Pattern Recognition 95 (2019) 191–200

Contents lists available at ScienceDirect

Pattern Recognition

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

Semi-supervised learning for hierarchically structured networks

Myungjun Kim, Dong-gi Lee, Hyunjung Shin*

Department of Industrial Engineering, Ajou University, Suwon 16499, South Korea

ARTICLE INFO

Article history:
Received 15 October 2018
Revised 9 April 2019
Accepted 15 June 2019
Available online 16 June 2019

MSC:
00-01
99-00

Keywords:
Hierarchical graph integration
Hierarchical networks
Hierarchically structured networks
Semi-supervised learning

ABSTRACT

A set of data can be obtained from different hierarchical levels in diverse domains, such as multi-levels of genome data in omics, domestic/global indicators in finance, ancestors/descendants in phylogenetics, genealogy, and sociology. Such layered structures are often represented as a hierarchical network. If a set of different data is arranged in such a way, then one can naturally devise a network-based learning algorithm so that information in one layer can be propagated to other layers through interlayer connections. Incorporating individual networks in layers can be considered as an integration in a serial/vertical manner in contrast with parallel integration for multiple independent networks. The hierarchical integration induces several problems on computational complexity, sparseness, and scalability because of a huge-sized matrix. In this paper, we propose two versions of an algorithm, based on semi-supervised learning, for a hierarchically structured network. The naïve version utilizes existing method for matrix sparseness to solve label propagation problems. In its approximate version, the loss in accuracy versus the gain in complexity is exploited by providing analyses on error bounds and complexity. The experimental results show that the proposed algorithms perform well with hierarchically structured data, and, outperform an ordinary semi-supervised learning algorithm.

1. Introduction

Semi-supervised learning (SSL) incorporates both labeled and unlabeled data in cases where labeled data are scarcely given compared to vast amounts of unlabeled data. Among the several approaches to SSL, the graph-based approach has received significant attention because of its success in machine learning and numerous practical applications. Graph-based SSL begins with constructing a graph (or network) with nodes and edges, each representing data points and similarities between them, respectively. The key concept behind the model is the 'label propagation' [1], where the influence of labels of nodes is propagated to neighboring unlabeled nodes through the edges. It is regarded as the 'smoothness' assumption, implemented by a graph Laplacian transformed from the similarity matrix. By minimizing an objective function composed of smoothness and loss terms, SSL predicts values of unknown labels. The theory of SSL is well-established, and it particularly clarifies difficult cases when there is only a little amount of labeled data. Nevertheless, applications of SSL are not limited for as long as a dataset can be represented in the form of a graph.

In relation to the following problem, multiple sources of data can exist. Consider an example from the protein function prediction. Nodes of proteins can be connected in several different ways based on heterogeneous types of multiple sources, such as such as protein-protein interactions, genetic interactions, and co-participation in a protein complex [2,3]. If the interest is on cancer survivability prediction, the sequencing results of patients, clinical treatment history, and electronic medical records can be sources of multiple graphs [4,5]. Because different graph sources are independent but contain complementary information, the total information can be enhanced by combining available graphs. In this regard, there have been extensive studies on the theory of multiple graph integration. In [2,3,6–9], the authors suggested various types of convex combination of graphs, whereas in [10–13], graphs are combined by incorporating the notion of multi-view learning. Nowadays, sources of data have become considerably diverse, to the extent that studies dealing with multiple data have raised more attention than ever before.

In the meantime, there are some exceptional cases in which the direct application of SSL is not appropriate, i.e., when the given multiple data are entangled in a certain hierarchical structure. Extending from an ordinary network, a hierarchically structured network consists of multiple layers of networks where two adjacent networks are connected. Within a layer, nodes are connected through the edges of the intralayer but between layers a new set of edges defines the interlayer connection. From such

* Corresponding author.
E-mail addresses: junkim930@ajou.ac.kr (M. Kim), ldg1226@ajou.ac.kr (D.-g. Lee), shin@ajou.ac.kr (H. Shin).
URL: http://www.alphaminers.net (H. Shin)

https://doi.org/10.1016/j.patcog.2019.06.009
0031-3203/

connections, the influence of labels in one layer can propagate to the nodes in the layer itself, as well as to neighboring layers, thereby providing external information. For instance, multi-omics data are rather bounded to each other in a hierarchical manner by obeying the central dogma from genome to proteome bypassing a series of more omics, such as epigenome and transcriptome [14]. If an era changes, the network also varies, but those of adjacent eras remain related in a chronological order. A series of human networks can be obtained from genealogies, which are arranged one after another through time, such as the network of ancestors followed by that of descendants [15]. Similar examples can be found for phylogenetics, sociology, etc. Recently, this notion was further extended to social network analysis: networks of bibliographical references of the first order, the second order, and so on, and networks of online interactions in successive time slots [16–18]. To take an instance from the financial domain, a network of individual stocks in a domestic market can be laid at a lower layer than that of global composite indicators, such as KOSPI, DJI, NASDAQ, WTI, exchange rates, and price of gold [19,20].

Recently, the hierarchical structure of networks has attracted many researchers' interest in network representation [21–23], network embedding [24,25], network dynamics [26,27], community detection [28–31], property diagnostics for hierarchically structured networks [22,32], and so on. Those work provide meaningful insights to observe, analyze, and represent the layered structure of networks. But if we add a certain function on the network so that it can provide inference or prediction via its hierarchical structure, it would benefit more pragmatic usage of peculiarities of the network structure. There have been some researches that further moved on from observation/diagnosis of the network structure. For instance, propagation of label information along with the edges on intra-layer or inter-layer of a hierarchical network: graph integration method [2,3,8] as a stopgap for hierarchical structure of financial networks [19,20], graph bipartization with co-linkage regularization [33], graph diffusion using Laplacian kernel for hierarchically structured networks [34]. Although the respective work contributes to seek an insight of the peculiar structure of the network, there still remain limitations. The first approach is only restricted to a couple of layers (i.e. domestic-global networks of financial indicators). On the other hand, in the second approach, the information flow from/to different layers is blocked therefore it cannot be regarded that it utilizes the structural benefit of the network. And in the third approach, the hierarchical structure of network is represented as Laplacian kernels which have strong points that it is convenient to incorporate many networks in the layered structure as a regular form of kernel. However, if network is represented as a kernel, it is required to compensate computational complexity since a kernel is a full-matrix. This may be a weak point of the approach since it has to confront scalability issues. Actually, it readily reaches N^3 to make a graph diffusion kernel where N is the number of nodes in a network [35,36], which seriously incurs computational burden when the layers of network are stacked.

One of the methods to boil down the problem is to keep graph Laplacian intact—not converting to a kernel matrix, and directly use it to take benefit of sparse connection of network. Graph-based SSL works so. However, there exist no solid method that can be applied to a hierarchically structured network. In this paper, we propose two versions of SSL frameworks for a hierarchically structured network—naïve version and its approximated version. The naïve version solves label propagation problem through graph Laplacian by means of well-established sparse matrix computation methods. In its approximated version, the solution is obtained via low rank approximation which further exploits benefit of sparsity of graph Laplacian. And accordingly, analysis on approximation error bounds is provided along with empirical experiments.

The remainder of the paper is organized as the follows. In Section 2, the graph-based SSL for a plain network is reviewed. In Section 3, the method of learning a hierarchically structured network under the framework of SSL is presented. The section also describes an analysis on the algorithm and some entailed limitations. In Section 4, experimental results are presented, and the conclusions of the study are discussed in Section 5.

