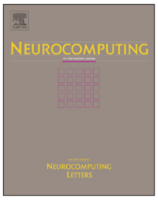
Contents lists available at ScienceDirect

Neurocomputing

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

ELSEVIER



An on-line sequential learning method in social networks for node classification

Yongjiao Sun*, Ye Yuan, Guoren Wang

Northeastern University, Shenyang 110004, China

ARTICLE INFO

Article history:
Received 6 August 2013
Received in revised form
19 April 2014
Accepted 22 April 2014

Keywords:
Extreme learning machine
Directed acyclic graph
OS-ELM
Node classification
Social network

ABSTRACT

Social networks have been a common platform for human interactions due to the rapid development of Internet. Along with the rising demand in network analysis, the issue of node classification have became an important research field. This article aims to address the task of node classification using an on-line sequential learning method with the role of links and node features. Compared to the conventional classification methods, we should not only use the node features for node classification, but also consider the interaction among the linked nodes. In this paper, we assume that the nodes have been partially labeled in a social network, and we use these labeled nodes to predict the categories of unlabeled nodes. Based on OS-ELM learning method, three node classification approaches are proposed. First, considering the influence of other nodes, we combine the linkage information with the features of the trained nodes to learn the classifier. Then, for reducing the learning time, we present a method to refine the node features. Finally, according to graph structures, we present an optimization node classification method. Extensive experiments were conducted to verify the performance of our proposed methods.

1. Introduction

With the emergence of online social networks and a vast increase in the volume of Internet information, the active users of many large-scale online social networks [19–21], such as Twitter, Facebook, microblog, increase rapidly. According to Facebook, there are more than 955 million active users in 2012. The development of social network motivates researches in social network analysis, in which node classification is a very important issue and has a great theoretical significance and application value [1–4].

In existing node classification methods, link-based classification is different from traditional content-based classification. Link-based classification uses not only node features, but also the information of network structure [22,23]. This kind of classification is not limited by the independent and identically distributed assumption, but permits existing various dependence relationships among nodes. Therefore, link-based classification is more suitable for many applications in real world. In this paper, we use both link information and node features to address node classification.

Machine learning is a standard method for classification. In this paper, we adopt OS-ELM [7,17] which is an online sequential extreme learning machine (ELM), to classify the nodes in a social network. ELM [8–11] is an easy-to-use and effective learning method for batch learning, while OS-ELM can train data one-by-one or chunk-by-chunk with varying or fixed chunk length.

Usually, the problem of node classification in social networks is divided into two cases. The first case is that the training set and the test set are in the same social network instance, while the data is partially labeled. In this case, we need to predict unlabeled data from labeled one [6,15,25,26]. The second case is that the training set and the test set are in different social network instances, while training set is fully labeled. For this case, we need to use fully labeled data to learn classifier in order to predict the labels of other social network instances [27,28]. In this paper, we mainly focus on the study of first case.

Specifically, in our problem, we assume that the structure of the social network is a directed acyclic graph (DAG) [5]. In DAG, the structure information can reflect the relationship among nodes. For example, in the DAG shown in Fig. 1, *a*, *b*, *c*, *d*, *e*, *f* and *g* are users in a social network, and they are connected with each other by interests. Node *e* has been labeled and nodes *a*, *b*, *c*, *d*, *f* and *g* are unlabeled. According to the link relationship among these nodes, we can easily find out the ancestor-descendant relationship (called A-D relationship) of node *e*, in which *a* and *b* are the ancestor nodes

* Corresponding author.
E-mail address: sunyongjiao@ise.neu.edu.cn (Y. Sun).

http://dx.doi.org/10.1016/j.neucom.2014.04.074
0925-2312/© 2014 Elsevier B.V. All rights reserved.

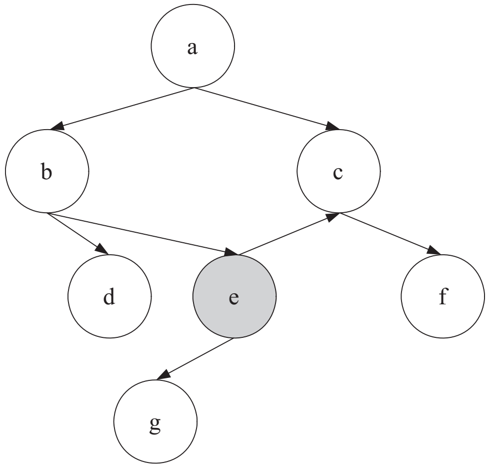


Fig. 1. Node relationship.

of node *e*, while *c*, *f* and *g* are descendant nodes. Therefore, except for node *d*, others are likely to affect the classification of node *e*. Based on this observation, in this paper, we first present a baseline node classification method (called BNC) for node classification using both A-D relationship and node features by OS-ELM. However, by analyzing the computational complexity of this method, we can know that it takes $O(D|V|)$ time to learn the classifier, where D denotes the number of training data, and $|V|$ denotes the total number of node objects. Next, we propose an improvement approach in order to reduce the computational cost. Conceptually, the baseline method needs to scan the global graph in each iteration for training. However, not all A-D nodes are able to generate a great influence on training data. Therefore, we propose an improved node classification method (called INC) by pruning little-influence nodes. Little-influence nodes, treated as little-influence samples in general purpose classification algorithm, may correspond to noise. Then, their removal will increase the classification error. Therefore, though INC is able to reduce the computation cost, it cannot guarantee an accurate classification. In order to obtain a higher accuracy for classification and less computation cost, we propose an optimization node classification method (called ONC) in the final. A process multiplexing approach is used to solve the problems of accurate classification and efficiency. Experimental results show that our algorithms boost the overall classification accuracy and reduce the running time.

The remainder of the paper is organized as follows. We formally define node classification in social networks and give a review of ELM and OS-ELM in Section 2. We propose a baseline node classification algorithm based on link information and node feature. After that, an improved algorithm and an optimization method are proposed in Section 3. We discuss the results of performance tests on real datasets in Section 4 and the conclusion of our work in Section 5.

2. Preliminaries and definitions

In this section, we first briefly review an on-line sequential learning machine OS-ELM that is used in our task. Then, we define some necessary concepts of our problem.

