Large-Scale Adaptive Semi-Supervised Learning via Unified Inductive and Transductive Model

[Extended Abstract]

De Wang *
University of Texas at Arlington
Department of Computer Science and Engineering,
500 UTA Boulevard
Arlington, Texas,76019-0015
wangdelp@gmail.com

Feiping Nie *
University of Texas at Arlington
Department of Computer Science and Engineering,
500 UTA Boulevard
Arlington, Texas,76019-0015
feipingnie@gmail.com

Heng Huang †
University of Texas at Arlington
Department of Computer Science and Engineering,
500 UTA Boulevard
Arlington, Texas,76019-0015
heng@uta.edu

ABSTRACT

Most semi-supervised learning models propagate the labels over the Laplacian graph, where the graph should be built beforehand. However, the computational cost of constructing the Laplacian graph matrix is very high. On the other hand, when we do classification, data points lying around the decision boundary (boundary points) are noisy for learning the correct classifier and deteriorate the classification performance. To address these two challenges, in this paper, we propose an adaptive semi-supervised learning model. Different from previous semi-supervised learning approaches, our new model needn't construct the graph Laplacian matrix. Thus, our method avoids the huge computational cost required by previous methods, and achieves a computational complexity linear to the number of data points. Therefore, our method is scalable to large-scale data. Moreover, the proposed model adaptively suppresses the weights of boundary points, such that our new model is robust to the boundary points. An efficient algorithm is derived to alternatively optimize the model parameter and class probability distribution of the unlabeled data, such that the induction of classifier and the transduction of labels are adaptively unified into one framework. Extensive experimental results on six real-world data sets show that the proposed semi-supervised learning model outperforms other related methods in most cases.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

General Terms

Algorithms

* Authors contributed equally to this paper. † Corresponding author.

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623731 .

Keywords

Semi-supervised learning; Large-scale semi-supervised learning; Unified inductive and transductive model

1. INTRODUCTION

In most data mining applications, the data are generally abundant, however, labeled data is often scarce. Labeling data is a tedious work, and costs huge amount of time and money. In this situation, how to fully utilize the abundant unlabeled data becomes very important.

Semi-supervised learning is a learning paradigm that suits for this situation, where both the labeled data and unlabeled data are used to learn the prediction model. There are two types of semi-supervised learning models: transductive learning models and inductive learning models. Transductive semi-supervised learning methods learn the labels of unlabeled data by propagating the label from labeled data to unlabeled data. The drawback of this kind of methods is that they can not be used for out-of-sample testing (*i.e.*, new testing data from not included in the unlabeled data). So when a new testing data arrives, such methods need to merge those new testing data into the previous data we have, and then reconstruct the whole model based on the merged data. Obviously, such a way is very inefficient for the testing of new out-of-sample data.

Inductive semi-supervised learning methods learn a classifier using both labeled data and unlabeled data. Then the learned classifier can be used for the classification of both unlabeled data using for training and also new out-of-sample testing data. In view of the convenience of out-of-sample testing, inductive semi-supervised learning methods are more attractive in practice.

Many graph based learning approaches have been proposed in recent years [1, 3, 5, 6, 13, 14, 15, 20]. Some of the most representative graph based semi-supervised learning models are: local and global consistency (LGC) [18], random walk (RW) [19], and gaussian field harmonic function (GFHF) [21], Laplacian regression [12], and semi-supervised discriminant analysis [2]. All of these models utilize the Laplacian graph and propagate the labels over the graph. Therefore, in order to use these models, an $n \times n$ graph Laplacian matrix has to be built beforehand.

However, the computational cost of building the $n \times n$ graph Laplacian matrix is at least $O(n^2)$. Such computational cost is daunting in the circumstance of large-scale data, where the number of data can easily reach billion level. So such graph based algorithms are not scalable to large-scale data.

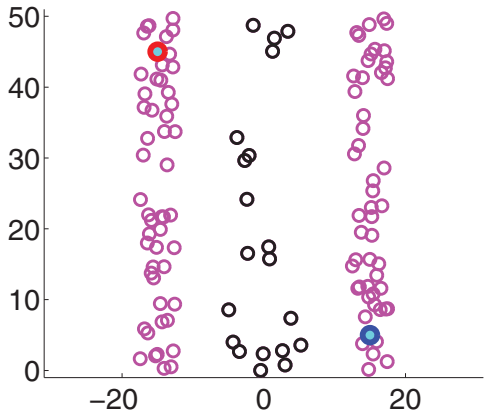


Figure 1: Data samples with boundary points (the black circles). The red circle point and the blue circle point are the labeled data points. All the rest data are unlabeled.

On the other hand, when we do classification, there are many data points lie around the decision boundary, which we call boundary points in the paper. These data points are very noisy for learning the correct classifier, and thus will deteriorate the classification performance of the learned classifier.

To address the above two challenges, a large-scale adaptive semi-supervised learning model is proposed in this paper. The proposed semi-supervised learning model has many good characteristics which will be discussed in detail after we introduce the model.

2. NEW LARGE-SCALE SEMI-SUPERVISED LEARNING MODEL

2.1 Motivation

Most semi-supervised learning methods are based on graph Laplacian matrix, like in the following works: local and global consistency (LGC) [18], random walk (RW) [19], and gaussian field harmonic function (GFHF) [21], Laplacian regression [12], and flexible manifold embedding (FME) [8].

One major drawback of such kind of methods is that the computational cost of constructing the $n \times n$ graph Laplacian matrix is very high, which is at least $O(n^2)$. What's more, if we use the gaussian kernel to construct the graph, the bandwidth parameter σ should be tuned carefully in order to achieve good performance. This makes the Laplacian graph based semi-supervised learning methods impractical to solve large-scale applications in which the number of samples n is often more than billion. If we can develop semi-supervised learning methods without constructing the graph Laplacian matrix, most of the computational cost can be avoided, such that the method can be applied to large-scale data.

On the other hand, when we do classification, there exist many *bad* data points that lie around the decision boundary. Consider the situation demonstrated in Figure 1. The red circle point and the blue circle point are the labeled data points. Considering the distribution of the data, the ideal decision boundary to classify these data points into two classes should be close to the vertical line $x = 0$. However, there are many data points lie around the decision boundary (the black circles in the figure). We call these points boundary points in the following. These boundary points will blur the clear distribution of the whole data, and are very noisy for learning the correct classifier. If these boundary points dominated the loss function, the learned classifier maybe distorted far way from the ground truth. In the example of Figure 1, if we consider too much of these boundary points, the learned classifier may become a horizontal line close to $y = 25$. If we do not consider these boundary points, and consider only the remaining clearly classified points, the correct vertical decision boundary can be easily discovered. Therefore, in order to learn the correct classifier, boundary points should be considered less. In fact, boundary points widely existed in all data sets. Especially when the class number of the data is large, boundary points existed around every decision boundary of each two classes. Therefore, it is very important to develop algorithms that is adaptive to boundary points. By adaptive, we mean that the algorithm can automatically distinguish the boundary points and clearly classified points, and pay less attention to boundary points while learning the classifier.

To address the above two important challenges, in this paper, we propose a new semi-supervised learning model.

