Robust Multi-Class Transductive Learning with Graphs

Wei Liu
Electrical Engineering Department
Columbia University
wliu@ee.columbia.edu

Shih-Fu Chang
Electrical Engineering Department
Columbia University
sfchang@ee.columbia.edu

Abstract

Graph-based methods form a main category of semi-supervised learning, offering flexibility and easy implementation in many applications. However, the performance of these methods is often sensitive to the construction of a neighborhood graph, which is non-trivial for many real-world problems. In this paper, we propose a novel framework that builds on learning the graph given labeled and unlabeled data. The paper has two major contributions. Firstly, we use a nonparametric algorithm to learn the entire adjacency matrix of a symmetry-favored k-NN graph, assuming that the matrix is doubly stochastic. The nonparametric algorithm makes the constructed graph highly robust to noisy samples and capable of approximating underlying submanifolds or clusters. Secondly, to address multi-class semi-supervised classification, we formulate a constrained label propagation problem on the learned graph by incorporating class priors, leading to a simple closed-form solution. Experimental results on both synthetic and real-world datasets show that our approach is significantly better than the state-of-the-art graph-based semi-supervised learning algorithms in terms of accuracy and robustness.

1. Introduction

Semi-supervised learning, typically *transductive learning*, deals with classification tasks through utilizing both labeled and unlabeled examples [15]. The classification methods usually handle the situations when a few labeled data together with large amounts of unlabeled data are available. This learning scenario has been increasingly popular in a lot of practical problems, since it is quite feasible to obtain unlabeled data by an automatic procedure but quite expensive to identify the labels of data.

Among current research on semi-supervised classification, Transductive Support Vector Machines (TSVMs) [6] aimed at optimizing margins of both labeled and unlabeled examples. [3] implemented the cluster assumption which favors decision boundaries for classification passing through low-density regions in the input sample space. A bunch of graph-based approaches [14][16][2][1][10][11] put forward various graph-based optimization frameworks with similar regularization terms.

The paramount foundation of semi-supervised learning is an appropriate assumption about data distribution. Two commonly adopted assumptions are the *cluster assumption* and the *manifold assumption*. The former assumes that samples associated with the same structure, typically a cluster or a submanifold, are very likely to have the same label [6][3]. The latter often implies that nearby sample points on manifolds are likely to take the same label. Notice that the cluster assumption is global whereas the manifold assumption is often local. Numerous semi-supervised learning methods such as [14][16][1][10] exploit such a manifold assumption to pursue smooth classification or prediction functions along manifolds. Specifically, all these methods represent both labeled and unlabeled instances by a graph, and then utilize its graph Laplacian matrix to characterize the manifold structure.

1.1. Motivation

Although graph-based semi-supervised learning has been studied extensively, it often lacks sufficient robustness in real-world learning tasks because of the sensitivity of graphs. The quality of graphs is very sensible to the topological structure, the choice of weighting functions and the related parameters. These factors will considerably influence the performance of semi-supervised learning approaches. Therefore, suitable methods for graph construction are needed.

Let us consider the classical two moons toy problem to explain our motivation for proper graph construction. As shown in Fig. 1, we are given a set of points in a two-moon shape plus some extra points which are either outliers or from rare classes. We simply consider these extra points as noisy points. Two points on the upper moon and lower moon are labeled as '+1' and '-1', respectively. Intuitively, the points in the upper moon should be labeled as '+1' while those in the lower moon should be '-1'. Our target is to

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

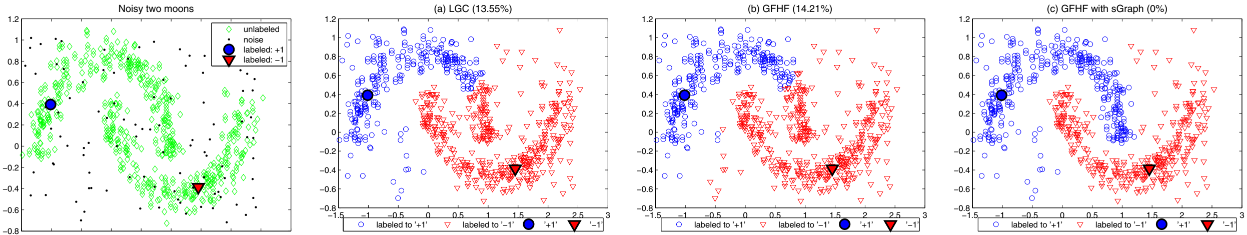

Figure 1. Noisy two moons given two labeled points. (a) LGC [14] with 13.55% error rate using a 10-NN graph; (b) GFHF [16] with 14.21% error rate using a 10-NN graph; (c) GFHF with zero error rate using a symmetry-favored 10-NN graph.

build a graph to highlight proximities among data points residing on two main submanifolds (i.e., two moons) so that the graph is relatively robust to the noisy points.

Using the traditional k-NN graph ($k = 10$), we run two well-know semi-supervised learning algorithms: the *Local and Global Consistency* (LGC) method [14] and the *Gaussian Field and Harmonic Function* (GFHF) method [16] on this toy problem. We only have ground truth labels for the points on two moons, so we evaluate classification performance on these on-manifold points. The visual classification results are displayed in Fig. 1(a)-(c), from which we observe quite a few errors when using the k-NN graph but zero mistake when using the symmetry-favored k-NN graph that will be proposed in Section 2. With the new graph, the performance of GFHF is significantly improved, with a gain of 100% over the old graph.[1] This phenomenon illustrates that graph quality is really critical to semi-supervised learning, and the same method might lead to very different results using different graph construction schemes.

As the weighting functions also influence the outcome performance, this paper proposes to automatically learn the entire adjacency matrix of a graph in a nonparametric mode, and then establishes an algorithmic framework, robust multi-class graph transduction, to integrate graph learning and multi-class label propagation.

2. Graphs and Graph Laplacians

Graph-based methods presume that data are represented in the form of undirected or directed graphs. Graph-based semi-supervised learning frequently exploits undirected graphs. In this paper, we aim at learning a set of real-valued label prediction functions taking as input an undirected weighted graph $G = (V, E, W)$. V is a set of vertices with each of them representing a data point, $E \subseteq V \times V$ is a set of edges connecting adjacent data points, and $W : E \rightarrow \mathbb{R}^+$ is a weighting function to measure the strength of connections.

