LAPLACIAN AFFINITY PROPAGATION FOR SEMI-SUPERVISED OBJECT CLASSIFICATION

Yun Fu [1,2*], *Zhu Li* [3], *Xi Zhou* [1,2], *and Thomas S. Huang* [1,2]

[1]Beckman Institute, University of Illinois at Urbana-Champaign (UIUC), Urbana, IL 61801, USA
[2] Dept. of ECE, University of Illinois at Urbana-Champaign (UIUC), Urbana, IL 61801, USA
[3] Multimedia Research Lab (MRL), Motorola Labs, Schaumburg, IL 60196, USA

ABSTRACT

We solve the semi-supervised multi-class object classification problem by a graph-based learning algorithm, called Laplacian Affinity Propagation (LAP). The idea is to model and train both labeled and unlabeled data by constructing a local neighborhood affinity graph in a smoothness formulation of Laplacian matrix, based on graph mincuts or harmonic energy minimization. The unknown labels for unlabeled data are inferred from an optimized graph embedding procedure subject to the labeled data. Such label-to-unlabel propagation scheme can provide a closed form solution via a learning framework that is flexible for any new design. LAP integrates embedding and classifier together and gives smooth labels with respect to the underlying manifold structure formed by the training data. Object classification experiments on COIL database demonstrate the effectiveness and applicability of such algorithm.

Index Terms— Semi-supervised learning, local affinity, graph embedding, spectral clustering, object classification.

1. INTRODUCTION

Labelling large size of training database is often tedious and even intractable in many learning-based problems, such as object classification and high-dimensional data clustering. In addition, some particular databases are lack of sufficient labelling for some reasons in practice. Can partially labeled training data still give comparable results to the fully labeled ones? Semi-supervised learning [18, 15] algorithms are such kind of techniques that learn from both labeled and unlabeled samples [17, 13, 16] for the purpose of pattern classification or data mining. The state-of-the-art algorithms of transductive learning [19] or inductive learning work by formulating the general assumption that nearby points and points belonging to the same manifold structure should have similar labels [12, 5]. This assumption is quite consistent with that of the recently developed manifold learning techniques. Those algorithms, such as Locally Linear Embedding (LLE) [2] and Laplacian Eigenmaps (LE)[3], model the local neighborhood structure with an affinity graph and find the low-dimensional manifold space through global nonlinear embedding. It is straightforward to extend manifold learning methods to a semi-supervised dimensionality reduction framework.

Many related works have been presented recently for clustering, information retrieval, and pattern recognition. Comprehensive reviews and discussions are available in both [13] and [4]. Graph Mincuts (GM) [16] and Harmonic Energy Minimization (HEM) [14] achieve binary classification by minimizing a harmonic energy function. Locality Preserving Indexing (LPI) [11] extends the LE framework to a linear embedding way of document clustering. Locally Embedded Analysis (LEA) [1] extends the LLE objective to the linear submanifold learning and dimensionality reduction. Linear Neighborhood Propagation (LNP) [5] learns the neighborhood linear reconstruction weights for modeling and predicts the labels of unlabeled data from the labeled data.

In view of the foregoing work, we extend the GM and HEM framework to a more general multiple object classification scenario and present a semi-supervised algorithm, called Laplacian Affinity Propagation (LAP). The idea is to model both labeled and unlabeled data by constructing a locality affinity graph in a general smoothness formulation of Laplacian matrix. The unknown labels for unlabeled data are predicted from an optimized graph embedding procedure subject to the labeled data. Such label-to-unlabel propagation can provide a closed form solution based on a flexible learning framework. LAP integrates both embedding and classifier and gives smooth labels with respect to the underlying manifold structure formed by the labeled and unlabeled data. We test the proposed algorithm on benchmark data sets for the applications of data clustering and object classification.

It is worthwhile to highlight several key advantages of LAP framework. First, LAP is for multi-class problem and has closed form solution. Secondly, the framework can be conveniently extended to out-of-sample cases. Finally, LAP considers the semi-supervised case for graph embedded manifold learning which is ignored in conventional methods.