2.1. Review of OS-ELM

ELM is a very efficient learning method for classification and it has some salient features, such as powerful generalization performance and fast learning speed [12–14]. ELM solves the problem of traditional SLFNs learning method with lower learning speed for batch learning. Another learning method OS-ELM is developed from ELM, which is able to learn data one-by-one or chunk-by-chunk. The process of OS-ELM algorithm is shown as follows.

Given N arbitrary samples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbf{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbf{R}^m$, since standard SLFN with ℓ hidden nodes can approximate these N samples with zero error [9], the output of a SLFN with ℓ hidden nodes can be formalized by

$$f_{\ell}(\mathbf{x}) = \sum_{i=1}^{\ell} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}), \quad \mathbf{x} \in \mathbf{R}^n, \mathbf{a}_i \in \mathbf{R}^n \tag{1}$$

where $\mathbf{a}_i$ and b_i are the learning parameters of hidden nodes, β_i is the output weight, and $G(\mathbf{a}_i, b_i, \mathbf{x})$ is the output of the ith hidden node with respect to the input $\mathbf{x}$. As for additive hidden node, $G(\mathbf{a}_i, b_i, \mathbf{x}) = g(\mathbf{a}_i \cdot \mathbf{x} + b_i)$, $b_i \in R$, where $\mathbf{a}_i$ is the input weight, b_i is the bias of the ith hidden node and $\mathbf{a}_i \cdot \mathbf{x}$ is the inner product. As for RBF hidden node $G(\mathbf{a}_i, b_i, \mathbf{x}) = g(b_i \| \mathbf{x} - \mathbf{a}_i \|)$, $b_i \in R^+$, where $\mathbf{a}_i$ and b_i are the center and impact factor of ith RBF node. R^+ is the set of all positive real values.

After selecting the type of node, the activation function, and the hidden node number ℓ, OS-ELM can be implemented with data $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \dots, N\}$ arrived one-by-one or chunk-by-chunk. Therefore, two steps OS-ELM algorithm can be represented as follows.

Initialization phase: Given a training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \dots, N\}$, a small chunk of the training data $\aleph_0 = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N_0}$ is used to initialize the learning, where $N_0 \geq \ell$.

- For RBF hidden nodes, input weights is center $\mathbf{a}_i$ and impact factor is b_i, where $i = 1, \dots, \ell$; for additive hidden nodes, input weights $\mathbf{a}_i$ and bias b_i are assigned randomly.
- Calculating the initial output matrix $\mathbf{H}_0$.

$$\mathbf{H}_0 = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \dots & G(\mathbf{a}_N, b_N, \mathbf{x}_1) \\ \vdots & \cdots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_{N_0}) & \cdots & G(\mathbf{a}_N, b_N, \mathbf{x}_{N_0}) \end{bmatrix}_{N_0 \times \ell} \tag{2}$$

- Estimate the initial output weight $\beta^{(0)}$
- Set $k=0$, where k is the number of chunks that is trained currently.

Sequential learning phase: Present the $(k+1)$th chunk of new observations:

$$\aleph_{k+1} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=\left(\sum_{j=0}^{k} N_j\right)+1}^{\sum_{j=0}^{k+1} N_j} \tag{3}$$

and N_{k+1} denotes the number of observations in the $(k+1)$th chunk.

- Calculate the partial output matrix $\mathbf{H}_{k+1}$,

$$\mathbf{H}_{k+1} = \begin{bmatrix} G\left(\mathbf{a}_1, b_1, \mathbf{x}_{\left(\sum_{j=0}^{k} N_j\right)+1}\right) & \dots & G\left(\mathbf{a}_N, b_N, \mathbf{x}_{\left(\sum_{j=0}^{k} N_j\right)+1}\right) \\ \vdots & \cdots & \vdots \\ G\left(\mathbf{a}_1, b_1, \mathbf{x}_{\sum_{j=0}^{k+1} N_j}\right) & \cdots & G\left(\mathbf{a}_N, b_N, \mathbf{x}_{\sum_{j=0}^{k+1} N_j}\right) \end{bmatrix}_{N_{k+1} \times \ell} \tag{4}$$

- where $\mathbf{H}_{k+1}$ is calculated by the $(k+1)$th chunk of data $\aleph_{k+1}$. Set $\mathbf{T}_{k+1} = \left[\mathbf{t}_{\left(\sum_{j=0}^{k} N_j\right)+1}, \dots, \mathbf{t}_{\left(\sum_{j=0}^{k+1} N_j\right)}\right]^T_{N_{k+1} \times m}$.
- Calculate the output weight $\beta^{(k+1)}$.

$$\beta^{(k+1)} = \beta^{(k)} + \mathbf{K}_{k+1}^{-1} \mathbf{H}_{k+1}^T \left(\mathbf{T}_{k+1} - \mathbf{H}_{k+1} \beta^{(k)}\right) \tag{5}$$

From Eq. (5), we find that $\mathbf{K}_{k+1}^{-1}$ rather than $\mathbf{K}_{k+1}$ is used to compute $\beta^{(k+1)}$ from $\beta^{(k)}$. The update formula for $\mathbf{K}_{k+1}^{-1}$ is derived using the Woodbury formula [24]:

$$\mathbf{K}_{k+1}^{-1} = \mathbf{K}_k^{-1} - \mathbf{K}_k^{-1}\mathbf{H}_{k+1}^T\left(\mathbf{I}+\mathbf{H}_{k+1}\mathbf{K}_k^{-1}\mathbf{H}_{k+1}^T\right)^{-1} \times \mathbf{H}_{k+1}\mathbf{K}_k^{-1} \tag{6}$$

$\beta^{(k+1)}$ can be written by $\mathbf{P}_{k+1}$,
Then, $\beta^{(k+1)}$ can be obtained.

$$\beta^{(k+1)} = \beta^{(k)} + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^T\left(\mathbf{T}_{k+1} - \mathbf{H}_{k+1}\beta^{(k)}\right) \tag{7}$$

where $\mathbf{P}_{k+1} = \mathbf{K}_{k+1}^{-1}$.

- Set $k = k+1$. Go to the beginning of Sequential Learning Phase.

