Transductive Classification on Heterogeneous Information Networks with Edge Betweenness-based Normalization

Phiradet Bangcharoensap
Tokyo Institute of Technology
phiradet.b@ai.cs.titech.ac.jp

Tsuyoshi Murata
Tokyo Institute of Technology
murata@cs.titech.ac.jp

Hayato Kobayashi
Yahoo Japan Corporation
hakobaya@yahoo-corp.jp

Nobuyuki Shimizu
Yahoo Japan Corporation
nobushim@yahoo-corp.jp

ABSTRACT

This paper proposes a novel method for transductive classification on heterogeneous information networks composed of multiple types of vertices. Such networks naturally represent many real-world Web data such as DBLP data (author, paper, and conference). Given a network where some vertices are labeled, the classifier aims to predict labels for the remaining vertices by propagating the labels to the entire network. In the label propagation process, many studies reduce the importance of edges connecting to a high-degree vertex. The assumption is unsatisfactory when reliability of a label of a vertex cannot be implied from its degree. On the basis of our intuition that edges bridging across communities are less trustworthy, we adapt edge betweenness to imply the importance of edges. Since directly applying the conventional edge betweenness is inefficient on heterogeneous networks, we propose two additional refinements. First, the centrality utilizes the fact that networks contain multiple types of vertices. Second, the centrality ignores flows originating from endpoints of considering edges. The experimental results on real-world datasets show our proposed method is more effective than a state-of-the-art method, GNetMine. On average, our method yields 92.79 $\pm$ 1.25% accuracy on a DBLP network even if only 1.92% of vertices are labeled. Our simple weighting scheme results in more than 5 percentage points increase in accuracy compared with GNetMine.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; E.1 [**Data**]: Data Structures—*Graphs and networks*

Keywords

Transductive Classification, Heterogeneous Information Network, Edge Betweenness Centrality

WSDM'16, February 22–25, 2016, San Francisco, CA, USA.

ISBN 978-1-4503-3716-8/16/02...$15.00

DOI: http://dx.doi.org/10.1145/2835776.2835799

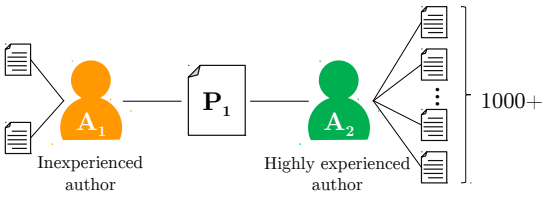


Figure 1: Bibliographic network countering the degree-based normalization. The research theme of the paper P_1 should conform to that of the highly experienced author A_2 rather than that of the inexperienced author A_1 because the research theme of A_2 is more well-established than A_1

1. INTRODUCTION

In machine learning, supervised learning has been a prominent paradigm. It makes predictions on the basis of a limited amount of data samples and their expected outputs. In many cases, manually labeling all data in a given dataset is a great burden as it is hard to scale, expensive, and prone to human-error. In contrast, a huge pool of unlabeled data always exists with lower cost, especially in this era of big data. *Semi-supervised learning* (SSL) that makes use of such unlabeled data has been gaining more attention.

Many Web data can be described as a network, where vertices represent entities of the system, and edges encode interactions or relationships between the entities. Networks can consist of multiple types of vertices such as bibliographic networks (author, paper, and conference), social networks (user, content, and page), and e-commerce purchasing logs (buyer, product, and seller). These kinds of networks are generally referred to as *heterogeneous information networks*.

We bring the two lines of research together by proposing a novel method for transductive semi-supervised classification on heterogeneous information networks. Given a network where a limited number of vertices hold initial labels, the classifier generally propagates the initial label information to the whole network. The classifier does not produce any decision functions or general rules to serve further unseen data. This paradigm is generally called *transductive* classification. The propagation process of many graph-based classifiers is based on the *smoothness* assumption — if two vertices are linked by a strong tie, then their outputs are likely to be close [5]. Several studies reduce influence of edges in accordance with the degrees of their endpoints to suppress

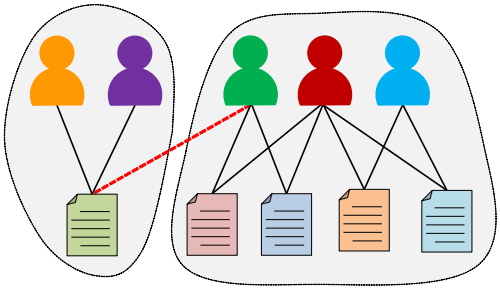


Figure 2: Network with an inter-community edge denoted by the red dashed line. The reliabilities of labels propagating through such inter-community edge should be discounted.

popular vertices from dominating the propagation process [16, 15, 17]. While edge weight normalization seems like a simple heuristic, it is one of the deciding factors for boosting the predictive performance. We observe significant improvement in accuracy by changing the normalization method to our proposed one.