2.2 Adaptive Semi-Supervised Learning

In [7], the label is obtained by the label propagation procedure and then used in a regression model. Inspired by [7], in this paper, we propose a new adaptive semi-supervised learning model where the label matrix Y is used as weights and optimized simultaneously with W. Our new model aims to solve the following objective function:

$$\min_{W,b,Y} \left\|X_l^T W + \mathbf{1}_{nl} b^T - Y_l\right\|_F^2 + \sum_{i=1}^{n}\sum_{k=1}^{c} y_{ik}^r \left\|x_i^T W + b^T - t_k^T\right\|_F^2$$
$$s.t.\ \ \forall i, y_{ik} \in [0,1], \sum_{k=1}^{c} y_{ik} = 1 \qquad (1)$$

where $X_l \in \Re^{d\times nl}$ is the labeled data set, nl is the number of labeled data, d is the number of feature, $\mathbf{1}_{nl}$ is a column vector of size nl whose elements are all one, $W \in \Re^{d\times c}$ is the model parameter matrix that need to be learned, c is the number of classes, $b \in \Re^{c\times 1}$ is the regression bias, $Y_l \in \Re^{nl\times c}$ is the label of labeled data; y_{ik} is the probability of the i-th unlabeled data belongs to the k-th class, which should be in a value between [0,1], $Y \in \Re^{n\times c}$ is the matrix formed by y_{ik}, which should also be learned along with W, n is the number of unlabeled data, $x_i \in \Re^{d\times 1}$ is the i-th unlabeled data, $t_k \in \Re^{c\times 1}$ is a class indicator vector for k-th class, where the k-th element of t_j: $t_{jk} = 1$, and the rest elements are zeros. r is an adaptive parameter that need to be tuned, and $r \geq 1$.

The first term in the objective function is the total loss of labeled data, the second term is the loss of unlabeled data, weighted by the probability distribution matrix Y. There is only one parameter, *i.e.* r, in our proposed ASL model. In our further analysis in the experiment section, we will show that the range of r can be fixed, and a reasonable interval is suggested.

It is interesting to note that the parameter actually serves for multiple purposes: **from the macro level, r balance the two terms in the objective function, decides how much unsupervised information is used; from the micro level, after r is fixed, the weights of boundary points will be suppressed to make the model robust to boundary points.** Following are some more detailed analysis for the conclusion.

From a macro point of view, r serves as a tradeoff between the first term (supervised part with label information available) and the second term (unsupervised part without label information). Note that we do not need another tradeoff parameter before the second term because r is able to balance the two terms. When r becomes large, y_{ik}^r will become small since it is a probability which is definitely less than 1, thus the weight of the second term become small.

In the extreme case, when r approaches infinity, the second term vanishes to zero, so only the first term counts, which means the objective function reduces to supervised learning.

From a micro point of view, r automatically adjust the importance (weight: y_{ik}^r in the objective function Eq. (1)) of each data instance. Consider the i-th data instance: if it is clearly classified, y_{ik} ($k = 1, ..., c$, the probability for x_i belongs to different class) will show obvious magnitude difference. For boundary points, however, y_{ik} ($k = 1, ..., c$) will be more likely equal to one another. Without loss of generality, assuming a binary classification problem, and $r = 2$. For clearly classified points, y_{i1} and y_{i2} would be one large and one small, say $y_{i1} = 0.9$ and $y_{i2} = 0.1$, then $y_{i1}^r = 0.9^2 = 0.81$, $y_{i2}^r = 0.1^2 = 0.01$. Thus, those clear classified points still have large weights in total and contribute a lot to the second term in the objective function. For boundary points, however, y_{i1} and y_{i2} would be more likely equal. Assuming $y_{i1} = y_{i2} = 0.5$, then $y_{i1}^r = y_{i2}^r = 0.5^2 = 0.25$. Thus, those boundary points will have small weights and contribute much less to the second term in the objective function. The above analysis is based on binary classification. Actually, in multi-class problems, for boundary points, y_{ik} ($k = 1, ..., c$) may tend to be $\frac{1}{c}$. Therefore, when c becomes larger, y_{ik} becomes smaller, and y_{ik}^r will be much smaller than the case in binary classification.

From the above analysis, we can see that: when r increases, y_{ik}^r will decrease for both clearly classified points and boundary points. However, the weights of boundary points are suppressed much more faster than clearly classified points. In this way, our model will relatively pay more attention to the clearly classified points and pay less attention to the boundary points. This makes our model adaptive and robust to boundary points, therefore, a better classifier can be learned.

It is innovative to use the class probability matrix of unlabeled data to do inductive learning. In our model, the induction of the classifier W is dependent on the class probability matrix Y of unlabeled data, and the transduction of labels to unlabeled data is dependent on the classifier W. The two steps are unified together by simultaneously optimize W and Y. This adaptive procedure is expected to benefit both the induction of classifier and also the transduction of labels to unlabeled data. However, in previous inductive semi-supervised learning methods, only the feature matrix X of unlabeled data is utilized to learn the classifier. And in those previous methods, an additional step is needed after the model is learned in order to predict the label of unlabeled data.

We summarize the characteristics of our model as following:

(1) **Computational efficient**: different from common graph-based semi-supervised learning methods, our model avoids the computational expensive step of constructing the graph Laplacian matrix.

(2) **Adaptive and robust to boundary points**: our model can adaptively adjust the weights of each data point. Boundary points will get much smaller weights than other points. This makes our model pay less attention to boundary points, and thus robust to boundary points.

(3) **Adaptive optimization procedure**: we simultaneously optimize the model parameter W and the class probability matrix Y of unlabeled data. This adaptive procedure is expected to benefit both the induction of classifier and also the transduction of labels to unlabeled data.

(4) **Only one parameter**: there is only one parameter to be tuned in our model, and the single parameter serves for multiple purposes. A reasonable range of the parameter can be given to facilitate the tuning procedure.

In the following section, an efficient iterative algorithm will be derived to solve the proposed objective function.

3. OPTIMIZATION ALGORITHM

In this section, we derive an efficient iterative algorithm that alternatively optimize over the model parameter W, b and the class probability matrix Y.

(1) When the model parameters W and b are fixed, we solve the class probability matrix Y. Since the first term becomes a constant, the objective function is reduced to:

$$\min_{Y} \sum_{i=1}^{n} \sum_{k=1}^{c} y_{ik}^r \left\| x_i^T W + b^T - t_k^T \right\|_2^2$$
$$s.t. \qquad \forall i, y_{ik} \in [0,1], \sum_{k=1}^{c} y_{ik} = 1 \tag{2}$$

Denote $p_{ik} = \left\| x_i^T W + b^T - t_k^T \right\|_2^2$, Eq. (2) can be written as:

$$\min_{Y} \sum_{i=1}^{n} \sum_{k=1}^{c} y_{ik}^r p_{ik}$$
$$s.t. \quad \forall i, y_{ik} \in [0,1], \sum_{k=1}^{c} y_{ik} = 1 \tag{3}$$

Obviously, Eq. (3) can be decoupled between samples. So it is equivalent to solving:

$$\min_{y_{i\cdot}} \sum_{k=1}^{c} y_{ik}^r p_{ik}$$
$$s.t. \quad \forall i, y_{ik} \in [0,1], \sum_{k=1}^{c} y_{ik} = 1 \tag{4}$$

where $y_{i\cdot}$ represents the i-th row of Y.