2. Graph-based semi-supervised learning

In graph-based SSL [37], a dataset can be represented by a graph $G(V, E)$ that consists of nodes (V) and edges (E). Given a graph $G(V, E)$ for n data points, nodes represent data points with $V = \{x_1, x_2, \ldots, x_N\}$, and edges represent similarities between data points. The similarities are given by the weight matrix, W, where elements, W_{ij}, represent the strength of the connection between nodes x_i and x_j. For the label set $Y = \{Y_l, Y_u\}$, SSL labels $Y_l \in \{-1, 1\}$ for labeled nodes whereas $Y_u = 0$ for unlabeled ones. Usually, $n_u \gg n_l$, for n_u and n_l as the number of unlabeled and labeled data, respectively. Through the learning process, it determines the output vector $f = (f_1, f_2, \ldots, f_n)^T$ by minimizing the following quadratic objective functional [37]:

$$\min_f \ (f-y)^T(f-y) + \mu f^T L f \tag{1}$$

where L is the graph Laplacian [38] defined as $D - W$, with $D = diag(d_i)$ and $d_i = \sum_j W_{ij}$. In (1), the first term is the loss for consistency with actual labels for labeled nodes, and the second term is the smoothness for consistency with the geometry of the data. The parameter μ is for a trade-off between the two terms [37]. Because (1) is a convex problem, the analytical solution is easily calculated by its partial derivative with respect to f:

$$f = (I + \mu L)^{-1} Y \tag{2}$$

where I is the identity matrix.

3. Proposed method

3.1. Representation of hierarchically structured networks

Suppose there are K number of datasets that are hierarchically structured. Let $G(V, E, S)$ denote the layered graph, where V is the node set, E is the edge set, and S is the set of K layers, $S = \{S_1, S_2, \ldots, S_K\}$. With respect to K distinct layers, n_k stands for the number of data points for each layer, and hence, $N = n_1 + n_2 + \ldots + n_k$ in total. Then, the weight matrix, W, is an $N \times N$ block tri-diagonal matrix, where diagonal blocks represent individual intralayer edges, and banded diagonal blocks represent interlayer edges between adjacent layers. The weight matrix contains $3K - 2$ non-zero blocks with K diagonal blocks and $2K - 2$ rectangular banded diagonal blocks. Fig. 1(a) depicts a hierarchically structured network. Fig. 1(b) and (c) shows structures of the corresponding weight matrix and the graph Laplacian.

3.2. Naïve SSL for hierarchically structured networks

Before applying SSL to a hierarchically structured network, it should first be noted that there are two types of label propagations: through edges within a layer and those between layers. To exert control for different propagations, it is desirable to separate the total weight matrix, W, into two parts, intrarelations and interrelations. Let $W^{\{S_p, S_q\}}$ be a matrix for a sub-block of W associated with layer S_p and S_q, masking other blocks to zeros. Accordingly,

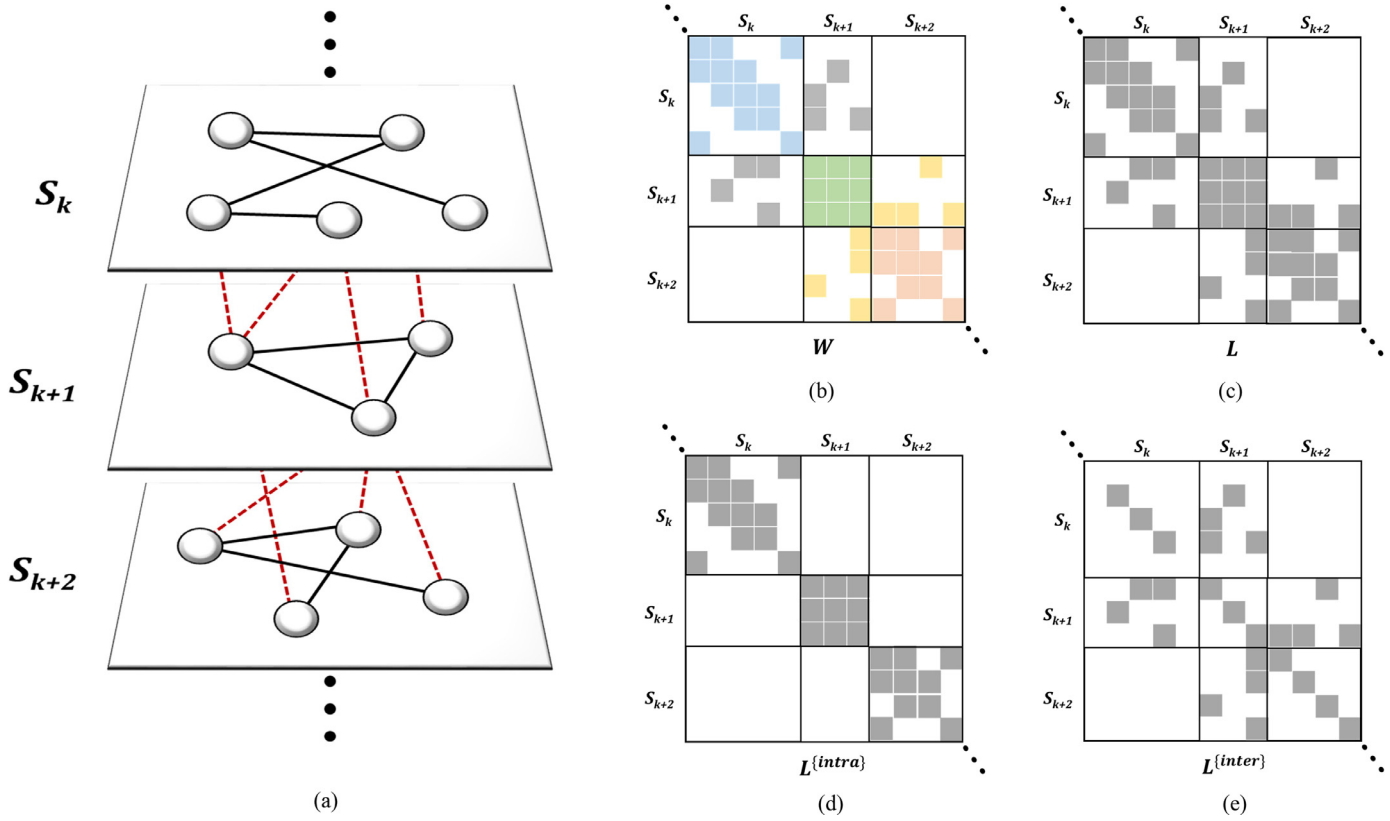


Fig. 1. (a) A hierarchically structured network with both intralayer and interlayer connections. (b) Corresponding block tri-diagonal structure of the weight matrix. (c) Structure of the graph Laplacian. (d) Structure of the graph Laplacian constructed with $W^{\{intra\}}$. (e) Structure of the graph Laplacian constructed with $W^{\{inter\}}$.

we have

$$\begin{aligned} W &= \sum_{S_p,S_q}^{K} W^{\{S_p,S_q\}} \\ &= \sum_{S_p=S_q}^{K} W^{\{S_p,S_q\}} + \sum_{S_p\neq S_q}^{K} W^{\{S_p,S_q\}} \\ &= W^{\{intra\}} + W^{\{inter\}}. \end{aligned} \tag{3}$$

Note that $S_p = S_q$ denotes $W^{\{S_p,S_q\}}$ as the weight matrix for the intralayer and $S_p \neq S_q$ for the interlayer relationship. The summation of the former becomes $W^{\{intra\}}$ with K diagonal blocks and that of the latter becomes $W^{\{inter\}}$, which consists of $2K-2$ banded diagonal blocks. Accordingly, the objective function for a hierarchically structured network parallel to (1) is defined as

$$\min_f \ (f-y)^T(f-y) + f^T\left(\mu_a L^{\{intra\}} + \mu_b L^{\{inter\}}\right) f, \tag{4}$$

where $L^{\{intra\}}$ and $L^{\{inter\}}$ are the graph Laplacians of corresponding weight matrices. Fig. 1(d) and (e) depicts their respective structures. Similarly, with (2), the solution is obtained as a closed form,

$$f = \left(I + \mu_a L^{\{intra\}} + \mu_b L^{\{inter\}}\right)^{-1} Y. \tag{5}$$

The parameters $\mu_a(\geq 0)$ and $\mu_b(\geq 0)$ are *smoothness-loss* trade-off parameters for intraconnections and interconnections, respectively. When $\mu_b = 0$,the equation reduces to (2).

