, in order to calculate $P(y_i \mid x_i)$, we need to estimate both $P(x_i|y_i)$ and $P(y)$. In this section, we focus on estimating the class conditional probability $P(x_i|y_i)$, and the estimation of $P(y)$ will be discussed in the next section.

We first form an affinity matrix $W \in \mathbb{R}^{n\times n}$ with $W_{ij} = \varphi(x_i, x_j)$, where $\varphi(x_i, x_j)$ is a non-negative function measuring the direct similarity between x_i and x_j. Then define D as the diagonal matrix, where $D_{ii} = \sum_{j=1}^{n} w_{ij}, i = 1,\ldots n$, and $S = D^{-1/2}WD^{-1/2}$. Finally define f^+ and f^- as two n-dimensional vectors. The element of f^+ (f^-) is set to 1 iff the corresponding point is a positive (negative) labeled one.

2.2 The Ideal Case

To start with, let us first consider the ideal case where the two classes are far apart. In this case, we have the following equation:

$$\begin{aligned} P(x) &= P(y=1)P(x|y=1) + P(y=0)P(x|y=0) \\ &\approx P(y_x)P(x|y_x) \end{aligned} \tag{2}$$

where y_x is the observed class label of data point x.

Based on this assumption, if x_i and x_j are from two different classes, the corresponding $W_{ij} = 0$. Therefore if we knew the labels of all the examples and put together the examples from the same class, the affinity matrix W, and thus the symmetric matrix S would be block-diagonal. To be specific, let

$$W = \begin{bmatrix} W_1 & 0 \\ 0 & W_0 \end{bmatrix} \tag{3}$$

where W_1 and W_0 represent the sub-matrices corresponding to the positive and negative examples respectively, and 0 represents zero matrix. If the total number of positive (negative) examples in the training set is n_1 (n_0), W_1 (W_0) is an $n_1 \times n_1$ ($n_0 \times n_0$) square matrix. Let D_1 and D_0 be two diagonal matrices, the diagonal elements of which are the row sums of W_1 and W_0. Then S can be written as

$$S = \begin{bmatrix} S_1 & 0 \\ 0 & S_0 \end{bmatrix} = \begin{bmatrix} D_1^{-1/2} W_1 D_1^{-1/2} & 0 \\ 0 & D_0^{-1/2} W_0 D_0^{-1/2} \end{bmatrix} \tag{4}$$

The following theorem connects the class conditional probabilities with the diagonal elements of D.

Theorem 1. If $\varphi(x_i, x_j) = \phi\left((x_i - x_j)/\sigma_n\right)/(\sigma_n)^d$, where σ_n is a positive parameter and the function $\phi(\cdot)$ satisfies the following conditions: $\phi(u) \geq 0$, $\int \phi(u)du = 1$, $\sup_u \phi(u) < \infty$, $\lim_{\|u\|\to\infty} \phi(u) \prod_{i=1}^{d} u_i = 0$, $\lim_{n\to\infty} \sigma_n = 0$, $\lim_{n\to\infty} n(\sigma_n)^d = \infty$, as the number of examples n goes to infinity, D_{ii}/n_{y_i} converges to $P(x_i|y_i)$.

The proof of the theorem is straightforward and therefore we put it in the appendix. Notice that this theorem is similar to a result in kernel density estimation. The difference is that in kernel density estimation, we only have labeled data from a single class; while in our situation, we have both labeled and unlabeled data, and we could estimate the class conditional distributions of the two classes at the same time.

Suppose that the labeled data are noise-free. According to Theorem 1, we can use D_{ii} to approximate the class conditional probability of x_i given the observed label y_i. However, for the unlabeled points, we do not know if D_{ii} corresponds to $P(x_i|y_i = 1)$ or $P(x_i|y_i = 0)$. To address this problem, we can make use of the eigenvectors of S.