When $r = 1$, obviously, the optimal solution of Eq. (4) is:

$$y_{ik} = 1, \text{if } \ k = k^*;$$
$$y_{ik} = 0, \text{if } \ k \neq k^*; \tag{5}$$

where $k^* = \arg\min_k p_{ik}$.

When $r > 1$, we solve Eq. (4) in the following way. The Lagrangian function of Eq. (4) is:

$$\sum_{k=1}^{c} y_{ik}^r p_{ik} - \beta(\sum_{k=1}^{c} y_{ik} - 1) \tag{6}$$

where β is the Lagrangian multiplier. In order to get the optimal solution of the subproblem, we set the derivative of Eq. (6) with respect to y_{ik} to zero. Thus, we get:

$$y_{ik} = (\frac{\beta}{r p_{ik}})^{\frac{1}{r-1}} . \tag{7}$$

Substituting Eq. (7) into the constraint $\sum_{k=1}^{c} y_{ik} = 1$, we get the closed form solution of Y as following:

$$y_{ik} = (\frac{1}{p_{ik}})^{\frac{1}{r-1}} / \sum_{k=1}^{c} (\frac{1}{p_{ik}})^{\frac{1}{r-1}} \tag{8}$$

(2) When the class probability matrix Y is fixed, we solve the model parameter W and b. Note that the second term in the objective function in Eq. (1) sums over the number of unlabeled points n and the number of classes c. If we directly taking the derivative of the second term with respect to W, the resulting algorithm would need to iterate over n and c, which would be slow.

It is interesting to note that we can write the objective function in Eq. (1) into compact matrix representation in the following way:

$$\min_{W,b} \left\| X_l^T W + \mathbf{1}_{nl} b^T - Y_l \right\|_F^2 - 2Tr(F(W^T X + b\mathbf{1}_n^T))$$
$$+ Tr(W^T X + b\mathbf{1}_n^T) S (W^T X + b\mathbf{1}_n^T)^T \tag{9}$$

where $F = Y^r \in \Re^{n\times c}$ (here Y^r denotes to perform power operation on each element in Y), $S \in \Re^{n\times n}$ is a diagonal matrix with the i-th diagonal element equal to $s_{ii} = \sum_{k=1}^{c} F_{ik}$.

Setting the derivative of Eq. (9) with respect to b to zero, we get:

$$b = q[(Y_l^T - W^T X_l)\mathbf{1}_{nl} + (F^T - W^T XS)\mathbf{1}_n] \quad (10)$$

where q is a scalar, $q = \frac{1}{nl+\mathbf{1}_n^T S\mathbf{1}_n}$.

Setting the derivative of Eq. (9) with respect to W to zero, and plug in Eq. (10), we get:

$$[X_l(I_{nl} - Q_1)X_l^T + X(I_n - Q_2)SX^T - X_lQ_3X^T - XQ_4X_l^T]W = X_l(I_{nl} - Q_1)Y + X(I_n - Q_2)F - X_lQ_3F - XQ_4Y \quad (11)$$

where $Q_1 = q\mathbf{1}_{nl}\mathbf{1}_{nl}^T$, $Q_2 = qS\mathbf{1}_n\mathbf{1}_n^T$, $Q_3 = q\mathbf{1}_{nl}\mathbf{1}_n^T$, $Q_4 = q\mathbf{1}_n\mathbf{1}_{nl}^T$.

Denote $C = X_l(I_{nl}-Q_1)X_l^T+X(I_n-Q_2)SX^T-X_lQ_3X^T-XQ_4X_l^T$, and $A = X_l(I_{nl} - Q_1)Y + X(I_n - Q_2)F - X_lQ_3F - XQ_4Y$, then the optimal solution of W is:

$$W = C^{-1}A \quad (12)$$

The optimization procedure is summarized in Algorithm 1. The two step optimization procedure alternatively optimize the model parameter W and the class probability matrix Y of unlabeled data. The induction of the classifier W is dependent on the class probability matrix Y of unlabeled data, and the transduction of labels to unlabeled data is dependent on the classifier W. This adaptive procedure is expected to benefit both the induction of classifier and also the transduction of labels to unlabeled data. It is novel to use the class probability matrix of unlabeled data to do inductive learning.

Because the algorithm get the minimum in each updates of W, b and Y, so the objective value decreases in each updates. What's more, the objective function is lower bounded by zero. Therefore, it is obvious that our algorithm will converge. We will show in the experiment section that the algorithm actually converges quite fast on all data sets.

Algorithm 1 Algorithm to solve the problem (1).

Initialize W.
repeat
  Update Y: using Eq. (5) if $r = 1$; using Eq. (8) if $r \neq 1$.
  Update W and b by Eq. (12) and Eq. (10), respectively.
until Converges

3.1 Complexity Analysis and Scalability

The major computational cost lies in our algorithm lies in the updating of $W = C^{-1}A$, where c is a $d \times d$ matrix. The computational cost of matrix inverting is $O(d^3)$. In fact, the computation of matrix inverse can be avoided. Note that we are aiming to compute $C^{-1}A$, which is the solution of the following problem:

$$\min_W W^T CW - 2W^T A \quad (13)$$

The solution of this minimization problem can be get iteratively using gradient descent method using the updating formula: $W_{t+1} = W_t - \alpha(CW_t - A)$, with a computational cost of $O(Td^2c)$, where T is the number of iterations, c is the number of columns in A.

In addition, getting the $d \times d$ matrix C cost $O(nd^2)$, getting the $d \times c$ matrix A cost $O(ndc)$. Therefore, the total computational cost of our algorithm is upper bounded by $O(nd^2) + O(ndc) + O(Td^2c)$. Consider in real situation c is always smaller in magnitude compared to d and n, the total computational cost can also be written as $O(nd^2) + O(Td^2)$. **This shows that the computational cost of our algorithm is linear with respect to the number of data samples n. Therefore our algorithm is able to scale to large-scale data.**

However, for those graph based semi-supervised learning methods, without taking into consideration of the huge computation cost for tuning the gaussian kernel bandwidth parameter and the running time of the algorithms themselves, constructing the Laplacian graph matrix already takes at least $O(n^2)$. In the big data applications, the number of data n can easily reach billion level. So such graph based algorithms are not scalable to large-scale data.

4. EXPERIMENTAL RESULTS

4.1 Data Sets Descriptions

In order to show the effectiveness of the proposed adaptive semi-supervised learning method, experiments are conducted on six real world data sets: AR [4], YALE-B [17], MSRC-V1 [16], PIE [11], FERET [10] and ORL [9]. These six data sets are all comprising of human faces, represented using gray scale pixel values. Figure 2 demonstrate some sample images from each data set. Important statistics are summarized in Table 1.

