Pattern Recognition Letters 45 (2014) 17–25

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Label propagation through minimax paths for scalable semi-supervised learning ☆

Kye-Hyeon Kim, Seungjin Choi *

Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), 77 Cheongam-ro, Nam-gu, Pohang 790-784, Republic of Korea

ARTICLE INFO

Article history:
Received 24 July 2013
Available online 13 March 2014

Keywords:
Label propagation
Minimax path
Semi-supervised learning

ABSTRACT

Semi-supervised learning (SSL) is attractive for labeling a large amount of data. Motivated from *cluster assumption*, we present a path-based SSL framework for efficient large-scale SSL, propagating labels through *only a few important paths* between labeled nodes and unlabeled nodes. From the framework, *minimax paths* emerge as a minimal set of important paths in a graph, leading us to a novel algorithm, *minimax label propagation*. With an appropriate stopping criterion, learning time is (1) *linear* with respect to the number of nodes in a graph and (2) *independent* of the number of classes. Experimental results show the superiority of our method over existing SSL methods, especially on large-scale data with many classes.

1. Introduction

Collecting a large amount of data has been increasingly easier and cheaper than before, while assigning class labels still requires expensive human efforts. When labels are only given for a small number of "labeled data", semi-supervised learning (SSL) [1] is attractive for labeling the remaining "unlabeled data" automatically. Not only learning from insufficient label information, SSL further exploits intrinsic cluster or manifold structures from plenty of unlabeled data.

We consider semi-supervised multi-class classification on a *partially-labeled sparse graph*: nodes correspond to data points; edges connect pairs of sufficiently close data points; class labels are given for a few "labeled nodes". Many graph-based SSL methods have been proposed (e.g., [2–5]), but their large computational costs limit applicability to real-world problems.[1]

In this paper, we present an efficient path-based SSL method: labels of labeled nodes are propagated into unlabeled nodes through only a few important paths lying in high-density regions, so-called *minimax paths*, thereby reducing the computational cost significantly. When *cluster assumption* holds [8], i.e., points connected via paths through high-density regions tend to have the same label, labels are propagated within the same cluster, not further into different clusters, so that our method can perform robust classification.

We briefly introduce existing work on graph-based SSL (Section 2), and propose our path-based SSL framework (Section 3). From the framework, minimax paths emerge as a minimal set of paths lying in high-density regions, leading to our *minimax label propagation* algorithm that propagates labels through only minimax paths (Section 4). Compared to existing SSL methods, some important contributions of our work are as follows (Section 5):

1. With an appropriate stopping criterion, our method requires $\mathcal{O}(N)$ time and space for a graph of N nodes and C classes.
2. The computational cost is *independent of the number of classes*.

That is, our method is good for large-scale data with many classes. On a large graph of $N = 10^6$ nodes and $C = 10^4$ classes, our method ran several orders of magnitude faster than existing SSL methods, while achieving comparable classification performance (Section 6).

2. Semi-supervised classification on partially-labeled graphs

We briefly introduce graph-based SSL. A dataset, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, forms a graph, $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, such that each node in $\mathcal{G}$ corresponds to each data point $\boldsymbol{x}_i$, and the nodes are linked by edges, $\mathcal{E} = \{(i,j)\}$. Each node belongs to one of C classes, $\mathcal{C} = \{1, \ldots, C\}$, but the true class labels, $\mathcal{Y} = \{y_i \in \mathcal{C}\}_{i=1}^N$, are known for only the first N_l ($\ll N$) *labeled nodes*, $\{\boldsymbol{x}_i\}_{i=1}^{N_l}$. Graph-based SSL predicts labels for the remaining *unlabeled nodes*, $\{\boldsymbol{x}_i\}_{i=N_l+1}^{N}$.

☆ This paper has been recommended for acceptance by F. Tortorella.

* Corresponding author. Tel.: +82 54 279 2259; fax: +82 54 279 2299.
E-mail addresses: fenrir@postech.ac.kr (K.-H. Kim), seungjin@postech.ac.kr (S. Choi).

[1] On a sparse graph of N nodes and C classes, (1) $\mathcal{O}(CN^2)$ time in [2] to solve C max-flow problems by recent algorithms, e.g., [6]; (2) $\mathcal{O}(CN^2)$ time in [3,4] to solve C sparse linear systems by iterative methods, e.g., [7]; (3) $\mathcal{O}(N^3)$ time in [5] due to a dense kernel matrix included in linear systems. All those methods require $\mathcal{O}(CN)$ space to store the solutions.

http://dx.doi.org/10.1016/j.patrec.2014.02.020
0167-8655/

SSL is commonly performed on a *K-nearest neighbor (K-NN) graph*, connecting two nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (i.e., $(i,j) \in \mathcal{E}$) only if they are K-NN of each other in the dataset. K is usually small (5-20), and it bounds the number of edges incident to a node, so-called *degree*. Not only for computational efficiency by retaining sparsity, k-NN graphs also prevent incorrect information propagation between semantically unrelated nodes [9].