Despite the progress made in the theory and practice for learning with graphs [15], the way to establish high-quality graphs is still an open problem. In this section, we propose a simple graph construction scheme as the building block of our learning framework.

2.1. Symmetry-Favored Graph Construction

Consider a sample set $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^d$ in which, without loss of generality, the first l samples are labeled and the remaining $u = n - l$ ones are unlabeled. In the graph G each vertex v_i essentially corresponds to each instance $\mathbf{x}_i$, so we also refer to $\mathbf{x}_i$ as a vertex. Then we put an edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ if $\mathbf{x}_i$ is among the k nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the k nearest neighbors of $\mathbf{x}_i$. The graph G is thus a k-NN graph. Although there are other strategies for setting up edges over data points, it turns out that a k-NN graph has advantages over others (e.g., a h-neighborhood graph) as shown in [4]. One of main advantages is that a k-NN graph provides a better adaptive connectivity.

Let us start by defining an asymmetric $n \times n$ matrix:

$$\mathbf{A}_{ij} = \begin{cases} \exp\left(-\frac{d(\mathbf{x}_i,\mathbf{x}_j)^2}{\sigma^2}\right), & \text{if } j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where the set $\mathcal{N}_i$ saves the indexes of k nearest neighbors of point $\mathbf{x}_i$ and $d(\mathbf{x}_i, \mathbf{x}_j)$ is some distance metric between $\mathbf{x}_i$ and $\mathbf{x}_j$. Typically, $d(,)$ refers to the Euclidean distance. The parameter σ is empirically estimated by $\sigma = \sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{x}_{i_k})/n$ where $\mathbf{x}_{i_k}$ is the k-th nearest neighbor of $\mathbf{x}_i$. Such an estimation is simple and sufficiently effective, which will be verified in the experimental section. Based on matrix $\mathbf{A}$, we can define the weighted adjacency matrix of G:

$$\mathbf{W}_{ij} = \begin{cases} \mathbf{A}_{ij} + \mathbf{A}_{ji}, & \text{if } j \in \mathcal{N}_i \text{ and } i \in \mathcal{N}_j \\ \mathbf{A}_{ji}, & \text{if } j \notin \mathcal{N}_i \text{ and } i \in \mathcal{N}_j \\ \mathbf{A}_{ij}, & \text{otherwise} \end{cases} \tag{2}$$

Obviously, $\mathbf{W} = \mathbf{A} + \mathbf{A}^T$ is symmetric with $\mathbf{W}_{ii} = 0$ (to avoid self loops). This weighting scheme favors the *symmetric* edges $(\mathbf{x}_i, \mathbf{x}_j)$ such that $\mathbf{x}_i$ is among k-NNs of $\mathbf{x}_j$ and $\mathbf{x}_j$ is simultaneously among k-NNs of $\mathbf{x}_i$. As done in eq. (2), weights of those symmetric edges are enlarged explicitly due to the reasonable consideration that two points connected by a symmetric edge are prone to be on the same

[1] LGC gets the same error rate using the symmetry-favored k-NN graph.

submanifold. In contrast, traditional weighting schemes [4] treat all edges in the same manner, which define the weighted adjacency matrix by $\max\{\mathbf{A}, \mathbf{A}^T\}$. We call the k-NN graph constructed through eq. (2) the *symmetry-favored* k-NN graph or k-NN sGraph in abbreviation. The proposed symmetry-favored graph is relatively robust to noise as it reinforces the similarities between points on manifolds.

2.2. Graph Laplacians

Suppose that $\mathcal{H}(G)$ is a linear space of real-valued functions defined on the vertex set V of the graph G, on which we define an inner-product between $f, g \in \mathcal{H}(G)$ as $< f, g >_{\mathcal{H}(G)} = \sum_{i=1}^{n} f(v_i)g(v_i)$. If we consider f as a label prediction function $f : V \to [0, 1]$, we will hope that f varies smoothly along the edges of G since adjacent vertices are very likely to have similar labels. The smooth semi-norm used in most graph-based semi-supervised learning approaches is the following functional

$$\|f\|_G^2 = \frac{1}{2}\sum_{i,j=1}^{n}(f(v_i) - f(v_j))^2\mathbf{W}_{ij}, \tag{3}$$

which sums weighted variations of the function f on all edges of the graph G. This semi-norm has been proved geometrically meaningful and hereby applied to many fields to measure the smoothness of functions.

To delve into the smooth norm $\|f\|_G$, we elicit the well-know *graph Laplacian* matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{4}$$

where $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal degree matrix such that $\mathbf{D}_{ii} = \sum_{j=1}^{n}\mathbf{W}_{ij}$. $\mathbf{D}_{ii}$ is actually related to the neighborhood density of sample $\mathbf{x}_i$. We introduce the discrete Laplace operator: $\Delta : \mathcal{H}(G) \to \mathcal{H}(G)$ by

$$(\Delta f)(v_i) = f(v_i) - \sum_{j=1}^{n}\frac{\mathbf{W}_{ij}}{\mathbf{D}_{ii}}f(v_j), \tag{5}$$

and a scaling operator $\mathcal{D} : \mathcal{H}(G) \to \mathcal{H}(G)$ by $(\mathcal{D}f)(v_i) = \mathbf{D}_{ii}f(v_i)$. Then, we have the following theorem. Apparently, $\|f\|_G^2 = \mathbf{f}^T\mathbf{L}\mathbf{f} \geq 0$ and the graph Laplacian matrix $\mathbf{L}$ is thus positive semi-definite.

Theorem 1. *Represent each $f \in \mathcal{H}(V)$ as a vector $\mathbf{f} = [f(v_1), \cdots, f(v_n)]^T \in \mathbb{R}^n$. Then*

$$\|f\|_G^2 = < \mathcal{D}f, \Delta f >_{\mathcal{H}(G)} = \mathbf{f}^T\mathbf{L}\mathbf{f}. \tag{6}$$

3. Robust Multi-Class Graph Transduction

This section will address multi-class transductive learning tasks using nonparametric graph adjacency matrices. Suppose that there are c classes $\{\mathcal{C}_k\}_{k=1}^{c}$ occurred in the labeled data $(\mathbf{x}_i, \mathbf{Y}_{i.})_{i=1}^{l}$. The notation $\mathbf{Y}_{i.} \in \mathbb{R}^{1\times c}$ indicates the class assignment of point $\mathbf{x}_i$, i.e., $\mathbf{Y}_{ik} = 1$ if $\mathbf{x}_i \in \mathcal{C}_k$ and otherwise $\mathbf{Y}_{ik} = 0$. In each class $\mathcal{C}_k$, we have $l_k = |\mathcal{C}_k|$ samples.