2.2. Problem definition

In this section, we give some necessary definitions and present the problem formulation. Since social networks can be represented as a graph structure, for explicitly depicting the problem of the issue, we assume that the graph structure is a DAG.[1] The definition of DAG is given as follows.

Definition 2.1. *DAG:* Let $\overrightarrow{G} = (V, \overrightarrow{E})$ denote a DAG, where V is a set of nodes and $\overrightarrow{E}$ is a set of directed edges. An arbitrary directed edge from node v_i to v_j is denoted by $\langle v_i, v_j \rangle$, where $v_i, v_j \in V$ and $\langle v_i, v_j \rangle \neq \langle v_j, v_i \rangle$.

Definition 2.2. *A-D relationship:* Given two nodes v_i and v_j, where v_i and v_j are in $\overrightarrow{G}$, if there is a path from v_i to v_j, then there exists an ancestor-descendant relationship between them, i.e. v_i is an ancestor of v_j.

In real social networks, many user nodes can label themselves to identify the occupation or interests. However, in many cases, it is difficult for each user to identify its own node, such as Bioinformatics and paper-coauthored networks. Therefore, we need to use partially labeled nodes as a training set to learn classifier for predicting the attributes of unlabeled nodes. Next, a partially labeled network is defined as follows.

Definition 2.3. *Partially labeled network:* Let $\overrightarrow{G} = (V_L, V_U, \overrightarrow{E}, T_L, W)$ denote a partially labeled network. V_L represents a set of labeled nodes and V_U represents a set of unlabeled nodes, where $V_L \cup V_U = V$. The categories of labeled nodes is represented by T_L, while W is a weight matrix associated with nodes in V.

In traditional content-based node classification, the solution always uses the node features to classify the unlabeled nodes. But, the classification accuracy is low without considering the influence of unlabeled nodes. In order to obtain a higher accuracy over node classification, in our approach, besides attributes of labeled nodes, we also take into account the A-D relations in the classification.

Definition 2.4. *Ancestor:* Given a node $v \in \overrightarrow{G}$, the ancestor nodes $\{P_i(v) | i = 1, 2, \ldots k\}$ represents the nodes that have paths to v. Specifically, i is the path length from $P_i(v)$ to v and k is the path length from the farthest root node to v, where $P_i(v)$ denotes the set of ancestors that have i distances to v.

According to Definition 2.4, we can compute the ancestor nodes which can affect the trained nodes. We consider these nodes, since they all have some relationships with trained nodes. Based on the above concepts, we can define the problem studied in this paper. For a partially labeled social network, our target is to discover the categories of all unlabeled nodes in a DAG. The formal definition of node classification in DAG is given as follows:

Definition 2.5. *Node classification:* Given a partially labeled network $\overrightarrow{G} = (V_L, V_U, \overrightarrow{E}, T_L, W)$, the corrected V_L by A-D relationship is

$$s : (V_L, P_i(v), NE_j(v)) \rightarrow V_l^* \tag{8}$$

where $\{P_i(v), NE_j(v)\} \in V_U$. The predictive function of node classification in DAG can be formulated with new labeled nodes V_l^* by

$$f_N : \overrightarrow{G} = (V_L^*, V_U, \overrightarrow{E}, T_L, W) \rightarrow T \tag{9}$$

3. The node classification algorithms

In this section, we will describe the different methods to solve the node classification problem against traditional approach. We start by describing a simple method, *BNC*, base on Definition 2.5 to learn classifiers.

3.1. BNC method

For a node classification issue, a straight forward method is to use the labeled node features to directly learn classifier by ELM. Then, using the trained classifier, we can predict categories of all the unlabeled nodes. However, the method has a low predictive accuracy due to not considering the influence of other nodes. Therefore, in this subsection, we will propose a classification method which considers A-D relationship based on OS-ELM.

As we know, the classification results are erroneous if we simply use node features for classification. In order to solve the problem, the BNC method uses the features of ancestor to correct the features of trained node, since the impacts of these features have a transitive relation. Considering the example of Fig. 1, b and d are the ancestor and neighbor of e respectively, where $x_b = [x_{b1}, x_{b2}, \ldots, x_{bn}]$, $x_e = [x_{e1}, x_{e2}, \ldots, x_{en}]$ and $x_d = [x_{d1}, x_{d2}, \ldots, x_{dn}]$ are the corresponding feature vectors. In DAG, if there exists a directed edge between two nodes, then we can confirm that there are links between attributes (user b follows user e in Twitter). The idea is inspired by the PageRank algorithm [16]. In PageRank approach, each Web page is regarded as a network node, and the links between Web pages are regarded as the directed edges. If there is an edge from node i to node j, i will contribute to j and the ranking of j will be uploaded. Based on PageRank approach, in our problem, if we assume that the link is from b to e, then we say that the contribution of b is transferred to e partially, denoted by $x_{bi} \rightarrowtail x_{ej}$.

Rule 1. *Let $\overrightarrow{G}$ be a DAG structure of a social network, while v_i and v_j are arbitrary two nodes in $\overrightarrow{G}$. If there exists a path from v_i to v_j, then all the nodes over the path can affect v_j.*

Rule 2. *Let $\overrightarrow{G}$ be a DAG structure of a social network, while v_i, v_j and v_k are arbitrary three nodes in $\overrightarrow{G}$. If there exists a path from v_i to v_j, while v_k is the neighbor of v_j, then v_k can gain partial contributions which belong to v_j from its parents.*

In DAG, if there is an edge between any two nodes, then the contribute transfer has been impacted. Considering the A-D relationship, we use path length to measure the loss and gain of each node contribution. Therefore, we select a universal loss function[2] for the correction of each node feature.

$$L(\mathbf{x}) = e^{-d(\mathbf{x})} \tag{10}$$

[1] DAG is a special case of complex graphic.

[2] Other loss functions can also be used in our problem.

where $d(\mathbf{x})$ is a distance vector in which each element denotes a distance between an unlabeled node to a labeled node.

