Graph Transduction via Alternating Minimization

Jun Wang JWANG@EE.COLUMBIA.EDU
Department of Electrical Engineering, Columbia University

Tony Jebara JEBARA@CS.COLUMBIA.EDU
Department of Computer Science, Columbia University

Shih-Fu Chang SFCHANG@EE.COLUMBIA.EDU
Department of Electrical Engineering, Columbia University

Abstract

Graph transduction methods label input data by learning a classification function that is regularized to exhibit smoothness along a graph over labeled and unlabeled samples. In practice, these algorithms are sensitive to the initial set of labels provided by the user. For instance, classification accuracy drops if the training set contains weak labels, if imbalances exist across label classes or if the labeled portion of the data is not chosen at random. This paper introduces a propagation algorithm that more reliably minimizes a cost function over both a function on the graph and a binary label matrix. The cost function generalizes prior work in graph transduction and also introduces node normalization terms for resilience to label imbalances. We demonstrate that global minimization of the function is intractable but instead provide an alternating minimization scheme that incrementally adjusts the function and the labels towards a reliable local minimum. Unlike prior methods, the resulting propagation of labels does not prematurely commit to an erroneous labeling and obtains more consistent labels. Experiments are shown for synthetic and real classification tasks including digit and text recognition. A substantial improvement in accuracy compared to state of the art semi-supervised methods is achieved. The advantage are even more dramatic when labeled instances are limited.

1. Introduction

Graph transduction refers to a family of algorithms that achieve state of the art performance in semi-supervised learning and classification. These methods tradeoff a classification function's accuracy on labeled examples against a regularizer term that encourages the function to remain smooth over a weighted graph connecting the data samples. The weighted graph and the minimized function ultimately propagate label information from labeled data to unlabeled data to provide the desired transductive predictions. Popular algorithms for graph transduction include the Gaussian fields and harmonic functions based method (*GFHF*) (Zhu et al., 2003) as well as the local and global consistency method (*LGC*) (Zhou et al., 2004). Other closely related methods include the manifold regularization framework proposed in (Sindhwani et al., 2005; Belkin et al., 2006) where graph Laplacian regularization terms are combined with regularized least squares (*RLS*) or support vector machine (*SVM*) function estimation criteria. These methods lead to graph-regularized variants denoted as Laplacian RLS (*LapRLS* and Laplacian SVM (*LapSVM*) respectively. For certain synthetic and real data problems, graph transduction approaches do achieve promising performance. However, this article identifies several realistic settings and labeling situations where this performance can be compromised. An alternative algorithm which generalizes the previous techniques is proposed by defining a joint iterative optimization over the classification function *and* a balanced label matrix.

Even if one assumes the graph structures used in the above methods faithfully describe the data manifold, graph transduction algorithms may still be misled by problems in the label information. Figure 1 depicts several cases where the label information leads to invalid graph transduction solutions for all the aforementioned algorithms. The top row of Figure 1 shows a

Appearing in *Proceedings of the 25^{th} International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

separable pair of manifolds where unbalanced label information affects the propagation results. Although a clear separation region is visible between the two manifolds, the imbalance in the labels misleads the previous algorithms which prefer assigning points to the larger class. In the bottom row of Figure 1, a non-separable problem is shown where two otherwise separable manifolds are peppered with noisy outlier samples. Here, the outliers do not belong to either class but once again interfere with the propagation of label information. In both situations, conventional transductive learning approaches such as *GFHF*, *LGC*, *LapRLS*, and *LapSVM* fail to give acceptable labeling results.

In order to handle such situations, we extend the graph transduction optimization problem by casting it as a *joint* optimization over the classification function *and* the labels. The optimization is solved iteratively and remedies the instability previous methods seem to have vis-a-vis the initial labeling. In our novel framework, initial labels simply act as the starting value of the label matrix variable which is incrementally refined until convergence. The overall minimization over the continuous classification function and the binary label matrix proceeds by an alternating minimization over each term separately and converges to a local minimum. Moreover, to handle the imbalanced labels issue, a node regularizer term is introduced to balance the label matrix among different classes. These two fundamental changes to the graph transduction problem produce significantly better performance on both artificial and real datasets.

The remainder of this paper is organized as the follows. In Section 2, we revisit the graph regularization framework of (Zhou et al., 2004) and extend it into a bivariate graph optimization problem. A corresponding algorithm is provided that solves the new optimization problem by iterative alternating minimization. Section 3 provides experimental validation for the algorithm on both toy and real classification datasets, including text classification and digital recognition. Comparisons with leading semi-supervised methods are made. Concluding remarks and a discussion are then provided in Section 4.