3.1. Learning the Graph Adjacency Matrix

Now we consider the problem of learning the graph adjacency matrix. From Theorem 1, we can see that the smooth norm emphasizes neighborhoods of high densities (large degrees $\mathbf{D}_{ii}$). Sampling is usually not uniform in practice, so over-emphasizing the neighborhoods with high densities may occlude the information in sparse regions in the smooth norm. To compensate for this effect, degree-normalization is often enforced, which is traditionally implemented by either $\mathbf{D}^{-1}\mathbf{W}$ or a symmetric normalization $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ [14]. However, the first one gives an asymmetric adjacency matrix, while the second one still holds inhomogeneous degrees.

In this paper, we choose to enforce the degree constraint that all vertices in the graph have the same degree $\mathbf{D}_{ii} = 1$, i.e., $\mathbf{W}\mathbf{1} = \mathbf{1}$. Since the adjacency matrix is always symmetric and non-negative, setting $\mathbf{W}\mathbf{1} = \mathbf{1}$ makes it a doubly-stochastic matrix [5]. Such kind of matrices exhibit good performance on spectral clustering [12] because their top eigenvectors tend to be piecewise constant.

Furthermore, we try to learn the doubly-stochastic adjacency matrix from training examples. We do not assume any function form for $\mathbf{W}$. Instead, we utilize only the natural assumption that $\mathbf{W}$ is close to the initial adjacency matrix $\mathbf{W}^0$ defined via eq. (2). There are three merits of learning the doubly-stochastic adjacency matrix: 1) it offers a nonparametric form for the neighborhood graph G, flexibly representing data lying in compact clusters or intrinsic low-dimensional submanifolds; 2) it is highly robust to noise, e.g., when a noisy sample $\mathbf{x}_j$ invades the neighborhood of $\mathbf{x}_i$ on some cluster/manifold, the unit-degree constraint makes the weight $\mathbf{W}_{ij}$ absolutely small compared to the weights between $\mathbf{x}_i$ and closer neighbors; 3) it provides the "balanced" graph Laplacian with which the smooth functional norm penalizes prediction functions on each item uniformly, resulting in uniform label propagation when applied to transductive learning.

It is desirable that we can infuse semi-supervised information into $\mathbf{W}$. Suppose a pair set $\mathcal{T} = \{(i,j)|i = j \text{ or } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are differently labeled}\}$ and define its matrix form

$$\mathbf{T}_{ij} = \begin{cases} 1, \text{ if } (i,j) \in \mathcal{T} \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

Clearly, we have $|\mathcal{T}| = n + l^2 - \sum_{k=1}^{c} l_k^2$. In particular, we require $\mathbf{W}_{ij} = 0$ for $(i,j) \in \mathcal{T}$ or equivalently require $\sum_{(i,j)\in\mathcal{T}}\mathbf{W}_{ij} = 0$ due to $\mathbf{W} \geq 0$. This constraint is very intuitive since it removes self loops and all possible edges connecting differently labeled points. It is worthwhile to point out that we only employ the differently labeled relationship, not the same labeled relationship. Because points sharing the same label may be very far or very close, we cannot make any decisions on the weights between them.

So far, we formulate learning doubly-stochastic $\mathbf{W}$ subject to differently labeled information as follows

$$\begin{aligned} \min \quad & \mathcal{G}(\mathbf{W}) = \frac{1}{2}\|\mathbf{W} - \mathbf{W}^0\|_F^2 \\ s.t. \quad & \sum_{(i,j)\in\mathcal{T}} \mathbf{W}_{ij} = 0 \\ & \mathbf{W}\mathbf{1} = \mathbf{1},\ \mathbf{W} = \mathbf{W}^T,\ \mathbf{W} \geq 0 \end{aligned} \tag{8}$$

where $\|.\|_F$ stands for the Frobenius norm. Eq. (8) falls into an instance of quadratic programming (QP). For efficient computation, we divide the QP problem into two convex sub-problems

$$\begin{aligned} \min \quad & \mathcal{G}(\mathbf{W}) = \frac{1}{2}\|\mathbf{W} - \mathbf{W}^0\|_F^2 \\ s.t. \quad & \sum_{(i,j)\in\mathcal{T}} \mathbf{W}_{ij} = 0,\ \mathbf{W}\mathbf{1} = \mathbf{1},\ \mathbf{W} = \mathbf{W}^T \end{aligned} \tag{9}$$

and

$$\min \mathcal{G}(\mathbf{W}) = \frac{1}{2}\|\mathbf{W} - \mathbf{W}^0\|_F^2 \quad s.t. \mathbf{W} \geq 0 \tag{10}$$

Right now, we find a simple solution to the latter sub-problem: $\mathbf{W} = \lceil \mathbf{W}^0 \rceil_{\geq 0}$ in which the operator $\lceil \mathbf{W}^0 \rceil_{\geq 0}$ zeros out all negative entries of $\mathbf{W}^0$. The operator is essentially a conic subspace projection operator.

In order to solve the sub-problem eq. (9), we take the Lagrangian as follows

$$\begin{aligned} \mathcal{G}(\mathbf{W}, r, \boldsymbol{\mu}) = & \frac{1}{2}\|\mathbf{W} - \mathbf{W}^0\|_F^2 - r \sum_{(i,j)\in T} \mathbf{W}_{ij} \\ & -\boldsymbol{\mu}^T(\mathbf{W}\mathbf{1} - \mathbf{1}) - \boldsymbol{\mu}^T(\mathbf{W}^T\mathbf{1} - \mathbf{1}), \end{aligned}$$

and designate $\partial\mathcal{G}/\partial\mathbf{W} = 0$ to obtain

$$\mathbf{W} = \mathbf{W}^0 + r\mathbf{T} + \boldsymbol{\mu}\mathbf{1}^T + \mathbf{1}\boldsymbol{\mu}^T, \tag{11}$$