E-mail: yunfu2@ifp.uiuc.edu. This research was funded in Part by the U.S. Government VACE program, and in part by the NSF Grant CCF 04-26627. The views and conclusions are those of the authors, not of the US Government or its Agencies.

2. LAPLACIAN AFFINITY PROPAGATION

2.1. Algorithm Objective

Consider a given data set $\mathcal{X} = \{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^n$, our goal is to find an associated label set $\mathcal{Y} = \{y_i : y_i \in \mathbb{N}\}_{i=1}^n$ to describe the underlying affinity of $\mathcal{X}$ as much as possible. Suppose a small part of the data with size m ($m < n$) has already been labeled as $\mathcal{Y}_l = \{y_i : y_i = l_i\}_{i=1}^m$. The problem is re-formulated to find the unknown label set $\mathcal{Y}_u = \{y_i : y_i \in \mathbb{N}\}_{i=m+1}^n$ subject to known label $\mathcal{Y}_l$ and data $\mathcal{X}$. Here we have $\mathcal{Y} = \mathcal{Y}_l \cup \mathcal{Y}_u$ and $\mathcal{X} = \mathcal{X}_l \cup \mathcal{X}_u$ for $\mathcal{X}_l = \{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^m$ and $\mathcal{X}_u = \{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^D\}_{i=m+1}^n$.

In matrix and column vector forms, we rewrite the above representations as $\mathbf{X} = [\mathbf{X}_l\ \mathbf{X}_u]$ and $\mathbf{y} = [\mathbf{y}_l;\ \mathbf{y}_u]$, where $\mathbf{X}_l = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_m]$, $\mathbf{X}_u = [\mathbf{x}_{m+1}\ \mathbf{x}_{m+2}\ \cdots\ \mathbf{x}_n]$, $\mathbf{y}_l = [y_1\ y_2\ \cdots\ y_m]^T$ and $\mathbf{y}_u = [y_{m+1}\ y_{m+2}\ \cdots\ y_n]^T$. Define $n \times n$ affinity matrix $\mathbf{A} = \{a_{ij}\}_{i,j=1}^n = f(\mathcal{X})$, as a function of $\mathcal{X}$, denoting the affinity weights for each pair of data points, diagonal matrix $\mathbf{D}(i,i) = \sum_j a_{ji}$, and Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$. To solve this problem, we define the *Laplacian Affinity Propagation* (LAP) algorithm conforming to the following objective function [14, 3, 4, 5, 15],

$$\mathbf{y}_u = \arg\min_{\mathbf{y}_l} \mathbf{y}^T g(\mathbf{L})\mathbf{y}. \tag{1}$$

where $g(\cdot)$ is a smoothness function of matrix $\mathbf{L}$. e.g. $g(\mathbf{L}) = \mathbf{L}$ [14], $g(\mathbf{L}) = \mathbf{L}^2$, or $g(\mathbf{L}) = \exp(-\mathbf{L})$ [4]. The local affinity of the original data space is propagated via Laplacian affinity matrix to the 1-D label space after the embedding. That is why we call this algorithm "Laplacian affinity propagation".

The basic idea of LAP is to generalize the common formulation for GM, HEM, LNP, and LE, since they all share a similar latent objective function in Equation 1; and apply semi-supervised learning technique to object classification scenario. Particularly, LAP has several good properties: (1) It aims at semi-supervised learning for multi-class problems; (2) The constraint on solving the objective function is straightforwardly the known labels instead of an empirical constraint that defined for conveniently solving mathematical formulations; (3) LAP is a hybrid framework integrating both embedding and classifier. The embedded space of LAP consists of 1-D discriminant labels for classifying the original data; (4) Out-of-sample problem is well-defined in LAP.

