Toward the Optimization of Normalized Graph Laplacian

Article *in* IEEE Transactions on Neural Networks · February 2011

DOI: 10.1109/TNN.2011.2107919 · Source: PubMed

Meng Wang
University of South China

Towards the Optimization of Normalized Graph Laplacian

Bo Xie, Meng Wang, *Member, IEEE,* and Dacheng Tao, *Member, IEEE,*

***Abstract*—Normalized graph Laplacian has been widely used in many practical machine learning algorithms, e.g., spectral clustering and semi-supervised learning. However, all of them use the Euclidean distance to construct the graph Laplacian, which does not necessarily reflect the inherent distribution of the data. In this paper, we propose a method to directly optimize the normalized graph Laplacian by using pair-wise constraints. The learned graph is consistent with equivalence and non-equivalence pair-wise relationships, and thus it can better represent similarity between samples. Meanwhile, our approach, unlike metric learning, automatically determines the scale factor during the optimization. The learned normalized Laplacian matrix can be directly applied in spectral clustering and semi-supervised learning algorithms. Comprehensive experiments demonstrate the effectiveness of the proposed approach.**

***Index Terms*—graph, Laplacian, semi-supervised learning, metric learning.**

I. Introduction

GRAPH-BASED methods have been widely used in machine learning and data mining fields. In these methods, data points are represented as vertices and their pair-wise relationships are modeled as edges. Normalized Laplacian matrix, which is an important graph representation, has many appealing characteristics. For example, the number of its zero eigenvalues equals to the number of connected components and the corresponding eigenvectors are indicators of each sample's label. Existing studies have also shown its superiority over Laplacian matrix in many algorithms from both theoretical and empirical perspectives [1], [2]. In semi-supervised learning, normalized Laplacian matrix can be used to estimate the smoothness of labels over the graph which is explored in the learning scheme.

Despite its usefulness, the construction of normalized graph Laplacian has not been extensively studied. Generally, the edge between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is estimated via a Gaussian function as

$$w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \tag{1}$$

where σ is a radius parameter. However, this method has two weaknesses: 1) Euclidean distance may not reflect the inherent distribution of the data; and 2) the setting of the radius parameter σ is a problem. Meanwhile, clustering and classification tasks are usually associated with certain prior information such as labeled samples or pair-wise constraints, i.e., some pair of samples should be closer/farther away. Therefore, in this paper we propose a normalized graph Laplacian construction approach that can take this information into account. We directly learn the normalized graph Laplacian matrix by optimizing its consistency with pair-wise constraints. Our approach has two advantages: 1) it is a general method as we only assume several constraints are available, and thus it can be applied in many different tasks; and 2) the matrix can learn its scale automatically and does not need to tune the radius parameter in order to obtain similarity measurements.

II. Related work

Graph-based semi-supervised learning algorithms [3]–[6] are usually implemented based on a graph, where the vertices are labeled and unlabeled samples and the edges reflect the similarities of sample pairs. The labels are usually formulated to satisfy the following two conditions: 1) they should be close to the given truths on the labeled vertices, and 2) they should be smooth on the whole graph. Blum et al. [7] transformed the problem into Mincut, in which positive and negative samples act as sources and sinks. Then the problem is equivalent to finding a minimum removal of edges that blocks all paths from sources to sinks. An equivalent formation is a Markov random field in the binary case. Zhu et al. [3] have derived a graph-based regularization framework from Gaussian random field. Zhou et al. [4] proposed a Learning with Local and Global Consistency (LLGC) algorithm, which can be solved with the normalized graph Laplacian. Other than the graph-based model, [8] proposed to use generalized point charge models for semi-supervised learning. And in [9], least squares SVM were extended to semi-supervised learning problems.

Spectral clustering is a promising graph-based clustering approach [1], [10], [11]. It accomplishes the clustering by exploring the information embedded in the eigenvectors of normalized graph Laplacian. The eigenvectors preserve locality over the graph and serve as a new representation of the data, in which natural clustering is more apparent. Moreover, spectral clustering can be viewed as a relaxation of graph cut formulation, and in this point of view the problem is to find a partition of the graph such that the edges connecting different groups have small weights and those within the same groups have large weights. Shi and Malik [1] were motivated in this perspective and proposed normalized cut. Ng et al. [10] proposed another spectral clustering algorithm based on the first k eigenvectors of the symmetric normalized graph Laplacian and a k-means clustering process.

Bo Xie is with the Nanyang Technological University, Singapore (zixu1986@gmail.com).

Meng Wang is with School of Computing, National University of Singapore, Singapore (eric.mengwang@gmail.com).

Dacheng Tao is with the University of Technology, Sydney, Australia (dacheng.tao@ieee.org).