3.3. An exact solution using matrix sparsity

The naïve solution of a hierarchically structured network involves a matrix inversion of a huge sized matrix. The computation of (5) can be highly resource demanding, but fortunately, the matrix to inverse is sparse because both $L^{\{intra\}}$ and $L^{\{inter\}}$are sparse block matrices. This means it can be efficiently solved by applying one of the well-established inversion methods for sparse matrices. Here, the Woodbury formula [39] was employed:

$$\left(A + UBU^T\right)^{-1} = A^{-1} - A^{-1}U\left(B^{-1} + U^TA^{-1}U\right)^{-1}U^TA^{-1}, \tag{6}$$

where is A is an $n \times n$ invertible matrix, B is an $m \times m$ invertible matrix, and U is an $n \times m$ rectangular matrix.

To rewrite the right-side term in (5) using the Woodbury formula, let $A = I + \mu_a L^{\{intra\}}$, B be the identity matrix, I. Moreover, let $L^{\{inter\}}$ be decomposed to UU^T, where U is the incidence matrix for the Laplacian. Here, U is an $n \times p$ matrix, where p is the number of edges. In the proposed method, elements of U are defined below with the weight w_{ij}:

$$U_{ik} = \begin{cases} \sqrt{\mu_b w_{ij}} & \text{if } v_i \sim v_j \text{ for } i > j \\ -\sqrt{\mu_b w_{ij}} & \text{if } v_i \sim v_j \text{ for } i < j \\ 0 & \text{otherwise.} \end{cases}$$

If $B^{-1} + U^TA^{-1}U$ is invertible, then, the Woodbury formula (6) works. That $I + U^TA^{-1}U$ is invertible is shown as follows.

Proposition. *Let $A = I + \mu_a L^{\{intra\}}$, and U be the incidence matrix of $L^{\{inter\}}$. Then, the matrix $I + U^T A^{-1}U$ is invertible.*

Proof. To prove $I + U^TA^{-1}U$ is invertible, it is sufficient to show that it is a positive definite (PD). We first show that A is PD. The $L^{\{intra\}}$ is a positive semi-definite and $\mu_a \geq 0$. By including an identity matrix, I, A becomes a PD. Therefore, its inverse A^{-1} is a PD. From this, we prove

$$x^T\left(I + U^TA^{-1}U\right)x = x^Tx + (Ux)^TA^{-1}(Ux) > 0,$$

for any non-zero vector $x \in \mathbb{R}^m$ because $x^Tx > 0$. Hence, $I + U^TA^{-1}U$ is invertible.

The solution of our naïve version (5) is rewritten as

$$f = A^{-1}Y - A^{-1}U\left(I + U^TA^{-1}U\right)^{-1}U^TA^{-1}Y. \tag{7}$$

It is interesting to see that (7) explicitly decomposes the solution (5) into two parts of intralayer and interlayer connections. The first term $A^{-1}Y$ in (7) is exactly the same as (2), which only accounts for the label propagation via intralayer connections. The remaining terms explain the effect through interlayer connections for prediction.

3.4. An approximated solution

Learning for a hierarchically structured network may come with great computational cost. In (5), the computational cost for the required inversion is $O(N^3)$ for the total of N data points. If layers are continuously stacked, then the matrix to the inverse becomes excessively large and will demand a substantial amount of computational time. If this is not the case, then it is advantageous to exploit an approximated solution because the hierarchically structured networks inherently demand greater amounts of resources than plain networks. In (7), the matrix A is huge but its inversion is easily obtained because it is a block diagonal matrix. Note that A is a combination matrix of an identity matrix and $L^{\{intra\}}$. For the block diagonal matrix, the inversion is used to dispose corresponding inversions of S_k blocks on its diagonal. However, the second term in (7) is considerably complicated because propagations through intraconnections and interconnections are entangled in the parentheses. As previously stated, the inversion for intraconnections is simple. Hence, $L^{\{inter\}}$ in (5) is further examined. The banded blocks in $L^{\{inter\}}$ are rectangular because the number of nodes in adjacent layers, S_k to S_{k+1}, are different. Thus, interconnections between them are represented as a rectangular matrix (see Fig. 1).

Although there are numerous inversion algorithms [34,40–43] for block tri-diagonal matrices, all of them are developed for "square-banded" diagonal block matrices, which results in the inapplicability of these algorithms to our problem. It may be recalled that the matrix under consideration does not necessarily have square banded diagonal blocks. Therefore, we employed the Nyström method [44], which is an efficient low rank approximation method applicable to any matrix with rectangular banded diagonal blocks provided it is positive semi-definite. The approximation is performed by randomly sampling (without replacement) N_c, which are the columns of the original matrix. The reconstructed matrix is calculated using the sampled column matrix. The Nyström approximation, $\hat{L}^{\{inter\}}$, is given by

$$\hat{L}^{\{inter\}} = CQ^{-1}C^T \approx L^{\{inter\}}, \tag{8}$$

where C is the sampled column matrix from $L^{\{inter\}}$, and Q is the intersection of C and its corresponding rows in $L^{\{inter\}}$. They are defined as

$$C = L^{\{inter\}}S \text{ and } Q = S^TL^{\{inter\}}S,$$

respectively, where S is a sampling matrix having dimension of $N \times N_c$, defined as

$$S = \begin{cases} S_{ij} = 1, & \text{if i}^{\text{th}} \text{ column is chosen for j}^{\text{th}} \text{ column of } C \\ S_{ij} = 0, & \text{otherwise.} \end{cases}$$

The Nyström method gives the benefit of reduced computational cost. The sampling size, N_c, is a user specified parameter that trades-off computational time and performance for precision. Moreover, there are several variants and extensions of the Nyström method depending on the manner of sampling: non-uniform sampling [45,46] or adaptive sampling [47,48]. Nevertheless, it is known that they may come at a greater computational cost, but with trivial performance increase compared to uniform sampling, which is typically used in practice [49].

By (8), the solution in (5) is approximated by

$$\hat{f} = \left(I + \mu_aL^{\{intra\}} + \mu_bCQ^{-1}C^T\right)^{-1}Y \tag{9}$$

Moreover, by matching the left-hand side of (6) with the right-hand side of (9), i.e.,B with μ_bQ^{-1}, and U with C, the approximated solution for sparsity (7) becomes

$$\hat{f} = A^{-1}Y - A^{-1}C\left(\frac{1}{\mu_b}Q + C^TA^{-1}C\right)^{-1}C^TA^{-1}Y. \tag{10}$$

3.5. Approximation error bounds

In this section, we provide the deviation bounds because of the approximation for predictive outputs, $\|f - \hat{f}\|$. From (5), (8), and (9), the deviation is incurred by the difference between two Laplacians, $E = L^{\{inter\}} - \hat{L}^{\{inter\}}$. Additionally, if we let $T = I + \mu_aL^{\{intra\}} + \mu_bL^{\{inter\}}$, the inversion in (9) can be rewritten as

$$\left(I + \mu_aL^{\{intra\}} + \mu_b\hat{L}^{\{inter\}}\right)^{-1} = (T - \mu_bE)^{-1}. \tag{11}$$

Technically, the deviation arises from the scaled difference, E, by the parameter μ_b. Taking the difference of inverses between (5) and (11), we have

$$\begin{aligned} T^{-1} - (T - \mu_bE)^{-1} &= T^{-1} - \left(T\left(I - \mu_bT^{-1}E\right)\right)^{-1} \\ &= T^{-1} - \left(I - \mu_bT^{-1}E\right)^{-1}T^{-1} \\ &= \left(I - \left(I - \mu_bT^{-1}E\right)^{-1}\right)T^{-1}. \end{aligned}$$

Observing that

$$I - \left(I - \mu_bT^{-1}E\right)^{-1} = -\mu_bT^{-1}E\left(I - \mu_bT^{-1}E\right)^{-1}$$

from $(I - \mu_bT^{-1}E)(I - \mu_bT^{-1}E)^{-1} = I$, we have

$$\begin{aligned} T^{-1} - (T - \mu_bE)^{-1} &= -\mu_bT^{-1}E\left(I - \mu_bT^{-1}E\right)^{-1}T^{-1} \\ &= -\mu_bT^{-1}E\left(T\left(I - \mu_bT^{-1}E\right)\right)^{-1} \\ &= -\mu_bT^{-1}E(T - \mu_bE)^{-1}. \end{aligned}$$

Taking the matrix norm on both sides yields

$$\begin{aligned} \left\|T^{-1} - (T - \mu_bE)^{-1}\right\|_p &= \mu_b\left\|T^{-1}E(T - \mu_bE)^{-1}\right\|_p \\ &\le \mu_b\left\|T^{-1}\right\|_p\left\|(T - \mu_bE)^{-1}\right\|_p\|E\|_p \end{aligned}$$

where $\|\cdot\|_p$ is the induced p-norm. Thus, the upper bound is affected by the column selection in the Nyström method, which is an inherent characteristic of both the original and approximated graphs, and the parameter μ_b, which controls the interlayer propagation.