2.2. LAP Solution

Suppose the matrix $g(\mathbf{L})$ is partitioned into following 4 matrix blocks [14] with size of $m \times m$ for $\mathbf{L}_l$, $(n-m) \times (n-m)$ for $\mathbf{L}_u$, $m \times (n-m)$ for $\mathbf{L}_{lu}$, and $(n-m) \times m$ for $\mathbf{L}_{ul}$,

$$g(\mathbf{L}) = \begin{bmatrix} \mathbf{L}_l & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_u \end{bmatrix}. \tag{2}$$

The embedding of $\mathbf{y}_u$ is derived from a closed form solution,

$$\mathbf{y}_u = -\mathbf{L}_u^{-1}\mathbf{L}_{ul}\mathbf{y}_l. \tag{3}$$

To overcome the intractable case when $\mathbf{L}_u$ is singular or nearly singular, a regularization parameter r is added so that $\widetilde{\mathbf{L}}_u = \mathbf{L}_u + r \cdot \mathrm{trace}(\mathbf{L}_u) \cdot \mathbf{I}$ is invertible. Here $\mathbf{I}$ is the $(n-m) \times (n-m)$ identity matrix.

2.3. Out-of-Sample Extension

For practical applications, we consider the case of large size training data in the identical distribution. To label an out-of-sample, e.g. $\mathbf{x}_t$, we add $\mathbf{x}_t$ to the given data set $\mathcal{X}$ and have a new data matrix $\widetilde{\mathbf{X}} = [\mathbf{X}_l\ \mathbf{X}_u\ \mathbf{x}_t]$. Form the $(n+1) \times (n+1)$ matrices $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{D}}$ and $\widetilde{\mathbf{L}}$. Partition $\mathbf{L}$ into blocks, we have

$$g(\widetilde{\mathbf{L}}) = \begin{bmatrix} \widetilde{\mathbf{L}}_l & \widetilde{\mathbf{L}}_{lu} & \mathbf{L}_{lt} \\ \widetilde{\mathbf{L}}_{ul} & \widetilde{\mathbf{L}}_u & \mathbf{L}_{ut} \\ \mathbf{L}_{tl} & \mathbf{L}_{tu} & \mathbf{L}_t \end{bmatrix}, \tag{4}$$

where $\widetilde{\mathbf{L}}_l$, $\widetilde{\mathbf{L}}_u$, $\widetilde{\mathbf{L}}_{lu}$, and $\widetilde{\mathbf{L}}_{ul}$ are different matrices from $\mathbf{L}_l$, $\mathbf{L}_u$, $\mathbf{L}_{lu}$, and $\mathbf{L}_{ul}$ because of the influence of $\mathbf{x}_t$ on $\mathbf{A}$. The sizes of $\mathbf{L}_{lt}$, $\mathbf{L}_{ut}$, $\mathbf{L}_{tl}$, $\mathbf{L}_{tu}$, and $\mathbf{L}_t$ are $m \times 1$, $(n-m) \times 1$, $1 \times m$, $1 \times (n-m)$, and 1×1 respectively. According to the solution in Equation 3, we have

$$\mathbf{y}_t = -\frac{[\mathbf{L}_{tl}\ \mathbf{L}_{tu}]\,\mathbf{y}}{\mathbf{L}_t}. \tag{5}$$

We should be aware of the dependence of $\mathbf{L}_{tl}$ and $\mathbf{L}_{tu}$ on the change of $\mathbf{L}_l$, $\mathbf{L}_u$, $\mathbf{L}_{lu}$, and $\mathbf{L}_{ul}$ after new data added.

2.4. Affinity Matrix

It is flexible to design new embedding metric via LAP framework. The essential trick to form the affinity matrix is to measure and encode the connectivity among the original data points $\{\mathbf{x}_i\}_{i=1}^n$. This connectivity can be defined in two ways:

(1). *k-nearest neighbors*: The case when $\mathbf{x}_i$ is one of the k-NNs of $\mathbf{x}_j$ and vice versa.

(2). *ϵ-nearest neighbors* : The case when $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon$.

After building the connection graph, three straightforward ways for setting the weights are listed as follows[11, 13, 2]:

(1). *Gaussian kernel*: Set $\mathbf{A}[i,j] = a_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/t)$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected, otherwise $\mathbf{A}[i,j] = a_{ij} = 0$

(2). *0-1 weights* : Set $\mathbf{A}[i,j] = a_{ij} = 1$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected, otherwise $\mathbf{A}[i,j] = a_{ij} = 0$.