Despite the two problems mentioned above, many other approaches optimize the objective function with regularization on graph Laplacian. In [12], graph Laplacian regularization is incorporated in optimization scheme to explore the low dimensional manifold structure in the Gaussian Mixture Model framework. Laplacian regularized optimization is also applied in active learning and image retrieval [13].

Although the effectiveness of graph Laplacian optimization have been demonstrated in many applications [4], [5], [14]–[16], a problem that receives less attention is the graph construction. A typical approach is to apply distance metric learning, which aims to construct a distance metric with pairwise relationships among samples. Eric Xing [17] proposed a global convex optimization approach which minimizes the distances of samples with equivalence constraints while keeping samples with non-equivalence constraints separate. In Goldberger et al.'s Neighborhood Components Analysis (NCA) approach [18], they directly optimize soft leave-one-out cross validation error of k-nearest neighbors to obtain a good metric. Weinberger came up with Large Margin Nearest Neighbor algorithm (LMNN) [19] to emphasize the local structure of data. Our method can also be viewed as a distance metric learning algorithm as the normalized graph Laplacian will be built based on a learned distance metric. But in comparison with the existing distance metric learning algorithms, our approach has the following two characteristics: 1) our method is designed to directly optimize the consistency between normalized graph Laplacian and the constraint information; 2) most of the existing distance metric learning methods need to further convert distance between samples into similarity via a radius parameter, whereas our approach does not need this parameter.

III. Normalized Graph Laplacian Optimization

We define normalized graph Laplacian and formulate an objective function to learn a good normalized graph Laplacian using the pair-wise constraints. Minimization of the objective function is solved by gradient descent.

A. Normalized Graph Laplacian

Suppose we have n samples $\{\mathbf{x}_i\}_1^n$ and w_{ij} indicates the similarity measure between $\mathbf{x}_i$ and $\mathbf{x}_j$ (w_{ii} is set to 0). As mentioned in Section I, most frequently it is defined based on the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ and a Gaussian function, such as Eq. 1.

The normalized graph Laplacian is defined as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{2}$$

where $\mathbf{D}$ is a diagonal matrix with its i-th element along the diagonal equals to the sum of the i-th row of $\mathbf{W}$, i.e., $d_i = \sum_{j=1}^{n} w_{ij}$. Therefore, the ($i$,$j$)-th element of the normalized graph Laplacian is

$$l_{ij} = \delta_{ij} - \frac{w_{ij}}{\sqrt{d_i d_j}} \tag{3}$$

where $\delta_{ij} = 1$ if $i = j$, otherwise $\delta_{ij} = 0$. Here l_{ij} is related to the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. When $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, l_{ij} is small, and l_{ij} is greater if $\mathbf{x}_i$ and $\mathbf{x}_j$ are far away.

B. The learning of normalized graph Laplacian

We define a transformation matrix $\mathbf{A}$ and the new similarity is

$$w_{ij} = \exp\left(-\left\|\mathbf{A}\left(\mathbf{x}_i - \mathbf{x}_j\right)\right\|^2\right) \tag{4}$$

We assume that there are a set of equivalence constraints and a set of non-equivalence constraints. The constraints are usually generated with prior knowledge. For example, typically an equivalence constraint means the two samples should belong to the same class or cluster, and contrarily a non-equivalence constraint means the samples belong to different classes or clusters. Denote by $\mathcal{S}$ and $\mathcal{D}$ the sets of equivalence and non-equivalence constraints, i.e.,

$$\mathcal{S} = \{(i,j)|\mathbf{x}_i, \mathbf{x}_j \text{ belong to the same class/cluster}\}$$
$$\mathcal{D} = \{(i,j)|\mathbf{x}_i, \mathbf{x}_j \text{ belong to different classes/clusters}\}$$

A typical strategy is to force points with equivalence constraints close and points with non-equivalence constraints far away from each other. Accordingly, we can minimize

$$\sum_{(i,j)\in\mathcal{S}} l_{ij} - \sum_{(i,j)\in\mathcal{D}} l_{ij} \tag{5}$$

The above formulation can also be derived from the label smoothness over the graph. Consider a binary classification problem and denote by $\mathbf{f} = [f_1, f_2, \cdots, f_n]^T$ with $f_i = \{-1, 1\}$ the vector of samples' labels. The label smoothness assumption means that nearby samples should have high probability to share the same label, and this leads to the minimization of the term

$$\begin{aligned}\mathbf{f}^T\mathbf{L}\mathbf{f} &= \mathbf{f}^T\mathbf{f} - \mathbf{f}^T\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{f}\\ &= \sum_{i=1}^{n} f_i^2 - \sum_{i,j=1}^{n} f_i f_j \frac{w_{ij}}{\sqrt{d_i d_j}}\\ &= \frac{1}{2}\left[\sum_{i=1}^{n} d_i\left(\frac{f_i}{\sqrt{d_i}}\right)^2 - 2\sum_{i,j=1}^{n}\frac{f_i f_j w_{ij}}{\sqrt{d_i d_j}} + \sum_{j=1}^{n} d_j\left(\frac{f_j}{\sqrt{d_j}}\right)^2\right]\\ &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2 \end{aligned} \tag{6}$$