In the case of $p = 2$, $\|\cdot\|_2$ denotes the largest singular value. Using this we have

$$\left\|T^{-1}\right\|_2 = \left(\frac{1}{\lambda_{min}(T)}\right) = 1$$

$$\left\|(T - \mu_bE)^{-1}\right\|_2 = \left(\frac{1}{\lambda_{min}(T - \mu_bE)}\right) = 1$$

where $\lambda_{min}(\cdot)$ denotes the smallest eigenvalue. Thus, with the induced two-norm, the bound becomes

$$\left\|T^{-1} - (T - \mu_bE)^{-1}\right\|_2 \le \mu_b\|E\|_2$$

in which the largest bound for the matrix inverse with the Nyström method is simply the induced two-norm of the deviation

scaled with the parameter μ_b. For the bounds on the predictive output, we have

$$\begin{aligned}\|f-\hat{f}\|_2 &= \|T^{-1}y-(T-\mu_b E)^{-1}y\|_2 \\ &\le \mu_b\|E\|_2\|y\|_2 \\ &= \mu_b\|E\|_2\sqrt{n_l}. \end{aligned} \tag{12}$$

By observing (12), the bound can be tight as $\mu_b \to 0$, $E \to 0$, or $n_l \to 0$. The first case indicates that there are no interlayer connections, whereas the second case occurs when the Nyström method gives an acceptable approximation for $L^{\{inter\}}$. Note that when the sampling size, N_c, equals N, then the deviation becomes zero. For the case $n_l \to 0$, it is advantageous to regard the proposed method as a robust approximation, particularly for SSL, because n_l is usually scarce in SSL ($n_u \gg n_l$).

3.6. Computational complexity

The primal advantage of the proposed method is the reduction in computational cost, which comes from the use of both the Woodbury formula and Nyström method. The former is known to be effective when obtaining A^{-1} is inexpensive and the entire matrix is sparse [50]. It is useful to see Fig. 1(b) and recall that $A = I + \mu_a L^{\{intra\}}$ from (6). In our method, because A is a block diagonal matrix, computing its inverse is inexpensive. Actually, the inverse computation is dominated by the largest layer. In addition, our matrix is block tri-diagonal and is therefore sparse, having $K^2 - 3K + 2$ zero blocks, where K is the number of layers as shown in Fig. 1(b). On the other hand, the latter approximation further reduces the complexity of $O(N^3)$ to $O(N_c \cdot N^2)$, and usually, $N_c \ll N$. Therefore, the overall cost is summarized as

$$O\left((\max\{n_1, n_2, \ldots, n_K\})^3 + N_c \cdot N^2\right)$$

where n_k denotes the size of layer S_k.

4. Experiments and results

The experimental results of the proposed method are presented in this section. The experiment consists of an artificial dataset and two real-world datasets. We compare the performance of two versions of the proposed method, naïve hierarchical SSL (NH-SSL) and approximate hierarchical SSL (AH-SSL), with that of plain networks, kernel-based SSL over multilayer graphs (KB-SSL) [34], and SSL with co-linkage regularization (CO-SSL) based on [33]. For CO-SSL, although it is not directly aimed for hierarchically structured networks, it is still applicable in a sense that it computes similarity based on co-linkage of interlayer relations between two nodes. Accordingly, the approximation and efficiency of the proposed approximation approach are shown.

4.1. Artificial data – performance comparison

To generate an artificial hierarchically structured network, the bottom layer is first constructed by drawing 200 data points from 10 different Gaussian distributions. The means and variances were randomly set ($-3 < \mu_i < 3$, $0 < \sigma < 5$). The half of the Gaussians belong to '+1' class, and '−1' class, otherwise. To pile up the succeeding layers, the similar procedure was repeated for 10 different Gaussians with random means and variances. The intralayer connection was calculated by the following formula

$$W_{ij} = e^{-\frac{\|x_i - x_j\|}{\alpha^2}}$$

with parameter α set as the median length of connected edges. The edges were removed by thresholding smaller ones than 0.8 × avg. W_{ij}. In order to create interlayer connections between adjacent layers, 20 data points were selected from the upper layer and connected to the k-nearest neighbors of the lower layer, belonging to the same class; k was set to 10. The number of layers varied over 3, 5, and 10. The parameters μ_a and μ_b in (5) and (9) were determined after validating all the combinations of (μ_a, μ_b) in the range of {0.01, 0.1, 1, 10, 100} × {0.01, 0.1, 1, 10, 100}. The number of sampled columns, N_c, in (9) was set as 20% of N. The results were obtained with different portions of labeled data points, 5%, 10%, 20%, 40%, 60%, and 80% for each layer. Thereafter, the two versions of the proposed method, NH-SSL and AH-SSL, were compared with plain networks, KB-SSL, and CO-SSL. The performance was measured by the area under receiving operating characteristic curve (AUC) [51], and the entire experiment was repeated 100 times.

Table 1
Performance comparison in terms of average AUC.

(a) 3 Layers

% of labels	NH-SSL	AH-SSL	KB-SSL	CO-SSL	Plain network
5	0.60	0.59	0.57	0.56	0.53
10	0.64	0.62	0.60	0.58	0.55
20	0.66	0.65	0.62	0.60	0.57
40	0.69	0.68	0.64	0.63	0.59
60	0.70	0.69	0.65	0.64	0.60
80	0.71	0.70	0.66	0.65	0.61

(b) 5 Layers

% of labels	NH-SSL	AH-SSL	KB-SSL	CO-SSL	Plain network
5	0.63	0.61	0.58	0.56	0.53
10	0.66	0.64	0.61	0.57	0.55
20	0.70	0.68	0.64	0.59	0.56
40	0.73	0.72	0.68	0.63	0.59
60	0.74	0.73	0.69	0.64	0.60
80	0.75	0.74	0.70	0.65	0.61

(c) 10 Layers

% of labels	NH-SSL	AH-SSL	KB-SSL	CO-SSL	Plain network
5	0.73	0.72	0.72	0.64	0.60
10	0.76	0.75	0.74	0.67	0.63
20	0.78	0.77	0.75	0.70	0.66
40	0.79	0.78	0.77	0.73	0.67
60	0.80	0.79	0.78	0.74	0.67
80	0.81	0.80	0.78	0.75	0.68

The overall results are shown in Table 1. First, it is seen that hierarchically structured networks achieved better performance that plain networks. The pairwise comparison of AH-SSL/NH-SSL with the plain network showed a statistically significant difference (both p-values are less than 0.0001). Meanwhile, when we compare the proposed method with other methods (in third and fourth column), both NH-SSL and AH-SSL achieves higher average AUC values. This conveys that the proposed method takes hierarchical structure of the network more effectively than other relevant algorithms. It is also notable to see that both comparing algorithms outperform the plain network, which portrays the benefit of using hierarchically structured networks.

Fig. 2 presents the individual AUC comparison for 100 experiments for 3, 5, and 10 layers. A point in the scatter plot located above the diagonal line means that the algorithm on the vertical axis performs better. The figure shows that most of the dots lie slightly above the diagonal line. This implies that in comparison between AH-SSL and NH-SSL, although it is observed that the AUCs of the former were slightly higher than those of the latter, the gap is trivial (p-value = 0.23). Thus, AH-SSL is compatible with NH-SSL in practice, particularly when execution time becomes a critical factor.