We argue that the degree-based method is not effective when reliability of a label of a vertex cannot be implied from its degree. Figure 1 illustrates a bibliographic network consisting of two authors and their papers. Author A_1 tends to be inexperienced. In contrast, author A_2 is a top author who has published thousands of papers. Suppose we aim to infer the research theme of the paper P_1 from its neighbors, A_1 and A_2, on the basis of the smoothness assumption. Following the degree-based normalization, the label of P_1 tends to conform with present label of A_1, rather than the high profile author A_2, especially when the confidences of labels from A_1 and A_2 are comparable. We argue that this is not always reasonable. An author publishing many papers tends to have more well-established research interests than a young author. Thus, the research theme of the paper P_1 should conform to the research theme of A_2. This example shows that the degree-based method is not always effective.

This paper provides a new insight into the edge weight normalization. Instead of edges originating from high degree vertices, we propose that edges bridging across communities, called *inter-community edges*, are less reliable for making label inference. Generally, the term community means a set of vertices that are densely connected internally and loosely connected with vertices outside. Figure 2 illustrates an example of an inter-community edge denoted by the red dashed line. The most well-known measure capturing the characteristic is *edge betweenness centrality*. It has been applied in many applications, for example community detection [11], biological function investigation in protein interaction networks [8], and topology-controlling for wireless sensor networks [7]. We discover that naïvely applying the conventional edge betweenness in heterogeneous networks is ineffective because it is defined on the basis of an assumption that networks are homogeneous. We enhance the centrality by considering that networks contain multiple types of vertices.

The main contribution of this paper is twofold: (1) this paper sheds new light on employing the concept of the edge betweenness centrality in the edge weight normalization (2) we further improve the centrality to make it suitable for heterogeneous networks. With the proposed normalization, we gain more than 5 percentage points increase in accuracy from a state-of-the-art method, GNetMine [16], on dense networks derived from the DBLP dataset. It highlights that the edge weight normalization has a serious effect on accuracy of graph-based classification.

This paper first gives an overview of transductive learning in Section 2. It then details the proposed method in Section 3. Section 4 presents experiment results of the proposed method in comparison with existing state-of-the-art research on various datasets and settings. Finally, Section 5 concludes this paper and provides potential extensions.

2. RELATED WORK

In 2003, Zhu et al. proposed a SSL method based on a Gaussian random field model, popularly known as *label propagation* (LP) [27]. It tries to maintain smoothness of labels over the given graph and strictly preserve the initial labels by optimizing a cost function. In practical cases, some initial labels are possibly inaccurate because of human errors in a dataset preparation process. Zhou et al. proposed *local and global consistency* (LGC) [26] which allows initial labels to be slightly changed. Another contribution is that it is the very first method to normalize edge weights in accordance with the degree of their endpoints, while previous research did not normalize the weights [23, 27, 3]. This degree-based normalization has been exploited in *spectral clustering* before [6, 20]. *Adsorption* [2] and *modified adsorption* [24] have been proposed. They reduce the importance of vertices with respect to the Shannon entropy of their edges, which becomes inversely proportional to their degrees in unweighted networks. In 2014, Gong et al. proposed a new concept of smoothness called *local smoothness* via ReLISH [12]. It regularizes labels of vertices weakly connected to neighbors to prevent erroneous label propagation.

The methods presented above assume that an input network contains a single type of vertices called *homogeneous information network*. Researchers have recently become increasingly interested in mining heterogeneous networks. In 2010, Ji et al. proposed GNetMine [16], which generalizes LGC to heterogeneous networks. GNetMine inherits the degree-based edge weight normalization from LGC. However, GNetMine considers the types of edges when it measures degrees of vertices. In 2011, Ji et al. proposed RankClass [15], which combines ranking and classification in order to analyze more accurately. RankClass normalizes edge weights in two ways: the degree-based and the confidence-based method. In 2014, Luo et al. proposed HetPathMine [17], which extends GNetMine by incorporating the concept of *meta-path* [22]. The term meta-path means a path composed of a sequence of relations defined between different vertex types. The method measures the significance of meta-paths and utilizes them for classification. It inherits the degree-based normalization as well. Jacob et al. proposed to compute a latent representation of vertices in a space that is common to all types of vertices, and deduce labels from the representation [13]. Their method does not normalize the influence of edges.

Overall, some previous works do not normalize edge weights, and some normalize them in accordance with the degrees of their endpoints. We originally propose to penalize inter-community edge. The proposed normalization can be easily plugged into the mentioned methods.

3. PROPOSED METHOD

3.1 Problem Definition

This section begins by formally defining heterogeneous information networks. Next, it will specify the transductive classification problem on heterogeneous networks. We use capital bold letters for matrices (e.g. $\mathbf{A}$), and lower case bold letters for column vectors (e.g. $\mathbf{a}$). Capital bold letters with subscripts represent elements of the matrix (e.g. $\mathbf{A}_{ij}$ is the element at the i^{th} row and j^{th} of column of $\mathbf{A}$). Functions are denoted by capital non-bold letters (e.g. F). Sets are denoted by capital non-bold italic letters (e.g. S).