The above principle has been widely explored in many different algorithms [4], [20], [21]. It is not difficult to prove that the above equation is equivalent to the Eq. 5, i.e.,

$$\mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{(i,j)\in\mathcal{S}} l_{ij} - \sum_{(i,j)\in\mathcal{D}} l_{ij} \tag{7}$$

A problem is that when constraints are too few, the optimization may encounter overfitting problem. Here we add a regularizer such that the learned normalized graph Laplacian does not change too much from that obtained using Euclidean distance. Denote by $\mathcal{N}_i$ the neighborhood of sample $\mathbf{x}_i$, which is obtained in the Euclidean space. The final objective function is

$$Q(\mathbf{A}) = \sum_{(i,j)\in\mathcal{S}} l_{ij} - \sum_{(i,j)\in\mathcal{D}} l_{ij} + \lambda\sum_{j=1}^{n}\sum_{i\in\mathcal{N}_j} l_{ij} \tag{8}$$

where λ is a weighting factor in regularization that is used to tune the effect of unlabeled data. We can see that the third term actually forces the neighborhoods of samples not to change too much.

C. Optimization Using Gradient Descent

We use gradient descent to solve the optimization problem. The derivative of l_{ij} with respect to the transformation matrix $\mathbf{A}$ can be derived as

$$\frac{\partial l_{ij}}{\partial \mathbf{A}} = \frac{1}{2\sqrt{d_i d_j}} \left(\frac{w_{ij}}{d_i} \frac{\partial d_i}{\partial \mathbf{A}} + \frac{w_{ij}}{d_j} \frac{\partial d_j}{\partial \mathbf{A}} - 2 \frac{\partial w_{ij}}{\partial \mathbf{A}} \right) \tag{9}$$

where

$$\frac{\partial w_{ij}}{\partial \mathbf{A}} = -2 w_{ij} \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j \right) \left(\mathbf{x}_i - \mathbf{x}_j \right)^T \tag{10}$$

and

$$\frac{\partial d_i}{\partial \mathbf{A}} = \sum_{j=1}^{n} \frac{\partial w_{ij}}{\partial \mathbf{A}} \tag{11}$$

This means we need to store $\partial w_{ij} / \partial \mathbf{A}$ for all j which may introduce very large storage cost. To avoid this issue, we substitute Eq. 10 into Eq. 11 and obtain

$$\begin{aligned} \frac{\partial d_i}{\partial \mathbf{A}} =& -2\mathbf{A} \sum_{j=1}^{n} w_{ij} \left(\mathbf{x}_i - \mathbf{x}_j \right) \left(\mathbf{x}_i - \mathbf{x}_j \right)^T \\ =& -2\mathbf{A} \Big[\mathbf{X} \mathrm{diag} \left(\mathbf{W}_i \right) \mathbf{X}^T + d_i \mathbf{x}_i \mathbf{x}_i^T \\ & - \mathbf{x}_i \left(\mathbf{X} \mathbf{W}_i \right)^T - \left(\mathbf{X} \mathbf{W}_i \right) \mathbf{x}_i^T \Big] \end{aligned} \tag{12}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ is the data matrix, $\mathbf{W}_i$ is the i-th column of $\mathbf{W}$ and diag$(\cdot)$ denotes a diagonal matrix with the diagonal elements from the input vector.

Finally, the updating rule is

$$\mathbf{A}_{t+1} = \mathbf{A}_t - \eta_t \frac{\partial Q}{\partial \mathbf{A}_t} \tag{13}$$

The transformation matrix $\mathbf{A}$ is initialized as $\mathbf{I}/\delta$, where δ is the median of all pair-wise Euclidean distances between samples.

For faster convergence, we adopt an adaptive learning rate for gradient descent. More specifically, if the current update reduces the objective function, we increase the learning rate η_t by a factor of 2; if it overshoots, we redo the update, decrease the learning rate η_t by half and update again. Since it is guaranteed that $Q(\mathbf{A}_{t+1}) \leq Q(\mathbf{A}_t)$, and $Q(\mathbf{A})$ is lower bounded by 0, the iterative process is guaranteed to converge. But it is worth mentioning that here we can also adopt other iterative solution method such as Levenberg-Marquardt algorithm. An illustration of the algorithm is shown in Fig. 1.