which always satisfies the symmetric property $\mathbf{W} = \mathbf{W}^T$. To fulfill $\sum_{(i,j)\in\mathcal{T}} \mathbf{W}_{ij} = 0$, we let $r = -t^0 - 2\frac{\mathbf{1}^T\mathbf{T}\boldsymbol{\mu}}{|\mathcal{T}|}$ where $t^0 = \sum_{(i,j)\in\mathcal{T}} \mathbf{W}^0_{ij}/|\mathcal{T}|$. After plugging r in eq. (11) and multiplying the two sides by $\mathbf{1}$, we obtain

$$\boldsymbol{\mu} = \widehat{\mathbf{T}}(\mathbf{1} - \mathbf{W}^0\mathbf{1} + t^0\mathbf{T}\mathbf{1}),$$

in which

$$\widehat{\mathbf{T}} = \frac{\mathbf{I}}{n} - \frac{\mathbf{1}\mathbf{1}^T}{2n^2} - \frac{(\mathbf{T} - \frac{|\mathcal{T}|}{2n}\mathbf{I})\mathbf{1}\mathbf{1}^T(\mathbf{T} - \frac{|\mathcal{T}|}{2n}\mathbf{I})}{n\mathbf{1}_l^T\mathbf{T}_{ll}^2\mathbf{1}_l + nu - \frac{|\mathcal{T}|}{2}n^2 - \frac{|\mathcal{T}|^2}{2}},$$

where $\mathbf{1}_l$ is an l-dimensional vector of 1's and $\mathbf{T}_{ll}$ is the top-left $l \times l$ submatrix of $\mathbf{T}$. It follows that the multiplier $\boldsymbol{\mu}$ depends on the initial weight matrix $\mathbf{W}^0$ and matrix $\mathbf{T}$ containing semi-supervised information, so we also denote it as $\boldsymbol{\mu}^0(\mathbf{W}^0, \mathbf{T})$. We rewrite eq. (11) with solved $\boldsymbol{\mu}^0$ by

$$\begin{aligned} \mathbf{W} = \mathcal{P}(\mathbf{W}^0, \mathbf{T}) = \mathbf{W}^0 - & \left(t^0 + \frac{2\mathbf{1}^T\mathbf{T}\boldsymbol{\mu}^0}{|\mathcal{T}|}\right)\mathbf{T} \\ & +\boldsymbol{\mu}^0\mathbf{1}^T + \mathbf{1}{\boldsymbol{\mu}^0}^T, \end{aligned} \tag{12}$$

where $\mathcal{P}(\mathbf{W}^0, \mathbf{T})$ denotes the projection operator that computes the affine subspace $\mathbf{W}$ from the initial weight matrix $\mathbf{W}^0$ controlled by $\mathbf{T}$.

Incorporating the above derivations, we tackle the original QP problem in eq. (8) by successively alternating between two sub-problems in eq. (9) and (10). The solution path is sketched as follows

$$\mathbf{W}^0 \xrightarrow{\mathcal{P}()} \mathbf{W}^1 \xrightarrow{\lceil\rceil_{\geq 0}} \mathbf{W}^1 \xrightarrow{\mathcal{P}()} \mathbf{W}^2 \xrightarrow{\lceil\rceil_{\geq 0}} \mathbf{W}^2 \longrightarrow \cdots$$

in which the input $\mathbf{W}^0$ to eq. (9) or (10) is always substituted by the current solution $\mathbf{W}^m$. While computing $\mathcal{P}(\mathbf{W}^m, \mathbf{T})$ $(m \geq 1)$, we update t^0 and $\boldsymbol{\mu}^0$ according to current $\mathbf{W}^m$:

$$t^m = \sum_{(i,j)\in\mathcal{T}} \mathbf{W}^m_{ij}/|\mathcal{T}|,\ \boldsymbol{\mu}^m = \widehat{\mathbf{T}}(\mathbf{1} - \mathbf{W}^m\mathbf{1} + t^m\mathbf{T}\mathbf{1}). \tag{13}$$

Due to Von-Neumann's successive projection lemma [8], the alternate projection process will converge onto the intersect of the affine and conic subspaces given by $\mathcal{P}(\mathbf{W}, \mathbf{T})$ and $\lceil \mathbf{W} \rceil_{\geq 0}$, respectively. Importantly, the Von-Neumann's lemma ensures that alternately solving sub-problems eq. (9) and (10) with the current solution as input is theoretically guaranteed to converge to the global optimal solution of the target problem eq. (8).

We describe the nonparametric adjacency matrix learning algorithm in Table 1, which converges within a few iterations in practice. Our algorithm is able to deal with all multi-class transductive learning tasks, while the hyperparameter learning algorithm proposed in [13] is neither nonparametric (assuming a parametric form for $\mathbf{W}$) nor capable of addressing multi-class problems. Even though it can handle two-class problems, the hyperparameter learning process is quite tedious. One key advantage of our algorithm is that it has no explicit dependence on the number of classes since it only utilizes the differently labeled information $\mathbf{T}$.

3.2. Constrained Multi-Class Label Propagation

We now turn our attention to the multi-class transductive learning algorithm built upon the learned graph adjacency matrix $\mathbf{W}$ and its resulting graph Laplacian $\mathbf{L}$. Given the multi-class labeled data $(\mathbf{x}_i, \mathbf{Y}_{i.})_{i=1}^l$, we aim at learning a

Table 1. Algorithm 1.

```
Alg. 1. Nonparametric Adjacency Matrix Learning
INPUT:  the initial adjacency matrix $\mathbf{W}^0$
        the differently labeled information $\mathbf{T}$
        the maximum iteration number $MaxIter$.
LOOP:   $m = 1, \cdots, MaxIter$
            $\mathbf{W}^m = \mathcal{P}(\mathbf{W}^{m-1}, \mathbf{T})$ (using eq. (13)(12))
            If $\mathbf{W}^m \geq 0$ stop LOOP;
            else $\mathbf{W}^m = \lceil \mathbf{W}^m \rceil_{\geq 0}$.
OUTPUT: $\mathbf{W} = \mathbf{W}^m$.
```

class assignment $\mathbf{F}_{i.} \in \mathbb{R}^{1\times c}$ for each unlabeled data point $\mathbf{x}_i$ ($i = l+1, \cdots, n$), and the final classifier is $y(\mathbf{x}_i) = \arg\max_{1\leq k\leq c} \mathbf{F}_{ik}$. In a matrix form, we denote the global classification result by $\mathbf{F} = [\mathbf{F}_l^T, \mathbf{F}_u^T]^T \in \mathbb{R}^{n\times c}$ in which $\mathbf{F}_l = \mathbf{Y}_l$ is the known class assignment and $\mathbf{F}_u \in \mathbb{R}^{u\times c}$ is the target variable. Note that $\mathbf{F}_u$ is in a soft range of the hard label values 0 and 1, so we call it the soft label matrix. From another view, each column $\mathbf{F}_{.k}$ in $\mathbf{F}$ accounts for each class $\mathcal{C}_k$ and its corresponding function form $f_k \in \mathcal{H}(G)$ should minimize the smooth norm proved by Theorem 1. Consequently, we obtain a cost function under the multi-class setting

$$\mathcal{Q}(\mathbf{F}) = \sum_{k=1}^{c} \|f_k\|_G^2 = \sum_{k=1}^{c} \mathbf{F}_{.k}^T \mathbf{L} \mathbf{F}_{.k} = tr(\mathbf{F}^T\mathbf{L}\mathbf{F}), \quad (14)$$

where $tr()$ stands for the matrix trace operator.