Therefore, for any two adjacent nodes i and j, if the path between them is from i to j, then the formulation of corrected vector is

$$\mathbf{s}(\mathbf{x}_j) = \mathbf{x}_j + \mathbf{x}_i e^{-d(\mathbf{x}_{i,j})} \tag{11}$$

Since the contribution only happens between the same node attributes, for any trained node, Eq. (11) can be rewritten as

$$\mathbf{s}(\mathbf{x}) = \mathbf{x} \sum_{l=1}^{L} (1 + \sum e^{-d(\mathbf{x})}) \tag{12}$$

where L is the number of path and $\sum e^{-d(\mathbf{x})}$ is all the contribution from the ancestors of node $\mathbf{x}$ in a path.

Given these, we can give the BNC method in Algorithm 1.

Algorithm 1. BNC algorithm.

```
Input:
1:      Graph $\vec{G}$;
2:      Training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m,\ i = 1, \ldots, N\}$;
Output:
3:      The output weight $\beta$ of OS-ELM;
4:      $k = 0$;
5:      for each training data do
6:        Calculate the corrected vector $\mathbf{x}_i^*$.
7:        Calculate the hidden layer output matrix $\mathbf{H}_k$ about $\mathbf{x}_i^*$;
8:        Estimate the initial output weight $\beta^{(k)}$;
9:        $k = k + 1$;
10:     end for
```

The process of this method is similar to OS-ELM. The difference is that we need to calculate the corrected node features in each reachable path (line 7). Therefore, it takes $O(D|V|)$ time to learn the classifier. If $D = |V|$, then the computational complexity will be $O(|V|^2)$. The response time is extremely large and unacceptable. In order to reduce the computational costs, we propose an improvement method to solve the problem in the next subsection.

3.2. INC method

Algorithm 1 is to correct a trained node in each iteration and use the corrected node to learn the classifier for output weight β, while the process of corrected node needs to scan the whole reachable paths.

In order to reduce the response time, we should use an observation from the BNC method.

Observation 3.1. *When the path length reaches a definite value, the prediction results will not change even if the path length keeps increasing. According to this phenomenon, we can confirm that the impact of path length on classification accuracy has its limits.*

Therefore, we have the following pruning rules to reduce path lengths in the training.

Rule 3. *For an arbitrary node, the corrected vector is impacted by loss function. Therefore, the affect of loss function is becoming smaller with the increase of path length. When the path length approaches a certain length, the change of node feature is negligible.*

Based on Rule 3, we can obtain a new conclusion. The node vector, parameter $\mathbf{a}_i$ and b_i are the input values of OS-ELM, while β is the only output value. Thus we have

Rule 4. *The change of output weight β can be neglected if the node contribution is negligible.*

Considering the correction function s, we only need to find out a lower-bound of corrected vector when the change of a node contribution is very small, since the real values of corrected vector can be obtained by Eq. (12). Thus, based on Rule 4, we can know that the path length could be shorter. This is because further nodes have a lower influence (contribution). For instance, user b follows user e and user e follows user g in Twitter, in which user b directly affects the classification result of user e, while user b has a lower influence on user g.

According to above analysis, we make some adjustment for the path length. Let ψ represents an arbitrary reachable path and $v = v_1, v_2, \ldots, v_n$ are the nodes over ψ. For arbitrary two nodes v_i and v_j, the path length between them is denoted by $l_{i,j}$, where v_i and v_j are neither the root nodes nor leaf nodes associated with ψ. Assume that the relationship of contribution is $v_i \rightarrowtail v_j$ and v_j is a labeled node. In a classification, based on Rule 3, the neglected change of β is denoted by ϵ. The neglected change of path length δ can be obtained by the following formula:

$$\delta = arg\ \mathbf{s}(arg\ G(\epsilon)) \tag{13}$$

The new path length can be easily obtained as

$$l^* = l - \delta \tag{14}$$

where l is the a dimension of $d(\mathbf{x})$.

Theoretically, since ϵ is a posteriori error, ϵ can be set by experiments. However, the new path length may be different for each node. Practically, the new path length l^* may be longer if v_j have many ancestors. Thus, we cannot set the same value for each node. For example, path $\psi_{i,j}$ is one of the paths reaching v_j. If v_j have many ancestors, then the path length that we set may not satisfy ϵ. The way of overcoming this problem is that we need to extend $\psi_{i,j}$ for v_j. However, in our problem, we choose exponential function as loss function for ancestors. Therefore, we would like to extend the value of l^* to guarantee ϵ.

For obtaining a better response time, we have improved BNC method and proposed INC method. However, INC method is proposed based on experience. To an effective learning method, the process of path length selection should be automatic without human intervention. Therefore, in the next subsection, we will propose an optimization node classification method according to path information.

3.3. ONC method

In previous subsections, we illustrate details of our node classification methods, including BNC and INC. We can know that these two methods have different advantages. BNC method can obtain a higher classification accuracy, while INC can reduce the response time. If we can take the advantages of these two methods, then the performance of the node classification will be improved significantly. To this end, in this subsection, we will describe our optimization classification method.

First, we talk about an example for node classification based on BNC. Fig. 2 shows a DAG subgraph in which two nodes a and i are used to learn. In the learning process, a is the first node that joins in the learning sequence. BNC algorithm adopts the depth first search approach to compute the transferred contribution from the nodes with A-D relationships. In Fig. 2, ψ_1, ψ_2 and ψ_3 represent three relationship paths for a. However, we can easily know that there are partial overlapping paths, such as $\psi_{f,d}$, when node i joins in the classifier. The learning will generate unnecessary computation costs if we take the same strategy as a. For this case, we propose a two-step method to solve the problem.

- *Initialization:* The ONC method starts at the first labeled node that joins into classifier with the depth-first search. All the

Please cite this article as: Y. Sun, et al., An on-line sequential learning method in social networks for node classification, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.04.074

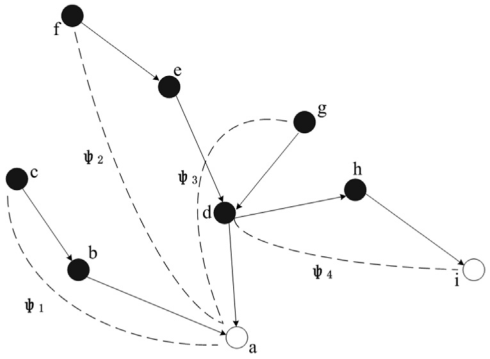


Fig. 2. Path selection.

nodes in the search are marked as its corresponding path length to indicate the layer in the search tree, and add a link that points to each node.