SSL performs prediction such that *two nodes linked in $\mathcal{G}$ are likely to have the same predicted label*. Let a real-valued vector $\boldsymbol{f}_i \in \mathbb{R}^C$ represent "soft assignments" of each node $\boldsymbol{x}_i$ to C classes. Also, let a binary 1-of-C-coding vector $\boldsymbol{y}_i \in \{0,1\}^C$ represent a "hard assignment" of a labeled node $\boldsymbol{x}_i$ to its true class label y_i, such that $[\boldsymbol{y}_i]_c = 1$ if $c = y_i$ and $[\boldsymbol{y}_i]_c = 0$ otherwise. [3] proposed minimizing an objective function,

$$E(f) = \sum_{(i,j)\in\mathcal{E}} w(i,j)\left\|\boldsymbol{f}_i - \boldsymbol{f}_j\right\|^2 \quad \text{s.t.} \quad \boldsymbol{f}_i = \boldsymbol{y}_i \quad \text{for} \quad i = 1,\ldots,N_l, \tag{1}$$

where $w(i,j)$ denotes the similarity between two data points, defined as $w(i,j) = \exp(-\beta\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2)$ for some $\beta > 0$. After minimization, predicted labels can be chosen as $f_i = \arg\max_{c=1,\ldots,C}[\boldsymbol{f}_i]_c$ for unlabeled nodes.

Minimizing Eq. (1) (or similar variants, e.g., [4]) requires large computational cost: $\mathcal{O}(CN^2)$ time and $\mathcal{O}(CN)$ space. Various approximation methods have been proposed for improving scalability. Approximation by M ($\ll$N) centroids requires $\mathcal{O}(M^2CN)$ time and $\mathcal{O}(MCN)$ space, where M determines the trade-off between the scalability and the approximation quality [10–13]. In manifold learning, similar techniques using Nyström approximation were proposed [14,15]. Other approaches for scalable SSL include dividing each dimension of data into grid [16], enforcing a small number of support vectors through sparsified constraints [17], and aggregating predictions along each dimension [18].

3. Path-based SSL framework: L_p norm aggregation over paths

We propose a novel approach for SSL based on *paths* between nodes. First, we define a set of *all possible paths* between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in $\mathcal{G} = (\mathcal{X}, \mathcal{E})$:

$$\mathcal{A}_{ij} = \{\boldsymbol{a} = (a_0, a_1, \ldots, a_m) | m \geqslant 1,\ (a_\ell, a_{\ell+1}) \in \mathcal{E} \text{ for all } \ell,\ a_0 = i,\ a_m = j,\ a_1, \ldots, a_{m-1} \neq j\}. \tag{2}$$

A path $\boldsymbol{a} \in \mathcal{A}_{ij}$ connects $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ through consecutive edges, $(a_0,a_1),(a_1,a_2),\ldots,(a_{m-1},a_m)$. We assign an *intermediate cost* to each edge, defined as $c(a_\ell, a_{\ell+1}) = \left\|\boldsymbol{x}_{a_\ell} - \boldsymbol{x}_{a_{\ell+1}}\right\|$. Then we define *total cost* of a path as the L_p norm of the intermediate costs along the path:

$$\|c(\boldsymbol{a})\|_p = \left[\sum_\ell c(a_\ell, a_{\ell+1})^p\right]^{1/p} \tag{3}$$

and the *path-based similarity* as the sum of scores over all possible paths between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$s_{ij} = \sum_{\boldsymbol{a}\in\mathcal{A}_{ij}} \exp\left\{-\frac{1}{T}\|c(\boldsymbol{a})\|_p\right\}, \tag{4}$$

where the score of each path follows an exponential decay with the total cost.[2] Our SSL framework is to compute $\boldsymbol{f}_i$ for unlabeled nodes by aggregating $\boldsymbol{y}_j$ of labeled nodes, weighted by s_{ij} between them:

$$\boldsymbol{f}_i = \sum_{j=1}^{N_l} s_{ij}\boldsymbol{y}_j = \sum_{j=1}^{N_l}\sum_{\boldsymbol{a}\in\mathcal{A}_{ij}} \exp\left\{-\frac{1}{T}\|c(\boldsymbol{a})\|_p\right\}\boldsymbol{y}_j. \tag{5}$$

[2] Since we allow revisits of intermediate nodes, $\mathcal{A}_{ij}$ is an infinite set. However, Eq. (4) still converges in some cases, e.g., when $p = 1, s_{ij}$ becomes the same form as a partition function $\mathcal{Z}$ proposed in [19].