In each iteration, the time complexity for computing derivative is $O(d^2 n(|\mathcal{S}| + |\mathcal{D}|))$, where $|\mathcal{S}|$ and $|\mathcal{D}|$ are the numbers of equivalence and non-equivalence constraints, respectively. We set the total number of iterations as $T = 50$.

IV. Experiments

To demonstrate the effectiveness of our approach, we conducted clustering and classification experiments with 8 different datasets: Yale [22], UMIST [23], USPS digits, UCI letters, Ecoli, Glass, Haberman and Libras [24]. These datasets present data from various sources and cover different applications such as face recognition and text classification. In clustering, we compared our approach with different clustering methods based on Euclidean distance and combined with distance metric learning, and we also compared a semi-supervised clustering technique. In classification, we compared our approach with different semi-supervised learning methods based on Euclidean distance and combined with distance metric learning. We also investigated the effect of the exploring unlabeled data (the third term in Eq. 8).

```
1   Initialization
    1.1  Set $t = 0$, $\eta_t = 1$ and initialize $\mathbf{A}_t$ as a diagonal matrix $\frac{1}{\delta}\mathbf{I}$.
    1.2  Construct normalized graph Laplacian $\mathbf{L}_t$ as Eq. 4 and 2.
2   Metric Update
    2.1  Let $\mathbf{A}_{t+1} = \mathbf{A}_t - \eta_t \frac{\partial Q}{\partial \mathbf{A}} \big|_{\mathbf{A}=\mathbf{A}_t}$
    2.2  If $Q(\mathbf{A}_{t+1}) > Q(\mathbf{A}_t)$, set $\eta_{t+1} = 0.5\eta_t$ and $\mathbf{A}_{t+1} = \mathbf{A}_t$; otherwise, $\eta_{t+1} = 2\eta_t$.
3   Let $t = t + 1$. If $t > T$, quit iteration and output $\mathbf{A}$ and $\mathbf{L}$; otherwise, go to step 2.
```

Fig. 1. The normalized graph Laplacian learning algorithm

A. Experimental settings

The Yale dataset consists of faces from 15 individuals, with 11 images for each person. The images cover a wide range of facial expressions and viewpoints. The UMIST database contains 564 images taken from 20 people. These people vary in race, sex and appearance and are presented with different poses from profile to frontal views. All the face images are cropped with reference to the eyes and normalized to 20×20 pixel arrays with 256 gray levels per pixel. Each image is then reshaped to a 400 dimension vector in a fixed order.

The USPS handwritten digit images contain two parts, namely a training part and a testing part, and in our experiments we used the testing part which contains 2007 images. There are 10 classes (digit "0" to "9") and each image is 16×16 in resolution which results in 256-dimensional feature vector. The UCI letter dataset contains 20,000 black-and-white capital letters in the English alphabet, varied in fonts and random distortion. Here we extracted letters A to D with a total of 3,096 samples with each represented by a 16-dimension feature vector. The Ecoli, Glass, Haberman and Libras datasets are all from UCI Machine Learning Repository, and details can be found in [24]. We have summarized the basic characteristics of datasets in Table I.

In our experiments, we first randomly selected training samples. Since the sizes of these datasets vary widely, it is inappropriate to find a fixed number-setting for different datasets. Therefore, we selected different percentages (the percentages are set to 2%, 4%, 6%, 8% and 10%) of samples for training for each dataset. Then we generated constraints accordingly. For example, if a dataset has 100 data points and the sampling percentage is set to 10%, then we have 10 selected training samples and 45 equivalence and non-equivalence constraints will be generated in total. Experiments were conducted with 20 trials and averaged results are reported.

For clustering, we compared the following methods:

1) k-means with Euclidean distance measure.
2) Spectral clustering with Euclidean distance measure, and the radius parameter (see Eq. 1) is tuned to its optimal value.
3) k-means with distance metric learning by the Probabilistic Global Distance Metric Learning (PGDML) algorithm [17].
4) Spectral clustering with distance metric learned by the PGDML algorithm, and the radius parameter (see Eq. 1) is tuned to its optimal value.
5) Metric Pairwise Constrained k-means (MPCKmeans) [14]. It is a semi-supervised clustering method that takes sample constraints into account in the clustering process.
6) Spectral clustering with optimized normalized graph Laplacian, i.e., our proposed approach. The parameter λ is simply set to $1/N$, where N is the size of neighborhood considered for regularization (see Eq. 8). In our experiments we set N to 20.

These six methods are denoted by "K-means+Euclidean", "Spectral+Euclidean", "K-means+PGDML", "Spectral+PGDML", "MPCKmeans" and "Spectral+OptLaplacian", respectively. For spectral clustering we adopted the method proposed in [10]. Balanced Rand Index [17] was used as the performance evaluation metric.