It is easy to show that the largest eigenvalue of S_1 and S_0 is 1, and if W_1 and W_0 form a connected graph respectively, the corresponding eigenvectors would be $v_1 = D_1^{1/2} \cdot \vec{1}$ and $v_0 = D_0^{1/2} \cdot \vec{1}$ [Chung, 1997]. Based on v_1 and v_0, we can construct two eigenvectors of S with eigenvalue 1:

$$v^+ = \begin{bmatrix} v_1^T & \vec{0} \end{bmatrix}^T, v^- = \begin{bmatrix} \vec{0} & v_0^T \end{bmatrix}^T \quad (5)$$

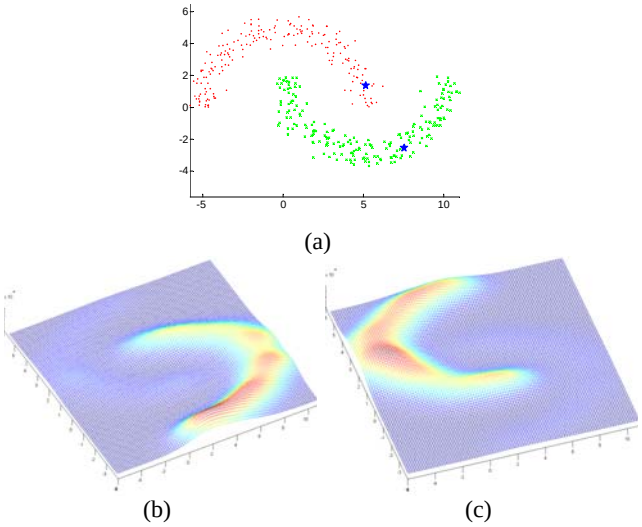
where $\vec{0}$ is a zero vector. Notice that if we square v^+ and v^- by elements to get $(v^+)^2$ and $(v^-)^2$, and then add them up, we get

$$(v^+)^2 + (v^-)^2 = D_1 \cdot \vec{1} + D_0 \cdot \vec{1} = D \cdot \vec{1} \quad (6)$$

Obviously, $(v^+)^2$ and $(v^-)^2$ correspond to $P(x_i|y_i=1)$ and $P(x_i|y_i=0)$ respectively, and their non-zero elements are equal to D_{ii}.

To get v^+ and v^-, we perform $f^+ \leftarrow S \cdot f^+$ and $f^- \leftarrow S \cdot f^-$ until convergence. Since the initial value of f^+ is not orthogonal to v^+ (the elements of f^+ and v_1 are non-negative), f^+ will converge to v^+. Similarly, f^- will converge to v^-. Therefore, upon convergence, $(f_i^+)^2$ ($(f_i^-)^2$) is in proportion to the class conditional probability of the positive (negative) class. After normalizing $(f_i^+)^2$ ($(f_i^-)^2$) so that it sums to 1, we have an empirical estimation of $P(x_i|y_i=1)$ ($P(x_i|y_i=0)$), which converges to its true value as n goes to infinity.

Figure 1 gives an example of density estimation in the ideal case. Figure 1(a) shows the training data, where the two moons represent two classes, and each class has one labeled example marked as star. Figure 1(b) and 1(c) show the estimated class conditional distributions of the two classes.



(a)

(b) (c)

Figure 1. Density Estimation in the Ideal Case. (a): training data; (b) and (c) class conditional distributions

2.3 The General Case

In the general cases, the two classes are not far apart, and we have the following theorem.

Theorem 2. If $\varphi(x_i, x_j)$ satisfies the conditions in Theorem 1, as the number of examples n goes to infinity, D_{ii}/n converges to $P(x_i|y_i=1)P(y=1) + P(x_i|y_i=0)P(y=0)$

The proof to this theorem is quite similar to Theorem 1. So we omit the details here. It can be seen easily that Theorem 1 is a special case of Theorem 2 when the two classes are far apart, i.e.

$$\begin{aligned} \lim_{n\to\infty} D_{ii}/n &\approx P(x_i|y_i=1)P(y=1), \text{ if } y_i = 1 \\ \lim_{n\to\infty} D_{ii}/n &\approx P(x_i|y_i=0)P(y=0), \text{ if } y_i = 0 \end{aligned} \quad (7)$$

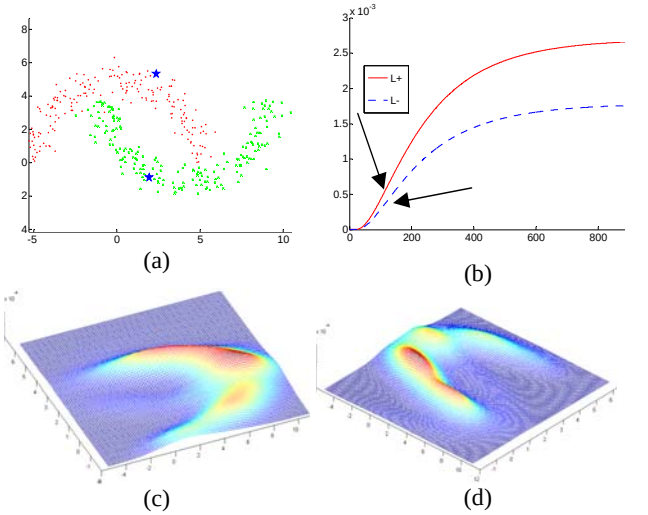
Equation (7), together with the fact that $\lim_{n\to\infty} n_1/n = P(y=1)$, leads to Theorem 1.