It suffices to assume the class posterior probabilities for the labeled data be $p(\mathcal{C}_k|\mathbf{x}_i) = \mathbf{F}_{ik} = \mathbf{Y}_{ik} = 1$ if $\mathbf{x}_i \in \mathcal{C}_k$ and $p(\mathcal{C}_k|\mathbf{x}_i) = \mathbf{F}_{ik} = \mathbf{Y}_{ik} = 0$ otherwise. If we knew class priors $\boldsymbol{\omega} = [p(\mathcal{C}_1), \cdots, p(\mathcal{C}_c)]^T$ ($\boldsymbol{\omega}^T\mathbf{1}_c = 1$) and regarded soft labels $\mathbf{F}_{ik}$ as posterior probabilities $p(\mathcal{C}_k|\mathbf{x}_i)$, we would have the equation

$$\begin{aligned} p(\mathcal{C}_k) &= \sum_{i=1}^{n} p(\mathbf{x}_i)p(\mathcal{C}_k|\mathbf{x}_i) = \sum_{i=1}^{n} \frac{p(\mathcal{C}_k|\mathbf{x}_i)}{n} \\ &\cong \frac{\sum_{i=1}^{n} \mathbf{F}_{ik}}{n} = \frac{\mathbf{1}^T\mathbf{F}_{.k}}{n}, \quad (15) \end{aligned}$$

where the marginal probability density $p(\mathbf{x}_i) \propto \mathbf{D}_{ii} = 1$ is approximated with $1/n$. To address multi-class problems, our motivation is to let the soft labels $\mathbf{F}_{ik}$ carry the main properties of the posteriors $p(\mathcal{C}_k|\mathbf{x}_i)$. Hence, we impose two hard constraints $n\boldsymbol{\omega}^T = \mathbf{1}^T\mathbf{F}$ and $\mathbf{F}\mathbf{1}_c = \mathbf{1}$ (due to $\sum_k p(\mathcal{C}_k|\mathbf{x}_i) = 1$, $\mathbf{1}_c$ is a c-dimensional 1-entry vector). Taking advantage of the cost function in eq. (14), a constrained multi-class label propagation is established as follows

$$\begin{aligned} \min\nolimits_{\mathbf{F}} \quad & tr(\mathbf{F}^T\mathbf{L}\mathbf{F}) \\ s.t. \quad & \mathbf{F}_l = \mathbf{Y}_l,\ \mathbf{F}\mathbf{1}_c = \mathbf{1},\ \mathbf{F}^T\mathbf{1} = n\boldsymbol{\omega} \quad (16) \end{aligned}$$

which reduces to

$$\begin{aligned} \min \quad & \mathcal{Q}(\mathbf{F}_u) = tr(\mathbf{F}_u^T\mathbf{L}_{uu}\mathbf{F}_u) + 2tr(\mathbf{F}_u^T\mathbf{L}_{ul}\mathbf{Y}_l) \\ s.t. \quad & \mathbf{F}_u\mathbf{1}_c = \mathbf{1}_u,\ \mathbf{F}_u^T\mathbf{1}_u = n\boldsymbol{\omega} - \mathbf{Y}_l^T\mathbf{1}_l \quad (17) \end{aligned}$$

where $\mathbf{L}_{uu}$ and $\mathbf{L}_{ul}$ are sub-matrices of $\mathbf{L} = \begin{bmatrix} \mathbf{L}^{ll} & \mathbf{L}^{lu} \\ \mathbf{L}^{ul} & \mathbf{L}^{uu} \end{bmatrix}$, and $\mathbf{1}_l$ and $\mathbf{1}_u$ are l- and u-dimensional 1-entry vectors, respectively.

Note that the soft labels stored in $\mathbf{F}_u$ generated by eq. (17) behave like pseudo-probabilities since we don't impose $\mathbf{F}_u \geq 0$. We drop the non-negative constraint since we can get a closed-form solution without it. And a negative soft label is not bad: it makes the competing positive label more dominant. We state the following theorem.

Theorem 2. *The solution of the problem in eq. (17) equals the solution of the simpler problem:*

$$\begin{aligned} \min \quad & \mathcal{Q}(\mathbf{F}_u) = tr(\mathbf{F}_u^T\mathbf{L}_{uu}\mathbf{F}_u) + 2tr(\mathbf{F}_u^T\mathbf{L}_{ul}\mathbf{Y}_l) \\ s.t. \quad & \mathbf{F}_u^T\mathbf{1}_u = n\boldsymbol{\omega} - \mathbf{Y}_l^T\mathbf{1}_l \quad (18) \end{aligned}$$

Proof. The Lagrangian corresponding to eq. (18) is

$$\begin{aligned} \mathcal{Q}(\mathbf{F}_u, \boldsymbol{\lambda}) = & tr(\mathbf{F}_u^T\mathbf{L}_{uu}\mathbf{F}_u) + 2tr(\mathbf{F}_u^T\mathbf{L}_{ul}\mathbf{Y}_l) \\ & -\boldsymbol{\lambda}^T(\mathbf{F}_u^T\mathbf{1}_u - n\boldsymbol{\omega} + \mathbf{Y}_l^T\mathbf{1}_l). \end{aligned}$$