For classification, we compared the following methods:

1) Learning with Local and Global Consistency (LLGC) [4] with Euclidean distance, and the radius parameter is tuned to its optimal value. As previously introduced, LLGC is a graph-based semi-supervised learning algorithm that exploits normalized graph Laplacian.
2) LLGC with distance metric learned by the PGDML algorithm, and the radius parameter is tuned to its optimal value.
3) LLGC with distance metric learned by the NCA algorithm [18], and the radius parameter is tuned to its optimal value.
4) LLGC with distance metric learned by the LMNN algorithm [19], and the radius parameter is tuned to its optimal value.
5) Transductive SVM [25]. It is a semi-supervised classification approach.
6) LLGC with optimized normalized graph Laplacian, i.e., our proposed approach. The parameter λ is with the same setting as the method (6) in clustering.

These six methods are denoted by "LLGC+Euclidean", "LLGC+PGDML", "LLGC+NCA", "LLGC+LMNN", "TSVM" and "LLGC+OptLaplacian", respectively. To implement Transductive SVM in the multi-class problems, we used one-vs-all approach, i.e., we trained an SVM classifier for each class against all other classes, and in prediction the class with largest confidence score was selected. It is worth noting that the LMNN method requires the number of training samples per class to be at least two, so the results of "LLGC+LMNN" will not be illustrated if the requirement is not satisfied.

TABLE I
CHARACTERISTICS OF DATASETS

Dataset	Class#	Sample#	Dataset	Class#	Sample#
USPS	10	2007	LETTER	4	3096
UMIST	20	564	YALE	15	165
Ecoli	8	336	Glass	7	214
Haberman	2	306	Libras	15	360

B. Experimental results

Figures 2 and 3 illustrate the clustering and classification results, respectively. We first compare "Spectral+OptLaplacian" and "Spectral+Euclidean" in clustering, and we can see that the first method is better in nearly all cases. It only performs worse on the Yale and Liras datasets with some settings. This may be attributed to the over fitting of the learning of optimization. For classification we can see that "LLGC+OptLaplacian" performs better than "LLGC+Euclidean" in almost all cases. This demonstrates that the normalized graph Laplacian constructed using our approach is better than that constructed with Euclidean distance. Among the compared methods, we can see that in both clustering and classification our approaches achieved the best results in nearly all the cases. This demonstrates the effectiveness of our proposed method for learning normalized graph Laplacian. In Fig. 2 we can see that, in some cases our performance curves even degrade when the number of constraints increase. This can be attributed to the imbalance of non-equivalence and equivalence constraints. In our current experiment setting, the non-equivalence constraints will be much more than the equivalence constraints if there is fairly large number of classes. Consider a C-class classification problem and there are n training samples selected for training in each class. It can be computed that the numbers of non-equivalence and equivalence are $Cn(C-1)n/2$ and $Cn(n-1)/2$, respectively, which results in a ratio of $\frac{(n-1)}{(C-1)n}$. The Fig. 2 (a), (e) and (f), which have obvious performance degrade, correspond to 10-, 8-, and 15-class classification problems, respectively. One method to address the problem is to add a tunable parameter in Eq. 8 to modulate the impacts of equivalence and non-equivalence constraints. Anyway, even with the imbalance problem, our algorithm still shows superior performance than other methods. We leave the investigation of the imbalance problem to our future work.

From Fig 3, we can see that in most cases, our method outperforms the LLGC method with different distance metric learning methods ("LLGC+PGDM", "LLGC+NCA" and "LLGC+LMNN"). In comparison with these distance metric learning methods, our approach has two advantages:

(1) We explore unlabeled data, whereas the metric learning methods only used labeled samples. This makes our method robust even with only few labeled pairs.

(2) These metric learning methods only learn distance metric and then need to convert it to similarity measurements to be applied to several algorithms such as spectral clustering (this process will involve a scaling factor σ). Thus a good distance metric with the criteria of these methods may not be good for several similarity-based algorithms. Our approach can avoid the problem by directly learning the normalized graph laplacian and similarity measurements (the scaling of distance