In the general cases, W tends to form one connected graph instead of two, and S only has one eigenvector that corresponds to eigenvalue 1. If we still iterate $f^+ \leftarrow S \cdot f^+$ and $f^- \leftarrow S \cdot f^-$ until convergence, both f^+ and f^- will converge to the same eigenvector. On the other hand, the operation of $f^+ \leftarrow S \cdot f^+$ and $f^- \leftarrow S \cdot f^-$ can be seen as the labeled data gradually spreading their information to nearby points. If the iteration steps are unlimited, every data point will be equally influenced by the positive and negative labeled data, leading to the same value of f^+ and f^-.

To solve this problem, in our algorithm, we have designed a stopping criterion, and the iteration process is stopped once the criterion is satisfied. To be more specific, when estimating the class conditional probabilities of the positive class, we could get an estimate of $P(x_i|y_i=1)$ in each iteration step (by normalizing $(f_i^+)^2$ so that it sums to 1). By summing up this probability for negative labeled examples, we have the average likelihood of these examples in the positive class: $L_+ = \left(\sum_{i=n_++1}^{n_l} P(x_i|y_i=1)\right)/n_-$. We stop the iteration when the second derivative of L_+ with respect to the iteration steps crosses 0. This criterion can be justified as follows: in the initial iteration steps, only a few negative data get positive score from their nearby positive labeled points, so the rate at which L_+ increases is very low; as the iteration proceeds, those negative data have accumulated high scores and propagate to the majority of negative points, so the rate gradually increases; finally, as f^+ begins to converge, its value at each data point becomes stable, so the rate decreases until it reaches 0. If we plot the curve of L_+ with respect to the number of iteration steps, the shape would be convex first, and then concave until convergence (Figure 2(b)). Notice that in the initial iteration steps, the positive points, which are far away from the positive labeled points but connected to them via some kind of manifold, cannot get positive scores. If the algorithm stops at this stage, it may not fully explore the data distribution and cause misclassification on certain clusters of data. Therefore we choose the transition point between convex and concave as the stopping point in order to trade off between prematurity and excessive propagation. The stopping criterion for the negative class can be derived similarly, i.e. $L_- = \left(\sum_{i=1}^{n_{l1}} P(x_i|y_i=0)\right)/n_{l1}$. A key point in our algorithm is that the estimation of the class conditional probabilities of the two classes is independent, i.e. the

numbers of iteration steps when the two stopping criterions are satisfied are not necessarily the same[1].

Figure 2 gives an example of density estimation in the general case showing the effectiveness of our criterion. This example is quite similar to the one shown in Figure 1 except that the two classes are not far apart. Figure 2(b) shows the value of L_+ (the upper curve) and L_- (the lower curve) in each iteration step. The arrows point to the positions in the curves where the two criterions are satisfied. Figure 2(c) and 2(d) show the estimated class conditional distributions of the two classes. Although there are small gaps in the middle of the distributions, the moon structure is recovered fairly well.



(a) (b) (c) (d)

Figure 2. Density Estimation in the Generation Case. (a): training data; (b): L_+ and L_- in each iteration; (c) and (d): class conditional distributions.

Note that the stopping criterion discussed above is based on simple heuristics. Currently we are trying to design a stopping criterion in a more principled manner.

3 Estimating Class Priors

In this section, we focus on estimating the class prior $P(y)$. Existing graph-based semi-supervised learning methods only use the labeled set to estimate the class priors, either explicitly [Zhu *et al.*, 2003] or implicitly [Zhou *et al.*, 2004]. Obviously, in real applications, the proportion of positive and negative labeled data is often far from the true class priors.