2. Graph Transduction

Consider the dataset $\mathcal{X} = (\mathcal{X}_l, \mathcal{X}_u)$ of labeled inputs $\mathcal{X}_l = \{\mathbf{x}_1, \cdots, \mathbf{x}_l\}$ and unlabeled inputs $\mathcal{X}_u = \{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$ along with a small portion of corresponding labels $\{y_1, \cdots, y_l\}$, where $y_i \in \mathcal{L} = \{1, \cdots, c\}$. For transductive learning, the objective is to infer the labels $\{y_{l+1}, \cdots, y_n\}$ of the unlabeled data $\{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$ where typically $l << n$. The graph transduction methods define an undirected graph represented by $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where the set of node or vertices is $\mathcal{X} = \{\mathbf{x}_i\}$ and the set of edges is $\mathcal{E} = \{e_{ij}\}$. Each sample $\mathbf{x}_i$ is treated as the node on the graph and the weight of edge e_{ij} is w_{ij}. Typically, one uses a kernel function k over pairs of points to recover weights, in other words $w_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ with the RBF kernel being a popular choice. The weights for edges are used to build a weight matrix which is denoted by $\mathbf{W} = \{w_{ij}\}$. Similarly, the node degree matrix $\mathbf{D} = diag\left([d_1, \cdots, d_n]\right)$ is defined as $d_i = \sum_{j=1}^{n} w_{ij}$. The binary label matrix $\mathbf{Y}$ is described as $Y \in \mathcal{B}^{n \times c}$ with $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ has label $y_i = j$ and $\mathbf{Y}_{ij} = 0$ otherwise. This article will often refer to row and column vectors of such matrices, for instance, ith row and jth column vectors of $\mathbf{Y}$ are denoted as $Y_{i\cdot}$ and $Y_{\cdot j}$, respectively. The graph Laplacian is defined as $\mathbf{\Delta} = \mathbf{D} - \mathbf{W}$ and the normalized graph Laplacian is $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

2.1. Consistent Label Propagation

Graph based semi-supervised learning methods propagate label information from labeled nodes to unlabeled nodes by treating all samples as nodes in a graph and using edge-based affinity functions between all pairs of nodes to estimate the weight of each edge. Most methods then define a continuous classification function $F \in \mathcal{R}^{n \times c}$ that is estimated on the graph to minimize a cost function. The cost function typically enforces a tradeoff between the smoothness of the function on the graph of both labeled and unlabeled data and the accuracy of the function at fitting the label information for the labeled nodes. Such is the case for a large variety of graph based semi-supervised learning techniques ranging from the the mincuts method (Blum & Chawla, 2001), the Gaussian fields and harmonic functions (*GFHF*) method, and the local and global consistency (*LGC*) method. A detailed survey of these methods is available in (Zhu, 2005).

In trading off smoothness for accuracy, both *GFHF* and *LGC* approaches attempt to preserve consistency on the data manifold during the optimization of the classification function. The loss function for both methods involves the additive contribution of two penalty terms the global smoothness Q_{smooth} and local fitness Q_{fit} as shown below:

$$\mathbf{F}^* = \arg\min_{\mathbf{F}} \mathcal{Q}(\mathbf{F}) = \arg\min_{\mathbf{F}} \left\{Q_{smooth}(\mathbf{F}) + Q_{fit}(\mathbf{F})\right\} \tag{1}$$

In particular, recall that *LGC* uses an elastic regularizer framework with the following cost function (Zhou

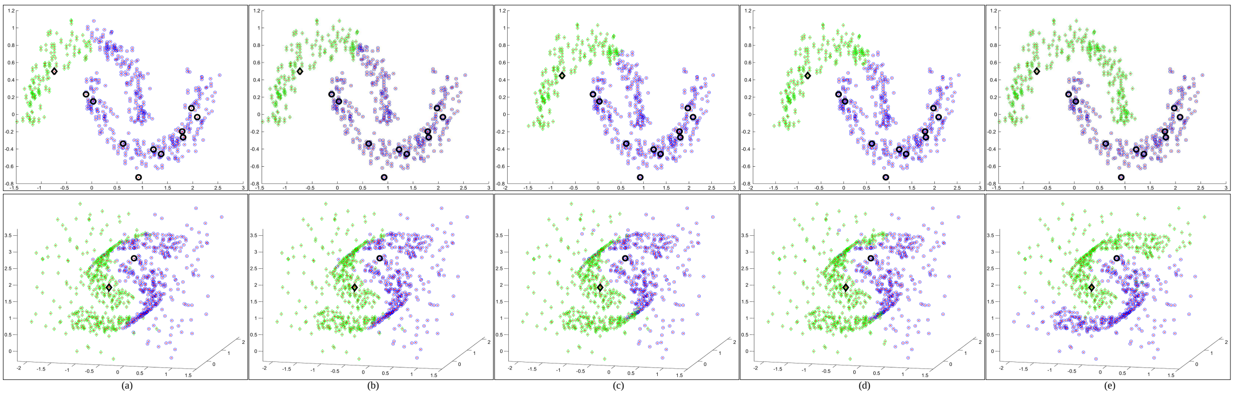

Figure 1. A demonstration with artificial data of the sensitivity graph transduction exhibits for certain initial label settings. The top row shows how imbalanced labels adversely affect even a well-separated 2D two-moon dataset. The bottom row shows a 3D two-moon data where graph transduction is again easily misled by the introduction of cloud of outliers. Large markers indicate known labels and the two-color small markers represent the predicted classification results. Columns depict the results from (a) the *GFHF* method (Zhu et al., 2003); (b) the *LGC* method (Zhou et al., 2004); (c) the *LapRLS* method (Belkin et al., 2006); (d) the *LapSVM* method (Belkin et al., 2006); and (e) Our method (*GTAM*).

et al., 2004).

$$\mathcal{Q}(\mathbf{F}) = \sum_{i,j=1}^{n} w_{ij} \left\| \frac{\mathbf{F}_{i\cdot}}{\sqrt{\mathbf{D}_{ii}}} - \frac{\mathbf{F}_{j\cdot}}{\sqrt{\mathbf{D}_{jj}}} \right\|^2 + \mu \sum_{i=1}^{n} \|\mathbf{F}_{i\cdot} - \mathbf{Y}_{i\cdot}\|^2 \tag{2}$$

where the coefficient μ balances global smoothness and local fitting penalty terms. If we set $\mu = \infty$ and use a standard graph Laplacian for the smoothness term, the above framework reduces to the harmonic function formulation as shown in (Zhu et al., 2003).