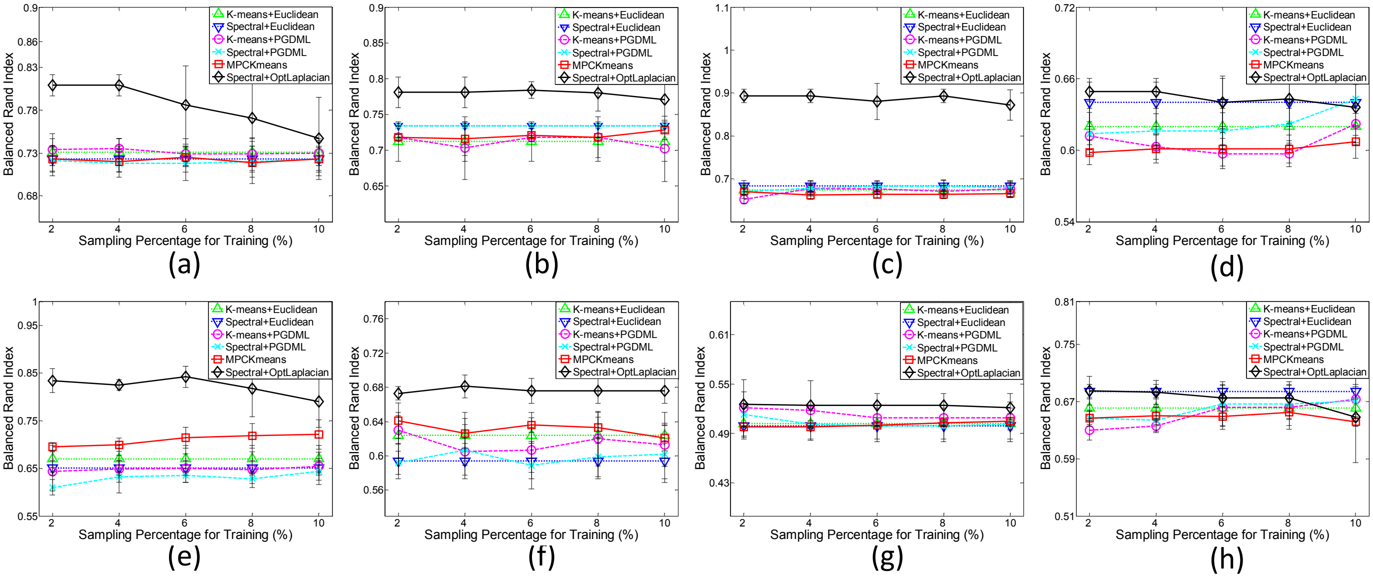

Fig. 2. Clustering performance comparison of different methods on the eight datasets: (a) USPS (b) LETTER (c) UMIST (d) YALE (e) Ecoli (f) Glass (g) Haberman (h) Libras. *(best viewed in color)*

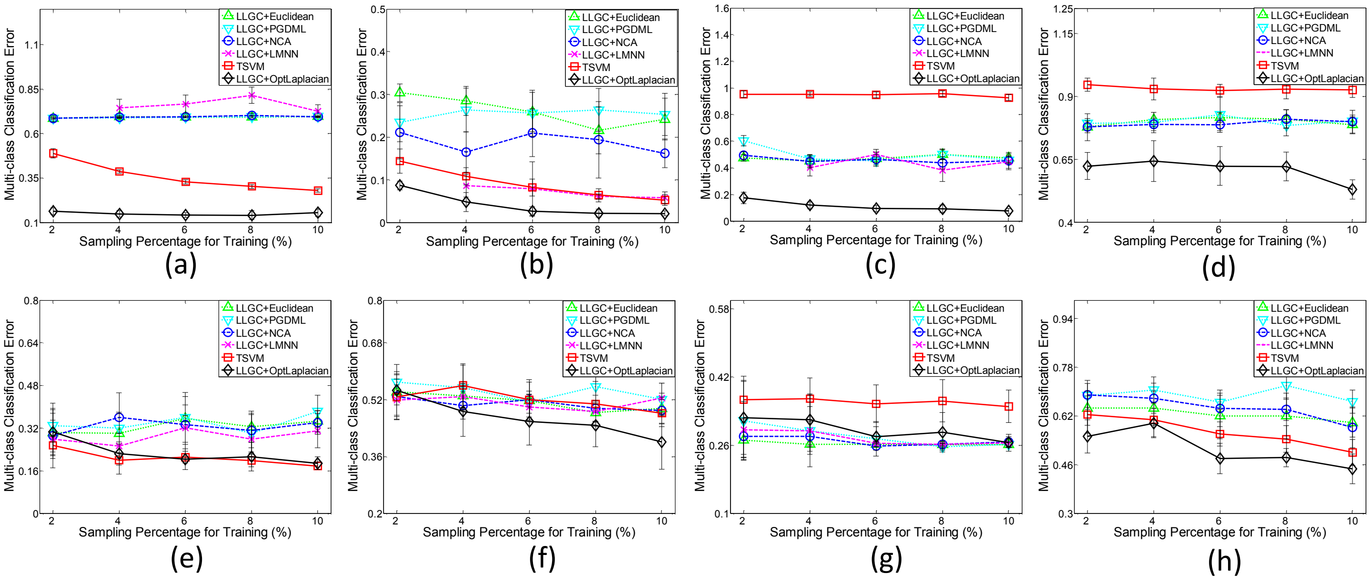

Fig. 3. Classification performance comparison of different methods on the eight datasets: (a) USPS (b) LETTER (c) UMIST (d) YALE (e) Ecoli (f) Glass (g) Haberman (h) Libras. **Note**: the LMNN method requires the number of training samples per class to be at least two, so the results of "LLGC+LMNN" will not be illustrated if the requirement is not satisfied. *(best viewed in color)*