In our algorithm, we use both the labeled and unlabeled data to estimate the class priors. According to Theorem 2, once we have estimated the class conditional probability $P(x_i|y_i)$, we can feed them into the following equations and form a linear regression problem, the solution of which is equal to the least squares estimate of $P(y=1)$.

$$D_{ii}/n = P(x_i|y_i=1)\hat{p} + P(x_i|y_i=0)(1-\hat{p}),\ i=1,\ldots,n \qquad (8)$$

However, when the number of labeled data is small, the estimated class conditional probabilities may not be very accurate, and thus $\hat{p}$ is not very reliable. To solve this problem, we use a beta distribution as the prior distribution for $P(y=1)$, the parameters of which are $\hat{p}$ and $1-\hat{p}$. Then the estimate of $P(y=1)$ based on the labeled set:

$$P(y=1) = \frac{\hat{p} + n_{l1}}{1+n_l},\ P(y=0) = 1 - P(y=1) \qquad (9)$$

which is equivalent to smoothing the proportion of the positive and negative examples in the labeled set. When the number of labeled data is small, unlabeled data can be fully exploited to compensate for the proportion in the labeled set that is not the same as the class priors; when the number of labeled data is large, labeled data will dominate the estimation of the class priors.

4 Prediction of New Testing Data

To classify a data point $x \in \mathbb{R}^d$ that is not present during the training stage, we first calculate its class conditional probabilities via the following formula:

$$P(x|y) = \sum_{i=1}^{n} \varphi(x, x_i) \cdot P(x_i|y) \qquad (10)$$

Based on the conditions in Theorem 1, we have $\int \varphi(x, x_i)dx = \int \phi\big((x-x_i)/\sigma_n\big)\big/(\sigma_n)^d \cdot d(x-x_i) = 1$;

$\int p(x|y)dx = \int \sum_{i=1}^{n} \varphi(x, x_i) \cdot p(x_i|y)dx = \sum_{i=1}^{n} p(x_i|y) = 1$.

Therefore, $p(x|y)$ is a valid probability distribution.

Using these class conditional probabilities and the class priors obtained during the training stage, we can calculate the posterior probability and make a prediction.

5 The Algorithm

The procedures for estimating $p(x_i|y_i)$ and $P(y)$ are summarized in Table 1 and Table 2 respectively.

6 Experimental Results

In this section, we present the comparative experimental results on two datasets: Cedar Buffalo binary digits database [Hull, 1994], and a document genre-classification dataset [Liu *et al.*, 2003]. Our algorithm is compared with two other graph-based semi-supervised learning methods: Gaussian random fields [Zhu *et al.*, 2003] and the local and global consistency method [Zhou *et al.*, 2004]. We did not compare with supervised learning methods, such as one nearest neighbor, since they have been proved to be less effective than Gaussian random fields based on experimental results [Zhu *et al.*, 2003].

We have designed two kinds of experiments: balanced and unbalanced. In the balanced case, the ratio of labeled points from each class is always the same as the class priors; in the unbalanced case, if not explained otherwise, we fix the total number n_l of labeled points, and perturb the number of

[1] We have also tried other stopping criterions, such as the one that stops when the first derivative of L_+ (L_-) crosses 0, the one that stops when L_+ (L_-) exceeds a certain threshold, etc. The current criterion performs the best among the different stopping criterions; hence it is used in our algorithm.

positive labeled points around $n_l/2$ with a Gaussian distribution of mean 0 and standard deviation $n_l/10$. In each experiment, we gradually increase the number of labeled data, perform 20 trials for each labeled data volume, and average the accuracy at each volume point.

1. Form the affinity matrix $W \in \mathbb{R}^{n\times n}$, where $W_{ij} = \varphi(x_i, x_j)$. Calculate D and S.
2. Initialize f^+ and f^-. The element of f^+ (f^-) is set to 1 if the corresponding point is a positive (negative) labeled one, and 0 otherwise.
3. Update $f^+ \leftarrow S \cdot f^+$, $f^- \leftarrow S \cdot f^-$.
4. Assign $P(x_i|y_i=1) = (f_i^+)^2$, $P(x_i|y_i=0) = (f_i^-)^2$, and normalize so that $\sum_{i=1}^{n} P(x_i|y_i=1) = 1$, $\sum_{i=1}^{n} P(x_i|y_i=0) = 1$.
5. Calculate the average likelihood of negative (positive) labeled points in the positive (negative) class: $L_+ = \left(\sum_{i=n_{l1}+1}^{n_l} P(x_i|y_i=1)\right)/n_{l0}$ ($L_- = \left(\sum_{i=1}^{n_{l1}} P(x_i|y_i=0)\right)/n_{l1}$). Go to step 4 unless one of the following conditions is satisfied:
   a. L_+ (L_-) remains at 0, and f^+ (f^-) has converged;
   b. L_+ (L_-) does not remain at 0, and the second derivative of L_+ (L_-) with respect to the iteration steps crosses 0.
6. Output $P(x_i|y_i=1)$ and $P(x_i|y_i=0)$.