- *Update:* The ONC method firstly conducts the depth-first search when the next node joins into classifier. If there exists one path which links to the marked node, then according to the path information of the node contained, we can compute the remaining path length from the marked nodes. Finally, ONC will add the new partial path into the path tree.

Let us reconsider the example in Fig. 2. In the figure, ψ_1, ψ_2 and ψ_3 consist of the paths tree in the initialization step. In the update step, ψ_4 is also added into the path tree, while $\psi_{f,d}$ and $\psi_{g,d}$ are pruned by the ONC algorithm. Algorithm 2 describes the details of the ONC algorithm.

Algorithm 2. ONC algorithm.

```
Input:
    Graph $\vec{G}$;
    Training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \dots, N\}$;
Output:
    The output weight $\beta$ of OS-ELM;
    $k=0$;
    for each training data do
        Search the whole path.
        if $\psi_{\mathbf{x}_{back}} \rightarrow \psi_{\mathbf{x}_{front}}$ then
            Calculate the remaining path length of $\psi_{\mathbf{x}_{back}}$.
            Add $\psi_{\mathbf{x}_{back}}$ into the path tree of $\psi_{\mathbf{x}_{front}}$.
        else
            Calculate the corrected vector $\mathbf{x}_i^*$.
        end if
        Calculate the hidden layer output matrix $\mathbf{H}_k$ about $\mathbf{x}_i^*$;
        Estimate the initial output weight $\beta^{(k)}$;
        $k=k+1$;
    end for
```

Compare with BNC methods, the ONC method solves the problem of large response time. By analyzing the computational complexity of the ONC method, we can know that it only takes $O(|V|+|E|)$ time to learn the classifier. In addition, the ONC method avoids the human intervention and keeps the classification accuracy.

4. Experiments

In this section, we test the effectiveness and efficiency of our proposed three node classification algorithms. We conduct extensive experiments by evaluating the efficiency of our techniques with varying parameters. Our methods are implemented on MATLAB 7.0 with AMD Athlon Dual Core processors 3.0 GHz and 4 GB main memory and programs are compiled in Java. We use GraphStream [29] to conduct the operation of DAG. GraphStream is a graph handling Java library that focuses on modeling of dynamic interaction networks of various sizes. The goal of the library is to provide a way to represent graphs and proposes several graph classes that allow to model directed and undirected graphs. We present our experiment results on real dataset to evaluate the effectiveness of our proposed algorithms.

4.1. Dataset

In this paper, we use three kinds of datasets, containing WebKB [18] and microblog, to evaluate the effectiveness of our algorithms.

The WebKB dataset consists of four sub-datasets which represent the web pages and the links between these pages from the universities, including Cornell, Washington, Texas and Wisconsin respectively. Each web page is marked one of five class labels, containing course, faculty, project, staff and student. The whole dataset consists of 877 web pages 1608 links. Each web page in the dataset is described by a 0/1-valued word vector indicating the corresponding word from the dictionary. There are 1703 unique words in the dictionary.

Sina Microblog is a large-scale online social network in China. The dataset of microblog is extracted by crawler. The dataset consists of 661 users and the whole network consists of 1972 links. The dataset is classified into four classes according to the interests of users. If there exists a following relationship between two users, we put a directed edge between them.

We extracted a connected sub-network of the DBLP dataset on four fields: database, data mining, information retrieval and artificial intelligence. The dataset contains 5768 papers, 12 conferences, 5975 authors and 3763 terms, with a total number of 123,154 links. The link can be connected if papers have some connection with authors, conferences and terms. For testing the accuracy of our methods, we use a labeled dataset of 2679 authors, 50 papers and all 12 conferences. We use DBLP dataset to test our methods in a larger social network.

4.2. Experiments setting

Our experiments mainly evaluate the following two aspects: (1) the accuracy and response time of methods BNC, INC, ONC, C-RMN (Community-based Relational Markov Networks) [30] and RC-ELM (Regularization Classification Method Based on ELM) [31] and (2) the effect of parameters l^* on the performance. For the first aspect, we compare our proposed algorithms with varying size of the training data in different datasets. For the second aspect, we mainly consider effect of path length on the performance of the INC method. The sigmoidal additive activation function is used in the simulations and the number of hidden node is set to be 180.

4.3. Results

We first only use OS-ELM to test the running time and classification accuracy of DAG nodes without considering the relationship of ancestor nodes, in order to compare with the performance of our proposed methods. Table 1 gives the experiments' results with varying training data size D in different datsets and the number of neurons is set to be 11. The results indicate that OS-ELM cannot fully demonstrate its capabilities in link-based classification. Though the results show a fast training speed, the classification accuracy is lower. For classification problem, classification accuracy is a more important factor. Thus, in the following

Please cite this article as: Y. Sun, et al., An on-line sequential learning method in social networks for node classification, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.04.074

experiments, we will compare our proposed methods with content-based OS-ELM method.

Next, we test the performance of our proposed algorithms. Fig. 3 shows the response time of our proposed algorithms in different size of training data respectively. In order to guarantee the INC method to obtain higher classification accuracy, we set the path length parameter $l^*=8$. From the five experiments, we can observe that the performance of the ONC method is better than the other two methods. With the increasing of training data size, the training time of the BNC method is increasing rapidly, while the ONC method appears a shortest response time. This is because that the BNC method needs to scan the ancestors of all nodes once in each learning iteration, while the ONC method only needs to scan the ancestors of all nodes once in the whole learning process. Based on the setting of l^*, the INC method costs less running time than the BNC method, since many paths are pruned. The experiments indicate that the performance of the ONC method is only relevant to the scale of trained graph and the performance of OS-ELM. Since the response time of C-RMN method is more than one minute, the results cannot be shown.