Definition 1. Heterogeneous information network. A heterogeneous information network, in this work, is defined as an unweighted undirected network $G = (V, E, W)$. V is a set of n vertices. The set is composed of t types of vertices denoted by $V_1 = \{v_{11}, \cdots, v_{1n_1}\}, \cdots, V_t = \{v_{t1}, \cdots, v_{tn_t}\}$, where $n_i = |V_i|$ and $n = \sum_{i=1}^{t} n_i$. When $t = 1$, the network G becomes a homogeneous information network. One vertex belongs to exactly one type, in other words, $\forall i, j : i \neq j \rightarrow V_i \cap V_j = \emptyset$. E is a set of m edges, composed of $e = (v_{ip}, v_{jq}) \in V \times V$. $W = \{\mathbf{W}_{11}, \cdots, \mathbf{W}_{tt}\}$ is a set of adjacency matrices denoted by $\mathbf{W}_{ij} \in \{0, 1\}^{n_i \times n_j}$, where $i, j \in \{1, \cdots, t\}$. An element $\bar{\mathbf{W}}_{ij,pq} = 1$ if two vertices $v_{ip} \in V_i$ and $v_{jq} \in V_j$ are linked together, otherwise zero. We say a network has r types of relationships when $|\{\mathbf{W}_{ij} \mid \mathbf{W}_{ij} \in W \wedge \mathbf{W}_{ij} \neq \mathbf{0}\}| = 2r$.

Definition 2. Transductive classification on heterogeneous networks. The inputs of the problem are a heterogeneous information network $G = (V, E, W)$ and a set of initial label information L. V_L is a set of labeled vertices. $\mathcal{C}$ is a set of possible labels, typically $|V_L| \ll |V|$. Without loss of generality, we assume that the possible labels are $\mathcal{C} = \{1, \cdots, |\mathcal{C}|\}$. The existence of a tuple (v, c) in L means the vertex v is initially labeled with the label c. In other words, c is an initial label of v. The set $V_U = V - V_L$ is the set of unlabeled vertices. This study assumes the given network contains no self-loops. The goal of the transductive classification task is to predict the labels of all unlabeled vertices V_U. First, the classifier computes soft label column vectors $\mathbf{f}_i^c \in \mathbb{R}^{n_i}$, for all $i \in \{1, \cdots, t\}$ and $c \in \mathcal{C}$. A column vector $\mathbf{f}_i^c = [f_{i1}^c, \cdots, f_{in_i}^c]^\top$ represents the confidence that each vertex in V_i should belong to a label c. A high value of f_{ip}^c shows that the classifier believes that a vertex v_{ip} tends to belong to the class c.

3.2 Basic Framework

The input of our proposed method is a heterogeneous information network $G = (V, E, W)$ and a set of labeled vertices L. First, the label information from L is represented as initial label column vectors $\mathbf{y}_i^c \in \{0, 1\}^{n_i}$, for $i \in \{1, \cdots, t\}$ and $c \in \mathcal{C}$. A column vector $\mathbf{y}_i^c = [y_{i1}^c, \cdots, y_{in_i}^c]^\top$ encodes the label information of all vertices $v_{ip} \in V_i$ regarding a label $c \in \mathcal{C}$. If a vertex v_{ip} is labeled to a class c, $(v_{ip}, c) \in L$, then $y_{ip}^c = 1$, otherwise zero. The proposed method propagates label information from the labeled vertices V_L to the entire network. Our proposed method has a trade-off between the two following constraints. First, the initial labels of seed vertices should be retained, called *fitting constraint*. Second, similar labels should be assigned to neighboring vertices, called *smoothness constraint*. To this end, the proposed method seeks the set of soft label matrix vectors that minimizes the cost function

$$\mathrm{J}(\mathbf{f}_1^c, \cdots, \mathbf{f}_t^c) = \sum_{i=1}^{t} \mathrm{E}(\mathbf{f}_i^c) \tag{1}$$

$$\mathrm{E}(\mathbf{f}_i^c) = \alpha_i \sum_{p=1}^{n_i} \left(f_{ip}^c - y_{ip}^c\right)^2 + \sum_{j=1}^{t} \lambda_{ij} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} \bar{\mathbf{W}}_{ij,pq} \left(f_{ip}^c - f_{jq}^c\right)^2, \tag{2}$$

where α_i and λ_{ij} are non-negative parameters balancing between the fitting and smoothness constraints, and $\bar{\mathbf{W}}_{ij,pq}$ is a normalized weight of an edge between vertices $v_{ip} \in V_i$ and $v_{jq} \in V_j$. For all i, j, the parameter λ_{ij} is equal to λ_{ji}. The edge weight normalization process is described in further detail in Section 3.3. A higher value of a normalized edge weight $\bar{\mathbf{W}}_{ij,pq}$ indicates that the soft labels of v_{ip} and v_{jq} tend to be more similar. The term *global cost function* will be used when referring to J(.), and *local cost function* for E(.). After the optimal $\mathbf{f}_i^{c*} = [f_{i1}^{c*}, \cdots, f_{in_i}^{c*}]^\top$ is found, for $i \in \{1, \cdots, t\}$ and $c \in \mathcal{C}$, the final predicted label of a vertex v_{ip} can be simply determined as

$$c_{ip} = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \, f_{ip}^{c*}. \tag{3}$$

