Efficient Label Propagation

Yasuhiro Fujiwara FUJIWARA.YASUHIRO@LAB.NTT.CO.JP
NTT Software Innovation Center, 3-9-11 Midori-cho Musashino-shi, Tokyo, Japan

Go Irie IRIE.GO@LAB.NTT.CO.JP
NTT Media Intelligence Laboratories, 1-1 Hikarinooka Yokosuka-shi, Kanagawa, Japan

Abstract

Label propagation is a popular graph-based semi-supervised learning framework. So as to obtain the optimal labeling scores, the label propagation algorithm requires an inverse matrix which incurs the high computational cost of $O(n^3+cn^2)$, where n and c are the numbers of data points and labels, respectively. This paper proposes an efficient label propagation algorithm that guarantees exactly the same labeling results as those yielded by optimal labeling scores. The key to our approach is to iteratively compute lower and upper bounds of labeling scores to prune unnecessary score computations. This idea significantly reduces the computational cost to $O(cnt)$ where t is the average number of iterations for each label and $t \ll n$ in practice. Experiments demonstrate the significant superiority of our algorithm over existing label propagation methods.

1. Introduction

Semi-supervised learning has been a dominant research topic in the machine learning area. Given a dataset consisting of both labeled and unlabeled data points, the task is to assign labels to the unlabeled subset. A number of semi-supervised learning methods have been proposed (Chapelle et al., 2010). One major framework, label propagation, was proposed by Zhou et al. (Zhou et al., 2003). Our goal in this paper is to develop an efficient algorithm for label propagation.

The key assumption in label propagation is that data points occupying the same manifold are very likely to share the same semantic label (Zhou et al., 2003). To this end, label propagation aims to "propagate" labels of the labeled data points to the unlabeled data points according to the intrinsic data manifold structures collectively revealed by a large number of data points. This implies that the label propagation algorithm can more successfully estimate the labels as the number of (labeled or unlabeled) data points increases. Obviously this results in higher computation time.

Theoretically, the labeling scores of the unlabeled data points are computed by minimizing the cost function where the optimal solution is obtained by means of the inverse of the adjacency matrix of a data graph (e.g., k-NN graph) (Zhou et al., 2003). Since the size of the adjacency matrix is generally $O(n^2)$, computing its inverse takes $O(n^3)$ time where n is the number of data points (Belkin et al., 2006). Consequently, $O(n^3 + cn^2)$ time is required to determine the labels for all unlabeled data points from c types of labels, which might be intractable for large-scale datasets. The original label propagation algorithm proposed by Zhou et al. uses the power method (Golub & Loan, 2012) to enhance computation speed (Zhou et al., 2003); the power method is the standard approach for label propagation. Even though the power method converges to the theoretically correct scores, practically, the algorithm terminates when the residual is less than some predetermined value (Xu et al., 2011). The labeling results (scores) after termination can differ from the theoretical ones in practice, resulting in unsatisfactory performance.

In this paper, we propose a new efficient label propagation algorithm. The key idea of our approach is to compute lower and upper bounding scores and thus iteratively prune unnecessary score computations in determining a label for each node. The resulting computation time of our approach falls to $O(cnt)$, where t is the average number of iterations for each label. In practice, $t \ll n$, so our approach is significantly faster than the original algorithm. Even though many approximation approaches have been proposed for efficient label propagation (Fergus et al., 2009; Kumar et al., 2009; Talwalkar et al., 2008; Yu & Yu, 2005; Zhu et al., 2003), one key advantage of our approach compared to them is that it guarantees the same label-

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

Table 1. Definition of main symbols.

Symbol	Definition
n	Number of data points
c	Number of labels
x_i	i-th data point
l_i	i-th label
$y(x_i)$	Label of data point x_i
$f(x_i\|l_j)$	i-th element of vector $\mathbf{f}_j$
$\overline{f}_t(x_i\|l_j)$	Upper bound of $f(x_i\|l_j)$ in the t-th iteration
$\underline{f}_t(x_i\|l_j)$	Lower bound of $f(x_i\|l_j)$ in the t-th iteration
$\mathbf{f}_i$	i-th column vector of matrix $\mathbf{F}$
$\mathbf{y}_i$	i-th column vector of matrix $\mathbf{Y}$
$\mathbf{W}$	$n \times n$ adjacency matrix of the k-NN graph
$\mathbf{S}$	Normalization matrix of $\mathbf{W}$
$\mathbf{F}$	$n \times c$ classification matrix
$\mathbf{Y}$	$n \times c$ initial label matrix

ing results as the optimal solution yielded by the inverse matrix computation. Moreover, our approach can handle several types of graphs such as the linear neighborhood graph and the sparse $\mathcal{L}^1$ graph (Wang & Zhang, 2008; Elhamifar & Vidal, 2011). On the other hand, the previous approaches do not have this property since their focus is on the graph Laplacian, where edge weights are forced to be non-negative (von Luxburg, 2007). Alexandrescu et al. collapsed multiple nodes having the same label before applying the power method to increase its speed (Alexandrescu & Kirchhoff, 2007). In addition, Subramanya et al. effectively used the cache in a parallel computing implementation by ordering the nodes for the power method (Subramanya & Bilmes, 2009). Since we iteratively compute the estimations similar to the power method, their approaches complement our approach. Experiments confirm that our approach is much faster than the existing methods. Note that our approach does not require users to set any inner-parameters whereas the previous approaches, including the power method, have inner-parameters that significantly impact the labeling results.

The remainder of this paper is organized as follows. Section 2 briefly reviews the original label propagation approach by Zhou et al. and the power method. Section 3 introduces the main ideas and details of our algorithm. Section 4 reviews the results of our experiments. Section 5 provides our conclusions.

2. Preliminary

In this section, we briefly review label propagation proposed by Zhou et al. (Zhou et al., 2003). Table 1 lists the main symbols and their definitions. $\mathbb{X} = \{x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_n\}$ represents a set of data points, and $\mathbb{L} = \{l_1, l_2, \ldots, l_c\}$ is the label set. The first m data points, $\{x_1, x_2, \ldots, x_m\}$, are labeled by $\{y(x_1), y(x_2), \ldots, y(x_m) | y(x_i) \in \mathbb{L}\}$ and the remaining data points are unlabeled. The goal of label propagation is to predict the labels of unlabeled data points which can be achieved as mentioned below.