Fig. 4 shows the classification accuracy of our proposed algorithms. In the experiments, we adopt cross-validation method to test the classification accuracy in turn and select the average value of all the test results as the final results. These figures show the classification accuracy of our proposed algorithms in different size of training data respectively. From the figures, we can know that the classification accuracy increases with the increasing of training data size. Compare with the methods BNC and INC, since the path length is set by us, some paths and nodes that need to be computed are lost. For this reason, the classification accuracy in the INC method is lower than BNC and ONC methods, in which BNC and ONC methods both scan the whole DAG in learning process. However, compare with the result in Table 1, the testing accuracy of all the methods are better than the ones in OS-ELM.

Table 1
The test only with OS-ELM.

Datasets	D	Training (ms)	Testing (ms)	Training (%)	Testing (%)
Cornell	50	10.8	2.52	0.92	0.38
	100	17.6	1.8	0.94	0.43
Washington	50	10.6	5.8	0.95	0.472
	100	18.3	4.9	0.96	0.65
Texas	50	11.6	3.13	0.95	0.272
	100	17.1	1.7	0.97	0.38
Wisconsin	50	13.6	10.1	0.92	0.37
	100	21.2	7.3	0.95	0.58
Microblog	50	27.4	19.6	0.87	0.32
	100	39.2	14.5	0.91	0.45

Then, we test the effect of path length with varying l^* based on the INC method. Fig. 5(a) shows the classification accuracy with varying l^* in different datasets. In this experiment, the number of training data is set to be 100. Likewise, we employ cross-validation method to test the classification accuracy in turn and select the average value of all the test results as the final results. From the figure, we can observe that the classification accuracy is very low when l^* is set to be 2. The phenomenon indicates that the ancestor nodes affect the node classification. With the increasing of parameter l^*, the classification accuracy increases rapidly. For microblog dataset, the classification accuracy keeps stable when $l^*=10$. By analyzing the dataset, we know that the max number of relations of each user in the social network is 9. Actually, the classification accuracy will not change if we set $l^*=9$. For the other datasets, we can also observe the same phenomenon. From the results, we can conclude that not all the relationship of ancestor nodes can affect the classification results even if the training data have many ancestors.

Next, we test the performance of depth-first search in DAG. In the experiments, we let all the nodes in different datasets join into learning and test the response time of feature correction. Fig. 5(b) shows the results of response time in different datasets. From the figure, we can observe that BNC and INC methods cost more time than ONC method does, since ONC method avoids the repeated computation on ancestors search.

Finally, we test the accuracy of our proposed algorithms in DBLP dataset. We show the classification accuracy of authors, papers and conferences in Fig. 6(a), (b) and (c). In order to test the problem of partially labeled, we randomly choose $\rho=(0.2\%, 1\%)$ of authors and papers as the labeled information for classification. The aim of randomly selecting is to avoid the influence of different information for classification results, and we use the average classification accuracy as our testing results. From the figure, we can observe that BNC and ONC methods still have a higher classification accuracy than INC method, and their classification

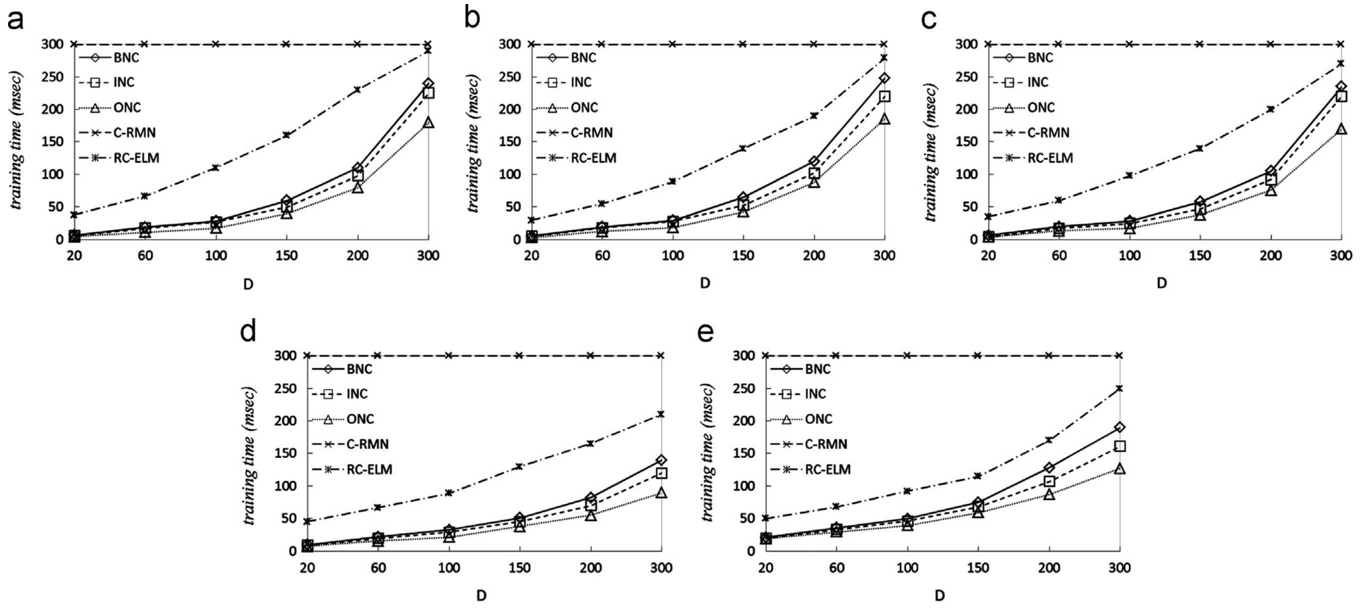


Fig. 3. Response time: (a) Cornell; (b) Washington; (c) Texas; (d) Wisconsin; and (e) Microblog.