While *LGC* and *GFHF* formulations remain popular and have been empirically validated in the past, it is possible to discern some key limitations. First, the optimization can be broken up into a separate parallel problems since the cost function decomposes into terms that only depend on individual columns of the matrix $\mathbf{F}$. Since each column of $\mathbf{F}$ indexes the labeling of a single class, such a decomposition reveals that biases may arise if the input labels are disproportionately imbalanced. In practice, both propagation algorithms tend to prefer predicting the class with the majority of labels. Second, both learning algorithms are extremely dependent on the initial labels provided in $\mathbf{Y}$. This is seen in practice but can also be explained mathematically by fact that $\mathbf{Y}$ is starts off extremely sparse and has many unknown terms. Third, when the graph contains background noise and makes class manifolds non-separable, these graph transduction approaches fail to output reasonable classification results. These difficulties were illustrated in Figure 1 and seem to plague many graph transduction approaches. However, the proposed method, graph transduction via alternating minimization (*GTAM*) appears resilient.

To address these problems, we begin will make modifications to the cost function in Equation 1. The first one is to explicitly show the optimization over both the classification function $\mathbf{F}$ and the binary label matrix $\mathbf{Y}$:

$$(\mathbf{F}^*, \mathbf{Y}^*) = \arg\min\nolimits_{\mathbf{F}\in\mathcal{R}^{n\times c}, \mathbf{Y}\in\mathcal{B}^{n\times c}} \mathcal{Q}(\mathbf{F}, \mathbf{Y}). \tag{3}$$

Where $\mathcal{B}^{n\times c}$ is the set of all binary matrices $\mathbf{Y}$ of size $n \times c$ that satisfy $\sum_j \mathbf{Y}_{ij} = 1$ and, for the labeled data $\mathbf{x}_i \in \mathcal{X}_l$, $\mathbf{Y}_{ij} = 1$ if $y_i = j$. More specifically, our loss function is:

$$\mathcal{Q}(\mathbf{F}, \mathbf{Y}) = \mathrm{tr}\left\{\mathbf{F}^T\mathbf{L}\mathbf{F} + \mu(\mathbf{F} - \mathbf{V}\mathbf{Y})^T(\mathbf{F} - \mathbf{V}\mathbf{Y})\right\} \tag{4}$$

where we have introduced the matrix $\mathbf{V}$ which is a node regularizer to balance the influence of labels from different classes. The matrix $\mathbf{V} = diag(\mathbf{v})$ is a function of the current label matrix $\mathbf{Y}$:

$$\mathbf{v} = \sum_{j=1}^{c} \frac{\mathbf{Y}_{\cdot j} \odot \mathbf{D}\vec{\mathbf{1}}}{\mathbf{Y}_{\cdot j}^T \mathbf{D}\vec{\mathbf{1}}} \tag{5}$$

where the symbol $\odot$ denotes Hadamard product and column vector $\vec{\mathbf{1}}$ represents $\vec{\mathbf{1}} = [1 \cdots 1]^T$. This node regularizer permits us to work with a normalized version of the label matrix $\mathbf{Z}$ defined as: $\mathbf{Z} = \mathbf{V}\mathbf{Y}$.

By definition, we see that the normalized label matrix satisfies $\sum_i \mathbf{Z}_{ij} = 1$. Using the normalized label matrix $\mathbf{Z}$ in a graph regularization allows labeled nodes with high degree to contribute more during the graph diffusion and label propagation process. However, the total diffusion of each class is kept equal and normalized to be one. Therefore, the influence of different classes is balanced even when if the given class labels are imbalanced. If class proportion information is known a priori, it can integrated by scaling the diffusion with the prior class proportion. However, because of the nature of graph transduction and the limitations in the size of labeled data, equal class balancing leads to generally more reliable solutions than class proportional weighting. This intuition is in line with prior work that uses of class proportion information in transductive inference such as (Chapelle et al., 2007) where class proportion is enforced as a hard constraint on the labels or in (Mann & McCallum, 2007) where such information is used as a regularizer. We next discuss the alternating minimization procedure which is the key modification to the overall framework.