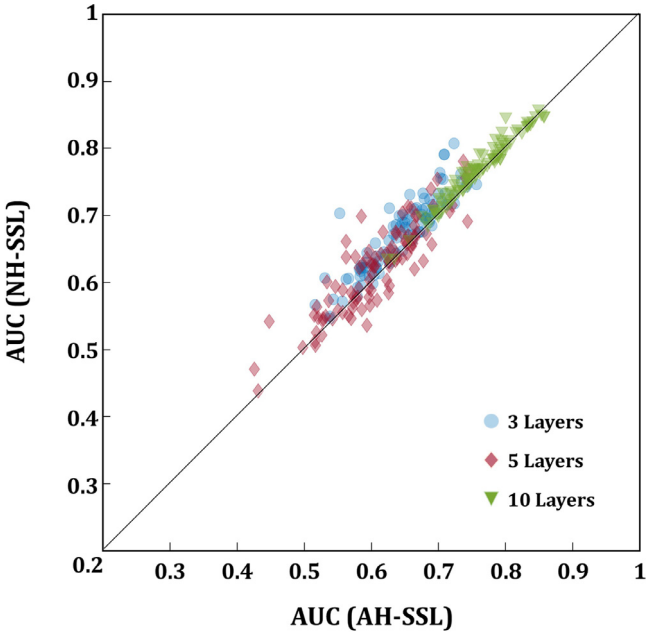


Fig. 2. Individual AUC comparison for NH-SSL and AH-SSL. Dots above diagonal line indicate higher AUC on the *y*-axis. Most of the dots lie slightly above the diagonal line.

Table 2
Computation time comparison between NH-SSL and AH-SSL.

Number of layers	NH-SSL (10^{-4})	AH-SSL (10^{-4})
2	2.58	39
5	10	19
10	30	45
20	250	87
50	2400	450
100	12,400	1600
200	71,200	6300
500	1,216,300	128,500

4.2. Artificial data – computation time comparison

The computational time (in seconds) of AH-SSL was compared with that of NH-SSL. We randomly generated block tri-diagonal weight matrices for 2, 5, 10, 20, 50, 100, 200, and 500 layers, each with 25–50 nodes.

Table 2 summarizes the computational time comparison between the two versions of the proposed method, and Fig. 3 depicts the results. The blue line represents NH-SSL, and red, AH-SSL. The gap in computational time increases with respect to the increase in the number of layers. Approximately, a one-fold magnitude difference was observed for 500 layers. This suggests that AH-SSL has the advantage of scalability.

4.3. Real-world problem I: disease co-occurrence prediction

The AH-SSL was applied to the prediction problem of co-occurring diseases. The hierarchical network consists of two layers of 1015 diseases and 319 symptoms. Table 3 summarizes the data with their sources. The disease layer was constructed by using information on shared proteins between diseases. Technically, a disease x_i is represented as a bit vector sized as 15,777 dimensions, with each bit indicating the existence of a relation between a protein and the disease. To calculate the intralayer relation of

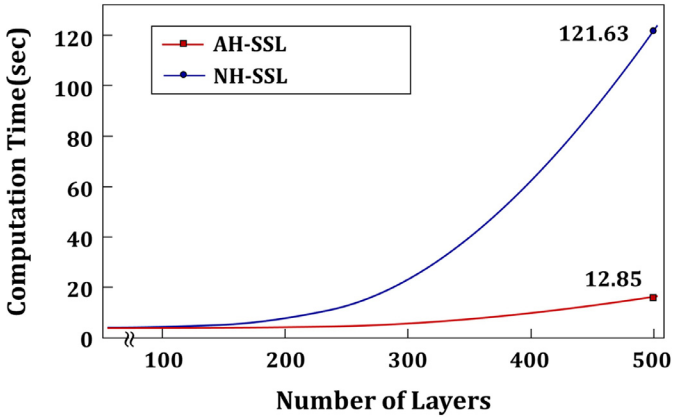


Fig. 3. Computational time comparison between NH-SSL and AH-SSL. The gap between the two methods increases with the increase in the number of layers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the disease layer, the Tanimoto similarity was used between two disease vectors, x_i and x_j:

$$W_{ij} = \frac{x_i \cdot x_j}{\|x_i\|^2 + \|x_j\|^2 - x_i \cdot x_j}.$$

The symptom layer was similarly built; 319 symptoms were represented as 1015 dimensional vectors. Each bit of the vector indicates diseases accompanying the symptom. The intralayer relation of the symptom layer was similarly calculated by using the Tanimoto similarity between two symptom vectors. For connections between the two layers, we simply used the presence/absence (1/0) of symptoms reported for a disease. Noting that a symptom can appear in multiple diseases, and vice versa, the connection can either be an *n*-to-1 or a 1-to-*n* mapping. The density of the Laplacian matrix, L, is 22%, where 17% is from $L^{\{intra\}}$ and 5% is from $L^{\{inter\}}$. Fig. 4(a) depicts the network of symptoms and diseases.

The prediction performance was measured on the disease layer. With the fixed labeled data portion in the bottom layer, the intervention from the top layer was controlled; 20% of the disease were labeled and fixed, and the portion of employed symptoms were varied from 0 to 100% at an interval of 20%. Note that a 0% symptom disease network means the plain network. Optimal values of μ_a and μ_b were searched over {0.01, 0.1, 1, 10, 100} × {0.01, 0.1, 1, 10, 100}. Approximately, 10% of N was set as the value of N_c. The performance of the proposed method was compared with that of KB-SSL, and CO-SSL, in which experiment was repeated 30 times.

The overall results are shown in Table 4. For plain network and CO-SSL, there is only one AUC value as it is invariant to the number of labels in different layers. By examining the second and third column, every increase in the intervention portion from the symptom layer resulted in higher AUC for the proposed method compared to plain network. Specifically, 20% of the intervention had already led to an improved AUC than that without intervention and the full intervention led the network to attain up to 0.74 AUC. Moreover, both NH-SSL and AH-SSL achieved higher AUC than KB-SSL and CO-SSL from the 20%, and 60% intervention, respectively. Note that even one additional layer laid on the plain network improved performance in this experiment. This result consolidates the advantages of using hierarchically structured network Table 4.

Table 3
Summary of data: symptom-disease hierarchical network.

Data	Number of data	Sources
Symptom-Disease	319 symptoms/1015 diseases	Supplementary information in [52]
Disease-Protein	1015 diseases/15,777 proteins	CTD, GAD, OMIM, PharmGKD, TTD
Disease Prevalence	1015 diseases	HuDiNe

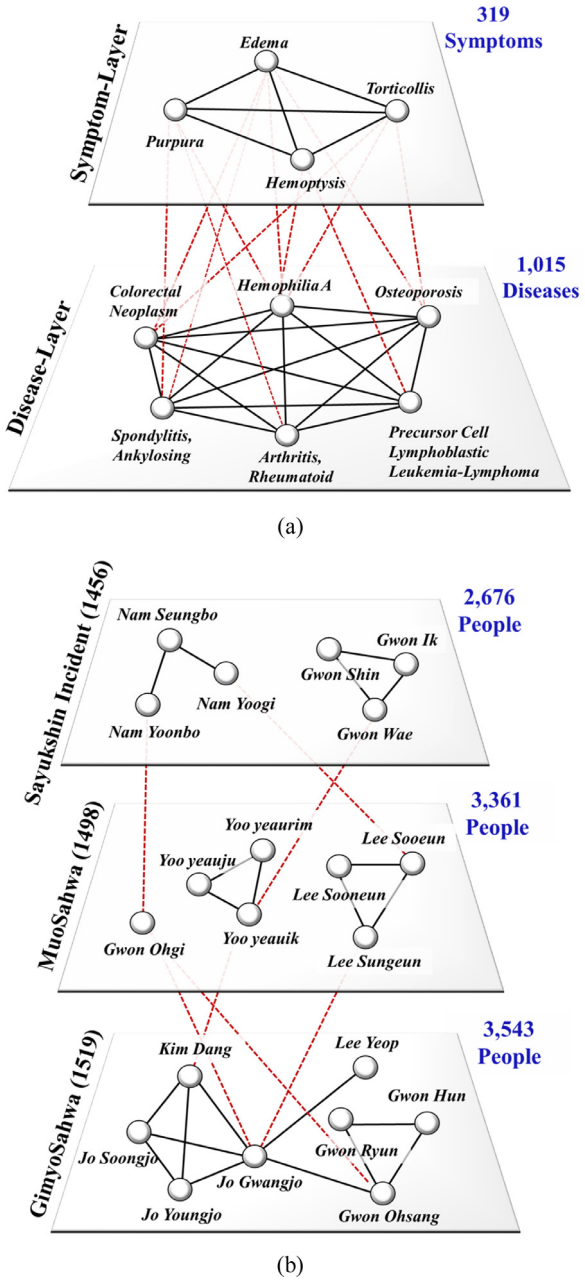


Fig. 4. Network diagrams for real-world datasets: (a) Two-layered hierarchically structured network of symptoms and diseases. (b) Three-layered hierarchically structured network consisting of people in contemporary era of political purges in medieval Korea.