Table 1: Data Sets Descriptions

	# sample	# feature	# class
AR	840	768	120
YALE-B	2414	1024	38
MSRC-V1	1799	1024	12
CMU-PIE	3329	1024	68
FERET	1400	1296	200
ORL	400	644	40

4.2 Experimental Settings

In order to evaluate the effectiveness of the proposed Adaptive Semi-supervised Learning (ASL) method, we compare it with some most representative state-of-the-art semi-supervised learning methods. Since our method is a unified model which can simultaneously perform transduction and induction, we compare our methods with both transductive semi-supervised learning methods and inductive semi-supervised learning methods.

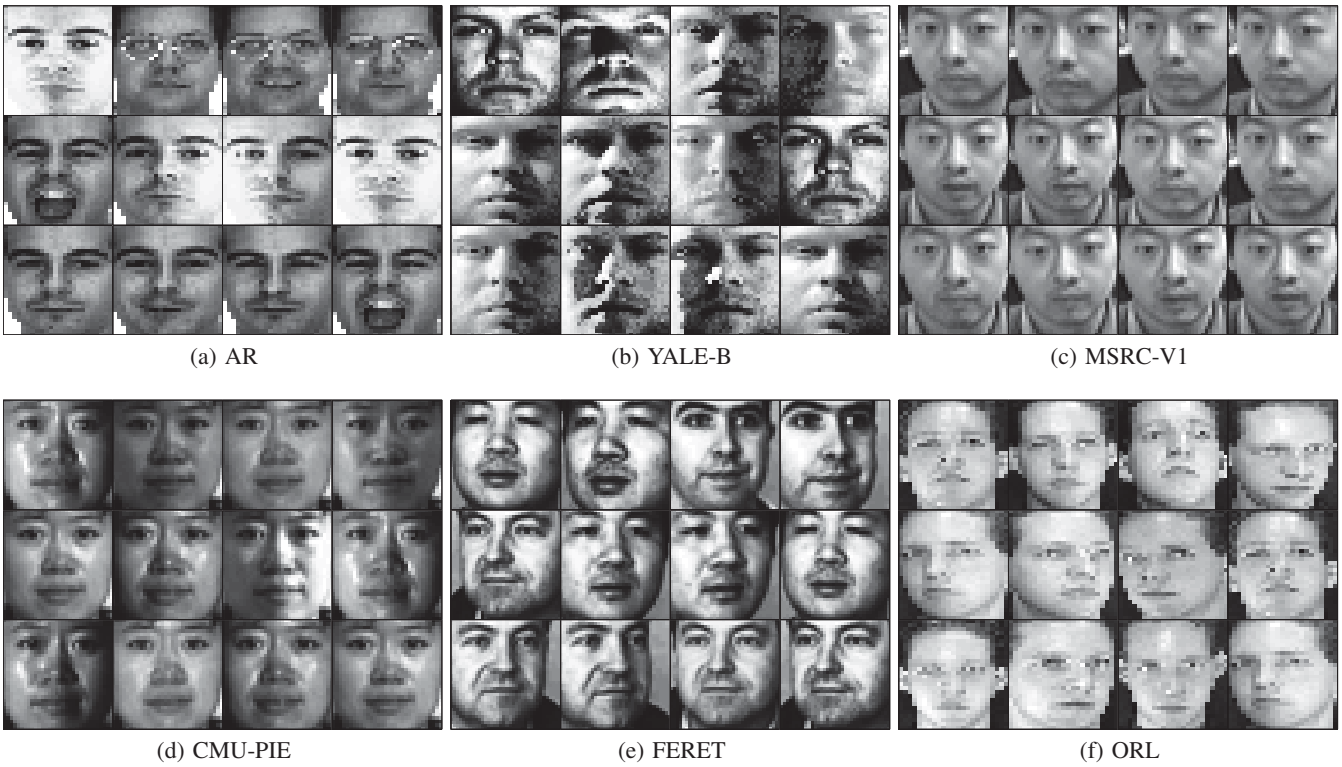
In this paper, we compare our method with three representative transductive semi-supervised learning methods: local and global consistency (LGC) [18], random walk (RW) [19], and gaussian field harmonic function (GFHF) [21].

Inductive semi-supervised learning methods learn a classifier using both labeled data and unlabeled data. Then the learned classifier can be used for the classification of both unlabeled data using for training and also new out-of-sample testing data. Compared with our method which can directly learn the labels of unlabeled data, common inductive semi-supervised learning methods need an additional step to get the labels for unlabeled data used in training.

Two representative inductive semi-supervised learning methods are compared with our method: flexible manifold embedding (FME) [8], and Laplacian regression (LapReg) [12]. LapReg aims to solve the manifold regularized problem, which has the following objective function:

$$\min_W ||Y_l - X_l^T W||_F^2 + \gamma_1 Tr(W^T XLX^T W) + \gamma_2 ||W||_F^2 \quad (14)$$

where Y_l,X_l are the labels and feature matrix of labeled data, respectively, X is formed by both labeled data and unlabeled data, L



(a) AR (b) YALE-B (c) MSRC-V1

(d) CMU-PIE (e) FERET (f) ORL

Figure 2: Sample face images from each data set.

is the graph Laplacian matrix constructed using X, γ_1 and γ_2 are regularization parameters to balance the three terms.

Semi-supervised discriminant analysis (SDA) [2] is also a commonly used semi-supervised learning method. It aims to learn a projection matrix to map the data in high dimension to a lower dimension subspace. Since our method aims to learn a classifier rather than projection matrix for dimension reduction, we do not compare with SDA in this paper.

In our experiment, we repeat every methods 20 times to compute the average classification accuracy and standard deviation. Each data are first preprocessed using PCA such that 95% of the energy are retained. Different number of labeled points are used for training to study the sensitivity of those methods to the number of labeled data. We randomly choose 1,3,5 labeled points from each class as labeled data, and the remaining as unlabeled data. For transductive methods, since they can not perform out-sample testing, only the accuracy for unlabeled data is computed. For inductive methods, 33% of data are leaved out for out-of-sample testing, then the remaining training data is splited into labeled data and unlabeled data. The accuracy for both unlabeled data and out-of-sample testing are computed.

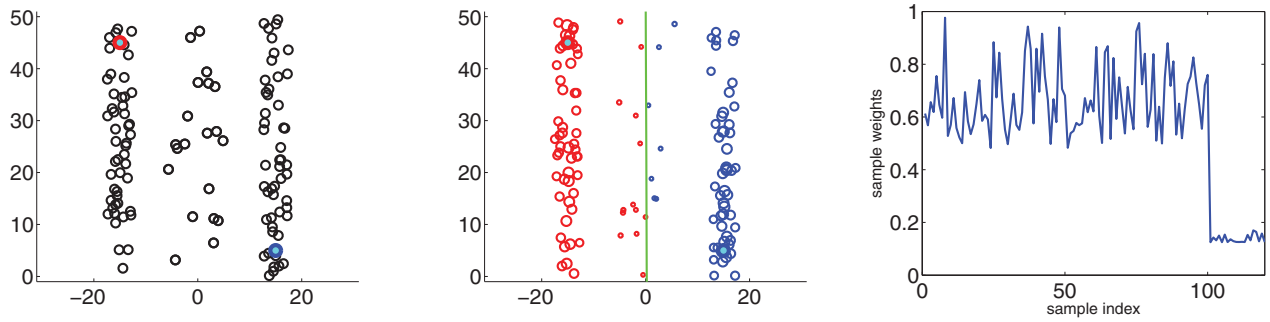
For our method, the parameter r is tuned from 1 to 2 with a step-size of 0.1. For LGC and RW, the tradeoff parameter α is tuned in [0.1:0.1:0.9,0.99] (*i.e.* from 0 to 0.99 with a step-size of 0.1, 0.99 is also included). GFHF is parameter-free. For LapReg and FME, both of them have two regularization parameters, and are tuned in $\{10^{-5}, 10^{-4}, ..., 10^{4}, 10^{5}\}$. The classification results of using the best tuned parameter are recorded.