Some path-based (dis) similarity measures (e.g., [21,20,22]) have been derived from similar frameworks. They are known to capture underlying, arbitrary-shaped clusters in a dataset very well (e.g., Fig. 1(b)), but computing them requires to aggregate scores of *all* possible paths between nodes, $\mathcal{A}_{ij}$, leading to $\mathcal{O}(N^3)$ time and $\mathcal{O}(N^2)$ space. Now we show that our framework can *selectively* aggregate scores of a few paths that are important for capturing clusters, thereby improving efficiency as well as reflecting underlying clusters (e.g., Fig. 1(c)).

The key ingredients of our framework is two parameters, T and p, in Eq. (4). The decay constant, $0 < T < \infty$, represents a preference for paths of smaller total costs. As T decreases, the score of a path of larger total cost decays much faster. Thus, only a few paths (whose total costs are smaller than others) have significant effects on s_{ij}, whereas all the other paths become relatively negligible. In short, *smaller T reduces the number of important paths* for computing s_{ij}.

The L_p norm determines the importance of a path. As p increases, larger intermediate costs have greater effect on the total cost (Fig. 2(a) and (b)), so that a *"compact path"* that consists of short edges has smaller L_p norm (i.e., more important for computing s_{ij}) than a *"loose path"* containing long edges. The idea comes from *cluster assumption* [8], saying that points connected via paths through high-density regions tend to have the same label. Since a cluster is a high-density region whose data points are connected via compact paths, *larger p makes s_{ij} within the same cluster tend to be larger*. Fig. 2(c) shows an example.

Fig. 3 illustrates the scores of paths according to varying T and p:

1. As p increases (from left to right), compact paths in the high-density region have higher scores than loose paths, leading to *robust classification* by penalizing the effects of labeled nodes in different clusters.
2. As T decreases (from top to bottom), fewer paths have significant scores and all the other paths become relatively negligible, brining *efficiency gains* by reducing the number of effective paths for computation.

4. Minimax label propagation

Now we consider the extreme case, $T \to 0$ and $p \to \infty$. When $T \to 0$, every $\exp(\cdots)$ in Eq. (4) becomes negligible, except for the score of the *smallest* L_p norm:

$$s_{ij} \to \max_{\boldsymbol{a}\in\mathcal{A}_{ij}} \exp\left(-\frac{1}{T}\|c(\boldsymbol{a})\|_p\right) \tag{6}$$

and Eq. (5) also converges as

$$\boldsymbol{f}_i \to \boldsymbol{y}_{j^*} \text{ or simply } f_i = y_{j^*}, \text{ where } j^* = \arg\max_{j=1,\ldots,N_l} s_{ij}, \tag{7}$$

i.e., $\boldsymbol{f}_i \in \mathbb{R}^C$ is simplified to the *integer class label*, $f_i \in \{1,\ldots,C\}$, which is propagated from a labeled node (denoted by $\boldsymbol{x}_{j^*}$) through *only one path* whose L_p norm is smallest. Since s_{ij} takes only the lowest L_p norm, Eq. (7) can be rewritten in terms of a distance measure, denoted by $d_{ij}^{(p)}$:

$$f_i = y_{j^*}, \text{ where } j^* = \arg\min_{j=1,\ldots,N_l}\left(d_{ij}^{(p)} := \min_{\boldsymbol{a}\in\mathcal{A}_{ij}}\|c(\boldsymbol{a})\|_p\right). \tag{8}$$

When $p = 1, d_{ij}^{(p)}$ is the shortest path distance, computed along the *shortest path* between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (e.g., bottom-left panel in Fig. 3). When $p \to \infty, d_{ij}^{(p)}$ is the *largest intermediate cost*, $\|c(\boldsymbol{a})\|_\infty = \max_\ell c(a_\ell, a_{\ell+1})$, computed along the *most compact path* between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (e.g., bottom-left panel in Fig. 3). The path is also referred to as *"minimax path"* [23], and the cost is called *"minimax distance"*, denoted by d_{ij}:

$$d_{ij} = \min_{\boldsymbol{a}\in\mathcal{A}_{ij}}\|c(\boldsymbol{a})\|_\infty = \min_{\boldsymbol{a}\in\mathcal{A}_{ij}}\left(\max_\ell c(a_\ell, a_{\ell+1})\right). \tag{9}$$