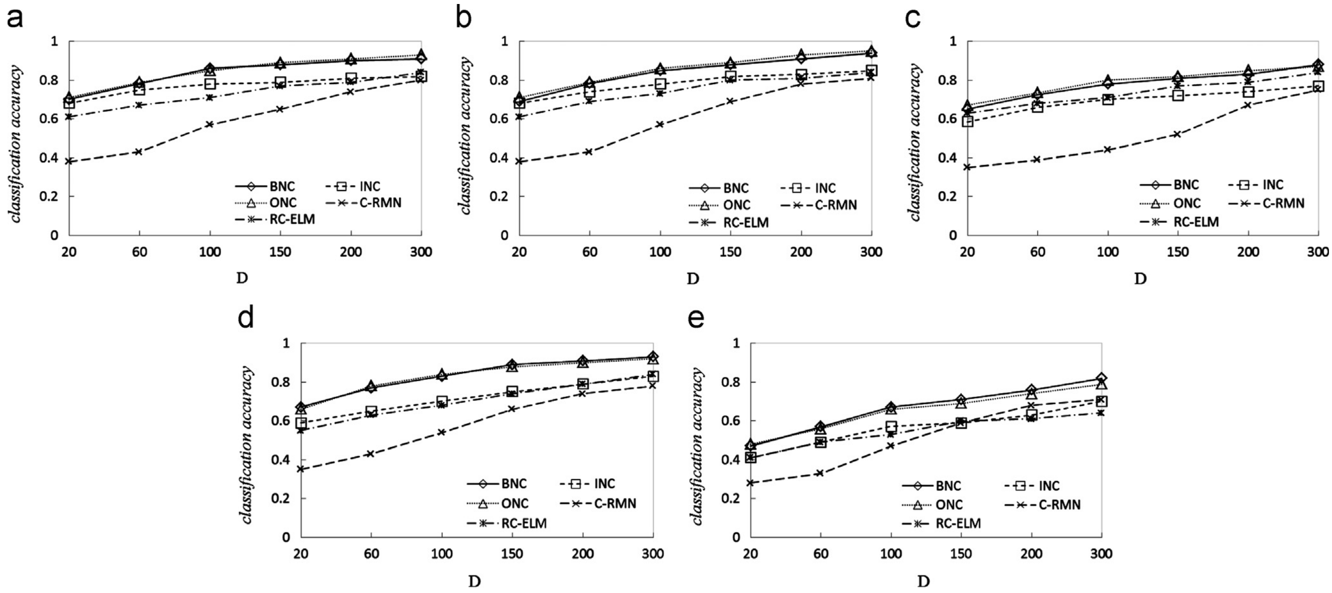


Fig. 4. Classification accuracy: (a) Cornell; (b) Washington; (c) Texas; (d) Wisconsin; and (e) Microblog.

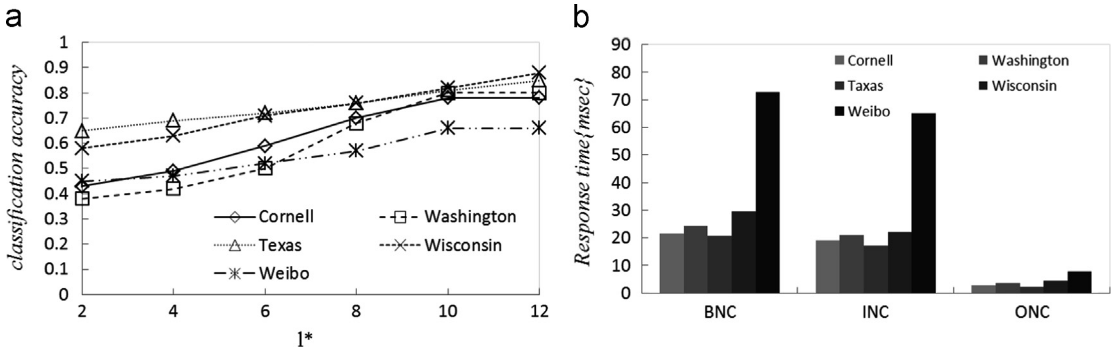


Fig. 5. Other experiments: (a) The effect of *l** and (b) the performance of depth-first search.

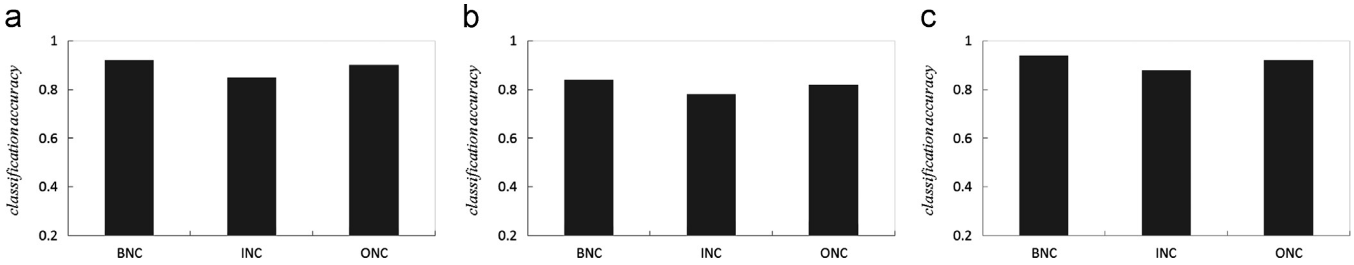


Fig. 6. Classification on DBLP: (a) Classification accuracy on authors; (b) classification accuracy on papers; and (c) classification accuracy on conferences.

accuracy is more than 90%. Although the classification accuracy of ONC method is slightly lower than BNC method, ONC method can obtain a faster running speed (ONC method scans partial nodes while BNC method scans the whole graph). Therefore, the testing results can verify the effectiveness of our methods in a large social network.

5. Conclusions

In this paper, we focus on studying node classification in social networks. We adopt DAG to construct the social network structure. Based on OS-ELM, we first propose the BNC method as a basic approach for node classification. Then, we improve the BNC method to obtain a better performance by setting the path length. Finally, we summarize the defects of previous two methods and propose an optimization method ONC without human intervention. Extensive experimental studies verify that the efficiency of ONC method is better than BNC and INC methods. In the future, we plan to study the node classification over joint distribution.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grant Nos. 61202087; the Open Foundation of WUHAN University Nos. SKLSE2012-09-40; the Fundamental Research Funds of the Central Universities No. N120404012; the National Basic Research Program of China under Grant No. 2011CB302200-G; the 863 Program under Grant

Please cite this article as: Y. Sun, et al., An on-line sequential learning method in social networks for node classification, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.04.074

No. 2012AA011004, and the Public Science and Technology Research Funds Projects of Ocean Grant No. 201105033.