2.2. Alternating Minimization Procedure

In our proposed graph regularization framework, the cost function involves two variables to be optimized. While simultaneously recovering both solutions is intractable due to the mixed integer programming problem over binary $\mathbf{Y}$ and continuous $\mathbf{F}$, we will propose a greedy alternating minimization approach. The first update of the continuous classification function $\mathbf{F}$ is straightforward since the resulting cost function is convex and unconstrained allowing us to recover the optimal $\mathbf{F}$ by setting the partial derivative $\frac{\partial \mathcal{Q}}{\partial \mathbf{F}}$ to be zero. However, since $\mathbf{Y} \in \mathcal{B}$ is a binary matrix and subject to linear constraints of the form $\sum_{j=1}^{c} \mathbf{Y}_{ij}$, the other step in our alternating minimization requires solving a linearly constrained max cut problem which is NP (Karp, 1972). Due to the alternating minimization outer loop, investigating guaranteed approximation schemes (Goemans & Williamson, 1995) to solve a constrained max cut problem for $\mathbf{Y}$ is unjustified due to the solution's dependence on the dynamically varying classification function $\mathbf{F}$ during the alternating minimization procedure. Instead, we use a greedy gradient based approach to incrementally update $\mathbf{Y}$, while keeping the classification function $\mathbf{F}$ at the corresponding optimal setting. Moreover, because the node regularizer term $\mathbf{V}$ normalizes the labeled data, we also interleave updates of $\mathbf{V}$ based on the revised $\mathbf{Y}$.

Minimization for F :
The classification function $\mathbf{F} \in \mathcal{R}^{n\times c}$ is continuous and its loss terms are convex allowing the minimum to be recovered by zeroing the partial derivative:

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial \mathbf{F}^*} = 0 \quad &\Longrightarrow \quad \mathbf{LF}^* + \mu(\mathbf{F}^* - \mathbf{VY}) = 0 \\ &\Longrightarrow \quad \mathbf{F}^* = (\mathbf{L}/\mu + \mathbf{I})^{-1}\mathbf{VY} = \mathbf{PVY} \end{aligned} \tag{6}$$

where we denote $\mathbf{P} = (\mathbf{L}/\mu + \mathbf{I})^{-1}$ as the propagation matrix.

Greedy minimization of Y:
To update $\mathbf{Y}$, first replace $\mathbf{F}$ in Eq. 4 by its optimal vlue $\mathbf{F}^*$ from to solution of Eq. 6.

$$\begin{aligned} \mathcal{Q}(\mathbf{Y}) &= \mathrm{tr}(\mathbf{Y}^T\mathbf{V}^T\mathbf{P}^T\mathbf{LPVY} \\ &+ \mu(\mathbf{PVY} - \mathbf{VY})^T(\mathbf{PVY} - \mathbf{VY})) \\ &= \mathrm{tr}\left(\mathbf{Y}^T\mathbf{V}^T\left[\mathbf{P}^T\mathbf{LP} + \mu(\mathbf{P}^T - \mathbf{I})(\mathbf{P} - \mathbf{I})\right]\mathbf{VY}\right) \end{aligned} \tag{7}$$

The optimization still involves the node regularizer $\mathbf{V}$ in Eq. 5, which depends on $\mathbf{Y}$ and normalizes the label matrix over columns. Due to the dependence on the current estimate of $\mathbf{F}$ and $\mathbf{V}$, only an incremental step will be taken greedily to reduce Equation 6. In each iteration, we find position (i^*, j^*) in the matrix $\mathbf{Y}$ and change the binary value of $\mathbf{Y}_{i^*j*}$ from 0 to 1. The direction with largest negative gradient guides our choice of binary step on $\mathbf{Y}$. Therefore, we need to evaluate $\| \bigtriangledown \mathcal{Q}_{\mathbf{Y}}\|$ and find the largest negative value to determine (i^*, j^*).

Note that setting $\mathbf{Y}_{i^*,j^*} = 1$ is equivalent to a similar operation on the normalized label matrix $\mathbf{Z}$ by setting $\mathbf{Z}_{i^*,j^*} = \epsilon, 0 < \epsilon < 1$, and $\mathbf{Y}, \mathbf{Z}$ have one to one correspondence. Thus, the greedy optimization of $\mathcal{Q}$ with respect to $\mathbf{Y}$ is equivalent to greedy minimizing of $\mathcal{Q}$ with respect to $\mathbf{Z}$. More formally: $\frac{\partial \mathcal{Q}}{\partial \mathbf{Y}} = \frac{\partial \mathcal{Q}}{\partial \mathbf{Z}}\frac{\partial \mathbf{Z}}{\partial \mathbf{Y}}$ and with straightforward algebra we see that:

$$(i^*, j^*) = \arg\min_{i,j} \frac{\partial \mathcal{Q}}{\partial \mathbf{Y}} = \arg\min_{i,j} \frac{\partial \mathcal{Q}}{\partial \mathbf{Z}} \tag{8}$$

Then we can rewrite the loss function using the variable $\mathbf{Z}$ as:

$$\mathcal{Q}(\mathbf{Z}) = \mathrm{tr}\left(\mathbf{Z}^T\left[\mathbf{P}^T\mathbf{LP} + (\mathbf{P}^T - \mathbf{I})(\mathbf{P} - \mathbf{I})\right]\mathbf{Z}\right) \tag{9}$$

and recover the gradient with respect to $\mathbf{Z}$ of the above loss function as:

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial \mathbf{Z}} &= \left(\mathbf{P}^T\mathbf{LP} + (\mathbf{P}^T - \mathbf{I})(\mathbf{P} - \mathbf{I})\right)\mathbf{Z} \\ &+ \left(\mathbf{P}^T\mathbf{LP} + (\mathbf{P}^T - \mathbf{I})(\mathbf{P} - \mathbf{I})\right)^T\mathbf{Z} \end{aligned}$$