First, a graph $\mathbb{G} = \{\mathbb{V}, \mathbb{E}\}$ is constructed where the set of nodes $\mathbb{V}$ is the set of data points $\mathbb{X}$, i.e., $\mathbb{V} = \mathbb{X}$. $\mathbb{E}$ is the set of edges whose weights reflect the similarities among data points. The k-NN graph scheme is the most popular approach for graph construction. In the k-NN graph, a node pair share an undirected edge if the two nodes are k-nearest neighbors (von Luxburg, 2007). This indicates that the number of edges is $O(n)$ and the graph is symmetric. Conventionally, the edge weight between point x_i and x_j, W_{ij}, is obtained by a Gaussian kernel (Bishop, 2007); $W_{ij} = \exp\{-||x_i - x_j||^2/2\sigma^2\}$ if an edge connects data point x_i to x_j, otherwise $W_{ij} = 0$. In this equation, σ is a hyperparameter. Many researchers have proposed efficient approaches for the k-NN graph construction (Chen et al., 2009; Connor & Kumar, 2010; Dong et al., 2011).

Next, the node scores are computed for each label to determine labels for unlabeled data points. In label propagation, the labeling scores are defined as the optimal solution that minimizes the cost function. The $n \times c$ size matrix $\mathbf{F}$ corresponds to a classification on data points $\mathbb{X}$ by labeling each data point; matrix $\mathbf{F}$ holds the labeling scores of all data points for each label. $\mathbf{Y}$ is an $n \times c$ size matrix where $Y_{ij} = 1$ if point x_i is initially labeled as $y(x_i) = l_j$ and $Y_{ij} = 0$ otherwise. Let $\mathbf{F}_i$ and $\mathbf{Y}_i$ be the i-th row vector of $\mathbf{F}$ and $\mathbf{Y}$, respectively, i.e., $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n]^T$ and $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n]^T$. The cost function $C(\mathbf{F})$ associated with classification matrix $\mathbf{F}$ is defined as follows:

$$\begin{aligned} C(\mathbf{F}) = \tfrac{1}{2} \textstyle\sum_{i,j=1}^{n} W_{ij} \left\| \frac{\mathbf{F}_i}{\sqrt{D_{ii}}} - \frac{\mathbf{F}_j}{\sqrt{D_{jj}}} \right\|^2 \\ + \left(\tfrac{1}{\alpha} - 1\right) \textstyle\sum_{i=1}^{n} \|\mathbf{F}_i - \mathbf{Y}_i\|^2 \end{aligned} \tag{1}$$

In Equation (1), $\mathbf{D}$ is a diagonal matrix where $D_{ii} = \sum_{j=1}^{n} W_{ij}$, and α is a constant parameter such that $0 < \alpha < 1$ (Zhou et al., 2003). The cost function is designed to enhance the accuracy of label prediction. The first and second terms in the cost function $C(\mathbf{F})$ correspond to the *smoothness constraint* and the *fitting constraint*, respectively. The smoothness constraint means that a good classifying function should not change too much between nearby points. The fitting constraint means good classification should not change too much from the initial label assignment. Minimizing the cost function yields the optimal $\mathbf{F}$ in the following closed form:

$$\mathbf{F} = (\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{Y} \tag{2}$$

where $\mathbf{I}$ is an identity matrix of size $n \times n$ and $\mathbf{S}$ is computed from matrix $\mathbf{W}$ as $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. Let $\mathbf{f}_i$ $(1 \leq i \leq c)$ be the i-th column vector of matrix $\mathbf{F}$ that corresponds to label l_i, and let $f(x_i|l_j)$ be the i-th element of vector $\mathbf{f}_j$ that corresponds to data point x_i. The label of data point x_i, $y(x_i)$, is obtained as follows:

$$y(x_i) = \arg\max_{1 \leq j \leq c} f(x_i|l_j) \tag{3}$$

Equation (2) indicates that the labeling score computation involves the matrix inversion operation; it requires $O(n^3)$ time to compute the inverse matrix (Belkin et al., 2006). Moreover, it takes $O(cn^2)$ time to compute the classification matrix $\mathbf{F}$ since matrix $\mathbf{F}$ is obtained as the product of matrix $(\mathbf{I}-\alpha\mathbf{S})^{-1}$ and $\mathbf{Y}$ whose sizes are $n\times n$ and $n\times c$, respectively. Consequently, the approach requires $O(n^3+cn^2)$ time to obtain labels from the graph which is prohibitively high.

Zhou et al. proposed to utilize the power method (Golub & Loan, 2012) to enhance the labeling speed, and this is the standard approach for label propagation. Their approach iteratively updates the labeling scores in the following form where $\mathbf{F}_t$ is the classification matrix of the t-th iteration (Zhou et al., 2003):

$$\mathbf{F}_{t+1}=\alpha\mathbf{S}\mathbf{F}_t+(1-\alpha)\mathbf{Y} \tag{4}$$

It is known that the scores yielded by the power method are equivalent to those by the optimal solution (Equation (2)) after convergence; $\mathbf{F}_\infty=\mathbf{F}$. However, in practice, the power method terminates the iterations when the residual is less than some predetermined value (Xu et al., 2011). This indicates that the power method approximately computes the labeling scores. Even though the power method is the standard approach for label propagation, it does not output the same labeling results as the the optimal solution.

3. Proposed method

This section presents our fast label propagation approach; it outputs the same labeling results as the optimal solution. First, we give an overview of the ideas underlying our approach and then provide a full description in Sections 3.1 to 3.4. We also give some theoretical analyses of its performance in Section 3.5. Finally, we show that we can handle other graph construction approaches, such as linear neighborhood graphs (Wang & Zhang, 2008) and sparse $\mathcal{L}^1$ graphs (Elhamifar & Vidal, 2011), as well as k-NN graphs in Section 3.6.

3.1. Ideas

The power method iteratively computes the labeling scores until convergence for all labels. In order to enhance the efficiency, we do not update the scores for all labels. Instead, our approach updates labeling scores for a subset of labels. Subset membership is determined by using the lower and upper bounds of labeling scores. This approach has several strong advantages. First, if there is no label to be updated, we terminate the iterations without waiting for convergence, unlike the power method. This implies that our approach needs fewer iterations than the power method. Second, we can obtain exactly the same labeling results as the optimal solution. This is because the lower/upper bounds allow us to safely discard unnecessary score computations. Finally, our approach does not require any user-defined inner-parameter. By contrast, the power method requires setting of the predetermined threshold for iteration termination, which induces a trade-off between efficiency and accuracy. That is, our approach is user-friendly.