References

[1] C. Desrosiers, G. Karypis, Within-network classification using local structure similarity, in: Proceedings of ECML/PKDD '09—The European Conference on Machine Learning and Knowledge Discovery in Databases, 2009, pp. 260–275.

[2] R. Heatherly, M. Kantarcioglu, B.M. Thuraisingham, Social network classification incorporating link type values, in: IEEE International Conference on Intelligence and Security Informatics (ISI), 2009, pp. 19–24.

[3] M. Ji, J. Han, M. Danilevsky, Ranking-based classification of heterogeneous information networks, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2011, pp. 1298–1306.

[4] M. Pennacchiotti, A.-M. Popescu, A machine learning approach to twitter user classification, in: Proceedings of the Fifth International Conference on Weblogs and Social Media (ICWSM), 2011.

[5] P.G. Franciosa, G. Gambosi, U. Nanni, The incremental maintenance of a depth-first-search tree in directed acyclic graphs, Inf. Process. Lett. 61 (2) (1997) 113–120.

[6] J. Karamon, Y. Matsuo, H. Yamamoto, M. Ishizuka, Generating social network features for link-based classification, in: European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), 2007, pp. 127–139.

[7] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411–1423.

[8] G.-B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of 2004 IEEE International Joint Conference on Neural Networks, vol. 2, Budapest, Hungary, 2004, pp. 985–990.

[9] G.-B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.

[10] G.-B. Huang, C.-K. Siew, Extreme learning machine: RBF network case, in: Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision, Kunming, China, 2004, pp. 1029–1036.

[11] G.-B. Huang, C.-K. Siew, Extreme learning machine with randomly assigned RBF kernels, Int. J. Inf. Technol. 11 (1) (2005) 16–24.

[12] N.-Y. Liang, P. Saratchandran, G.-B. Huang, N. Sundararajan, Classification of mental tasks from EEG signals using extreme learning machines, Int. J. Neural Syst. 16 (1) (2006) 29–38.

[13] C.-W.T. Yeu, M.-H. Lim, G.-B. Huang, A. Agarwal, Y.S. Ong, A new machine learning paradigm for terrain reconstruction, IEEE Geosci. Remote Sens. Lett. 3 (3) (2006) 382–386.

[14] R. Zhang, G.-B. Huang, N. Sundararajan, P. Saratchandran, Multi-category classification using extreme learning machine for microarray gene expression cancer diagnosis, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (3) (2007) 485–495.

[15] S.A. Macskassy, F.J. Provost, Classification in networked data: a toolkit and a univariate case study, J. Mach. Learn. Res. 8 (2007) 935–983.

[16] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Netw. 30 (1-7) (1998) 107–117.

[17] Y. Sun, Y. Yuan, G. Wang, An OS-ELM based distributed ensemble classification framework in P2P networks, Neurocomputing 74 (16) (2011) 2438–2443.

[18] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93–106.

[19] A.F. McDaid, N.J. Hurley, Detecting highly overlapping communities with model-based overlapping seed expansion, Adv. Soc. Netw. Anal. Min. (2010) 112–119.

[20] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007).

[21] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, in: Proceedings of the National Academy of Sciences of the United States of America (PNAS), vol. 105(4), 2008, pp. 1118–1123.

[22] C.C. Aggarwal, An introduction to Social Network Data Analytics, Social Network Data Analytics, Springer, 2011, pp. 1–15.

[23] B.-R. Dai, C.-Y. Lee, C.-H. Chung, A framework of recommendation system based on both network structure and messages, in: Advances in Social Network Analysis and Mining (ASONAM), 2011, pp. 709–714.

[24] G.H. Golub, C.F.V. Loan, Matrix Computations, 3rd edition, The Johns Hopkins University Press, Baltimore, 1996.

[25] Q. Lu, L. Getoor, Link-based classification, in: Proceedings of the 20th International Conferrence on Machine Learning (ICML), 2003, pp. 496–503.

[26] S.A. Macskassy, F. Provost, A simple relational classifier, in: Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM), 2003, pp. 64–76.

[27] B. Taskar, M.F. Wong, P. Abbeel, D. Koller, Link prediction in relational data, in: Neural Information Processing Systems (NIPS), 2003, pp. 659–666.

[28] R. Guimera, M. Sales-Pardo, Missing and spurious interactions and the reconstruction of complex networks, in: Proceedings of the National Academy of Sciences of the United States of America (PNAS), vol. 106(52), 2009, pp. 22073–22078.

[29] Y. Pigne, A. Dutot, F. Guinand, D. Olivier, A tool for bridging the gap between complex systems and dynamic graphs, in: Emergent Properties in Natural and Artificial Complex Systems (EPNACS), Workshop of the 4th European Conference on Complex Systems (ECCS), abs/0803.2093, 2007.

[30] H. Wan, Y. Lin, C. Jia, H. Huang, Community-based relational Markov networks in complex networks, in: Rough Sets and Knowledge Technology, 2011, pp. 301–310.

[31] L. Li, D. Liu, J. Ouyang, A new regularization classification method based on extreme learning machine in network data, J. Inf. Comput. Sci. 9 (12) (2012) 3351–3363.

Yongjiao Sun received the B.S. degree, M.S. degree and Ph.D. degree in Computer Science from Northeastern University, China in 2006, 2008 and 2012 respectively. Currently, he is an associate professor at computer science department of the Northeastern University. His main research interests include distributed data management, data analysis and probabilistic database.

Ye Yuan received the B.S. degree, M.S. degree and Ph.D. degree in Computer Science from Northeastern University, China in 2004 and 2007 respectively. Currently, he is an associate professor at the computer science department of Northeastern University, and he is also a visiting scholar at department of computer science and engineering in Hong Kong University of Science and Technology. His research interests include graph database, probabilistic database, P2P data management, and data piracy.

Guoren Wang received the B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the Northeastern University, Shenyang, China, in 1988, 1991, and 1996, respectively. Now, he is currently a Full Professor with the School of Information Science and Engineering, Northeastern University. His research interests include uncertain data management, data-intensive computing, visual media data management and analysis, distributed query processing and optimization techniques, and bioinformatics. He has published more than 100 research papers in international conference proceedings and journals.

Please cite this article as: Y. Sun, et al., An on-line sequential learning method in social networks for node classification, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.04.074