TABLE II

CLUSTERING PERFORMANCE COMPARISON OF LEARNING WITH AND WITHOUT NEIGHBORHOOD CONSISTENCY PROBLEM. THE RESULTS THAT ARE STATISTICALLY SUPERIOR TO THE COMPARED ONES ARE BOLDED (PAIR-WISE T-TEST WITH 95% SIGNIFICANCE LEVEL)

	Sampling Rate for Training									
	2%		4%		6%		8%		10%	
Dataset	*Method 1	*Method 2	Method 1	Method 2	Method 1	Method 2	Method 1	Method 2	Method 1	Method 2
USPS	**0.809±0.012**	0.798±0.015	**0.809±0.012**	0.796±0.026	**0.786±0.045**	0.746±0.055	0.771±0.056	0.755±0.040	0.747±0.048	0.742±0.078
LETTER	**0.781±0.021**	0.639±0.071	**0.781±0.021**	0.598±0.109	**0.784±0.011**	0.634±0.086	**0.780±0.026**	0.659±0.111	**0.771±0.046**	0.666±0.084
UMIST	0.893±0.016	**0.912±0.009**	**0.893±0.016**	0.830±0.128	**0.880±0.042**	0.855±0.039	**0.893±0.016**	0.869±0.032	**0.872±0.036**	0.842±0.047
YALE	**0.649±0.008**	0.643±0.007	0.649±0.008	**0.657±0.013**	0.640±0.022	**0.650±0.009**	0.643±0.021	0.650±0.010	**0.636±0.023**	0.614±0.048
Ecoli	0.834±0.025	0.832±0.027	0.825±0.011	0.806±0.060	**0.842±0.023**	0.727±0.082	**0.817±0.059**	0.754±0.056	0.790±0.054	**0.821±0.025**
Glass	**0.673±0.008**	0.657±0.039	**0.681±0.014**	0.614±0.077	**0.676±0.014**	0.635±0.084	**0.676±0.014**	0.599±0.085	**0.676±0.014**	0.636±0.062
Haberman	**0.525±0.002**	0.513±0.019	**0.524±0.000**	0.506±0.003	**0.524±0.000**	0.504±0.008	**0.524±0.000**	0.514±0.018	**0.521±0.007**	0.514±0.018
Libras	0.685±0.020	0.678±0.020	0.683±0.017	0.685±0.017	0.675±0.013	0.684±0.052	0.675±0.013	0.673±0.062	0.648±0.063	**0.694±0.028**

**Method 1 refers to the algorithm with neighborhood consistency term while Method 2 is without the term.*

TABLE III

CLASSIFICATION ERROR COMPARISON OF LEARNING WITH AND WITHOUT NEIGHBORHOOD CONSISTENCY PROBLEM. THE RESULTS THAT ARE STATISTICALLY SUPERIOR TO THE COMPARED ONES ARE BOLDED (PAIR-WISE T-TEST WITH 95% SIGNIFICANCE LEVEL)

	Sampling Rate for Training									
	2%		4%		6%		8%		10%	
Dataset	*Method 1	*Method 2	Method 1	Method 2	Method 1	Method 2	Method 1	Method 2	Method 1	Method 2
USPS	**0.163±0.019**	0.213±0.037	**0.149±0.020**	0.194±0.034	**0.142±0.014**	0.168±0.020	**0.141±0.025**	0.173±0.030	0.155±0.021	0.163±0.016
LETTER	**0.087±0.010**	0.112±0.037	**0.048±0.022**	0.079±0.017	**0.026±0.011**	0.063±0.019	**0.022±0.010**	0.054±0.012	**0.021±0.006**	0.056±0.018
UMIST	0.175±0.043	**0.152±0.028**	**0.119±0.026**	0.160±0.041	**0.095±0.024**	0.124±0.041	**0.094±0.029**	0.130±0.054	0.077±0.018	0.091±0.046
YALE	0.623±0.054	**0.569±0.041**	0.643±0.081	**0.596±0.075**	0.623±0.077	0.630±0.069	0.621±0.057	**0.589±0.049**	0.530±0.039	0.544±0.096
Ecoli	0.305±0.089	0.275±0.071	**0.224±0.077**	0.325±0.070	**0.203±0.039**	0.280±0.083	0.213±0.053	0.237±0.052	**0.188±0.024**	0.260±0.044
Glass	0.546±0.073	0.542±0.073	**0.487±0.070**	0.530±0.079	**0.458±0.065**	0.505±0.084	**0.448±0.060**	0.556±0.111	**0.401±0.077**	0.491±0.126
Haberman	0.324±0.098	0.374±0.093	**0.320±0.060**	0.377±0.120	**0.280±0.024**	0.380±0.085	0.290±0.047	0.313±0.058	**0.266±0.020**	0.279±0.020
Libras	**0.553±0.054**	0.599±0.047	0.596±0.048	0.597±0.045	**0.480±0.050**	0.526±0.058	0.483±0.030	0.501±0.061	**0.446±0.049**	0.529±0.072