Let $\partial\mathcal{Q}/\partial\mathbf{F}_u = 0$ and $\mathbf{F}_u^0 = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul}\mathbf{Y}_l$, we get

$$\mathbf{F}_u = \mathbf{F}_u^0 + \frac{1}{2}\mathbf{L}_{uu}^{-1}\mathbf{1}_u\boldsymbol{\lambda}^T,$$

where $\boldsymbol{\lambda}^T$ is solved via making $\mathbf{F}_u^T\mathbf{1}_u = n\boldsymbol{\omega} - \mathbf{Y}_l^T\mathbf{1}_l$

$$\boldsymbol{\lambda}^T = \frac{2}{\mathbf{1}_u^T\mathbf{L}_{uu}^{-1}\mathbf{1}_u}\left(n\boldsymbol{\omega}^T - \mathbf{1}_l^T\mathbf{Y}_l - \mathbf{1}_u^T\mathbf{F}_u^0\right).$$

We know $\mathbf{L}\mathbf{1} = 0$ which leads to $\mathbf{L}_{ul}\mathbf{1}_l + \mathbf{L}_{uu}\mathbf{1}_u = 0$. Thus we can derive

$$\mathbf{F}_u^0\mathbf{1}_c = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul}\mathbf{Y}_l\mathbf{1}_c = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul}\mathbf{1}_l = \mathbf{L}_{uu}^{-1}\mathbf{L}_{uu}\mathbf{1}_u = \mathbf{1}_u,$$

and afterwards have

$$\begin{aligned} \boldsymbol{\lambda}^T\mathbf{1}_c &= \frac{2}{\mathbf{1}_u^T\mathbf{L}_{uu}^{-1}\mathbf{1}_u}\left(n\boldsymbol{\omega}^T\mathbf{1}_c - \mathbf{1}_l^T\mathbf{1}_l - \mathbf{1}_u^T\mathbf{1}_u\right) \\ &= \frac{2}{\mathbf{1}_u^T\mathbf{L}_{uu}^{-1}\mathbf{1}_u}(n - l - u) = 0. \end{aligned}$$

Immediately, we find

$$\mathbf{F}_u\mathbf{1}_c = \mathbf{F}_u^0\mathbf{1}_c + \frac{1}{2}\mathbf{L}_{uu}^{-1}\mathbf{1}_u\boldsymbol{\lambda}^T\mathbf{1}_c = \mathbf{1}_u, \quad (19)$$

which means the hard constraint $\mathbf{F}_u\mathbf{1}_c = \mathbf{1}_u$ is naturally satisfied when the other hard constraint is explicitly satisfied. We complete the proof. □

Table 2. Algorithm 2.

```
Algorithm 2. Robust Multi-Class Graph Transduction
Step 1. Construct a $k$-NN sGraph $G(V, E, \mathbf{W}^0)$ upon
        a set of points $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^d$
        using eq. (1)(2).
Step 2. Run Algorithm 1 with the differently labeled information $\mathbf{T}$ in eq. (7) and $\mathbf{W}^0$, and obtain the
        nonparametric adjacency matrix $\mathbf{W}$.
Step 3. Compute graph Laplacian $\mathbf{L}$ from $\mathbf{W}$. Use $\mathbf{L}$,
        the known class assignment $\mathbf{Y}_l$, and the class
        prior $\boldsymbol{\omega}$ to infer the class assignment $\mathbf{F}_u$ on the
        unlabeled data $\{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$ with eq. (20).
        Predict their labels $y \in \{1, \cdots, c\}$ by
        $y(\mathbf{x}_i) = \arg\max_{1 \le k \le c} [\mathbf{F}_u]_{i-l,k}$ $(i = l+1, \cdots, n)$.
```

In the proof of Theorem 2, we have obtained the closed-form solution to eq. (18):

$$\mathbf{F}_u = \mathbf{F}_u^0 + \frac{\mathbf{L}_{uu}^{-1}\mathbf{1}_u}{\mathbf{1}_u^T \mathbf{L}_{uu}^{-1}\mathbf{1}_u}(n\boldsymbol{\omega}^T - \mathbf{1}_l^T \mathbf{Y}_l - \mathbf{1}_u^T \mathbf{F}_u^0), \quad (20)$$

where $\mathbf{F}_u^0 = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul}\mathbf{Y}_l$ is just the multi-class version of the harmonic function proposed in [16]. As a result, the established constrained label propagation formulation well addresses multi-class problems through engaging the class prior probabilities $\boldsymbol{\omega}$ in hard constraints, thus controlling the interactions among c label prediction functions $\mathbf{F}_{.k}$.

We summarize the whole algorithmic framework for robust multi-class transductive learning in Table 2. We call Algorithm 2 *Robust Multi-Class Graph Transduction* (RMGT). By default, the class priors are assumed to be uniform, i.e. $p(\mathcal{C}_k) = 1/c$. Actually, because of scarcity of the labeled data, uniform class proportions have shown better performance on classification tasks.

The computational complexity of constructing an sGraph (step 1 in RMGT) is $\mathcal{O}(kn^2)$. Learning the nonparametric graph adjacency matrix (step 2 in RMGT) needs about $\mathcal{O}(n^2)$ time cost, and the multi-class label prediction (step 3 in RMGT) spends about $\mathcal{O}(n^3)$.

4. Experiments

In this section, we evaluate the proposed novel algorithm robust multi-class graph transduction (RMGT) integrating k-NN sGraph (i.e. symmetry-favored graph) and nonparametric adjacency matrix learning on two toy problems and two real-world datasets. We compare RMGT with the state-of-the-art graph-based semi-supervised learning approaches: Local and Global Consistency (LGC) [14], Quadratic Criterion (QC) [2], Gaussian Fields and Harmonic Functions (GFHF) plus the post-processing Class Mass Normalization (CMN) [16], Laplacian Regularized Least Squares (LapRLS) [1], and Superposable Graph Transduction (SGT) [11], all of which can be directly applied to multi-class problems.

For fair comparisons, we used eq. (2) (with an empirical choice of σ suggested in section 2.1) to build the same k-NN graph and k-NN sGraph for all algorithms within one trial. The width of the RBF kernel for LapRLS was set by cross validation. Specially, SGT will degenerate to LGC under the situations of one labeled sample per class, and CMN usually improves the performance of GFHF. The regularization parameters associated with LGC, QC, LapRLS and SGT were tuned by cross validation. As an advantage, our algorithm RMGT doesn't introduce any regularization parameters and only introduces k. We carried out two versions of RMGT: *without* and *with* nonparametric adjacency matrix learning. We denote them as RMGT and RMGT(W), respectively.