3.2. Lower/upper bounds

We iteratively compute the lower and upper bounds for the labeling scores to efficiently obtain a label for each node. In the t-th iteration ($t=0,1,2,\ldots$), we compute the lower/upper bounds for label set $\mathbb{L}_t$; we detail the procedure used to obtain $\mathbb{L}_t$ in Section 3.3. Let $\mathbf{y}_i$ be the i-th column vector of matrix $\mathbf{Y}$, and let $y(x_i|l_j)$ be the i-th element of vector $\mathbf{y}_j$. $\mathbf{y}_i$ corresponds to scores of initially labeled nodes for label l_i. $y(x_i|l_j)=Y_{ij}$ is the initial label score of data point x_i with respect to label l_j. We here introduce *propagation score* $p_t(x_i|l)$ to obtain the lower/upper bounds. We iteratively compute propagation score $p_t(x_i|l)$ of data point x_i for label l in the t-th iteration as follows:

$$p_t(x_i|l)=\begin{cases} y(x_i|l) & (t=0)\\ \sum_{x_j\in\mathbb{X}}S_{ij}p_{t-1}(x_j|l) & (t\neq 0)\end{cases} \tag{5}$$

This equation indicates that (1) the propagation scores are initialized by the initial label setting if $t=0$ and (2) the propagation scores are incrementally updated from those of the previous iteration and the matrix $\mathbf{S}$. The lower bound of data point x_i for label l, $\underline{f}_t(x_i|l)$, is obtained by using the propagation scores as follows:

Definition 1 (Lower bound) *The lower bound of labeling score $f(x_i|l)$ in the t-th iteration is computed as follows:*

$$\underline{f}_t(x_i|l)=(1-\alpha)\Big\{\textstyle\sum_{\tau=0}^{t}\{\alpha^\tau p_\tau(x_i|l)\}+\frac{\alpha^{t+1}\underline{\sigma}\,\underline{p}_t(l)}{1-\alpha\underline{\sigma}}\Big\} \tag{6}$$

where $\underline{\sigma}$ and $\underline{p}_t(l)$ are defined as follows:

$$\underline{\sigma}=\min\nolimits_{1\leq i\leq n}\textstyle\sum_{x_j\in\mathbb{X}}S_{ij} \tag{7}$$

$$\underline{p}_t(l)=\min\nolimits_{1\leq i\leq n}p_t(x_i|l) \tag{8}$$

Before describing the lower bounding property of $\underline{f}_t(x_i|l)$, we introduce the following two lemmas which underlie the lower bounding property:

Lemma 1 ($\mathcal{L}^1$ norm of row elements) *The value of $\underline{\sigma}$ is not larger than* 1, *i.e.* , $\underline{\sigma}\leq 1$.

Proof As shown in (Zhou et al., 2003), the i-th eigenvalue of matrix $\mathbf{S}$, λ_i, is $-1\leq\lambda_i\leq 1$. Since $\mathbf{S}$ is clearly a non-negative matrix, we have the following inequality for the spectral radius of matrix $\mathbf{S}$, $\rho(\mathbf{S})$, from the Perron-Frobenius theorem (Golub & Loan, 2012):

$$\underline{\sigma}=\min\nolimits_{1\leq i\leq n}\textstyle\sum_{x_j\in\mathbb{X}}S_{ij}\leq\rho(\mathbf{S})=\max\nolimits_{1\leq i\leq n}|\lambda_i|\leq 1$$

Therefore, we have $\underline{\sigma}\leq 1$. □

Lemma 2 (Lower bounding difference) *For the $(t+\tau)$-th iteration where $\tau \geq 1$, we have $p_{t+\tau}(x_i|l) \geq \underline{p}_t(l)\underline{\sigma}^\tau$*

Proof We prove Lemma 2 by mathematical induction (Gunderson, 2010).

Initial step: From Equation (5), we have the following inequality in the $(t+1)$-th iteration:

$$p_{t+1}(x_i|l) = \sum_{x_j \in \mathbb{X}} S_{ij} p_t(x_j|l) \geq \underline{p}_t(l) \sum_{x_j \in \mathbb{X}} S_{ij} \geq \underline{p}_t(l)\underline{\sigma}$$

Inductive step: In the $(t+\tau-1)$-th iteration, we assume that $p_{t+\tau-1}(x_i|l) \geq \underline{p}_t(l)\underline{\sigma}^{\tau-1}$ holds. From Equation (5),

$$\begin{aligned} p_{t+\tau}(x_i|l) &= \textstyle\sum_{x_j \in \mathbb{X}} S_{ij} p_{t+\tau-1}(x_j|l) \\ &\geq \underline{p}_t(l)\underline{\sigma}^{\tau-1} \textstyle\sum_{x_j \in \mathbb{X}} S_{ij} \geq \underline{p}_t(l)\underline{\sigma}^\tau \end{aligned}$$

This completes the inductive step. Therefore, $p_{t+\tau}(x_i|l) \geq \underline{p}_t(l)\underline{\sigma}^\tau$ holds. □

By utilizing Lemma 1 and 2, we show the lower bounding property of $\underline{f}_t(x_i|l)$.

Lemma 3 (Lower bound) *For the labeling score of data point x_i, $\underline{f}_t(x_i|l) \leq f(x_i|l)$ holds in the t-th iteration.*

Proof From Equation (4), we have

$$\begin{aligned} \mathbf{F}_t &= \alpha\mathbf{S}\mathbf{F}_{t-1} + (1-\alpha)\mathbf{Y} = \alpha^2\mathbf{S}^2\mathbf{F}_{t-2} + (1-\alpha)(\alpha\mathbf{S}\mathbf{Y} + \mathbf{Y}) \\ &= \ldots = \alpha^t\mathbf{S}^t\mathbf{Y} + (1-\alpha)\textstyle\sum_{\tau=0}^{t-1}(\alpha^\tau\mathbf{S}^\tau\mathbf{Y}) \end{aligned}$$

Since (1) $\lim_{t\to\infty}(\alpha\mathbf{S})^t = 0$ as shown in (Zhou et al., 2003) and (2) the power method has the property of converging to the theoretically correct scores, we have

$$\begin{aligned} \mathbf{F} &= \lim_{t\to\infty}\{(\alpha\mathbf{S})^t\mathbf{Y} + (1-\alpha)\textstyle\sum_{\tau=0}^{t-1}(\alpha^\tau\mathbf{S}^\tau\mathbf{Y})\} \\ &= (1-\alpha)\textstyle\sum_{\tau=0}^{\infty}(\alpha^\tau\mathbf{S}^\tau\mathbf{Y}) \end{aligned}$$