4.3 Demonstration on Synthetic Data

In this section, we show the working mechanism of the proposed adaptive semi-supervised learning model on a synthetic data. The synthetic data (demonstrated in Figure 3(a)) is generated as following: 50 data points (corresponding to the samples in the left part of the figure) are sampled from a gaussian distribution with a mean of -15, and a standard deviation of 5 ; Another 50 data points (corresponding to the samples in the right part of the figure) are sampled from a gaussian distribution with a mean of 15, and also a standard deviation of 5. Then 20 boundary points are generated around the ideal decision boundary (*i.e.* the vertical line $x = 0$) with a mean of 0 and a standard deviation of 12. All the above data points are unlabeled data. After that, we add only one labeled point to each part (*i.e.*, red circle box in the left, blue circle box in the right). Note that we intentionally place the labeled point in the upper left corner and lower right corner. Compared to randomly label one point from each part, this labeling strategy will make it harder for a model to recover the correct decision boundary.

Figure 3 (c) shows the weights/importance of each unlabeled data point learned by our model . The weights of the i-th data point is $\sum_{k=1}^{c} y_{ik}^r$. If the weight of a data point is large, it will contribute more to the learning of the classifier. The last 20 samples are boundary points (points in the middle part of Figure 3 (a)). We can see that the weights of those boundary points are very small compared to other point. Therefore, our model can adaptively adjust the weights of each point, and thus robust to boundary points.

Figure 3 (b) shows the data after classification using our adaptive semi-supervised learning model. The size of each point is proportional to their weights. Boundary points in the middle part get smaller weights. The correct vertical decision boundary (the green line in the figure) is perfectly recovered, even with only one selected labeled point from each class, and the one labeled point is intentionally set to make it harder to recover the correct decision



(a) Original toy data (b) Toy data after classification using our model (c) Sample weights

Figure 3: Demonstration on Synthetic Data: Figure (a) shows the original data distribution, the red circle box and the blue circle box are the labeled points, all other points are unlabeled; Figure (b) shows the data after classification by our model, the vertical green line is the recovered decision boundary. The size of each point is proportional to their weights. Boundary points in the middle part get smaller weights; Figure (c) shows the weights/importance of each sample learned by our model. The last 20 samples are the boundary points in Figure (a)

boundary. This shows that our model is able to learn the correct classifier with the presence of boundary points.

4.4 Parameter Discussions

There is only one parameter, *i.e.* r, in our proposed ASL model. As discussed before, the parameter actually serves for multiple purposes. In order to achieve the best performance, a proper r should be used to balance the supervised part and unsupervised part in the objective function, and in the meantime, make our model robust to boundary points.

If r is too large, the y_{ik}^r will tend to be zero, and the unsupervised information is lost. So r can not be too large. That's why r is suggested to vary in [1,2] (Consider the class number is often larger than 10, $r = 2$ is large enough to suppress; when the class number is small, the range of r can be extended correspondingly. Generally, [1,4] is totally sufficient for all small class number problems.)

In the following, we will show how the change of parameter r influence the weights of data points , and the influence on classification performance.

4.4.1 r and Number of Important Points

When r varies, the contribution of each data instance to the model also changes. We define a data point as an important point if the sum of weights over all classes exceeds an threshold value, *i.e.* it satisfies the following condition:

$$\sum_{k}^{c} y_{ik}^r > t \tag{15}$$

where t is a pre-specified threshold. In our experiment, t is set as 0.25 since it is reasonable to assume that an important point should belong to one certain class with a probability larger than 0.5, so $\sum_{k}^{c} y_{ik}^r > 0.5^2 = 0.25$ when $r < 2$.

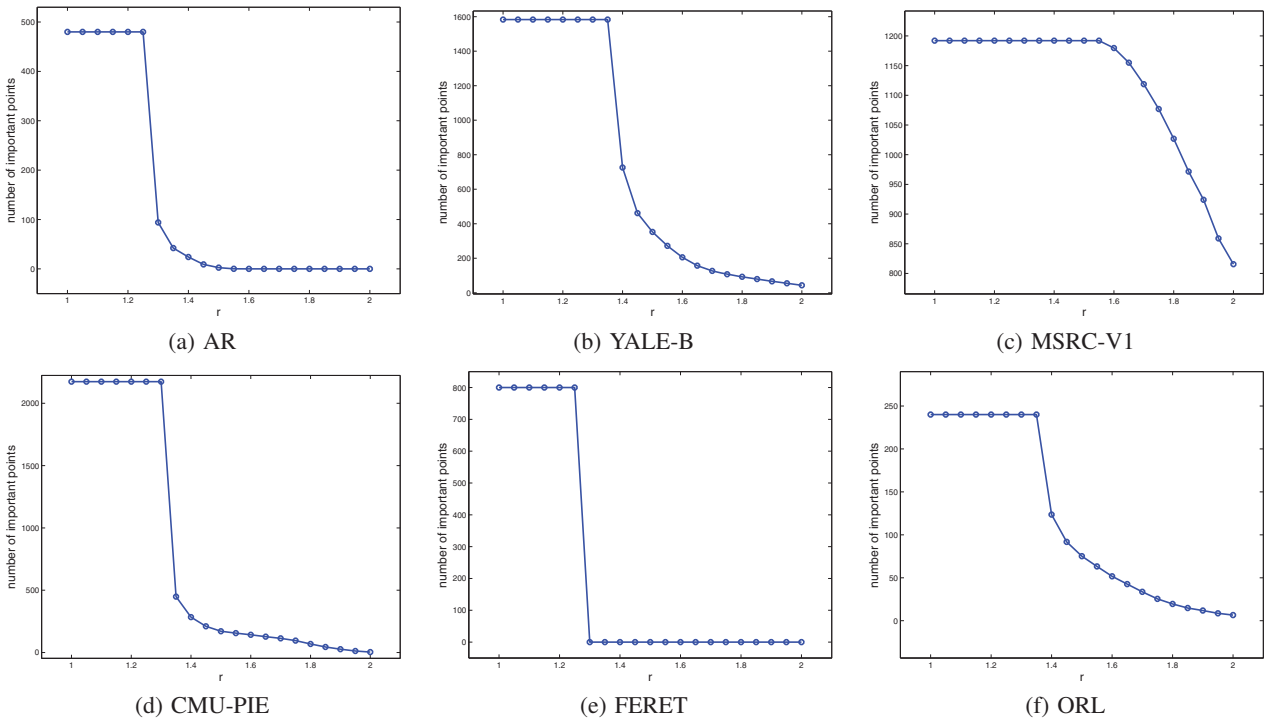
Figure 4 shows the number of important points using different r values. From the figure we can see that:

(1)When $r = 1$, all points are regarded as important points because $\sum_{k}^{c} y_{ik}^1 = 1$. In this case, all samples contributes equally to the loss function since their sum of weights over all class are the same.