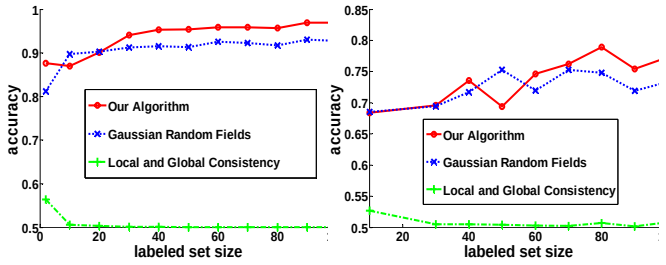
Table 1. Description of Estimation for $p(x_i|y_i)$

1. Solve the following linear regression problem for the least squares estimator $\hat{p}$ of $P(y=1)$:
   $$\hat{p} \cdot P(x_i|y_i=1) + (1-\hat{p}) \cdot P(x_i|y_i=0) = D_{ii}/n, i = 1,\ldots,n$$
2. Calculate the class priors as the smoothed proportion of the positive and negative examples in the labeled set
   $$P(y=1) = \frac{\hat{p} + n_{l1}}{1+n_l}, P(y=0) = 1 - P(y=1)$$

Table 2. Description of Estimation for $P(y)$

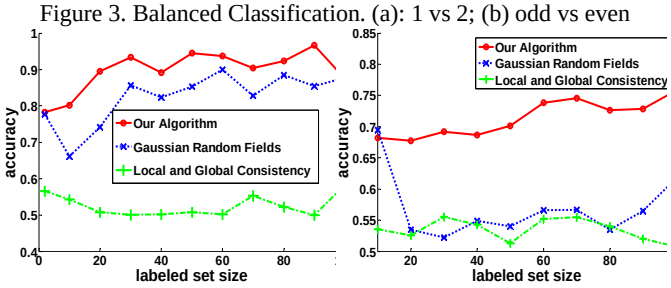
6.1 Cedar Buffalo Binary Digits Database

We first perform experiments on Cedar Buffalo binary digits database [Hull, 1994] including two classification tasks: classifying digits "1" vs "2", with 1100 images in each class; and odd vs even digits, with 2000 images in each class (400 images for each digit). The data we use are the same as those used in [Zhu *et al.*, 2003]. Here $\varphi(x_i, x_j) = (2\pi\sigma^2)^{-d/2} \exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)$, where σ is the average distance between each data point and its 10 nearest neighbors.



(a) (b)

Figure 3. Balanced Classification. (a): 1 vs 2; (b) odd vs even



(a) (b)

Figure 4. Unbalanced Classification. (a): 1 vs 2; (b) odd vs even

Figure 3(a) and 3(b) show the results of the two classification tasks in the balanced case. The performance of our algorithm is comparable with Gaussian random fields, and both of them are much better than the local and global consistency method. Figure 4(a) and 4(b) show the results in the unbalanced case. In this situation, the performance of Gaussian random fields is much worse than in the balanced case, while the performance of our algorithm is comparable to the balanced case. This is because the class mass normalization procedure adopted in Gaussian random fields depends on the labeled set only to estimate the class priors; while our algorithm makes use of both the labeled and the unlabeled set to estimate the class priors. Therefore, it is more robust against the perturbation in the proportion of the positive and negative data in the labeled set.

6.2 Genre Dataset

Genre classification is to classify the documents based on its writing styles, such as political articles and movie reviews. The genre dataset that we use consists of documents from 10 genres, including biographies (b), interview scripts (is), movie reviews (mr), product reviews (pr), product press releases (ppr), product descriptions on store websites (pd), political articles on newspapers (pa), editorial papers on politics (ep), news (n), and search results from multiple search engines using 10 queries (sr). We randomly select 380 documents from each category to compose the whole dataset of 3800 documents. Each document is processed into a "tf.idf" vector, which is generated based on the top 10,000 most frequent words in this dataset after stemming, with the header and stop words removed. Here $\varphi(x_i, x_j) = \exp\left(-\left(1 - (x_i \cdot x_j)/(\|x_i\|\|x_j\|)\right)/0.03\right)$, which is borrowed from [Zhu *et al.*, 2003] and roughly measures the similarity between documents. The only difference is that we keep all the edges instead of keeping edges for only 10

nearest neighbors. Next we perform experiments to compare the three algorithms. The results are provided in Figure 5 and Figure 6 respectively.