(3). *LLE weights* : Set $\mathbf{A}[i,:] = \frac{\mathbf{G}_i^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{G}_i^{-1}\mathbf{1}}$, when $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected, where Gram matrix $\mathbf{G}_i$ is defined in [2, 1], otherwise $\mathbf{A}[i,j] = a_{ij} = 0$.

From above definitions, we have two free parameters, k (or ϵ) and t, that need to tune in practice.

3. THEORETICAL JUSTIFICATION

To solve the Equation 1, we form the Lagrangian function

$$\mathcal{G}(\mathbf{y}, \lambda) = [\mathbf{y}_l^T\ \mathbf{y}_u^T] \begin{bmatrix} \mathbf{L}_l & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_u \end{bmatrix} \begin{bmatrix} \mathbf{y}_l \\ \mathbf{y}_u \end{bmatrix} + \lambda^T(\mathbf{y}_l - \ell), \tag{6}$$

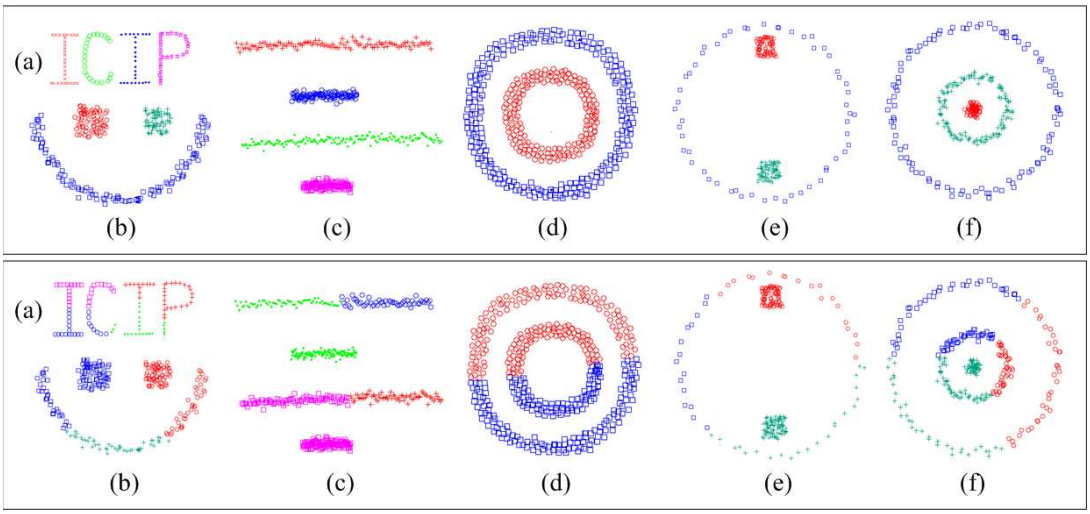

Fig. 1. Clustering results of LAP in the first row and K-means in the second row for 7 cases. (a) 4 clusters; (b) 3 clusters; (c) 4 clusters; (d) 2 clusters; (e) 3 clusters; (f) 3 clusters. The data of (b)(c)(e)(f) are from [7].

where λ is the Lagrange multiplier vector and ℓ (= $\mathbf{y}_l$) is the label vector for $\mathcal{Y}_l$. We convert this constrained optimization problem into an unconstrained one by taking the derivative $\partial\mathcal{G}(\mathbf{y},\lambda)/\partial\mathbf{y}_l = 0$. After some steps, it follows

$$\lambda^T = -2\mathbf{y}_l^T\mathbf{L}_l - \mathbf{y}_u^T\mathbf{L}_{ul} - \mathbf{y}_u^T\mathbf{L}_{lu}^T. \tag{7}$$

Substitute λ^T in Equation 6 by Equation 7 and take another derivative $\partial\mathcal{G}(\mathbf{y},\lambda)/\partial\mathbf{y}_u = 0$. We have

$$2\mathbf{L}_u\mathbf{y}_u + \mathbf{L}_{ul}\ell + \mathbf{L}_{lu}^T\ell = 0. \tag{8}$$

Since $g(\mathbf{L})$ as well as $\mathbf{A}$ is usually symmetric, we notice the fact that $\mathbf{L}_{lu} = \mathbf{L}_{ul}^T$ and $\mathbf{L}_{ul} = \mathbf{L}_{lu}^T$. After some algebraic steps, we finally get the Equation 3.