(2)When r increases, the number of important points will decrease. In this process, boundary points will become unimportant (contribute less to the objective function) with a small r value. This makes our model robust to boundary points. Some non boundary points may become unimportant with a relatively large r value. In the early stage of the increasing of r (say $r < 1.4$), the decreasing of important points is mainly because boundary points become unimportant. In the later stage of the increasing of r (say $r > 1.4$), the decrease of number of important points may also be due to that some non boundary points also become unimportant. Note that the number of important points on some data sets become zero in the later stage, but this does not mean unlabeled data points are not used in our model. Remember that the threshold we are using is 0.25, so the unlabeled data information is still utilized in the model with a relatively small weight. In the extreme case, when r approaches to infinity, all points will become totally unimportant with weight 0. In this case, unlabeled data points are not used in the model, which reduces to supervised learning.

Therefore, in order for the model to utilize the unlabeled data to the best degree, a proper r value should be chosen. The proper r value should: in the macro level, balance the supervised term and unsupervised term in the objective function; in the micro level, automatically adjust the weights of data points, so that the model become robust to boundary points, and thus learn a better classifier.

(3) On some data sets with large number of classes (FERET with 200 classes, AR with 120 classes), the number of important points will decrease drastically to zero. In the contrary, on some data sets with small number of classes (MSRC-V1 with 12 classes), the number of important points will decrease much slower, and still has many important points when $r = 2$. This is because when the class number is small, there tend to be less boundary points, and the probability y_{ik} will be relatively larger, so even with a large r value, there are still many important points. However, when the class number is large, there tend to be much more boundary points whose y_{ik} are small (tend to be $\frac{1}{c}$ in the worst case). So even with small r value, those boundary points become unimportant. In this case, the benefits of our proposed model, which suppresses the weights of boundary points, will be more obvious.



(a) AR (b) YALE-B (c) MSRC-V1

(d) CMU-PIE (e) FERET (f) ORL

Figure 4: The number of important points using different r values on six data sets. One labeled data point is used from each class.

4.4.2 r and Classification Accuracy

After the above analysis, we know that r will influence the learned model from both macro level (the weight of unsupervised term in the objective function) and micro level (suppress the weights of boundary points). Different model will lead to different classification results. In this section, we show how the r value will influence the classification performance.

Figure 5 shows the classification accuracy using different r value on the six data sets. From the figure we can see that: the classification accuracy changes with the parameter r, and the best classification results is achieved by different r value on different data sets.

On all the data sets, the best classfication accuracy is not achieved when $r = 1$, which is the common practice in many papers. This is because: by setting y_{ik}^r as data weights, our model is more robust to boundary points by suppressing the weights of boundary points. Therefore, by setting the weights of data as y_{ik}^r rather than the commonly used y_{ik} is more meaningful.

4.5 Classification Performance Comparison

In this section, we present extensive empirical study of the classification performance on six real world data sets. The classification accuracy and standard deviation for running different semi-supervised learning methods are reported in Table 2 to Table 7. From these tables, we can conclude that:

(1) The ASL method outperforms other comparing inductive semi-supervised learning methods (LapReg and FME) in most cases, and is significantly better than other transductive methods (LGC, RW, and GFHF) on all the six data sets. This justifies the effectiveness of the proposed ASL method. Because the ASL method is able to automatically balance the supervised part and unsupervised part in macro level. What's more, in the micro level, the ASL method can suppress the weights of boundary points, which makes the model robust to boundary points, and thus, learn a better classifier.

(2) In general, the inductive semi-supervised learning methods perform better than transductive methods.

(3) The performance of the three transductive methods are comparable to each other. The performance of these transductive methods is pretty good on the MSRC-V1 and ORL data sets. The reason maybe that the number of classes is small on these two data sets compared to other data sets. It seems that those transductive methods can not perform well when the data has large number of classes.

(4) The performance of LapReg and FME are comparable to each other. On the AR data set, the performance of FME is slightly better than the ASL method. This shows the effectiveness of using the manifold regularization to utilize the unlabeled data information. However, the drawback of LapReg and FME is that both of them have two regularization parameters, and the parameters' range is pretty large and not fixed (vary in $[10^{-5}, 10^5]$ or even larger). This makes it hard to tune the parameter.

(5) When the number of labeled data points from each class(*i.e.* kl) increases, the performance also become better on all data sets. Especially when number of labeled data points increase from 1 to 3, the classification accuracy improved significantly. Therefore, when the labeled information is scarce, to acquire more labeled data can be very helpful in semi-supervised learning.

4.6 Convergence Speed

In this section, we show the converge speed of the proposed algorithm empirically. Figure 6 shows the objective function value versus number of iterations. We can see that the proposed iterative algorithm converges in less than 20 iterations on all data sets.

In our experiment, running our algorithm 20 iterations only takes about 1 second on a laptop with 2.70GHz double core Intel Core i7 cpu, 16GB memory. Since the major computational cost in our al-

Table 2: Classification accuracy and standard deviation for running different semi-supervised learning methods 20 times on AR Data Set. "kl" represents the number of labeled points from each class used. The column "Unlabeled" record the classification accuracy of unlabeled data using transductive methods or inductive methods. The column "Testing" record the classification accuracy of out of sample testing data using inductive methods. "NA" represents the value is not available.

AR	kl=1		kl=3		kl=5	
	Unlabeled	Testing	Unlabeled	Testing	Unlabeled	Testing
LGC	22.42±1.57	NA	34.69±1.99	NA	41.92±3.53	NA
RW	20.40±1.60	NA	33.09±1.63	NA	40.40±2.65	NA
GFHF	20.89±1.43	NA	32.67±1.79	NA	40.96±2.41	NA
LapReg	75.56±1.64	73.65±1.94	92.63±1.72	92.92±1.81	94.67±1.89	94.77±1.09
FME	67.02±2.63	67.79±2.55	92.58±1.43	93.29±1.76	95.67±2.65	**95.83±1.23**
ASL	**77.36±2.23**	**76.08±2.81**	**93.15±1.60**	**93.46±1.67**	**96.83±1.96**	95.08±1.33

Table 3: Classification accuracy and standard deviation for running different semi-supervised learning methods 20 times on YALE-B Data Set.