4.1. Toy Problems

We conducted experiments on two synthetic datasets depicted in Fig. 2. The first one, Noisy Face Contour, which was used to evaluate spectral clustering [7], comprises of 266 points belonging to three classes and 61 uniformly distributed noise. The second one, Noisy Two Half-Cylinders, is more challenging, composed of 600 points from two classes plus 400 noisy points. We do not care about the labels of the noisy points, so we computed classification error rates only on non-noisy points whose labels are known in advance.

The existing graph-based methods exhibit worse results as the noisy points essentially destroy the graph structure so that labels are unnecessarily propagated along them. From Fig. 2, we can see all of LGC, QC and GFHF give mistakes but our method RMGT gives a correct classification when only one point in each class is labeled. We further show average error rates over 100 random trials in Table 3. Clearly, RMGT(W) demonstrates a substantial advantage over all the other algorithms. In particular, it accomplishes fully correct results on Noisy Face Contour when the k-NN sGraph is employed. What's more, all algorithms achieve a performance gain when switching k-NN graph to k-NN sGraph. Therefore, we claim that both the proposed graph construction scheme and the adjacency matrix learning algorithm are quite robust to noise. Even without adjacency matrix learning, the presented multi-class label propagation controlled by uniform class priors (i.e. RMGT) still outperforms all the other algorithms.

4.2. Handwritten Digit Recognition

We evaluate these graph-based semi-supervised learning algorithms on the USPS (test) handwritten digits dataset in which each example is a 16 × 16 image and there are ten types of digits 0, 1, 2, ..., 9 used as 10 classes. There are 160 examples for each class at least, summing up to a total

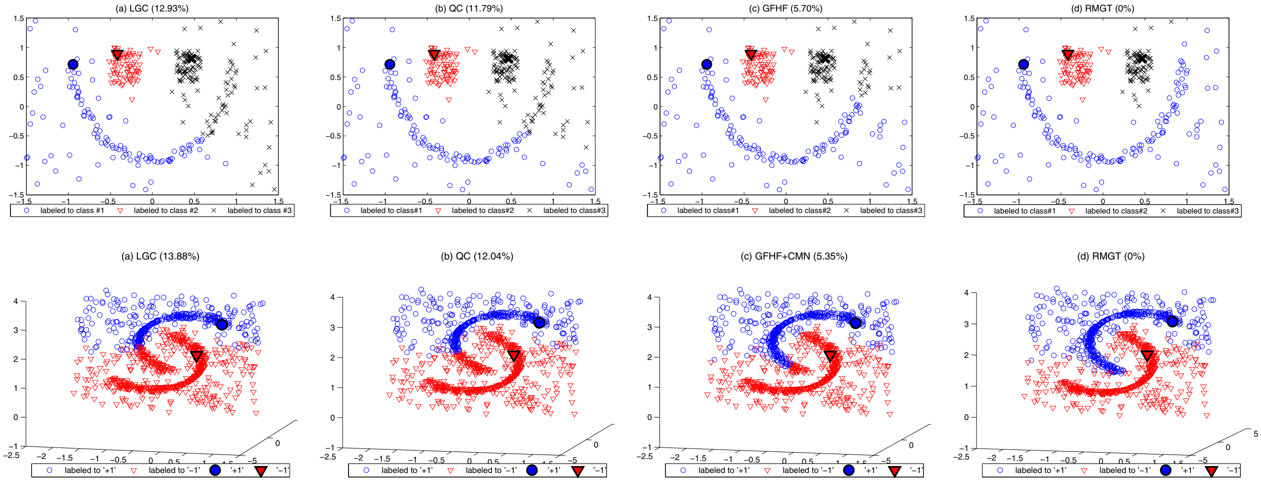

Figure 2. Semi-supervised classification results on two toy problems. The up line is Noisy Face Contour (3 classes plus noise) and the lower line is Noisy Two Half-Cylinders (2 classes plus noise). The percentages shown on top in each subfigures are the classification error rates of evaluated algorithms over non-noise data points.

Table 3. Average classification error rates on toy problems. There is only one labeled point per class.

Error Rate (%)	**Face Contour**		**Two Half-Cylinders**	
	10-NN Graph	10-NN sGraph	10-NN Graph	10-NN sGraph
LGC	8.55±3.80	7.27±4.24	13.87±10.44	11.71±9.79
QC	9.61±3.29	6.97±3.91	13.66±10.50	9.54±9.02
GFHF	8.61±5.15	5.11±4.48	15.26±8.79	11.63±8.54
GFHF+CMN	7.94±2.67	5.62±3.66	9.46±9.31	4.92±6.10
LapRLS	8.79±5.02	5.38±4.54	10.53±9.09	5.91±6.34
RMGT	**4.14±2.61**	**2.85±2.27**	**8.15±9.62**	**3.97±6.01**
RMGT(W)	**0.10±0.47**	**0±0**	**6.15±7.59**	**1.31±2.38**

Table 4. Average classification error rates on USPS.

Error Rate (%)	**20 Labeled Samples**		**100 Labeled Samples**	
	20-NN Graph	20-NN sGraph	20-NN Graph	20-NN sGraph
LGC	34.39±5.42	33.10±5.40	18.70±1.33	17.23±1.29
SGT	31.53±5.70	30.37±5.48	17.14±1.30	15.79±1.33
QC	36.25±5.33	34.24±5.18	20.45±1.27	18.77±1.22
GFHF	61.49±8.46	57.28±7.81	25.58±3.12	22.04±2.64
GFHF+CMN	33.91±5.27	32.31±5.41	18.96±1.88	17.07±1.89
LapRLS	37.45±5.20	37.22±5.28	18.84±1.85	17.68±1.86
RMGT	**27.99±4.49**	**25.76±5.24**	**15.98±1.00**	**14.48±0.85**
RMGT(W)	**25.94±4.82**	**22.52±5.08**	**14.05±0.91**	**12.20±0.86**

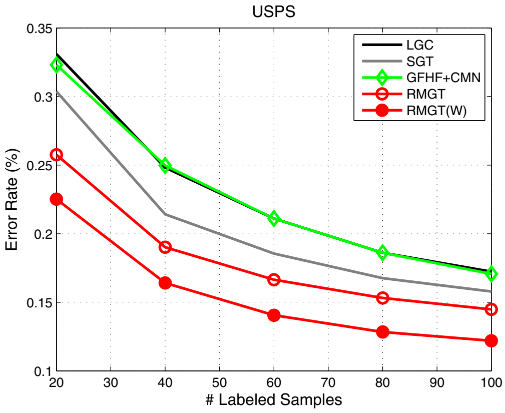

Figure 3. Average error rates vs. numbers of labeled samples.

of 2007. Fig. 4 shows 10 examples. We randomly chose labeled samples such that they contain at least one labeled sample for each class. Averaged over 20 trials, we calculated the error rates for all referred algorithms with the number of labeled samples increasing from 20 to 100. The results are displayed in Fig. 3 and Table 4. Again, we observe that RMGT(W) is significantly superior to the other methods, which demonstrates that the deployed graph construction and the more sophisticated graph learning technique improve graph-based semi-supervised classification performance very much.