4. EXPERIMENTS

4.1. Clustering

For data clustering, we apply LAP and K-means to 7 clustering problems shown in Figure 1. The Gaussian kernel is used to build the connection graph. The data of (b)(c)(e)(f) is from [7] and online available[1]. As for semi-supervised clustering, we assume the number of clusters is known. The data points are represented by their coordinates in the 2-D space. The results of different clusters found are shown via different symbols and colors. We plot the clustering results of LAP in the first row and K-means in the second row in Figure 1. It appears that LAP performs accurate and robust, even for clusters that do not form convex regions. LAP reliably finds clusterings consistent with subjective observations. Unfortunately, K-means easily fails in all the 7 cases though we repeat the clustering for 50 times in each run. We know that other methods, such as [8] and [7], can also give good results for these problems, but LAP is applicable to more scenarios of machine learning, such as object classification, since it can analyze high-dimensional data in a manifold learning manner.

[1] http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html

Fig. 2. Samples of COIL-20 object image database.

4.2. Object Classification

The COIL-20 database [9] is used in our experiments for object classification. This database consists of 20 different objects. Each object has 72 gray-scale images in total. The images of the objects were taken at pose intervals of 5 degrees around each objects. All the 1,440 images were normalized to 128×128. We further resize the images to 32×32 and rearrange them to vectors in column style with 1,024 dimension for each. Figure 2 shows the sample images of the 20 objects.

We select 4 and 6 images of each object to form two training sets. The sequence numbers of selected images in the database are [1, 18, 36, 54] and [1, 12, 24, 36, 48, 60] respectively. For comparison, we also test the object classification through k-Nearest Neighbor (K-NN), Principal Component Analysis (PCA) [10] plus K-NN, Linear Discriminant

Table 1. Algorithm comparison on COIL-20 with 4 images of each object for training and 68 images of each for testing.

Method	**Parameter** (k, σ^2, t)	**Dim.** $\sharp$ (T., B.)	**Error Rate** (%) ($\sharp$ / 1360)
K-NN	$k_{\mathrm{NN}} = 1$	1024	20.22 (275/1360)
PCA+K-NN	$k_{\mathrm{NN}} = 1$	20 (T)	20.51 (279/1360)
LDA+K-NN	$k_{\mathrm{NN}} = 1$	19 (T)	16.76 (228/1360)
GKPCA [6]	$\sigma^2 =?$	20 (T)	18.31 (249/1360)
SFA [6]	$\sigma^2 =?$	20 (T)	18.31 (249/1360)
LEA+NN	$k = 3$	533 (B)	18.31 (249/1360)
LAP	$k = 2, t = 0.04$	1	**15.15 (206/1360)**

Table 2. Algorithm comparison on COIL-20 with 6 images of each object for training and 66 images of each for testing.

Method	**Parameter** (k, σ^2, t)	**Dim.** $\sharp$ (T., B.)	**Error Rate** (%) ($\sharp$ / 1320)
K-NN	$k_{\mathrm{NN}} = 1$	1024	11.36 (150/1320)
PCA+K-NN	$k_{\mathrm{NN}} = 1$	20 (T)	14.09 (186/1320)
LDA+K-NN	$k_{\mathrm{NN}} = 1$	19 (T)	15.23 (201/1320)
GKPCA [6]	$\sigma^2 =?$	20 (T)	10.45 (138/1320)
SFA [6]	$\sigma^2 =?$	20 (T)	10.45 (138/1320)
LEA+NN	$k = 5$	274 (B)	9.47 (125/1320)
LAP	$k = 2, t = 0.05$	1	**5.45 (72/1320)**