4.4. Real-world problem II: political party classification

One of the interesting applications of the proposed method may be a study on history. An era/regime of a certain dynasty can be represented as a human network of contemporary people. Moreover, the network is interconnected to human networks of ancestors or offspring. In our experiment, the proposed method was applied to the classification of politicians during tragic purges, such as Sayukshin Incident (1456), MuoSahwa (1498), and GimyoSahwa (1519), in medieval Korea. The network was composed of three layers arranged in chronological order, each of historical figures during those periods. Table 5 summarizes the data and their sources. For both intralayer and interlayer connections, ancient literature on genealogy was used. The degree of kinship, k_{ij}, between persons i and j was converted to the similarity W_{ij} by

$$W_{ij} = \begin{cases} \frac{2}{1+e^{k_{ij}}}, & \text{if } k_{ij} < 9, \\ 0, & \text{otherwise.} \end{cases}$$

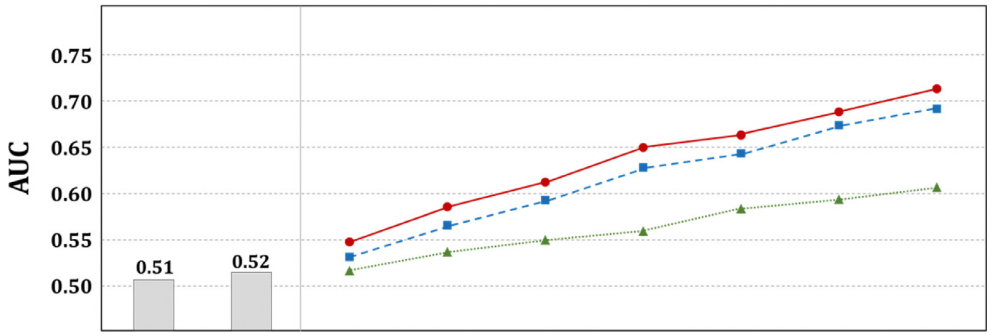
The edge has higher values for close relatives and lower values for far relatives. The experiment was repeated 100 times. For each of the purges, $p = 5\%$, 10%, 20%, 40%, 60%, 80%, 100% of labels were assigned to three other layers. For instance, if we were to measure performance on GimyoSahwa (the bottom) layer, labels on other two layers (Sayukshin Incident, and MuoSahwa) were varied according to the percentages. Then, 5-fold cross validation AUC was calculated as a performance measurement. The combinations of (μ_a, μ_b) were determined in the range of {0.01, 0.1, 1, 10, 100} $\times$ {0.01, 0.1, 1, 10, 100} and the number of sampled columns in (9), N_c, was set as 10% of N. For comparison, performances of plain network, KB-SSL, and CO-SSL were also measured.

Table 4
Performance comparison for disease co-occurrence prediction.

% of labeled symptoms	NH-SSL	AH-SSL	KB-SSL	CO-SSL	Plain network
20	0.63	0.62	0.57		
40	0.67	0.64	0.60		
60	0.69	0.68	0.63	0.67	0.59
80	0.72	0.70	0.65		
100	0.74	0.73	0.68		

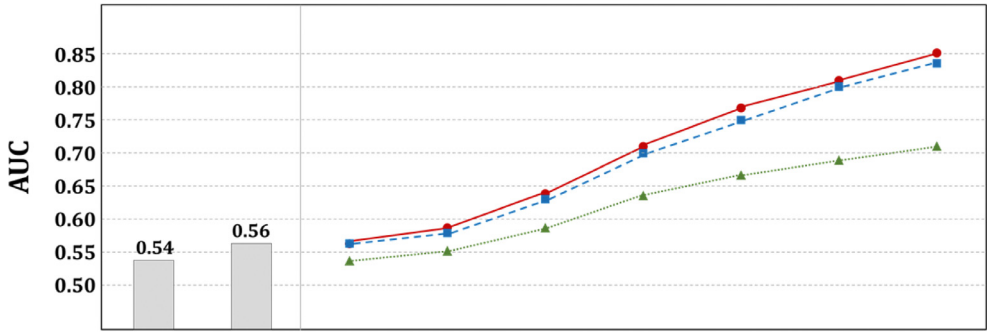
The overall AUC values for each of the purges are shown in Fig. 5. For plain network and CO-SSL, there is only single average AUC value, since the value does not vary with respect to the number of labels in other layers. For single networks and CO-SSL, AUC values are not too much distant from simple random guessing (AUC = 0.5). Although problems on the study of history are not easily predictable, the results indicate the 'no need to use' the prediction algorithm. On the other hand, when ancestor networks are employed, the performance increased up to an average of 0.85 AUC, in the case of MuoSahwa. These results indicate the potential use of the hierarchical network on such difficult problems.

Fig. 6(a) presents the individual AUC comparison. Most of the dots are above the diagonal line. The NH-SSL is marked with red triangles and AH-SSL is marked with blue circles. From the figure, the proposed method outperformed KB-SSL. Similarly, from Fig. 6(b), most dots align slightly above the diagonal line, indicating that AUC values of AH-SSL are lower than, but similar to the values of NH-SSL. This consolidates the compatibility of AH-SSL on historical data.



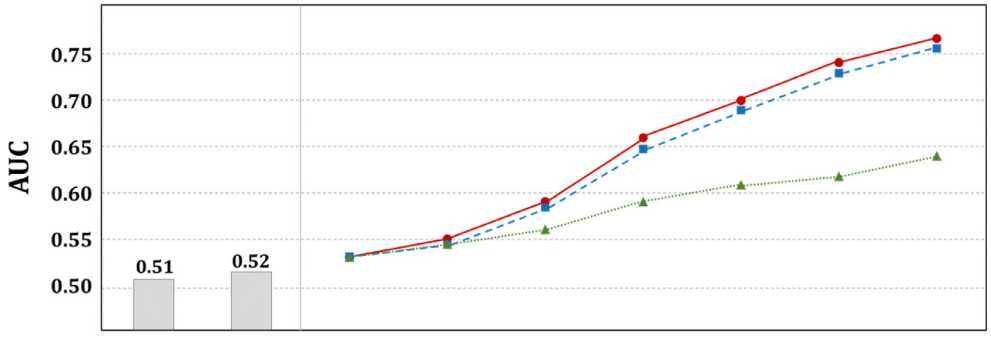
Plain	CO-SSL	5%	10%	20%	40%	60%	80%	100%
NH-SSL		0.55	0.58	0.62	0.63	0.66	0.68	0.71
AH-SSL		0.53	0.57	0.59	0.62	0.64	0.67	0.70
KB-SSL		0.52	0.54	0.55	0.56	0.58	0.59	0.61

(a) Sayukshin Incident



Plain	CO-SSL	5%	10%	20%	40%	60%	80%	100%
NH-SSL		0.56	0.58	0.64	0.71	0.77	0.81	0.85
AH-SSL		0.56	0.57	0.63	0.70	0.75	0.80	0.83
KB-SSL		0.54	0.55	0.58	0.64	0.67	0.69	0.71

(b)MuoSahwa



Plain	CO-SSL	5%	10%	20%	40%	60%	80%	100%
NH-SSL		0.53	0.55	0.59	0.66	0.70	0.74	0.77
AH-SSL		0.53	0.54	0.58	0.65	0.69	0.73	0.76
KB-SSL		0.53	0.54	0.56	0.59	0.61	0.62	0.64

(c) GimyoSahwa

Fig. 5. Overall performance comparison for three purges, (a) Sayukshin Incident (b) MuoSahwa, and (c) GimyoSahwa. The results show the better performance of using hierarchically structured networks.