The first term of the right hand side of Eq.(2) corresponds to the fitting constraint. A parameter α_i indicates trustworthiness of initial labels of seed vertices in a vertex type V_i. Suppose there is a bibliographic network containing vertices representing author, papers, and conferences such that some vertices are labeled with their research theme. Suppose we want to infer the research theme of every vertex. Generally, a research theme of an author is more ambiguous than that of a conference, so determining the initial labels of an author is prone to errors. Therefore, the parameter α_i of conference vertices should be higher than that of the authors. The second term enforces the smoothness constraint. The parameter λ_{ij} controls the degree of reliability of relations between vertices in V_i and V_j. A large λ_{ij} means labels of vertices in V_i can be faithfully inferred from labels of their neighbors in V_j. A user may set the parameter λ_{ij} of relationships between papers and conferences to be higher than that between papers and authors. These two parameters allow the method to treat each type of relationship uniquely, which is an advantage of learning on heterogeneous networks.

3.3 Edge Weight Normalization

We normalize the weight of an edge $e = (v_{ip}, v_{jq}) \in E$ as

$$\bar{\mathbf{W}}_{ij,pq} = \frac{1}{\mathrm{C}(e)} \mathbf{W}_{ij,pq}, \tag{4}$$

where C(.) is a *normalizing function*. By plugging Eq.(4) into Eq.(2), we obtain the complete definition of E(.). Our cost function is different from that of GNetMine, particularly the second term of E(.). The second term of E(.) of GNetMine is

$$\sum_{j=1}^{t} \lambda_{ij} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} \mathbf{W}_{ij,pq} \left(\frac{f_{ip}^c}{\sqrt{d_{ij,p}}} - \frac{f_{jq}^c}{\sqrt{d_{ji,q}}} \right)^2,$$

where $d_{ij,p}$ is the summation of the p^{th} row of $\mathbf{W}_{ij}$. GNetMine aims to reduce the impact of popular vertices. Hence, the normalizing terms, $d_{ij,p}$ and $d_{ji,q}$, are located below f_{ip}^c

and f_{jq}^c of vertices. In contrast, we aim to reduce the impact of inter-community edges. Therefore, our normalizing function C(.) is located below the weight of edges as Eq.(4).

Ideally, C(e) is expected to be high if the edge e is an inter-community edge. We hypothesize that labels propagating through such edge are less reliable for making inference. The normalization help us decrease erroneous flows propagating through inter-community edges.

The inter-community characteristic can be captured by *edge betweenness centrality* [11, 19]. It measures the influence of an edge over flows of information between vertices, assuming that the flows follow the shortest paths. Originally, the edge betweenness centrality is defined as

$$\mathrm{C_{B0}}(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}, \tag{5}$$

where $\sigma(s,t|e)$ is the number of the shortest paths from vertex s to vertex t passing through an edge e, and $\sigma(s,t)$ is the number of the shortest paths from s to t. By convention, let $\frac{0}{0} = 0$ in this equation. From now on, the edge centrality $\mathrm{C_{B0}}(.)$ is called *homogeneous edge betweenness centrality* because the centrality is defined on the basis of an assumption that the given network contains single type of vertices. The fact that some networks contain multiple types of vertices is not considered. In contrast, we propose an edge betweenness centrality for heterogeneous networks, called *heterogeneous edge betweenness centrality*. It has two important properties: excluding flows from endpoints and ignoring irrelevant flows, which means flows from vertices of non-target types. The proposed heterogeneous edge betweenness of an edge $e = (v_{ip}, v_{jq})$, where $v_{ip} \in V_i$ and $v_{jq} \in V_j$, is defined as

$$\mathrm{C_{B1}}(e) = 1 + \sum_{s \in V_i \setminus \mathrm{EP}(e)} \sum_{t \in V_j \setminus \mathrm{EP}(e)} \frac{\sigma(s,t|e)}{\sigma(s,t)}, \tag{6}$$

where $\mathrm{EP}(e)$ indicates the set of endpoints of the edge e, which is $\{v_{ip}, v_{jq}\}$.

The idea of excluding flows from endpoints is motivated by Freeman's vertex betweenness [9]. He defined the betweenness of a vertex u as

$$\mathrm{VB}(u) = \sum_{s,u,t \in V : s \neq u \neq t} \frac{\sigma(s,t|u)}{\sigma(s,t)}, \tag{7}$$

where $\sigma(s,t|u)$ is the number of the shortest paths from a vertex s to a vertex t passing through a vertex u. We observed that one important property of vertex betweenness is missing from edge betweenness — flows originating from the vertex u are not counted. Therefore, when calculating the edge betweenness of an edge, we propose to ignore flows originating from its endpoints. The experimental results presented in Section 4.2 ensure the benefit of this refinement.