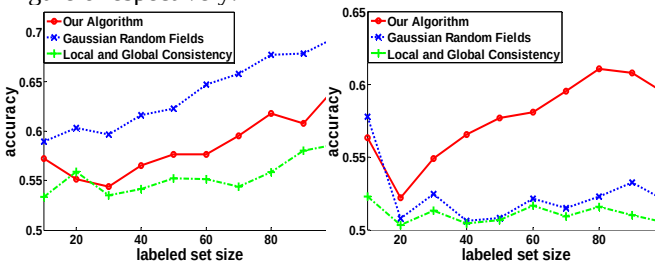


Figure 5. Classification between Random Partitions. (a): balanced; (b): unbalanced

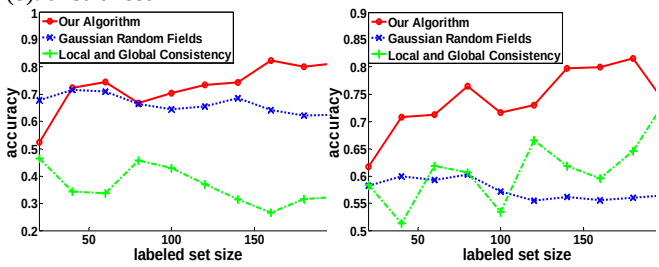


Figure 6. Unbalanced Classification. (a): pa vs other; (b) b vs other

For Figure 5, we randomly partition the 10 categories into two classes, i.e. pa, pr, sr, b, and is, vs mr, ppr, pd, ep and n. Figure 5(a) and 5(b) correspond to the balanced and unbalanced cases respectively. In the balanced case, Gaussian random fields is better than our algorithm and the local and global consistency method. This might be because the function $\varphi(x_i, x_j)$ does not have some of the nice properties required by Theorem 2. However, in the unbalanced case, Gaussian random fields tends to suffer a lot. On the contrary, our algorithm is quite robust despite of the perturbation.

In Figure 6, we try to classify pa and b against all the other categories. In these experiments, the class priors are 0.1 for the positive class and 0.9 for the negative class. However, here we provide equal numbers of positive and negative points in the labeled set. From the figures, we can see that the performance of our algorithm is rather stable, while the performance of both Gaussian random fields and the local and global consistency method is largely affected by the misleading labeled set, since they only depend on the labeled set to estimate the class priors, either explicitly or implicitly.

7 Discussion

7.1 Objective Function

It can be shown that in the ideal case, the two functions f^+ and f^- maximize the following objective function with the constraints that $\|f^+\|=1$ and $\|f^-\|=1$.

$$\infty\left(f^+\right)^T Sf^+ + \sum_{i=1,y_i=1}^{n_l} f_i^+ + \infty\left(f^-\right)^T Sf^- + \sum_{i=1,y_i=0}^{n_l} f_i^- \quad (11)$$

where ∞ in front of the first and third terms means that they have arbitrarily large weights. However, it is not quite clear how this objective function is related to the outputs of our algorithm in Table 1 for the general case. Right now, we are working in this direction.

7.2 Generalization to Multiple Classes

The proposed algorithm can be easily generalized to multiple classes. In the binary case, as mentioned in subsection 2.3, the estimation of the class conditional probabilities of the two classes is independent. Following the same line of reasoning, when we have multiple classes, we can use the labeled data to estimate the class conditional probability for each class in the same way as in Table 1. On the other hand, to estimate the class priors, we can formulate a similar linear regression problem as in Section 3, and get the least squares estimates of the class priors in the same way as in Table 2.

8 Conclusion and Future Work

In this paper, we propose a novel graph-based semi-supervised learning method to estimate both the class conditional probabilities and the class priors. It is a generative model, in contrast to existing graph-based methods, which are essentially discriminative. In the ideal case, the estimated class conditional probabilities have been proved to converge to the true value. In the general case, our algorithm can still output reasonable estimates of the class conditional probabilities. For data points outside the training set, the class conditional probabilities are estimated via kernel regression. When estimating the class priors, we effectively use the unlabeled data to make up for the labeled data with unrepresentative class prior distributions. Experimental results on two datasets demonstrate the superiority of our algorithm over recent existing graph-based semi-supervised learning methods, especially when the proportion in the labeled set is not the same as the class priors.