Let $\mathbf{A} = \mathbf{P}^T\mathbf{LP} + (\mathbf{P}^T - \mathbf{I})(\mathbf{P} - \mathbf{I})$. We simplify the representation of the gradient as:

$$\nabla_{\mathbf{Z}}\mathcal{Q} = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{Z} = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{VY} \tag{10}$$

As described earlier, we search the gradient matrix $\nabla_{\mathbf{Z}}\mathcal{Q}$ to find the minimal element for updating

$$\begin{aligned} (i^*, j^*) &= \arg\min\nolimits_{\mathbf{x}_i \in \mathcal{X}_u, 1 \le j \le c} \nabla_{\mathbf{Z}_{ij}} \mathcal{Q}^{t+1} \\ \mathbf{Y}^{t+1}_{i^*j^*} &= 1 \end{aligned} \tag{11}$$

Because of the binary nature of $\mathbf{Y}$, we simple set $\mathbf{Y}_{i^*j^*} = 1$ instead of using a continuous gradient approach. In $t+1$th iteration, after obtaining $\mathbf{Y}^{t+1}$, the node regularizer $\mathbf{v}^{t+1}$ can be recalculated with $\mathbf{Y}^{t+1}$ using Eq. 5.

The update $\mathbf{Y}$ is indeed greedy. Therefore, it could oscillate and backtrack from predicted labelings in previous iterations without convergence guarantees. We propose a straightforward way to guarantee convergence and avoid backtracking, inconsistency or unstable oscillation in the greedy propagation of labels. Once an unlabeled point has been labeled, its labeling can no longer be changed. Thus, we remove the most recently labeled point (i^*, j^*) from future consideration and only permit the algorithm to search for the minimal gradient entries corresponding to the remaining unlabeled examples. Thus, to avoid changing the labeling of previous predictions, the new labeled node $\mathbf{x}_{i^*}$ will be removed from $\mathcal{X}_u$ and added to $\mathcal{X}_l$.

In the following, we summarize the updating rules from step t to $t+1$ in the alternative minimization scheme. Although optimal $\mathbf{F}^*$ is being computed in each iteration as shown in Eq. 6, we do not explicitly need to update it. Instead, it is implicitly used in Eq. 8 to directly update $\mathbf{Y}$.

$$\begin{aligned}
\nabla_{(\mathbf{Z})}\mathcal{Q}^{t+1} &= \left(\mathbf{A}+\mathbf{A}^T\right) diag(\mathbf{v}^t)\mathbf{Y}^t \qquad (12)\\
(i^*, j^*) &= \arg\min\nolimits_{\mathbf{x}_i \in \mathcal{X}_u, 1\le j\le c} \nabla_{\mathbf{Z}_{ij}}\mathcal{Q}^{t+1}\\
\mathbf{Y}^{t+1}_{i^*j^*} &= 1\\
\mathbf{v}^{t+1} &= \sum_{j=1}^{c} \frac{\mathbf{Y}^{t+1}_{\cdot j} \odot \mathbf{D}\vec{\mathbf{1}}}{{\mathbf{Y}^{t+1}_{\cdot j}}^T \mathbf{D}\vec{\mathbf{1}}}\\
\mathcal{X}_l^{t+1} &\longleftarrow \mathcal{X}_l^t + \mathbf{x}_{i^*} \;\; ; \;\; \mathcal{X}_u^{t+1} \longleftarrow \mathcal{X}_u^t - \mathbf{x}_{i^*}
\end{aligned}$$

The procedure above repeats until all points have been labeled.

2.3. Algorithm Summary and Convergence

From the above discussion, our method is unique in that it optimizes the loss function over both continuous-valued $\mathbf{F}$ space and binary-valued $\mathbf{Y}$ space. Starting from a few given labels, the method iteratively and greedily updates the label matrix $\mathbf{Y}$, node regularizer $\mathbf{v}$, and gradient matrix $\nabla_{\mathcal{Q}}$. In each individual iteration, new labeled samples are obtained to drive a better graph propagation in the next iteration. In our approach, we directly acquire new labels instead of calculating $\mathbf{F}$ and then conducting a mapping to $\mathbf{Y}$, which is the regular procedure in other graph transduction methods like *LGC* and *GFHF*. This unique feature makes the proposed algorithm very efficient since we only update the gradient matrix $\nabla_{\mathbf{Z}}\mathcal{Q}$ in each iteration. Furthermore, similar to the graph superposition approach introduced in (Wang et al., 2008), the calculation of $\nabla_{\mathbf{Z}}\mathcal{Q}$ can be more efficient by conducting incremental updating as to the newly gained labels.