Therefore, for column vector $\mathbf{f}$ in matrix $\mathbf{F}$ and the corresponding column vector $\mathbf{y}$ in matrix $\mathbf{Y}$, we have the following equation from Equation (5):

$$\mathbf{f} = (1-\alpha)\textstyle\sum_{\tau=0}^{\infty}(\alpha^\tau\mathbf{S}^\tau\mathbf{y}) = (1-\alpha)\sum_{\tau=0}^{\infty}(\alpha^\tau\mathbf{p}_\tau(l))$$

where $\mathbf{p}_\tau(l)$ is an $n \times 1$ vector where the i-th element is $p_\tau(x_i|l)$. Consequently, the i-th element of vector $\mathbf{f}$, which corresponds to data point x_i, can be computed as follows:

$$f(x_i|l) = (1-\alpha)\textstyle\sum_{\tau=0}^{\infty}\{\alpha^\tau p_\tau(x_i|l)\}$$

From the above equation and Lemma 2, $\underline{f}_t(x_i|l)$ can be computed as follows:

$$\begin{aligned} f(x_i|l) &= (1-\alpha)\{\textstyle\sum_{\tau=0}^{t}(\alpha^\tau p_\tau(x_i|l)) + \sum_{\tau=1}^{\infty}(\alpha^{t+\tau}p_{t+\tau}(x_i|l))\} \\ &\geq (1-\alpha)\{\textstyle\sum_{\tau=0}^{t}(\alpha^\tau p_\tau(x_i|l)) + \alpha^t\underline{p}_t(l)\sum_{\tau=1}^{\infty}(\alpha^\tau\underline{\sigma}^\tau)\} \end{aligned}$$

Since $0 < \alpha < 1$ and $\underline{\sigma}^\tau \leq 1$ from Lemma 1, we have $\sum_{\tau=1}^{\infty}(\alpha^\tau\underline{\sigma}^\tau) = \frac{\alpha\underline{\sigma}}{1-\alpha\underline{\sigma}}$. As a result,

$$f(x_i|l) \geq (1-\alpha)\left\{\textstyle\sum_{\tau=0}^{t}\{\alpha^\tau p_\tau(x_i|l)\} + \frac{\alpha^{t+1}\underline{\sigma}\,\underline{p}_t(l)}{1-\alpha\underline{\sigma}}\right\} = \underline{f}_t(x_i|l)$$

which completes the proof. □

The upper bound can be obtained by exploiting the propagation scores as follows:

Definition 2 (Upper bound) *In the t-th iteration, the following equation gives the upper bound of labeling score $f(x_i|l)$, $\overline{f}_t(x_i|l)$:*

$$\begin{aligned} &\overline{f}_t(x_i|l) = \\ &(1-\alpha)\textstyle\sum_{\tau=0}^{t}\{\alpha^\tau p_\tau(x_i|l)\} + \alpha^{t+1}\{p_t(x_i|l) + \frac{n\delta_t\overline{S}(x_i)}{1-\alpha}\} \end{aligned} \tag{9}$$

In this equation, $\overline{S}(x_i)$ and δ_t are defined as follows:

$$\overline{S}(x_i) = \max\nolimits_{1\leq j\leq n} S_{ij} \tag{10}$$

$$\delta_t = \begin{cases} n & (t=0) \\ \sum_{x_i\in\mathbb{X}} \max\{p_t(x_i|l) - p_{t-1}(x_i|l), 0\} & (t\neq 0) \end{cases} \tag{11}$$

The upper bounding property of Definition 2 is based on the following two lemmas:

Lemma 4 ($\mathcal{L}^1$ norm of column elements) *Letting $\mathbf{e}_i$ be an $n \times 1$ vector of zeros with only the i-th element set to 1, we have $\mathbf{S}^\tau\mathbf{e}_i \leq n$.*

Proof Let $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$ be an $n \times n$ matrix composed of the eigenvectors of $\mathbf{S}$ where $\mathbf{u}_i$ is eigenvector of eigenvalue λ_i. In addition, let $\mathbf{D}$ be a diagonal matrix of eigenvalues such that $\mathbf{D} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Since matrix $\mathbf{S}$ is symmetric, we have $\mathbf{U}^{-1} = \mathbf{U}^T$. Therefore, $\mathbf{S}^\tau = (\mathbf{U}\mathbf{D}\mathbf{U}^{-1})^\tau = \mathbf{U}\mathbf{D}^\tau\mathbf{U}^T$. As shown in (Zhou et al., 2003), we have $-1 \leq \lambda_i \leq 1$. Therefore,

$$\mathbf{S}^\tau\mathbf{e}_i = \textstyle\sum_{1\leq j\leq n}\lambda_j^\tau\mathbf{u}_j\mathbf{u}_j^T\mathbf{e}_i \leq \sum_{1\leq j\leq n}1^\tau\mathbf{u}_j\mathbf{u}_j^T\mathbf{e}_i = n$$

As a result, we have $\mathbf{S}^\tau\mathbf{e}_i \leq n$. □

Lemma 5 (Upper bounding difference) *For the $(t+\tau)$-th iteration, $p_{t+\tau}(x_i|l) \leq p_t(x_i|l) + \tau n\delta_t\overline{S}(x_i)$ holds.*

Proof Letting $S_{ij}^{\tau-1}$ be the (i,j) element of matrix $\mathbf{S}^{\tau-1}$, from Equation (5), we have

$$\begin{aligned} &p_{t+\tau}(x_i|l) - p_{t+\tau-1}(x_i|l) \\ &= \textstyle\sum_{x_j\in\mathbb{X}}\sum_{x_k\in\mathbb{X}} S_{ij}S_{jk}^{\tau-1}\{p_t(x_k|l) - p_{t-1}(x_k|l)\} \\ &\leq \textstyle\sum_{x_k\in\mathbb{X}}\sum_{x_j\in\mathbb{X}} \overline{S}(x_i)S_{jk}^{\tau-1}\max\{p_t(x_k|l) - p_{t-1}(x_k|l), 0\} \\ &= \overline{S}(x_i)\textstyle\sum_{x_k\in\mathbb{X}}\max\{p_t(x_k|l) - p_{t-1}(x_k|l), 0\}\left(\sum_{x_j\in\mathbb{X}} S_{jk}^{\tau-1}\right) \end{aligned}$$

From Lemma 4, we have $\sum_{x_j\in\mathbb{X}} S_{jk}^{\tau-1} = \mathbf{S}^{\tau-1}\mathbf{e}_k \leq n$. Therefore, from Equation (11),

$$p_{t+\tau}(x_i|l) - p_{t+\tau-1}(x_i|l) \leq n\delta_t\overline{S}(x_i)$$

As a result,

$$p_{t+\tau}(x_i|l) \leq p_{t+\tau-1}(x_i|l) + n\delta_t\overline{S}(x_i) \leq \ldots \leq p_t(x_i|l) + \tau n\delta_t\overline{S}(x_i)$$

which completes the proof. □

The upper bounding property is introduced as follows:

Lemma 6 (Upper bound) *We have $\overline{f}_t(x_i|l) \geq f(x_i|l)$ for the labeling score of data point x_i in the t-th iteration.*

Proof As shown in the proof of Lemma 3,

$$f(x_i|l) = (1-\alpha)\{\sum_{\tau=0}^{t}(\alpha^\tau p_\tau(x_i|l)) + \sum_{\tau=1}^{\infty}(\alpha^{t+\tau} p_{t+\tau}(x_i|l))\}$$

Therefore, from Lemma 5, we have

$$f(x_i|l) \leq (1-\alpha)\{ \sum_{\tau=0}^{t}(\alpha^\tau p_\tau(x_i|l)) + \alpha^t p_t(x_i|l) \sum_{\tau=1}^{\infty} \alpha^\tau + \alpha^t n \delta_t \overline{S}(x_i) \sum_{\tau=1}^{\infty} \tau \alpha^\tau \}$$

Since $\sum_{\tau=1}^{\infty} \alpha^\tau = \frac{\alpha}{1-\alpha}$ and $\sum_{\tau=1}^{\infty} \tau\alpha^\tau = \frac{\alpha}{(1-\alpha)^2}$,

$$f(x_i|l) \leq (1-\alpha)\Big\{ \sum_{\tau=0}^{t}(\alpha^\tau p_\tau(x_i|l)) + \frac{\alpha^{t+1} p_t(x_i|l)}{1-\alpha} + \frac{\alpha^{t+1} n \delta_t \overline{S}(x_i)}{(1-\alpha)^2} \Big\} = \overline{f}_t(x_i|l)$$

Consequently, we have $\overline{f}_t(x_i|l) \geq f(x_i|l)$. □

As described in Section 3.1, we iteratively compute the lower/upper bounds. Note that we do not compute the bounds with Definition 1 and 2 in each iteration. Instead, we incrementally update the lower/upper bounds for efficient labeling by using the following property:

Lemma 7 (Incremental update) *In the t-th iteration, if the propagation score of data point x_i is obtained, the lower/upper bounds of the t-th iteration can be incrementally updated from those of the $(t-1)$-th iteration at $O(1)$ time as follows:*

$$\underline{f}_t(x_i|l) = \underline{f}_{t-1}(x_i|l) + (1-\alpha)\alpha^t \Big\{ p_t(x_i|l) + \frac{\underline{\sigma}(\alpha \underline{p}_t(l) - \underline{p}_{t-1}(l))}{1-\alpha\underline{\sigma}} \Big\} \quad (12)$$

$$\overline{f}_t(x_i|l) = \overline{f}_{t-1}(x_i|l) + \alpha^t \Big\{ p_t(x_i|l) - p_{t-1}(x_i|l) + \frac{n\overline{S}(x_i)(\alpha\delta_t - \delta_{t-1})}{1-\alpha} \Big\} \quad (13)$$

The proof of Lemma 7 is omitted due to space limits. However, this property can be shown by computing $\underline{f}_t(x_i|l) - \underline{f}_{t-1}(x_i|l)$ and $\overline{f}_t(x_i|l) - \overline{f}_{t-1}(x_i|l)$ from Definition 1 and 2, respectively. We can efficiently compute the lower/upper bounds in the iterations by utilizing Lemma 7.

For the convergence values of $\underline{f}_t(x_i|l)$ and $\overline{f}_t(x_i|l)$, we have the following property:

Lemma 8 (Convergence of the lower/upper bounds) *The lower/upper bounds converge to the exact labeling score. That is, $\underline{f}_\infty(x_i|l) = \overline{f}_\infty(x_i|l) = f(x_i|l)$ holds.*

Even though we omit the proof of this lemma due to space limits, it can be derived from Equation (6) and (9) by using the property of $p_t(x_i|l) \leq n$ obtained from Lemma 4. This lemma implies that the bounds are expected to tighten as the number of iterations increases. Furthermore, this lemma gives the theoretical guarantee that our approach outputs the same labeling results as the optimal solution.

3.3. Label set

In the t-th iteration, we compute the lower/upper bounds for label set $\mathbb{L}_t$ instead of all the labels to enhance the efficiency. In this section, we first define label set $\mathbb{L}_t$, and then introduce its theoretical property.

We obtain label set $\mathbb{L}_t$ by using the lower/upper bounds in each iteration. Formally, the label set in the t-th iteration, $\mathbb{L}_t$, is given as follows:

Definition 3 (Label set) *Letting $l_j \neq l_i$ and $t \neq 0$, label l_i is included in label set $\mathbb{L}_t$ if the following condition holds for data point x such that $x \in \mathbb{X}$:*

$$(1)\ \exists l_j \ \text{ s.t. } \ \underline{f}_{t-1}(x|l_i) \leq \overline{f}_{t-1}(x|l_j), \ \text{ and}$$
$$(2)\ \forall l_j \in \mathbb{L}, \ \ \overline{f}_{t-1}(x|l_i) \geq \underline{f}_{t-1}(x|l_j)$$

If $t = 0$, the label set $\mathbb{L}_t$ is initialized as $\mathbb{L}$, i.e., $\mathbb{L}_t = \mathbb{L}$.

We introduce the following two lemmas to describe the property of label set $\mathbb{L}_t$:

Lemma 9 (Labeled data) *If we have $\underline{f}_t(x|l_i) > \overline{f}_t(x|l_j)$ for all labels l_j such that $l_j \neq l_i$ and $l_j \in \mathbb{L}$, the label of data point x is determined as l_i by the optimal solution.*

Proof If $\underline{f}_t(x|l_i) > \overline{f}_t(x|l_j)$ holds for such label l_j, from Lemma 3 and 6, we have

$$f(x|l_j) \leq \overline{f}_t(x|l_j) < \underline{f}_t(x|l_i) \leq f(x|l_i)$$

Therefore, it is clear that $\max_{1 \leq k \leq c} f(x|l_k) = f(x|l_i)$. As a result, from Equation (3),

$$y(x) = \arg\max_{1 \leq k \leq c} f(x|l_k) = l_i$$

Therefore, the optimal solution labels data point x as l_i. □

Lemma 10 (Unlabeled data) *The optimal solution determines the label of data point x as not l_i if $\overline{f}_t(x|l_i) < \underline{f}_t(x|l_j)$ holds for a label l_j such that $l_j \neq l_i$ and $l_j \in \mathbb{L}$.*