YALE-B	kl=1		kl=3		kl=5	
	Unlabeled	Testing	Unlabeled	Testing	Unlabeled	Testing
LGC	42.47±2.11	NA	60.42±2.42	NA	68.16±1.48	NA
RW	43.98±2.43	NA	61.37±2.38	NA	68.69±1.42	NA
GFHF	43.75±1.96	NA	63.05±1.71	NA	70.32±2.12	NA
LapReg	51.36±2.88	50.88±3.11	86.09±1.73	86.51±1.92	94.90±0.82	95.02±1.07
FME	51.24±2.01	50.90±3.17	85.70±2.55	85.54±2.92	94.61±2.03	94.83±2.11
ASL	**59.35±2.91**	**58.98±8.08**	**96.06±1.86**	**96.13±1.92**	**99.14±0.54**	**99.00±0.85**

Table 4: Classification accuracy and standard deviation for running different semi-supervised learning methods 20 times on MSRC-V1 Data Set.

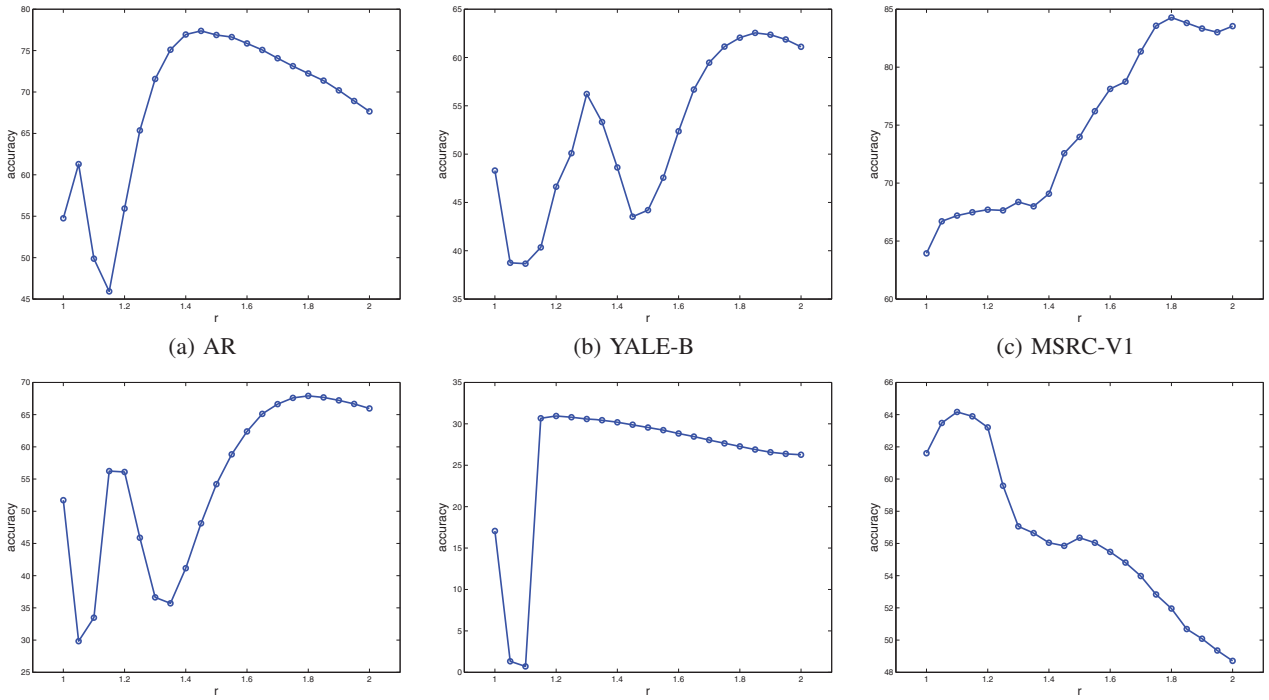
MSRC-V1	kl=1		kl=3		kl=5	
	Unlabeled	Testing	Unlabeled	Testing	Unlabeled	Testing
LGC	62.38±3.46	NA	88.34±5.38	NA	95.19±3.58	NA
RW	60.05±3.78	NA	88.98±5.29	NA	96.88±2.94	NA
GFHF	55.73±6.01	NA	89.73±4.39	NA	97.07±2.72	NA
LapReg	81.91±4.44	81.83±4.65	97.54±2.08	97.55±1.90	98.69±1.69	98.61±1.61
FME	80.72±4.34	80.45±4.50	97.17±2.64	97.28±2.57	99.29±0.88	99.22±0.98
ASL	**82.55±3.64**	**82.29±3.74**	**98.39±1.92**	**98.45±1.82**	**99.36±1.25**	**99.42±1.15**

Table 5: Classification accuracy and standard deviation for running different semi-supervised learning methods 20 times on CMU-PIE Data Set.

CMU-PIE	kl=1		kl=3		kl=5	
	Unlabeled	Testing	Unlabeled	Testing	Unlabeled	Testing
LGC	26.36±1.60	NA	44.76±0.85	NA	54.67±1.36	NA
RW	26.71±1.40	NA	44.03±1.15	NA	54.28±1.36	NA
GFHF	26.34±1.62	NA	45.33±1.72	NA	55.58±1.17	NA
LapReg	62.18±2.11	61.89±1.83	86.08±1.56	86.26±1.23	91.44±0.81	91.45±0.96
FME	60.31±1.53	60.68±1.73	86.68±1.78	85.36±1.66	91.35±1.05	91.31±1.42
ASL	**63.12±2.20**	**62.80±2.76**	**90.64±1.73**	**90.70±1.64**	**93.53±0.84**	**93.38±0.90**

Table 6: Classification accuracy and standard deviation for running different semi-supervised learning methods 20 times on FERET Data Set.

FERET	kl=1		kl=3		kl=5	
	Unlabeled	Testing	Unlabeled	Testing	Unlabeled	Testing
LGC	14.65±0.60	NA	26.74±1.13	NA	31.74±1.52	NA
RW	13.98±0.49	NA	25.02±1.14	NA	30.39±1.65	NA
GFHF	12.07±0.56	NA	21.57±1.38	NA	29.52±2.16	NA
LapReg	27.51±1.20	27.39±1.74	58.29±2.62	57.45±1.66	71.55±1.98	70.21±2.01
FME	26.14±1.42	26.64±1.95	57.61±2.63	56.58±2.51	70.50±2.61	71.10±2.27
ASL	**31.79±1.50**	**31.06±1.95**	**59.75±2.36**	**58.54±2.55**	**73.70±1.95**	**71.97±1.46**



(a) AR (b) YALE-B (c) MSRC-V1

(d) CMU-PIE (e) FERET (f) ORL

Figure 5: The classification accuracy using different r values on the six data sets. One labeled data point is used from each class.

Table 7: Classification accuracy and standard deviation for running different semi-supervised learning methods 20 times on ORL Data Set.

ORL	kl=1		kl=3		kl=5	
	Unlabeled	Testing	Unlabeled	Testing	Unlabeled	Testing
LGC	75.95±2.48	NA	**88.84±2.78**	NA	90.20±2.54	NA
RW	75.86±1.46	NA	86.55±2.49	NA	90.65±1.70	NA
GFHF	74.99±1.73	NA	86.54±2.66	NA	89.85±2.27	NA
LapReg	72.38±2.44	68.58±3.95	87.16±2.54	84.50±2.98	92.94±2.41	90.29±3.03
FME	73.65±2.86	69.08±4.29	87.72±2.54	85.21±3.00	93.00±2.88	90.08±2.43
ASL	**76.42±2.02**	**72.17±3.01**	87.16±3.16	**86.08±3.33**	**93.75±2.91**	**90.47±2.54**

gorithm lies in the inverse of a d by d matrix, when the algorithm is used on data with high dimensionality, the computational cost can be reduced by using PCA for dimensionality reduction beforehand.