Algorithm 1 Graph Transduction via Alternating Minimization (*GTAM*)

```
Input: data set $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$, labeled subset $\mathcal{X}_l = \{\mathbf{x}_1, \cdots, \mathbf{x}_l\}$, unlabeled subset $\mathcal{X}_u = \{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$, labels $\{y_1, \cdots, y_j, \cdots, y_l\}$, where $y_j \in \mathcal{L} = \{1, \cdots, l\}$. Affinity matrix $\mathbf{W} = \{w_{ij}\}$, node degree matrix $\mathbf{D}$, initial label matrix $\mathbf{Y}^0$;
Initialization:
iteration counter $t = 0$;
normalized graph Laplacian $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2}$;
propagation matrix $\mathbf{P} = (\mathbf{L}/\mu + \mathbf{I})^{-1}$;
matrix $\mathbf{A} = \mathbf{P}^T\mathbf{L}\mathbf{P} + (\mathbf{P}^T - \mathbf{I})(\mathbf{P} - \mathbf{I})$;
node regularizer $\mathbf{v}^0 = \sum_{j=1}^{c} \frac{\mathbf{Y}^0_{\cdot j} \odot \mathbf{D}\vec{\mathbf{1}}}{{\mathbf{Y}^0_{\cdot j}}^T \mathbf{D}\vec{\mathbf{1}}}$.
repeat
  Compute graph gradient:
      $\mathbf{Z}^t = diag(\mathbf{v}^t)\mathbf{Y}^t$, $\nabla_{\mathbf{Z}^t}\mathcal{Q} = (\mathbf{A} + \mathbf{A}^T)\mathbf{Z}^t$;
  Find the optimal element in $\nabla_{\mathbf{Z}^t}\mathcal{Q}$:
      $(i^*, j^*) = \arg\min_{\mathbf{x}_i \in \mathcal{X}_u, 1\le j\le c} \nabla_{\mathbf{Z}_{ij}}\mathcal{Q}$;
  Update label matrix to obtain $\mathbf{Y}^{t+1}$ by setting:
      $\mathbf{Y}^{t+1}_{i^*j^*} = 1$; also $\mathbf{y}_{i^*} = j^*$;
  Update node regularizer by:
      $\mathbf{v}^{t+1} = \sum_{j=1}^{c} \frac{\mathbf{Y}^{t+1}_{\cdot j} \odot \mathbf{D}\vec{\mathbf{1}}}{{\mathbf{Y}^{t+1}_{\cdot j}}^T \mathbf{D}\vec{\mathbf{1}}}$;
  Remove $\mathbf{x}_{i^*}$ from $\mathcal{X}_u$: $\mathcal{X}_u^{t+1} \longleftarrow \mathcal{X}_u^t - \mathbf{x}_{i^*}$;
  Add $\mathbf{x}_{i^*}$ to $\mathcal{X}_l$: $\mathcal{X}_l^{t+1} \longleftarrow \mathcal{X}_l^t + \mathbf{x}_{i^*}$;
  Update iteration counter: $t = t + 1$;
until $\mathcal{X}_u^t = \emptyset$
Output:
The labels of unlabeled samples $\{y_{l+1}, \cdots, y_n\}$.
```

Due to greedy assignment, the algorithm's can only loop the alternative minimization (or the gradient computation equivalently) at most $n-l$ times. The update of the graph gradient, finding the largest element in the gradient and the matrix algebra involved can be done in efficiently by only a single entry in $\mathbf{Y}$ is modified per loop. Each minimization step over $\mathbf{F}$ and $\mathbf{Y}$ thus requires $\mathcal{O}(n^2)$ time and the total runtime of the greedy *GTAM* algorithm is $\mathcal{O}(n^3 - \ln n)$. Empirically, the value of the loss function $\mathcal{Q}$ decreases rapidly in the first dozens of iterations and achieves steady convergence afterward. This phenomenon indicates that the label propagation loop could be early terminated through deriving the labels from the optimized $\mathbf{F}$ (Eq. 6) after gaining enough labels. The above algorithm chart summarizes the proposed *GTAM* method.

3. Experiments

In this section, we demonstrate the superiority of the proposed *GTAM* method in comparison to state of the art semi-supervised learning methods over both synthetic and real data. For instance, on the WebKB data, previous work shows that *LapSVM* and *LapRLS* are better than other semi-supervised approaches, such as Transductive SVMs *TSVM* (Joachims, 1999) and ∇*TSVM*. Therefore, we only compare our method with *LapRLS*, *LapSVM* and two related methods, $LapRLS_{joint}$ and $LapSVM_{joint}$ (Sindhwani et al., 2005). In all experiments, we used the same parameter settings reported in the literatures. The *GTAM* approach only requires a single μ parameter which controls the tradeoff between the global smoothness and local fitting terms in the cost function. Although our experiments show that *GTAM* is fairly robust to the setting of μ, we set $\mu = 99$ throughout all experiments.

For all real implementations of graph-based methods, there needs a construction method that builds a graph from the training data $\mathcal{X}$, which involves a procedure for computing the weight of links via a kernel or similarity function. Typically, these methods use RBF kernels for image recognition and cosine distances for text classification (Zhou et al., 2004; Ng et al., 2001; Chapelle et al., 2003; Hein & Maier, 2006). However, finding adequate parameters for the kernel or similarity function, such as RBF kernel size δ, is not always straightforward particularly if labeled data is scarce. The empirical experience has shown that the propagation results highly depend on the kernel parameter selection. Motivated by the approach reported in (Hein & Maier, 2006), we use an adaptive kernel size based on the mean distance of k-nearest neighborhoods ($k = 6$) for the experiments on real USPS handwritten digit data. On the WebKB data, we use the same graph construction suggested by (Sindhwani et al., 2005). For each dataset, the same graph is used built for all the compared transductive learning approaches.