Analysis (LDA) [10] plus K-NN, Gaussian Kernel PCA [6] (GKPCA), Spectral Feature Analysis (SFA) [6], and Locally Embedded Analysis (LEA) [1] plus NN. We take K-NN as the baseline classifier. For simplification, the k_{NN} is set to 1. Since LAP integrates the embedding and classifier, we do not need a classifier. The Gaussian kernel is used to build the connection graph for LAP. We choose a linear smoothness function. In the testing, we label the training data with integers $1, 2, \ldots, 20$. For the estimated labels of the unlabeled data, we simply round them to nearest integers. The results for the two cases of test are summarized in Table 1 and Table 2 respectively. All the reported results of those methods are obtained by best tuning the associated parameters listed in the tables. The results of SFA and GKPCA are from [6], but the parameters are not reported. We observe that LAP consistently outperforms all the listed methods with lowest error rate of 15.15% and 5.45% in the two tests. It also gives the lowest number of reduced dimension that is only 1. Moreover, SFA, LEA and GKPCA are comparable to each other and reasonably better than baselines in this scenario.

5. CONCLUSION

We solve the multi-class semi-supervised clustering and object classification using LAP algorithm. LAP defines the semi-supervised problem in a manifold learning manner and gives the solution in closed form. The solution can also be easily extended to out-of-sample cases. This brings many interesting research directions for future work, such as interactive image segmentation, face recognition, and information retrieval. Another point we need to address is the computation issue. When the number of training data is too large, similar to any graph embedding methods, LAP will suffer from the lower computation power and out of memory space in common PCs. We are planning to explore the algorithm based on block matrix decompositions to tackle this difficulty in future.

6. REFERENCES

[1] Y. Fu and T.S. Huang, *Locally Linear Embedded Eigenspace Analysis*, www.ifp.uiuc.edu/∼yunfu2/papers/LEA-Yun05.pdf.

[2] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, pp. 2323-2326, 2000.

[3] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 6, pp. 1373-1396, 2003.

[4] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and Semisupervised Learning on Large Graphs," *Proc. of Computational Learning Theory (COLT'04)*, 2004.

[5] F. Wang, J.D. Wang, C.S. Zhang, and H.C. Shen, "Semi-Supervised Classification Using Linear Neighborhood Propagation," *IEEE Conf. on CVPR'06*, vol. 1, pp. 160-167, 2006.

[6] F. Wang, J.D. Wang, and C.S. Zhang, "Spectral Feature Analysis," *IEEE Conf. on IJCNN'05*, vol. 3, pp. 1971-1976, 2005.

[7] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," *NIPS'04*, 2004.

[8] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and An Algorithm," *NIPS'01*, 2001.

[9] S.A. Nene, S.K. Nayar, and H. Murase, *Columbia Object Image Library (COIL-20)*, TR CUCS-005-96, 1996.

[10] A.M. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. on PAMI*, vol. 23, no. 2, pp. 228-233, 2001.

[11] D. Cai, X.F. He, and J.W. Han, "Document Clustering Using Locality Preserving Indexing," *IEEE Trans. on KDE*, vol. 17, no. 12, pp. 1624-1637, 2005.

[12] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo, "On Semi-Supervised Classification," *NIPS'04*, 2004.

[13] X.J. Zhu, *Semi-Supervised Learning with Graphs*, Doctoral Thesis, CMU-LTI-05-192, 2005.

[14] X.J. Zhu, Z.B. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *ICML'03*, 2003.

[15] X.J. Zhu, J. Lafferty, and Z. Ghahramani, *Semi-Supervised Learning: From Gaussian Fields to Gaussian Processes*, CMU Technical Report CMU-CS-03-175, 2003.

[16] A. Blum and S. Chawla, "Learning from Labeled and Unlabeled Data using Graph Mincuts," *ICML'01*, 2001.

[17] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf, "Learning with Local and Global Consistency," *NIPS'04*, 2004.

[18] X.H. Zhang and W.S. Lee, "Hyperparameter Learning for Semi-supervised Learning Algorithms," *NIPS'06*, 2006.

[19] T. Joachims, "Transductive Learning via Spectral Graph Partitioning," *ICML'03*, 2003.