In our experiments, we notice that in some cases, adding even a single labeled point into the labeled set brings about significant improvement in classification accuracy; while in other cases, adding many labeled points into the labeled set does not help improve the performance. Currently we are incorporating active learning into our framework. Particularly, we are interested in determining when to invoke active learning (not just which instances to label) in order to achieve the biggest gain while minimizing incremental labeling cost.

References

[Blum and Chawla, 2001] Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. *Proc. 18th Int. Conf. on Machine Learning, pp. 19-26.*

[Chung, 1997] Chung, F. R. K. (1997). *Spectral graph theory, regional conference series in mathematics.* American Mathematical Society.

[Grady and Funka-Lea, 2004] Grady, L., & Funka-Lea, G. (2004). Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. *Proc. 8th European Conf. on Computer Vision, workshop on Computer Vision Approaches to Medical Image Analysis and Mathematical Methods in Biomedical Image Analysis.*

[He *et al*., 2004] He, J., Li, M., Zhang, H. J., Tong, H., & Zhang, C. (2004). Manifold-ranking based image retrieval. *Proc. 12th ACM Int. Conf. on Multimedia, pp. 9-16.*

[Hein *et al*., 2007] Hein, M., Audibert, J.Y., & Luxburg, U. (2007). *Graph Laplacians and their convergence on random neighborhood graphs.* The Journal of Machine Learning Research, vol. 8, pp. 1325-1370.

[Hull, 1994] Hull, J. J. (1994). *A database for handwritten text recognition research.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, pp. 550-554.

[Joachims, 1999] Joachims, T. (1999). Transductive Inference for text classification using Support Vector Machines. *Proc. 16th Int. Conf. on Machine Learning, pp. 200-209.*.

[Liu *et al.*, 2003] Liu, Y., Carbonell, J., & Jin, R. (2003). A New Pairwise Ensemble Approach for Text Classification. *Proc. 14th European Conf. on Machine Learning.*

[Niu, Ji and Tan, 2005] Niu, Z. Y., Ji, D. H., & Tan, C. L. (2005). Word sense disambiguation using label propagation based semi-supervised learning. *Proc. 43rd Annual Meeting of the Association for Computational Linguistics, pp. 395-402.*

[Niyogi, 2008] Niyogi, P. (2008). Manifold regularization and semi-supervised learning: some theoretical analysis. *Technical Report TR-2008-01, Computer Science Dept., University of Chicago.*

[Zhou *et al*., 2004] Zhou, D., bousquet, O., Lal, T., Weston, J., & Schlkopf, B. (2004). Leaning with local and global consistency. *Proc. 18th Annual Conf. On Neural Information Processing Systems.*

[Zhu *et al.*, 2003] Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Proc. 20th Int. Conf. on Machine Learning, pp. 912-929.*

[Zhu *et al*., 2005] Zhu, X., & Lafferty, J. (2005). Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. *Proc. 22th Int. Conf. on Machine Learning, pp. 1052-1059.*

[Zhu, 2005] Zhu, X. (2005). *Semi-supervised learning with graphs.* Doctoral dissertation, School of Computer Science, Carnegie Mellon University.

Appendix

Proof of Theorem 1: suppose x_i is from the positive class:

$$\begin{aligned}
&\lim_{n\to\infty} D_{ii}/n_1 = \lim_{n\to\infty} \sum_{j=1}^{n} W_{ij} \Big/ n_1 \\
&= \lim_{n\to\infty} \sum_{j=1}^{n_1} \phi\big((x_i - x_j)/\sigma_n\big) \Big/ (n_1 V_n) \\
&\to E_{x|Y=1}\big[\delta(x_i, x)\big] \\
&= \int \delta(x_i, x)\, p(x|y=1)\, dx \\
&= p(x_i|y=1)
\end{aligned} \tag{11}$$

Equation (11) reduces the number of terms in the summation from n to n_1 since $W_{ij} = 0$ if x_j is from the negative class. $\delta(x_i, x_j)$ is a delta function at $x_i = x_j$. A corresponding proof applies if x_i is from the negative class.