The homogeneous edge betweenness, defined in Eq.(5), does not differentiate multiple types of vertices. Figure 3 shows an example of a heterogeneous network where the homogeneous edge betweenness exhibits an irrational result. The vertices with the prefixes A, P, and C represent author, paper, and conference vertices, respectively. Let us start by calculating $\mathrm{C_{B0}}(e_4)$ and $\mathrm{C_{B0}}(e_5)$ as

$$\mathrm{C_{B0}}(e_4) = 2\left[\sum_{s \in \{A_1,A_2,A_3,P_1\}} \sum_{t \in \{A_4,P_2,C_1\}} \frac{\sigma(s,t|e)}{\sigma(s,t)}\right] = 24,$$

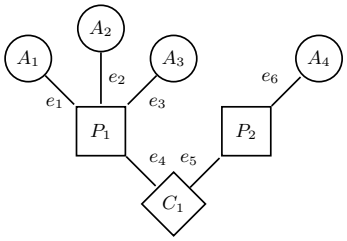


Figure 3: Heterogeneous bibliographic network containing authors (circles), papers (squares), and a conference (diamond).

$$\mathrm{C_{B0}}(e_5) = 2\left[\sum_{s \in \{A_4,P_2\}} \sum_{t \in \{A_1,A_2,A_3,P_1,C_1\}} \frac{\sigma(s,t|e)}{\sigma(s,t)}\right] = 20.$$

The $\mathrm{C_{B0}}(e_4)$ is higher than $\mathrm{C_{B0}}(e_5)$ even if both papers have edges dedicated to the conference. The only difference is the number of corresponding authors of papers P_1 and P_2. The shortest paths from $\{A_1, A_2, A_3\}$ and $\{A_4\}$ to $\{C_1\}$ are the crucial point of the different values. In the case of $\mathrm{C_{B0}}(e_4)$, the flows from three authors, $\{A_1, A_2, A_3\}$, to the conference are counted. In $\mathrm{C_{B0}}(e_5)$, the flow from only one author, $\{A_4\}$, is counted. This shows that a paper written by more authors has a higher value of $\mathrm{C_{B0}}(.)$.

Given the network as illustrated in Figure 3 and initial labels of P_1 and P_2, the label of C_1 can be implied from the labels of P_1 and P_2 under the smoothness assumption. As stated earlier, this research hypothesizes that edges in between communities are less reliable. If $\mathrm{C_{B0}}(e_4)$ and $\mathrm{C_{B0}}(e_5)$ are employed to measure the influence of the labels of P_1 and P_2 over the label of C_1 respectively, then the method tends to assign the initial label of P_2 to C_1 because $\mathrm{C_{B0}}(e_5) < \mathrm{C_{B0}}(e_4)$. This example reveals that a label of a paper written by many authors is less reliable based on the homogeneous betweenness. Obviously, this is not reasonable. Therefore, we propose to ignore flows or the shortest paths originating from irrelevant vertices, which means vertices of non-target types. In summary, one advantage of $\mathrm{C_{B1}}(.)$ over $\mathrm{C_{B0}}(.)$ is that $\mathrm{C_{B1}}(.)$ reduces the influence of irrelevant flows by ignoring the shortest paths originating from authors while considering paper-conference edges in this example.

In this section, two edge betweenness centralities have been introduced, $\mathrm{C_{B0}}(.)$ and $\mathrm{C_{B1}}(.)$. The normalizing function C(.) can be either of them.

3.4 Solving the Optimization Problem

Our optimization method is based on that of GNetMine. However, our cost function is different from theirs, as mentioned in Section 3.3. This section shows the method to optimize our cost function.

We find the set of soft label vectors, $\mathbf{f}_i^c$ for all $i \in \{1, \cdots, t\}$ and $c \in \mathcal{C}$, that optimize J(.) by *block coordinate descent* (BCD) method. It seeks the optimal solution by optimizing only one soft label vector $\mathbf{f}_i^c$ at one time, while keeping the remaining label vectors fixed at their last updated values. The process is repeated until convergence.