Table 5
Summary of data: hierarchical network of historical figures.

Data	Number of data	Sources
Historical Figures	9580 People	Genealogy of the Andong Gwon [53] and Munhwa Yoo [54] clans
Political Parties	355 People	The Annals of Joseon Dynasty (http://sillok.history.go.kr)

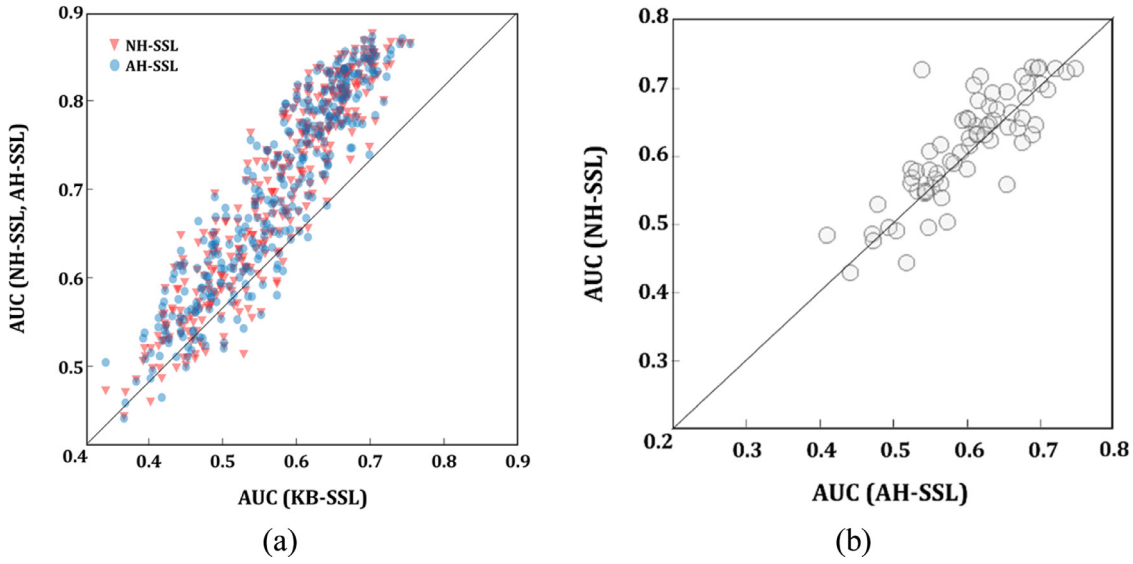


Fig. 6. (a) AUC for the NH-SSL and AH-SSL against KB-SSL. NH-SSL is marked with red triangles and AH-SSL is marked with blue circles. (b) AUC for the AH-SSL against NH-SSL. The left figure shows a better performance of the proposed methods compared to KB-SSL. The right figure consolidates the compatibility of AH-SSL, with most dots aligning slightly above the diagonal line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion

A hierarchically structured network benefits from the propagation of labels from one layer to another, providing additional information for predictions of interest. Such cases arise in several domains of study, including bioinformatics, history, financial domain, and social networks.

In this paper, we presented two versions of SSL algorithms for the hierarchically structured network—the naïve and approximated versions. The naïve algorithm solves the label propagation problem by facilitating the existing method for matrix sparseness. However, the structure, which stacks layers upon layers, incurs not only high sparsity, but also computational complexity and scalability. Hence, to resolve these difficulties, an approximation version was proposed. The approximation algorithm can reduce computational complexity and is therefore scalable. As most approximation algorithms tradeoff complexity and accuracy, the proposed approximation version similarly sacrifices a certain amount of accuracy. First, the error bound of the algorithm was analyzed, and thereafter, its complexity was presented. The approximation version reduces to the naïve version when its hyper-parameter is specified. The foregoing experiments on artificial and real-world problems validated the proposed algorithms. Moreover, the overall results indicate that in terms of predictions, the hierarchically structured network always outperforms the ordinary network. It was observed that the gap in accuracy between the naïve and approximation versions appears trivial.

It is regarded that the contribution of this study is related to its consideration of interconnections beyond intraconnections. It is significant that label information can be propagated to other networks in different layers by means of the proposed method.

However, a number of limitations, including future works, need to be addressed. First, in the approximation version, the sampling size 'N_c', which weighs on the loss of accuracy and reduction of complexity, is critical. Accordingly, considerable analyses and heuristics on the selection of its value should be further conducted. Second, it is required to further exploit additional domain problems. For instance, a check on whether the network-based time-series prediction is well suited to the proposed algorithm needs to be performed. In this regard, we believe that there are growing opportunities across diverse domains. Third, the integration of multiple hierarchical networks when multiple channels of data are given is reserved for our future study.

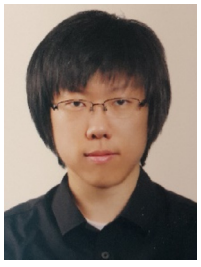
Acknowledgments

This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (no. 2018S1A5B6075104), Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (no. 2018-0-00440, ICT-based Crime Risk Prediction and Response Platform Development for Early Awareness of Risk Situation), and the Ajou University research fund.

References

[1] X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, Technical Report CMU-CALD-02-107, 2002.
[2] H. Shin, K. Tsuda, B. Schölkopf, Protein functional class prediction with a combined graph, Expert Syst. Appl. 36 (2009) 3284–3292.
[3] K. Tsuda, H. Shin, B. Schölkopf, Fast protein classification with multiple networks, Bioinformatics 21 (2005) ii59–ii65.
[4] D. Kim, J.-G. Joung, K.-A. Sohn, H. Shin, Y.R. Park, M.D. Ritchie, J.H. Kim, Knowledge boosting: a graph-based integration approach with multi-omics data and genomic knowledge for cancer clinical outcome prediction, J. Am. Med. Inform. Assoc. 22 (2014) 109–120.
[5] D. Kim, H. Shin, K.-A. Sohn, A. Verma, M.D. Ritchie, J.H. Kim, Incorporating inter-relationships between different levels of genomic data into cancer clinical outcome prediction, Methods 67 (2014) 344–353.
[6] A. Argyriou, M. Herbster, M. Pontil, Combining graph Laplacians for semi-supervised learning, Adv. Neural Inf. Process. Syst. 19 (2006) 67–74.
[7] V. Sindhwani, P. Niyogi, M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, in: Proceedings of ICML Workshop on Learning with Multiple Views, Citeseer, 2005, pp. 74–79.