5. CONCLUSION

In this paper, we propose an adaptive semi-supervised learning model. Different from previous semi-supervised learning, our proposed model needn't construct the graph Laplacian matrix. Thus, our method avoids the huge computational cost required by previous methods, and achieves a computational complexity linear to the number of data points. Therefore, our method is scalable to large-scale data. Moreover, the proposed model adaptively suppresses the weights of boundary points. This makes our model robust to boundary points. An efficient algorithm is derived to alternatively optimize the model parameter and class probability distribution of unlabeled data, such that the induction of classifier and the transduction of labels are adaptively unified in one framework. Our model only has one parameter need to be tuned, and a fixed range is also suggested. Extensive experimental results show that the proposed semi-supervised learning model outperforms other state-of-the-art methods in most cases.

6. ACKNOWLEDGMENTS

This research was partially supported by NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

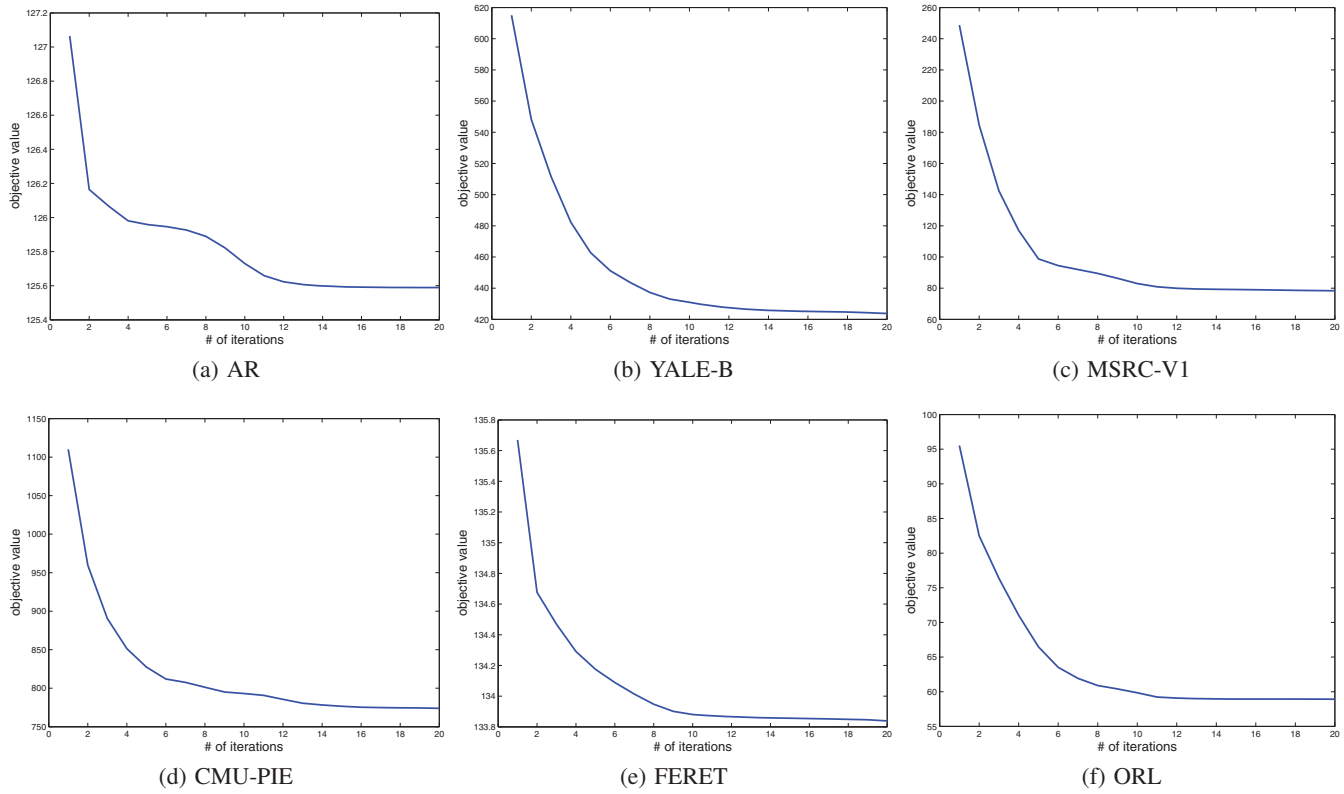
[2] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7. IEEE, 2007.

[3] X. Chang, F. Nie, Y. Yang, and H. Huang. A convex formulation for semi-supervised multi-label feature selection. In *The Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[4] A. M. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[5] F. Nie, S. Xiang, Y. Jia, and C. Zhang. Semi-supervised orthogonal discriminant analysis via label propagation. *Pattern Recognition*, 42(11):2615–2627, 2009.

[6] F. Nie, S. Xiang, Y. Liu, and C. Zhang. A general graph-based semi-supervised learning with novel class discovery. *Neural Computing and Applications*, 19(4):549–555, 2010.



(a) AR (b) YALE-B (c) MSRC-V1

(d) CMU-PIE (e) FERET (f) ORL

Figure 6: Objective value versus the number of iterations.

[7] F. Nie, D. Xu, X. Li, and S. Xiang. Semisupervised dimensionality reduction and classification through virtual label regression. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(3):675–685, 2011.

[8] F. Nie, D. Xu, I. W.-H. Tsang, and C. Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *Image Processing, IEEE Transactions on*, 19(7):1921–1932, 2010.

[9] ORL Face Database, 2007. `http://www.cam-orl.co.uk/facedatabase.html`.

[10] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1090–1104, 2000.

[11] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[12] V. Sindhwani, P. Niyogi, M. Belkin, and S. Keerthi. Linear manifold regularization for large scale semi-supervised learning. In *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*, volume 28, 2005.

[13] D. Wang, Y. Wang, F. Nie, J. Yan, W. Cai, A. Saykin, L. Shen, and H. Huang. Human connectome module pattern detection using a new multi-graph minmax cut model. In *Med Image Comput Comput Assist Interv*, 2014.

[14] H. Wang, C. Ding, and H. Huang. Directed Graph Learning via High-Order Co-linkage Analysis. In *Proceedings of the ECML PKDD*, pages 451–466, 2010.

[15] H. Wang, H. Huang, and C. Ding. Image annotation using multi-label correlated green's function. *IEEE Conference on Computer Vision*, pages 1–6, 2009.

[16] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 756–763. IEEE, 2005.

[17] Yale Univ. Face Database, 2002. `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`.

[18] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Proc. Neural Info. Processing Systems*, 2003.

[19] D. Zhou and B. Schölkopf. Learning from labeled and unlabeled data using random walks. In *Pattern Recognition*, pages 237–244. Springer, 2004.

[20] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2:3, 2006.

[21] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *Proc. Int'l Conf. Machine Learning*, 2003.