First, the local cost function E(.) is transformed into a matrix form. The first term of Eq.(2), can be rewritten as

$$\alpha_i \sum_{p=1}^{n_i} \left(f_{ip}^c - y_{ip}^c\right)^2 = \alpha_i \left(\mathbf{f}_i^c - \mathbf{y}_i^c\right)^\top \left(\mathbf{f}_i^c - \mathbf{y}_i^c\right). \tag{8}$$

The second term of the Eq.(2) can be rewritten as

$$\begin{aligned}
&\sum_{j=1}^{t}\lambda_{ij}\sum_{p=1}^{n_i}\sum_{q=1}^{n_j}\bar{\mathbf{W}}_{ij,pq}\left(f_{ip}^{c}-f_{jq}^{c}\right)^2\\
&=\sum_{j=1}^{t}\lambda_{ij}\sum_{p=1}^{n_i}\sum_{q=1}^{n_j}\bar{\mathbf{W}}_{ij,pq}\left({f_{ip}^{c}}^2+{f_{jq}^{c}}^2-2f_{ip}^{c}f_{jq}^{c}\right)\\
&=\sum_{j=1}^{t}\lambda_{ij}\left(\sum_{p=1}^{n_i}\sum_{q=1}^{n_j}\bar{\mathbf{W}}_{ij,pq}{f_{ip}^{c}}^2+\sum_{p=1}^{n_i}\sum_{q=1}^{n_j}\bar{\mathbf{W}}_{ij,pq}{f_{jq}^{c}}^2\right.\\
&\left.-2\sum_{p=1}^{n_i}\sum_{q=1}^{n_j}\bar{\mathbf{W}}_{ij,pq}f_{ip}^{c}f_{jq}^{c}\right)\\
&=\sum_{j=1}^{t}\lambda_{ij}\left(\sum_{p=1}^{n_i}f_{ip}^{c}\bar{\mathbf{D}}_{ij,pp}f_{ip}^{c}+\sum_{q=1}^{n_j}f_{jq}^{c}\bar{\mathbf{D}}_{ji,qq}f_{jq}^{c}\right.\\
&\left.-2\sum_{p=1}^{n_i}\sum_{q=1}^{n_j}f_{ip}^{c}\bar{\mathbf{W}}_{ij,pq}f_{jq}^{c}\right)\\
&=\sum_{j=1}^{t}\lambda_{ij}\left(\mathbf{f}_i^{c\top}\bar{\mathbf{D}}_{ij}\mathbf{f}_i^{c}+\mathbf{f}_j^{c\top}\bar{\mathbf{D}}_{ji}\mathbf{f}_j^{c}-2\mathbf{f}_i^{c\top}\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c}\right), \quad (9)
\end{aligned}$$

where $\bar{\mathbf{D}}_{ij}\in\mathbb{R}_{\geq 0}^{n_i\times n_i}$ is a diagonal matrix with diagonal elements $\bar{\mathbf{D}}_{ij,pp}=\sum_{q=1}^{n_j}\bar{\mathbf{W}}_{ij,pq}$, for $p\in\{1,\cdots,n_i\}$. Hence, by plugging Eq.(8) and Eq.(9) into Eq.(2), the local cost function can be alternatively defined as

$$\begin{aligned}
\mathrm{E}(\mathbf{f}_i^{c})=&\sum_{j=1}^{t}\lambda_{ij}\left(\mathbf{f}_i^{c\top}\bar{\mathbf{D}}_{ij}\mathbf{f}_i^{c}+\mathbf{f}_j^{c\top}\bar{\mathbf{D}}_{ji}\mathbf{f}_j^{c}-2\mathbf{f}_i^{c\top}\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c}\right)\\
&+\alpha_i\left(\mathbf{f}_i^{c}-\mathbf{y}_i^{c}\right)^{\top}\left(\mathbf{f}_i^{c}-\mathbf{y}_i^{c}\right). \quad (10)
\end{aligned}$$

Let us start by finding one optimal soft label vector while the others are fixed. It can be obtained by setting the first derivative of the local cost function with respect to $\mathbf{f}_i^{c}$ to zero. The first derivative of E(.) is

$$\begin{aligned}
&\frac{\partial}{\partial\mathbf{f}_i^{c}}\left[\sum_{j=1}^{t}\lambda_{ij}\left(\mathbf{f}_i^{c\top}\bar{\mathbf{D}}_{ij}\mathbf{f}_i^{c}+\mathbf{f}_j^{c\top}\bar{\mathbf{D}}_{ji}\mathbf{f}_j^{c}-2\mathbf{f}_i^{c\top}\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c}\right)\right.\\
&\left.+\alpha_i\left(\mathbf{f}_i^{c}-\mathbf{y}_i^{c}\right)^{\top}\left(\mathbf{f}_i^{c}-\mathbf{y}_i^{c}\right)\right]\\
=&\frac{\partial}{\partial\mathbf{f}_i^{c}}\left[\sum_{j=1,j\neq i}^{t}\lambda_{ij}\left(\mathbf{f}_i^{c\top}\bar{\mathbf{D}}_{ij}\mathbf{f}_i^{c}+\mathbf{f}_j^{c\top}\bar{\mathbf{D}}_{ji}\mathbf{f}_j^{c}-2\mathbf{f}_i^{c\top}\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c}\right)\right.\\
&\left.+\lambda_{ii}\left(2\mathbf{f}_i^{c\top}\bar{\mathbf{D}}_{ii}\mathbf{f}_i^{c}-2\mathbf{f}_i^{c\top}\bar{\mathbf{W}}_{ii}\mathbf{f}_i^{c}\right)+\alpha_i\left(\mathbf{f}_i^{c}-\mathbf{y}_i^{c}\right)^{\top}\left(\mathbf{f}_i^{c}-\mathbf{y}_i^{c}\right)\right]\\
=&\sum_{j=1,j\neq i}^{t}\lambda_{ij}\left(2\bar{\mathbf{D}}_{ij}\mathbf{f}_i^{c}-2\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c}\right)+\lambda_{ii}\left(4\bar{\mathbf{D}}_{ii}\mathbf{f}_i^{c}-4\bar{\mathbf{W}}_{ii}\mathbf{f}_i^{c}\right)\\
&+\alpha_i\left(2\mathbf{f}_i^{c}-2\mathbf{y}_i^{c}\right)\\
=&2\left(\sum_{j=1,j\neq i}^{t}\lambda_{ij}\bar{\mathbf{D}}_{ij}+\alpha_i\mathbf{I}+2\lambda_{ii}\left(\bar{\mathbf{D}}_{ii}-\bar{\mathbf{W}}_{ii}\right)\right)\mathbf{f}_i^{c}\\
&-2\left(\sum_{j=1,j\neq i}^{t}\lambda_{ij}\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c}+\alpha_i\mathbf{y}_i^{c}\right). \quad (11)
\end{aligned}$$