Proof If $\overline{f}_t(x|l_i) < \underline{f}_t(x|l_j)$ holds for such label l_j, from Lemma 3 and 6, we have

$$f(x|l_i) \leq \overline{f}_t(x|l_i) < \underline{f}_t(x|l_j) \leq f(x|l_j)$$

Therefore, it is clear that $\max_{1 \leq k \leq c} f(x|l_k) \neq f(x|l_i)$. Consequently, from Equation (3),

$$y(x) = \arg\max_{1 \leq k \leq c} f(x|l_k) \neq l_i$$

This indicates that data point x is not labeled as l_i by the optimal solution. □

From Lemma 9 and 10, we introduce the following property of label set $\mathbb{L}_t$:

Lemma 11 (Label set) *If label l_i is included in label set $\mathbb{L}_t$ and $t \neq 0$, there exists data point x whose label is not determined as or as not l_i by the lower and upper bounds.*

Proof If $l_j \neq l_i$ and $t \neq 0$, from Lemma 9 and 10, there are the following two conditions under which data point x is not determined to have/not to have label l_i by the lower/upper bounds: (1) there exists a label l_j such that $\underline{f}_{t-1}(x|l_i) \leq \overline{f}_{t-1}(x|l_j)$, and (2) $\overline{f}_{t-1}(x|l_i) \geq \underline{f}_{t-1}(x|l_j)$ holds for all labels $l_j \in \mathbb{L}$. These two conditions are equivalent to those of the label set $\mathbb{L}_t$ as shown in Definition 3. Consequently, the statement of Lemma 11 holds. □

Lemma 11 validates our approach with the property to output the same labeling results as the optimal solution.

3.4. Labeling algorithm

Algorithm 1 is the full description of our approach. We initially set $t := 0$ and $\mathbb{L}_t := \mathbb{L}$ (lines 1-2). If $t = 0$, we compute the lower and upper bounds from Equation (6) and (9), respectively (lines 5-9). Otherwise, we incrementally update the lower/upper bounds to enhance the processing speed from Lemma 7 (lines 10-16). We then compute label set $\mathbb{L}_{t+1}$ from Definition 3 (lines 18-24). We iteratively repeat these procedures until no label remains in $\mathbb{L}_{t+1}$ (line 28). We finally determine the label of each node from the lower bounds (lines 29-31).

Note that, our algorithm does not require any user-defined inner-parameters. Moreover, it terminates the iterations automatically unlike the power method. Therefore, our approach provides to the user with a simple way to determine labels with enhanced processing speed.

Algorithm 1 Proposed algorithm

```
1:  $t := 0$;
2:  $\mathbb{L}_t := \mathbb{L}$;
3:  repeat
4:    for each label $l_i \in \mathbb{L}_t$ do
5:      if $t = 0$ then
6:        for each data point $x_j \in \mathbb{X}$ do
7:          compute propagation score $p_t(x_j|l_i)$ by Equation (5);
8:          compute the lower/upper bounds by Equation (6) and (9);
9:        end for
10:     else
11:       for each data point $x_j \in \mathbb{X}$ do
12:         update propagation score $p_t(x_j|l_i)$ by Equation (5);
13:         update the lower/upper bounds by Equation (12) and (13);
14:       end for
15:     end if
16:   end for
17:   $\mathbb{L}_{t+1} := \emptyset$;
18:   for each data point $x_i \in \mathbb{X}$ do
19:     for each label $l_j \in \mathbb{L}_t$ do
20:       if $\exists l_k$ s.t. $\underline{f}_t(x_i|l_j) \leq \overline{f}_t(x_i|l_k)$ and $\forall l_k \in \mathbb{L}, \overline{f}_t(x_i|l_j) \geq \underline{f}_t(x_i|l_k)$ then
21:         add label $l_j$ to label set $\mathbb{L}_{t+1}$;
22:       end if
23:     end for
24:   end for
25:   if $\mathbb{L}_t \neq \emptyset$ then
26:     $t := t + 1$;
27:   end if
28: until $\mathbb{L}_t = \emptyset$
29: for each data point $x_i \in \mathbb{X}$ do
30:   $y(x_i) = \arg\max_{1 \leq j \leq c} \underline{f}_t(x_i|l_j)$;
31: end for
```

3.5. Theoretical analyses

We introduce theoretical analyses addressing labeling results and the computational cost of our approach.

Theorem 1 (Labeling results) *The labeling results of our approach are the same as those of the optimal solution.*

Proof We assume that we reach termination after t iterations. As shown in Algorithm 1, we determine the label of data point x as follows:

$$y(x) = \arg\max_{1 \leq k \leq c} \underline{f}_t(x|l_k)$$

In addition, we perform iterations until the label set contains no label. Therefore, let label l_i and l_j be $l_i, l_j \in \mathbb{L}$ and $l_i \neq l_j$, we have (1) $\forall l_j$, $\underline{f}_t(x|l_i) > \overline{f}_t(x|l_j)$ or (2) $\exists l_j$ such that $\overline{f}_t(x|l_i) < \underline{f}_t(x|l_j)$ after the iterations from Definition 3.

If $\underline{f}_t(x|l_i) > \overline{f}_t(x|l_j)$ holds $\forall l_j$, the label of data point x is determined as l_i by the optimal solution from Lemma 9. In addition, since $\underline{f}_t(x|l_j) \leq \overline{f}_t(x|l_j) < \underline{f}_t(x|l_i)$ holds from Lemma 3 and 6, we have

$$y(x) = \arg\max_{1 \leq k \leq c} \underline{f}_t(x|l_k) = l_i$$

Therefore, our approach also determines l_i as the label of data point x.

If we have $\overline{f}_t(x|l_i) < \underline{f}_t(x|l_j)$ for a label l_j, l_i is not determined as the label of data point x according to the optimal solution from Lemma 10. In addition, since $\underline{f}_t(x|l_i) \leq \overline{f}_t(x|l_i) < \underline{f}_t(x|l_j)$ holds from Lemma 3 and 6, we have

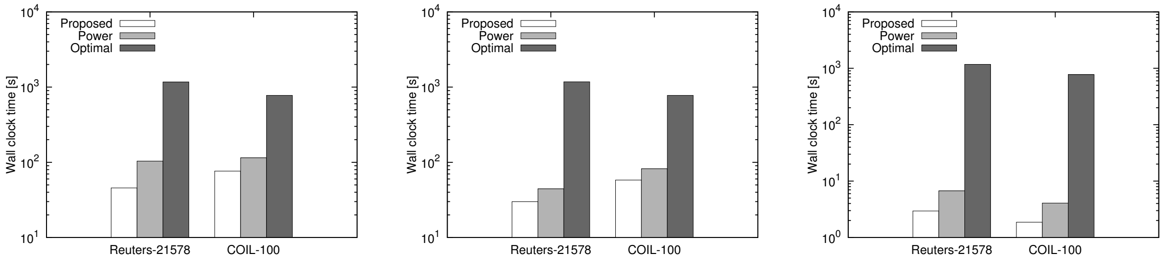
$$y(x) = \arg\max_{1 \leq k \leq c} \underline{f}_t(x|l_k) \neq l_i$$

As a result, our approach also determines the label of data point x as not l_i.