3.1. Two Moon Synthetic Data

Figure 1 illustrated synthetic experiments on 2D and 3D two-moon data. Despite the near-perfect classification results reported on such datasets in the literature (Zhou et al., 2004), we show how small perturbations to the problem can have adverse effects on prior algorithms. The prior methods are overly sensitive to locations of the initial labels, ratios of the two-class labels, and the level of ambient noise or outliers.

A more thorough experimental study is also possible for the two-moon data by exploring the effect of class imbalance. We start by fixing one class to have one observed label and select r labels from the other class. Here, r is also the imbalance ratio and the range we explore is $1 \leq r \leq 20$. These experiments use the 3D noisy two-moon data which contain 300 positive and 300 negative sample points as well as an additional 200 background noisy point. Multiple round tests (100 trails) are evaluated for each imbalance condition by calculating the average prediction accuracy on the relevant 600 samples. For a fair comparison, we use the same graph Laplacian, which is constructed using k-NN ($k = 6$) neighbors with RBF weights. Moreover, the parameter for *LGC* is set as $\alpha = 0.99$. The parameters for *LapRLS* and *LapSVM* are $\gamma_A = 1, \gamma_I = 1$. For *GTAM*, we set $\mu = 99$, which corresponding to $\alpha = 0.99$ in the *LGC* method.

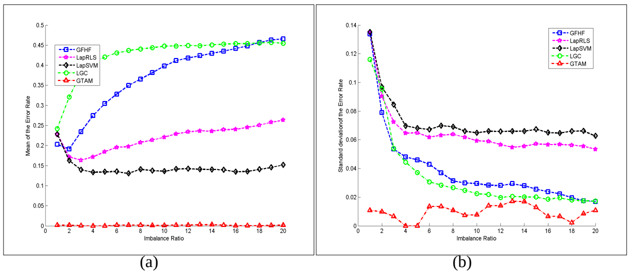

Figure 2. Performance comparison of *LGC*, *GFHF*, *LapRLS*, *LapSVM*, and *GTAM* on noisy 3D two moon data. Only one label is given for one class, while the other class has a varying number of labels, shown as imbalance ratio on the horizontal axis: (a) The mean of the test error; (b) The standard deviation of the test error.

Figure 2 demonstrates the performance advantage of the proposed *GTAM* approach versus the *LGC*, *GFHF*, *LapRLS*, and *LapSVM* methods. From the figure, we can conclude that all the four strawman approaches are extremely sensitive to the initial labels and label class imbalance since none of them can produce perfect accuracy and the error rates of *LGC* and *GFHF* are dramatically increased when the label class becomes more imbalanced *even though more information is being provided to the algorithm*. However, *GTAM* is clearly superior, achieving the best accuracy regardless of the imbalance ratio and despite contamination with noisy samples. In fact only 1 or 2 of the 100 trails for each individual setting of r were imperfect using the *GTAM* method.

3.2. WebKB Dataset

For validation on real data, we first evaluated our method using the WebKB dataset, which has been widely used in semi-supervised learning experiments (Joachims, 2003; Sindhwani et al., 2005). The WebKB dataset contains two document categories, *course* and *non-course*. Each document has two types of information, the webpage text content called *page* representation and *link* or pointer representation. For fair comparison, we applied the same feature extraction procedure as discussed in (Sindhwani et al., 2005), obtained 1051 samples with 1840-dimensional *page* attributes and 3000 *link* attributes. The graph was built based on cosine-distance neighbors with Gaussian weights (number of nearest neighbors is 200 as in (Sindhwani et al., 2005)). We compared our method with four of the best known approaches, *LapRLS*, *LapSVM*, and the two problem specific methods, $LapRLS_{joint}$, $LapSVM_{joint}$ reported in (Sindhwani et al., 2005). All the compared approaches used the same graph construction procedure and all parameter settings were set according to (Sindhwani et al., 2005), in other words $\gamma_A = 10^{-6}, \gamma_I = 0.01$. We varied the number of labeled data to measure the performance gain with increasing supervision. For each fixed number of labeled samples, 100 random trails were tested. The means of the test errors are shown in Figure 3.

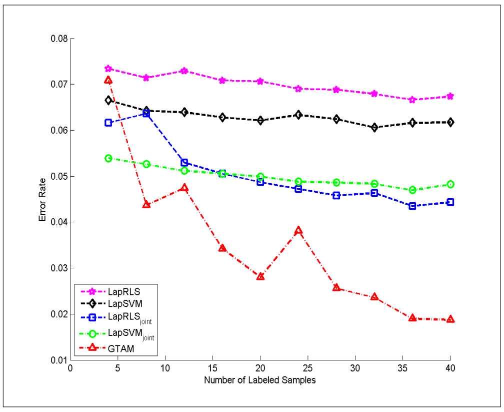

Figure 3. Performance comparison on text classification (WebKB dataset). The horizontal axis represents the number of randomly observed labels (guaranteeing there is at least one label for each class). The vertical axis shows the average error rate over 100 random trials.

As the Figure reveals, the proposed *GTAM* method achieved significantly better accuracy than all the other methods, except for the extreme case when only four labeled samples were available. The performance gain grows rapidly when the number of labeled samples increases, although in some cases the error rate does not drop monotonically.