By setting the first derivative of E(.) with respect to $\mathbf{f}_i^{c}$, as shown in Eq.(11), to zero, we obtain the optimal $\mathbf{f}_i^{c}$ that is the minimizer of the local cost function E(.) as

$$\begin{aligned}
\mathbf{f}_i^{c*}=&\left(\sum_{j=1,j\neq i}^{t}\lambda_{ij}\bar{\mathbf{D}}_{ij}+\alpha_i\mathbf{I}+2\lambda_{ii}\left(\bar{\mathbf{D}}_{ii}-\bar{\mathbf{W}}_{ii}\right)\right)^{-1}\\
&\left(\sum_{j=1,j\neq i}^{t}\lambda_{ij}\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c}+\alpha_i\mathbf{y}_i^{c}\right). \quad (12)
\end{aligned}$$

However, the above solution is based on matrix inversion, which is not suitable for a large sparse network because the inverse of a matrix tends to be dense even if the source matrix is sparse. The advantage of a sparse matrix structure cannot be taken. Thus, we present an iterative optimization algorithm that avoids matrix inversion. It is based on the Jacobi iterative method [21]. Given a linear system $\mathbf{Mx}=\mathbf{b}$, the approximate solution at step $k+1$ is

$$\mathbf{x}_i^{(k+1)}=\frac{1}{\mathbf{M}_{ii}}\left(\mathbf{b}_i-\sum_{j\neq i}\mathbf{M}_{ij}\mathbf{x}_j^{(k)}\right). \quad (13)$$

Hence, a soft label vector $\mathbf{f}_i^{c}$ at step $k+1$ is

$$\begin{aligned}
\mathbf{f}_i^{c(k+1)}=&\left(\sum_{j=1,j\neq i}^{t}\lambda_{ij}\bar{\mathbf{D}}_{ij}+\alpha_i\mathbf{I}+2\lambda_{ii}\bar{\mathbf{D}}_{ii}\right)^{-1}\\
&\left(\sum_{j=1,j\neq i}^{t}\lambda_{ij}\bar{\mathbf{W}}_{ij}\mathbf{f}_j^{c(k)}+\alpha_i\mathbf{y}_i^{c}+2\lambda_{ii}\bar{\mathbf{W}}_{ii}\mathbf{f}_i^{c(k)}\right). \quad (14)
\end{aligned}$$

Even though Eq.(14) involves matrix inversion, its results are still sparse because the inversion of a diagonal matrix is diagonal as well. The inversion of a diagonal matrix can be easily computed in linear time in accordance with the number of diagonal elements. Algorithm 1 summarizes the iterative optimization algorithm for determining predicted labels of all vertices $v_{ip}\in V$.

3.5 Computational Complexity

Section 3.4 presents an iterative optimization algorithm that aims to seek the optimal soft labels of all vertices. It first starts by initializing soft label vectors $\mathbf{f}_i^{c}$ at Line 1. Preliminarily, they are initialized to the corresponding initial label vectors $\mathbf{y}_i^{c}$. The total size of all soft label vectors is $n|C|$, so this step takes $\mathcal{O}(n|C|)$. At Line 2, edge weight matrices are normalized on the basis of their edge betweenness centrality as described in Section 3.3. Assuming centralities are given beforehand, the normalization needs $\mathcal{O}(m)$ since each edge is processed once. In 2008, Brandes introduced an $\mathcal{O}(nm)$ algorithm to compute edge betweenness [4]. It was adapted in this present study. Next, Lines 3-6 create vertex degree matrices $\bar{\mathbf{D}}_{ij}$ derived from the normalized edge weight matrices $\bar{\mathbf{W}}_{ij}$. Basically, the algorithm iterates through all edges, so the time complexity of this step is $\mathcal{O}(m)$. It then computes the first term of Eq.(14). The degree matrices of each vertex type are summed up together. Let us define r_i as the number of neighboring vertex types of a vertex type V_i such that $r_i=|\{j\mid\mathbf{W}_{ij}\in W\wedge\mathbf{W}_{ij}\neq\mathbf{0}\}|$ and $\sum_{i=1}^{t}r_i=2r$. This line takes $\sum_{i=1}^{t}(n_ir_i+n_i)$ steps.