[8] H. Shin, K. Tsuda, Prediction of protein function from networks, in: Semi-Supervised Learning, MIT Press, 2006, pp. 361–376.
[9] H. Shin, A.M. Lisewski, O. Lichtarge, Graph sharpening plus graph integration: a synergy that improves protein functional classification, Bioinformatics 23 (2007) 3217–3224.
[10] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational learning theory, ACM, 1998, pp. 92–100.
[11] M. Culp, G. Michailidis, K. Johnson, On multi-view learning with additive models, Ann. Appl. Stat. 3 (2009) 292–318.
[12] D. Zhou, C.J. Burges, Spectral clustering and transductive learning with multiple views, in: Proceedings of the 24th International Conference on Machine learning, ACM, 2007, pp. 1159–1166.
[13] G. Lin, K. Liao, B. Sun, Y. Chen, F. Zhao, Dynamic graph fusion label propagation for semi-supervised multi-modality classification, Pattern Recognit. 68 (2017) 14–23.
[14] K. Yugi, H. Kubota, A. Hatano, S. Kuroda, Trans-omics: how to reconstruct biochemical networks across multiple 'omic' layers, Trends Biotechnol. 34 (2016) 276–290.
[15] D.-g. Lee, S. Lee, M. Kim, H. Shin, Historical inference based on semi-supervised learning, Expert Syst. Appl. 106 (2018) 121–131.
[16] B. Oselio, A. Kulesza, A.O. Hero, Multi-layer graph analysis for dynamic social networks, IEEE J. Selected Topics Signal Process. 8 (2014) 514–523.
[17] P. Kazienko, P. Bródka, K. Musial, J. Gaworecki, Multi-layered social network creation based on bibliographic data, Social Computing (SocialCom), in: 2010 IEEE Second International Conference on, IEEE, 2010, pp. 407–412.
[18] A. Java, X. Song, T. Finin, B. Tseng, Why we twitter: understanding microblogging usage and communities, in: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, ACM, 2007, pp. 56–65.
[19] K. Park, H. Shin, Stock price prediction based on hierarchical structure of financial networks, in: International Conference on Neural Information Processing, Springer, 2013, pp. 456–464.
[20] K. Park, H. Shin, Stock price prediction based on a complex interrelation network of economic factors, Eng. Appl. Artif. Intell. 26 (2013) 1550–1561.
[21] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M.A. Porter, S. Gómez, A. Arenas, Mathematical formulation of multilayer networks, Phys. Rev. X 3 (2013) 041022.
[22] M. Kivelä, A. Arenas, M. Barthelemy, J.P. Gleeson, Y. Moreno, M.A. Porter, Multilayer networks, J. Complex Netw. 2 (2014) 203–271.
[23] P. Riba, J. Lladós, A. Fornés, Hierarchical graphs for coarse-to-fine error tolerant matching, Pattern Recognit. Lett. (2019).
[24] J. Lu, J. Xuan, G. Zhang, X. Luo, Structural property-aware multilayer network embedding for latent factor analysis, Pattern Recognit. 76 (2018) 228–241.
[25] S.F. Mousavi, M. Safayani, A. Mirzaei, H. Bahonar, Hierarchical graph embedding in vector space by graph pyramid, Pattern Recognit. 61 (2017) 245–254.
[26] S. Gomez, A. Diaz-Guilera, J. Gomez-Gardenes, C.J. Perez-Vicente, Y. Moreno, A. Arenas, Diffusion dynamics on multiplex networks, Phys. Rev. Lett. 110 (2013) 028701.
[27] S. Boccaletti, G. Bianconi, R. Criado, C.I. Del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, M. Zanin, The structure and dynamics of multilayer networks, Phys. Rep. 544 (2014) 1–122.
[28] C. De Bacco, E.A. Power, D.B. Larremore, C. Moore, Community detection, link prediction, and layer interdependence in multilayer networks, Phys. Rev. E 95 (2017) 042317.
[29] A. Tagarelli, A. Amelio, F. Gullo, Ensemble-based community detection in multilayer networks, Data Min. Knowl. Discov. 31 (2017) 1506–1543.
[30] X. Yang, W. Yu, R. Wang, G. Zhang, F. Nie, Fast spectral clustering learning with hierarchical bipartite graph for large-scale data, Pattern Recognit. Lett. (2018).
[31] Y. Yang, S. Han, T. Wang, W. Tao, X.-C. Tai, Multilayer graph cuts based unsupervised color–texture image segmentation using multivariate mixed student's t-distribution and regional credibility merging, Pattern Recognit. 46 (2013) 1101–1124.
[32] M. De Domenico, C. Granell, M.A. Porter, A. Arenas, The physics of spreading processes in multilayer networks, Nat. Phys. 12 (2016) 901.
[33] D. Zhou, T. Hofmann, B. Schölkopf, Semi-supervised learning on directed graphs, Adv. Neural Inf. Process. Syst. 18 (2005) 1633–1640.
[34] V.N. Ioannidis, P.A. Traganitis, Y. Shen, G.B. Giannakis, Kernel-based semi–supervised learning over multilayer graphs, in: 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2018, pp. 1–5.
[35] A.J. Smola, R. Kondor, Kernels and regularization on graphs, in: Learning Theory and Kernel Machines, Springer, 2003, pp. 144–158.
[36] S.V.N. Vishwanathan, N.N. Schraudolph, R. Kondor, K.M. Borgwardt, Graph kernels, J. Mach. Learn. Res. 11 (2010) 1201–1242.
[37] Y. Bengio, O. Delalleau, N. Le Roux, in: Label Propagation and Quadratic Criterion, Semi-Supervised Learning, MIT Press, 2006, pp. 183–206.
[38] F.R. Chung, Spectral graph theory, Am. Math. Soc. (1997).
[39] M.A. Woodbury, Inverting modified matrices, Mem. Rep. 42 (1950) 336.
[40] R.E. Bank, D.J. Rose, Marching algorithms for elliptic boundary value problems. I: the constant coefficient case, SIAM J. Numer. Anal. 14 (1977) 792–829.
[41] A.V. Terekhov, A fast parallel algorithm for solving block-tridiagonal systems of linear equations including the domain decomposition method, Parallel Comput. 39 (2013) 245–258.
[42] G. Meurant, A review on the inverse of symmetric tridiagonal and block tridiagonal matrices, SIAM J. Matrix Anal. Appl. 13 (1992) 707–728.
[43] N.M. Boffi, J.C. Hill, M.G. Reuter, Characterizing the inverses of block tridiagonal, block Toeplitz matrices, Comput. Sci. Discov. 8 (2014) 015001.
[44] C.K. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, Adv. Neural Inf. Process. Syst. 14 (2001) 682–688.
[45] P. Drineas, R. Kannan, M.W. Mahoney, Fast Monte Carlo algorithms for matrices II: computing a low-rank approximation to a matrix, SIAM J. Comput. 36 (2006) 158–183.
[46] P. Drineas, M.W. Mahoney, On the Nyström method for approximating a Gram matrix for improved kernel-based learning, J. Mach. Learn. Res. 6 (2005) 2153–2175.
[47] A.J. Smola, B. Schölkopf, Sparse greedy matrix approximation for machine learning, (2000).
[48] S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, J. Mach. Learn. Res. 2 (2001) 243–264.
[49] S. Kumar, M. Mohri, A. Talwalkar, Sampling methods for the Nyström method, J. Mach. Learn. Res. 13 (2012) 981–1006.
[50] W.W. Hager, Updating the inverse of a matrix, SIAM Rev. 31 (1989) 221–239.
[51] J.A. Swets, Signal Detection Theory and ROC Analysis in Psychology and Diagnostics: Collected Papers, Psychology Press, 2014.
[52] X. Zhou, J. Menche, A.-L. Barabási, A. Sharma, Human symptoms–disease network, Nat. Commun. 5 (2014) 4212.
[53] S. Lee, The impact of family background on bureaucratic reproduction in the thirteenth -to- fifteenth century Korea: a case study on the Kwon-ssi Sunghwabo, Daedong Munhwa Yeon'gu 81 (2013) 41–67.
[54] J. Choi, Genealogy and organization of the clan in the Chosôn dynasty, J. Korean Hist. 81 (1979) 37–79.



Myungjun Kim received M.S. degree from Ajou University in 2017 and is currently pursuing his Ph.D. degree at Graduate School of Industrial Engineering, Ajou University, South Korea. His research interest is on machine learning, especially semi-supervised learning, algorithms and applications for networks with multi-layered structure.

Dong-gi Lee received M.S. degree from Ajou University in 2016 and is currently pursuing his Ph.D. degree at Graduate School of Industrial Engineering, Ajou University, South Korea. His current research interest is on historical inference and biomedical informatics using various techniques of machine learning algorithms.

Hyunjung Shin received the Ph.D. degree in Data Mining from Seoul National University, and further majored in Machine Learning during her Post-Doc at Max Planck Institute (MPI) Tübingen in Germany. Since 2006, she joined Ajou University as a faculty member of the Department of Industrial Engineering. Currently, she provides academic services as a member of board of directors in Business Intelligence & Data Mining Society, Korean Institute of Information Scientists and Engineers (KIISE), Artificial Intelligence Society at KIISE, Korean Institute of Industrial Engineers, and Korean Society for Bioinformatics and Systems Biology. Also, she has been the vice chairman for Billing Software Inspection and Review Committee of Health Insurance Review and Assessment Service. Theory interest of her is focused on Machine Learning algorithms, particularly in Kernel and Semi-Supervised Learning methods. Her research activities range across diverse areas including network analytics, biomedical informatics, hospital fraud detection, etc.