**Method 1 refers to the algorithm with neighborhood consistency term while Method 2 is without the term.*

metric has been learned implicitly in this process). These two advantages make the performance of our approach better.

From Eq. 8 we can see that the method has explored unlabeled data, but we can also choose not to use unlabeled data by removing the third item in the equation. We conducted experiments without the neighborhood consistency term as mentioned above and reported performance comparison below. Tables II and III show the performance comparison of whether or not exploring unlabeled data in clustering and classification, respectively. We can see that in most cases our method performs better than the method without the neighborhood consistency term in both clustering and classification, and it thus validates the effectiveness of the neighborhood consistency term.

V. CONCLUSION

This paper proposed a unified scheme to learn normalized graph Laplacian with equivalence and non-equivalence constraints among samples. We formulated a regularization framework based on the consistency of the normalized graph Laplacian with the constraint information as well as neighborhood information, and it can be solved with an efficient gradient descent process. We have used the normalized graph Laplacian in both clustering and classification, and empirical results on a variety of datasets have demonstrated the effectiveness of our approach. In our future work, we will investigate the imbalance problem of constraints and we will also study the sparsification of graph Laplacian for further performance improvement.

REFERENCES

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000.

[2] J. Huang, "A combinatorial view of graph laplacians," Max Planck Institute, Tech. Rep. no. 144, 2005.

[3] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of 20th International Conference on Machine Learning*, 2003.

[4] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Proceedings of Advances of Neural Information Processing Systems*, 2004.

[5] T. Kato, H. Kashima, and M. Sugiyama, "Robust label propagation on multiple networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, 2009.

[6] D. Y. Yeung and H. Chang, "A kernel approach for semisupervised metric learning," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, 2007.

[7] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings of 18th International Conference on Machine Learning*, 2001.

[8] F. Wang and C. Zhang, "Semi-supervised learning based on generalized point charge models," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, 2008.

[9] M. Adankon, M. Cheriet, and A. Biem, "Semi-supervised least squares support vector machine," *IEEE Transactions on Neural Networks*, vol. 20, no. 12, 2009.

[10] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Proceedings of Advances of Neural Information Processing Systems*, 2001.

[11] A. Szymkowiak-Have, M. Girolami, and J. Larsen, "Clustering via kernel decomposition," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, 2006.

[12] X. He, D. Cai, Y. Shao, H. Bao, and J. Han, "Laplacian regularized gaussian mixture model for data clustering," *IEEE Transactions on Knowledge and Data Engineering*, to appear.

[13] X. He, "Laplacian regularized d-optimal design for active learning and its application to image retrieval," *IEEE Transactions on Image Processing*, vol. 19, no. 1, 2010.

[14] S. Basu, M. Bilenko, A. Banerjee, and R. Mooney, "Probabilistic semi-supervised clustering with constraints," in *Semi-Supervised Learning*. MIT Press, 2006.

[15] M. Wang, X. S. Hua, J. Tang, and R. Hong, "Beyond distance measurement: Constructing neighborhood similarity for video annotation," *IEEE Transactions on Multimedia*, vol. 11, no. 3, 2009.

[16] M. Wang, X. S. Hua, R. Hong, J. Tang, G. J. Qi, and Y. Song, "Unified video annotation via multi-graph learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 5, 2009.

[17] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proceedings of Advances of Neural Information Processing Systems*, 2003.

[18] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proceedings of Advances of Neural Information Processing Systems*, 2005.

[19] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proceedings of Advances of Neural Information Processing Systems*, 2005.

[20] M. Belkin and P. Niyogi, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *Neural Computation*, vol. 15, no. 6, 2003.

[21] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceedings of International Conference on Machine Learning*, 2003.

[22] P. Belhumeour, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 1997.

[23] D. Graham and N. Allinson, "Characterizing virtual eigen-signatures for general purpose face recognition," in *Face Recognition: From Theory to Applications*, vol. 163, 1998.

[24] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[25] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of International Conference on Machine Learning*, 1999.