3.3. USPS digit data

We also evaluated the proposed method in an image recognition task. Specifically, we used the data in (Zhou et al., 2004) for handwritten digit classification experiments. To evaluate the algorithms, we reveal a subset of the labels (randomly chosen and guaranteeing at least one labeled example is available for each digit). We compared our method with *LGC* and *GFHF*, *LapRLS*, and *LapSVM*. The error rates are calculated based on the average over 20 trials.

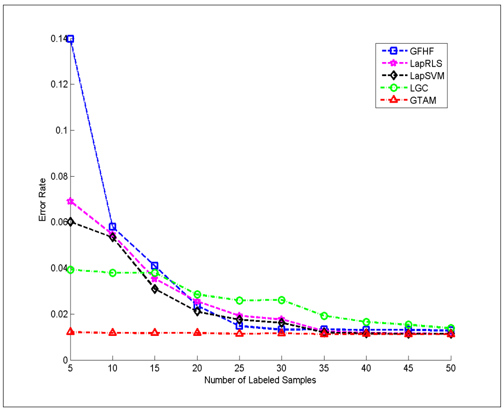

Figure 4. Performance comparison on handwritten digit classification (USPS database). The horizontal axis shows the total number of randomly observed labels (guaranteeing there is at least one label for each class). The vertical axis shows the average error rate over 20 random trials.

From Figure 4, we can conclude that *GTAM* significantly improved the classification accuracy, compared to the other approaches, especially when very few labeled samples are available. The mean accuracies of *GTAM* are consistently low for different numbers of labels and the standard deviation values are also very small (10^{-4} level). This demonstrates that the *GTAM* method is insensitive to the numbers and specified locations of the initially given labels. Only 1% of the test digit images were mislabeled. These failure cases are presented in Figure 5 and are often ambiguous or extremely poorly drawn digits. Compared to the performance on WebKB dataset shown in Figure 3, the USPS digit database experiments exhibit even more promising results. One possible reason is that USPS digit dataset has relatively more samples (3874) and a lower feature dimensionality (256), compared to WebKB dataset (which has 1840 samples in 4800 dimensions). Therefore the graph construction procedure is more reliable and the estimation of graph gradients in our algorithm is more robust.

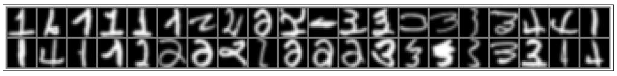

Figure 5. USPS handwritten digit samples which are incorrectly classified.

4. Conclusion and Discussion

Existing graph-based transductive learning methods hinge on good labeling information and can easily be misled if the labels are not distributed evenly across classes, if the choice of initial label locations is varied or if excessive noise or outliers corrupt the underlying manifold structure. These degenerate settings seem to plague real world problems as well, compromising the performance of state-of-the-art graph transduction methods. Our experiments over synthetic data sets (two moon data sets) and the real data sets (USPS digits and WebKB) confirm the shortcomings of the existing tools.

This article addresses these shortcomings and proposes a novel graph transduction method, graph transduction via alternating minimization (*GTAM*). Therein, both the classification function and the label matrix are treated as variables in a cost function that is iteratively minimized. While the optimal classification function can be estimated exactly, greedy optimization is applied to update the label matrix. The algorithm iterates an alternating minimization between both variables and is guaranteed to converge via a greedy scheme. In each individual iteration, through the graph gradient, the unlabeled node with the largest cost reduction is labeled. We gradually update the label matrix by adding more labeled samples while keeping the classification function at its optimal setting. Furthermore, we enforce normalization of the label matrix to avoid degeneracies. This results in an algorithm that can cope with all the aforementioned degeneracies and in practice achieves significant gains in accuracy while remaining efficient and cubic in the number of samples. Future work will include out of sample extensions of this method such that new data points can be added to the training and test set without requiring a full retraining procedure.

5. Acknowledgments

The authors would like to thank Mr. Yongtao Su and Shouzhen Liu for their valuable comments. We also thank Mr. We Liu and Deli Zhao for the collection of the artificial data.

References

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *JMLR*, *7*, 2399–2434.

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. *Proc. 18th ICML* (pp. 19–26).

Chapelle, O., Sindhwani, V., & Keerthi, S. (2007). Branch and Bound for Semi-Supervised Support Vector Machines. *Proc. of NIPS*.

Chapelle, O., Weston, J., & Scholkopf, B. (2003). Cluster kernels for semi-supervised learning. *Proc. NIPS*, *15*, 1.

Goemans, M., & Williamson, D. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, *42*, 1115–1145.

Hein, M., & Maier, M. (2006). Manifold denoising. *Proc. NIPS*, *19*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proc. of the ICML*, 200–209.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Proc. of ICML*, 290–297.

Karp, R. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, *43*, 85–103.

Mann, G., & McCallum, A. (2007). Simple, robust, scalable semi-supervised learning via expectation regularization. *Proc. of the ICML*, 593–600.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Proc. NIPS*, *14*, 849–856.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. *Proc. of ICML*.

Wang, J., Chang, S.-F., Zhou, X., & Wong, T. C. S. (2008). Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. *IEEE CVPR*. Alaska, USA.

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Scholkopf, B. (2004). Learning with local and global consistency. *Proc. NIPS* (pp. 321–328).

Zhu, X. (2005). *Semi-supervised learning literature survey* (Technical Report 1530). Computer Sciences, University of Wisconsin-Madison.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. *Proc. 20th ICML*.