4.3. Face Recognition

Now we draw our attention to the intensively studied topic, face recognition. Experiments were performed on a subset of 3160 facial images selected from 316 persons in `FRGC` version 2 [9]. We aligned all these faces according to the positions of eyes and mouth and cropped them to the fixed size of 64×72 pixels. We adopted grayscale values of images as facial features. Fig. 4 shows 10 face examples.

We randomly chose 316 $\sim$ 1000 images in this dataset as the labeled data, and kept the remaining samples as the unlabeled data. The chosen labeled samples always cover the total 316 classes (persons). By repeating the recognition process 20 times, we plot average recognition rates of five algorithms using the same 6-NN sGraph according to

Figure 4. Examples: 10 face images in the top line from one person and 10 digit images in the second line.

Table 5. Average recognition accuracy on FRGC.

Recognition Rate (%)	**316 Labeled Samples**		**1000 Labeled Samples**	
	6-NN Graph	6-NN sGraph	6-NN Graph	6-NN sGraph
LGC	61.66±1.08	63.28±1.06	70.25±1.17	73.95±1.24
SGT	61.66±1.08	63.28±1.06	71.33±1.02	74.53±0.99
QC	59.91±1.10	60.75±1.09	73.52±1.07	75.43±1.00
GFHF	59.48±1.09	61.35±1.03	72.81±1.08	76.01±1.07
GFHF+CMN	59.86±1.06	61.55±1.04	73.12±1.11	76.21±1.06
LapRLS	61.02±0.90	63.42±0.88	75.79±1.16	76.95±1.21
RMGT	**62.26±1.09**	**64.11±1.11**	**76.09±1.24**	**78.42±1.16**
RMGT(W)	**64.37±1.20**	**66.25±1.00**	**78.67±1.20**	**80.86±1.17**

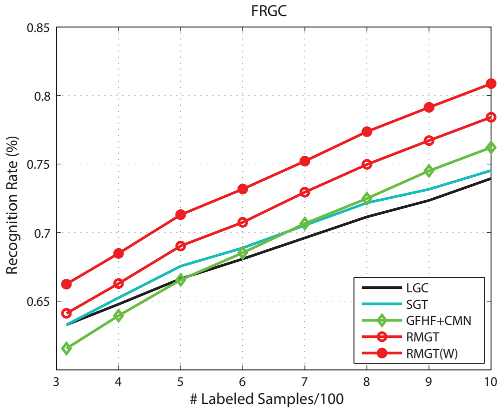

Figure 5. Average recognition rates vs. numbers of labeled samples.

the expanding labeled data size in Fig. 5. Given 316 and 1000 labeled samples, the average recognition rates of all compared algorithms are reported in Table 5, respectively. The results in Fig. 5 and Table 5 further confirm the effectiveness of the proposed RMGT algorithm, outperforming all the other compared algorithms.

We show that for this high-dimensional multi-class classification task, both of the proposed nonparametric adjacency matrix learning and constrained multi-class label propagation are substantially robust, whereas the compared methods either suffer from non-trivial high-dimensional noise or are vulnerable to the multi-class setting.

5. Conclusions

We have shown that the quality of graphs significantly affects the performance of graph-based semi-supervised learning. To this end, we proposed a novel graph-driven transductive learning framework which constructs a symmetry-favored k-NN graph and automatically learns its adjacency matrix. To address multi-class classification, we used the learned graph to enforce constrained multi-class label propagation by incorporating class priors judiciously. Integration of graph learning and multi-class label propagation makes the proposed framework extremely attractive for semi-supervised problems. We have shown through a series of experiments on synthetic and real-world datasets that our approach is substantially robust and effective.

Acknowledgements

Wei Liu wants to thank Zhouyu Fu, Junfeng He and Fuxin Li for fruitful discussions. The toy datasets are kindly provided by Zhenguo Li and Deli Zhao.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from examples. *JMLR*, 7:2399–2434, December 2006.

[2] Y. Bengio, O. Dellalleau, and N. L. Roux. Label propagation and quadratic criterion. *In O. Chapelle, B. Schlkopf and A. Zien (Eds.), Semi-supervised learning, MIT Press*, 2006.

[3] O. Chapelle, V. Sindhwani, and S. Keerthi. Branch and bound for semi-supervised support vector machines. *NIPS 19*, 2006.

[4] M. Hein and M. Maier. Manifold denoising. *NIPS 19*, 2006.

[5] R. A. Horn and C. A. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[6] T. Joachims. Transductive inference for text classification using support vector machines. *In Proc. ICML*, 1999.

[7] Z. Li, J. Liu, S. Chen, and X. Tang. Noise robust spectral clustering. *In Proc. ICCV*, 2007.

[8] J. V. Neumann. *Functional Operators Vol. II*. Princeton University Press, 1950.

[9] P. Philips, P. Flynn, T. Scruggs, and K. Bowyer. Overview of the face recognition grand challenge. *In Proc. CVPR*, 2005.

[10] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. *In Proc. ICML*, 2005.

[11] J. Wang, S.-F. Chang, X. Zhou, and S. T. C. Wong. Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. *In Proc. CVPR*, 2008.

[12] R. Zass and A. Shashua. Doubly stochastic normalization for spectral clustering. *NIPS 19*, 2006.

[13] X. Zhang and W. S. Lee. Hyperparameter learning for graph based semi-supervised learning algorithms. *NIPS 19*, 2006.

[14] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. *NIPS 16*, 2003.

[15] X. Zhu. *Semi-supervied Learning Literature Survey*. Computer Sciences Technical Report 1530, University of Wisconsin-Madison, 2005.

[16] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *In Proc. ICML*, 2003.