Consequently, the labeling results of the optimal solution and our approach are identical. □

Theorem 2 (Computational cost) *Our approach requires $O(cnt)$ time to obtain the labeling result.*

Proof For each label, our approach iteratively computes the propagation scores to obtain the lower/upper bounds. This process needs $O(nt)$ time since the number of edges in the graph is $O(n)$ as described in Section 2. The lower/upper bounds of data points in the iterations are obtained at $O(nt)$ time since it needs $O(1)$ time to compute the lower/upper bounds from the propagation scores as described in Lemma 7. As a result, it requires $O(cnt)$ time to obtain the lower/upper bounds from the propagation scores



(1) k-NN graph (2) linear neighborhood graph (3) sparse $\mathcal{L}^1$ graph

Figure 1. Labeling time of each approach.

in the iteration since the number of labels is c. In addition, our approach computes the label set from Definition 3 by using the lower/upper bounds, which needs $O(cnt)$ time. Consequently, our approach requires $O(cnt)$ time. □

3.6. Extension

In Section 3.1 to 3.5, we assume the use of k-NN graphs since it is the most popular graph structure for label propagation. In this section, we briefly describe the extension of our approach to handle other popular graph structures such as linear neighborhood graph (Wang & Zhang, 2008) and sparse $\mathcal{L}^1$ graph (Elhamifar & Vidal, 2011).

A linear neighborhood graph is constructed so that each node is represented as a linear combination of its local neighbor nodes, just like locally linear embedding (LLE) (Roweis & Saul, 2000). In a sparse $\mathcal{L}^1$ graph, nearest neighbors of each data points and corresponding edge weights are obtained by solving an $\mathcal{L}^1$-norm sparse optimization problem. Note that these graphs can have a negative edge weight. The major change of our approach is to compute the lower/upper bounds for these graph structures. More specifically, the bounds are computed as follows:

Definition 4 (Lower/upper bounds) *The following equations give the lower bound $\underline{f}_t(x_i|l)$ and the upper bound $\overline{f}_t(x_i|l)$ for linear neighborhood graph and sparse $\mathcal{L}^1$ graph:*

$$\underline{f}_t(x_i|l)=(1-\alpha)\textstyle\sum_{\tau=0}^{t}\{\alpha^\tau p_\tau(x_i|l)\}+\alpha^{t+1}\,\underline{p}_t(l)$$

$$\overline{f}_t(x_i|l)=(1-\alpha)\textstyle\sum_{\tau=0}^{t}\{\alpha^\tau p_\tau(x_i|l)\}+\alpha^{t+1}\left\{p_t(x_i|l)+\frac{\delta_t\overline{S}(x_i)}{1-\alpha}\right\}$$

While we omit the details of the above definition, it can be derived from the property of these graph structures such that $\sum_{x_j\in\mathbb{X}}S_{ij}=1$. The next section evaluates the labeling speed of our approach for linear neighborhood graphs and sparse $\mathcal{L}^1$ graphs as well as k-NN graphs.

4. Experimental evaluation

We performed experiments to compare the proposed approach to the optimal solution and the power method in terms of efficiency and effectiveness. The experiments used the following standard datasets.

Table 2. Number of average iterations in each approach.

Dataset	Approach	Graph		
		k-NN	linear	$\mathcal{L}^1$
Reuters-21578	Proposed	435.3	427.5	62.9
	Power	990.2	633.6	143.7
COIL-100	Proposed	588.1	646.3	112.7
	Power	879.3	913.5	247.1

- Reuters-21578 [1]: This dataset contains documents released by the Reuters newswire. Documents with multiple category labels were discarded. As a result, it contained $8,293$ documents of 65 categories. tf-idf was used as the document feature; it has $18,933$ dimensions.
- COIL-100 [2]: This dataset contains images of 100 objects; the number of object labels is 100. Images of the objects were taken at pose intervals of 5 degrees; 72 poses per object resulting in $7,200$ images. We resized all images to 32×32 and used RGB pixel values as the feature vector, resulting in $3,048$ dimensions.

In this section, "Proposed", "Power", and "Optimal" represent the results of the proposed approach, the power method, and the optimal solution, respectively. The results of the optimal solution are obtained by computing the inverse matrices. Following previous papers (Zhou et al., 2003; Xu et al., 2011), we set $\alpha = 0.99$ and stop iterating the power method when the residual drops below 10^{-4}. We conducted the experiments for the linear neighborhood graph and sparse $\mathcal{L}^1$ graph as well as k-NN graph, where 100 nearest neighbors were used to construct each graph [3]. In the experiments, 10 data points in each category/object were initially labeled. All experiments were conducted on a Linux 2.70 GHz Intel Xeon sever.

4.1. Efficiency

We evaluated the labeling time of each approach. Figure 1 shows the results. In addition, Table 2 details the number of

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[2] http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php

[3] We set the parameter $\lambda = 10$ on sparse $\mathcal{L}^1$ graph.

Table 3. Precision against the optimal solution.

Dataset	Approach	Graph		
		k-NN	linear	$\mathcal{L}^1$
Reuters-21578	Proposed	1.000	1.000	1.000
	Power	0.725	0.723	0.812
COIL-100	Proposed	1.000	1.000	1.000
	Power	0.899	0.982	0.980

Table 4. Classification accuracy.

Dataset	Approach	Graph		
		k-NN	linear	$\mathcal{L}^1$
Reuters-21578	Optimal	0.744	0.597	0.603
	Proposed	0.744	0.597	0.603
	Power	0.595	0.538	0.547
COIL-100	Optimal	0.533	0.891	0.902
	Proposed	0.533	0.891	0.902
	Power	0.531	0.889	0.900

average iterations needed by the proposed approach and the power method. In this table, "k-NN", "linear", and "$\mathcal{L}^1$" represent the results of each approach for k-NN graph, linear neighborhood graph, and sparse $\mathcal{L}^1$ graph, respectively.

Figure 1 shows that our approach is much faster than the previous approaches for all the types of graphs. Our approach is up to 410 and 2.3 times faster than the optimal solution and the power method, respectively. Since the optimal solution requires matrix inversion to obtain the labeling scores, it needs $O(n^3+cn^2)$ time as described in Section 2. On the other hand, our approach avoids computing the matrix inversion; it iteratively computes the lower/upper bounds to determine the labels in $O(cnt)$ time (Theorem 2). The power method also exploits iterative computation in a similar way to our approach. However, the power method computes the labeling scores for all the labels while our approach updates the lower/upper bounds only for selected labels. Furthermore, we terminate the iterations without waiting for convergence if no label remains to be updated (Algorithm 1). Therefore, as shown in Table 2, our approach needs fewer iterations than the power method. As a result, our approach has better labeling speed than the previous approaches.

4.2. Effectiveness

One major advantage of our approach is that it outputs the same labeling results as the optimal solution. The power method can obtain the exact labeling scores if it performs iterations until convergence. However, in practice, iterations are terminated to enhance the labeling speed, i.e., it approximately computes the labeling scores.

We evaluated the precision of the labeling results by our approach and the power method against the optimal solution. In this experiment, precision is the fraction of labeling results of an approach that match the labeling results of the optimal solution. Precision takes a value between 0 and 1, and, precision is 1 if the labeling results are identical to those of the optimal solution. Table 3 indicates the precision of each approach. In addition, Table 4 shows classification accuracy of each approach for ground-truth labels.

Table 3 shows that, as expected, the precision of our approach is 1 under all conditions examined. This is because our approach has the theoretical property that the labeling results of our approach are same as those of the optimal solution as shown in Theorem 1. In contrast, the power method has precision under 1; the power methods and the optimal solutions output different labeling results. This is because the power method terminates its iterative computation if the residual is less than the predetermined threshold. Precision is expected to improve if the threshold is set to a smaller score, however, this obviously reduces labeling speed. It is clear that setting the threshold forces a trade-off between precision and labeling speed.

As shown in Table 4, classification results of our approach is same as those of the optimal solution since our approach output the same labeling results as the optimal solution as shown in Table 3. Table 4 also indicates that the power method has lower classification accuracy than the optimal solution. As described in Section 2, the optimal solution gives the scores that minimize the cost function. Since the cost function is designed to improve classification accuracy, the optimal solution has high classification accuracy. However, the power method outputs different labeling results from the optimal solution as shown in Table 3. As a result, the power method has lower classification accuracy.

Table 4 along with Figure 1 indicates that the power method enhances labeling speed at the sacrifice of classification accuracy even though it is currently the standard approach to computing labeling scores. On the other hand, our approach achieves higher labeling speed than the previous approaches while its labeling results replicate those of the optimal solution. Furthermore, our approach does not require any inner-parameters to be set unlike the power method. This indicates that our approach is an attractive option for the research community in use in label propagation.

5. Conclusions

This paper proposed an efficient label propagation algorithm that gives the same labeling results as the optimal solution. Our approach computes lower and upper bounds of the labeling scores to prune unnecessary score computations. Experiments show that our approach can achieve high efficiency without sacrificing accuracy unlike the power method. Our approach can improve the effectiveness of future label-propagation-based applications.

References

Alexandrescu, Andrei and Kirchhoff, Katrin. Graph-based Learning for Phonetic Classification. In *ASRU*, pp. 359–364, 2007.

Belkin, Mikhail, Niyogi, Partha, and Sindhwani, Vikas. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2007.

Chapelle, Olivier, Scholkopf, Bernhard, and Zien, Alexander. *Semi-Supervised Learning*. The MIT Press, 2010.

Chen, Jie, ren Fang, Haw, and Saad, Yousef. Fast Approximate *k*NN Graph Construction for High Dimensional Data via Recursive Lanczos Bisection. *Journal of Machine Learning Research*, 10:1989–2012, 2009.

Connor, Michael and Kumar, Piyush. Fast Construction of k-Nearest Neighbor Graphs for Point Clouds. *IEEE Trans. Vis. Comput. Graph.*, 16(4):599–608, 2010.

Dong, Wei, Charikar, Moses, and Li, Kai. Efficient k-Nearest Neighbor Graph Construction for Generic Similarity Measures. In *WWW*, pp. 577–586, 2011.

Elhamifar, Ehsan and Vidal, René. Sparse Manifold Clustering and Embedding. In *NIPS*, pp. 55–63, 2011.

Fergus, Rob, Weiss, Yair, and Torralba, Antonio. Semi-supervised Searning in Gigantic Image Collections. In *NIPS*, pp. 522–530, 2009.

Golub, Gene H. and Loan, Charles F. Van. *Matrix Computations*. Johns Hopkins University Press, 2012.

Gunderson, David S. *Handbook of Mathematical Induction: Theory and Applications*. Chapman and Hall/CRC, 2010.

Kumar, Sanjiv, Mohri, Mehryar, and Talwalkar, Ameet. Sampling Techniques for the Nystrom Method. In *AISTATS*, pp. 304–311, 2009.

Roweis, Sam T. and Saul, Lawrence K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326, 2000.

Subramanya, Amarnag and Bilmes, Jeff A. Entropic Graph Regularization in Non-parametric Semi-supervised Classification. In *NIPS*, pp. 1803–1811, 2009.

Talwalkar, Ameet, Kumar, Sanjiv, and Rowley, Henry A. Large-scale Manifold Learning. In *CVPR*, 2008.

von Luxburg, Ulrike. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Wang, Fei and Zhang, Changshui. Label Propagation through Linear Neighborhoods. *IEEE Trans. Knowl. Data Eng.*, 20(1):55–67, 2008.

Xu, Bin, Bu, Jiajun, Chen, Chun, Cai, Deng, He, Xiaofei, Liu, Wei, and Luo, Jiebo. Efficient Manifold Ranking for Image Retrieval. In *SIGIR*, pp. 525–534, 2011.

Yu, Kai and Yu, Shipeng. Blockwise Supervised Inference on Large Graphs. In *Proc. of the 22nd ICML Workshop on Learning*, 2005.

Zhou, Dengyong, Bousquet, Olivier, Lal, Thomas Navin, Weston, Jason, and Schölkopf, Bernhard. Learning with Local and Global Consistency. In *NIPS*, 2003.

Zhu, Xiaojin, Ghahramani, Zoubin, and Lafferty, John D. Semi-supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML*, pp. 912–919, 2003.