Algorithm 1: Iterative optimization algorithm

Input : Heterogeneous network $G = (V, E, W)$
Initial label $\mathbf{y}_i^c$, for $i \in \{1, \cdots, t\}$ and $c \in \mathcal{C}$
λ_{ij} and α_i, for $i, j \in \{1, \cdots, t\}$
Output: Predicted label c_{ip}, for $i \in \{1, \cdots, t\}$ and $p \in \{1, \cdots, n_i\}$

1 $\mathbf{f}_i^c \leftarrow \mathbf{y}_i^c$, for $i \in \{1, \cdots, t\}$ and $c \in \mathcal{C}$
2 $\bar{\mathbf{W}}_{ij} \leftarrow$ `EdgeNormalize(`$\mathbf{W}_{ij}$`)`, for $\mathbf{W}_{ij} \in W$
3 $\mathbf{D}_{ij} \leftarrow \mathbf{0}^{n_i \times n_i}$, for $i, j \in \{1, \cdots, t\}$
4 **foreach** $(v_{ip}, v_{jq}) \in E$ **do**
5 $\quad \mathbf{D}_{ij,pp} \leftarrow \mathbf{D}_{ij,pp} + \bar{\mathbf{W}}_{ij,pq}$
6 **end**
7 $\mathbf{M}_i \leftarrow \sum_{j=1, j\neq i}^{t} \lambda_{ij} \bar{\mathbf{D}}_{ij} + \alpha_i \mathbf{I} + 2\lambda_{ii} \bar{\mathbf{D}}_{ii}$, for $i \in \{1, \cdots, t\}$
8 **repeat**
9 $\quad$ **foreach** $c \in \mathcal{C}$ **do**
10 $\quad\quad$ **foreach** $i \in \{1, \cdots, t\}$ **do**
11 $\quad\quad\quad \mathbf{f}_i^c \leftarrow$
$\quad\quad\quad \mathbf{M}_i{}^{-1} \left(\sum_{j=1, j\neq i}^{t} \lambda_{ij} \bar{\mathbf{W}}_{ij} \mathbf{f}_j^c + \alpha_i \mathbf{y}_i^c + 2\lambda_{ii} \bar{\mathbf{W}}_{ii} \mathbf{f}_i^{c(t)} \right)$
12 $\quad\quad$ **end**
13 $\quad$ **end**
14 **until** *convergence*
15 $c_{ip} \leftarrow \arg\max_{c \in \mathcal{C}} f_{ip}^c$, for $i \in \{1, \cdots, t\}$ and $p \in \{1, \cdots, n_i\}$

Its inequality is

$$\begin{aligned} \sum_{i=1}^{t} (n_i r_i + n_i) &\leq \sum_{i=1}^{t} \left(n_i \sum_{i=1}^{t} r_i \right) + \sum_{i=1}^{t} n_i. \\ &= 2 \sum_{i=1}^{t} (n_i r) + \sum_{i=1}^{t} n_i. \\ &= 2nr + n. \end{aligned}$$

Hence, Line 7 spends $\mathcal{O}(n(r+1))$. It then iteratively optimizes soft label vectors. At Lines 10-12, the soft label vector of a vertex type V_i and a class c is updated. It aggregates the multiplication of edge weights and the current soft label of their endpoints. This operation takes $\mathcal{O}(m_i)$ for each vertex type, in total $\mathcal{O}(m)$. Further, incorporating initial labels and dividing by $\mathbf{M}_i$, which needs $\mathcal{O}(n)$, are conducted. The update procedures are performed for all $c \in \mathcal{C}$ and repeated until convergence. Therefore, the algorithm takes $\mathcal{O}(\tau|C|(n+m))$, where τ is the number of iterations. At Line 15, the final prediction can be determined within $\mathcal{O}(n|C|)$. Finally, the time complexity of the iterative optimization algorithm, presented in Algorithm 1, is $\mathcal{O}(\tau|C|(n+m))$. In our experiment, τ was less than twenty.

4. EXPERIMENTS

This paper proposes to utilize edge betweenness in the edge weight normalization process. Furthermore, two refinements are proposed to make the centrality more suitable for heterogeneous networks. Two questions were raised:

1. Are the two refinements of the edge betweenness really useful?
2. How accurate is the method with the edge betweenness centrality compared with the existing methods?

Two set of experiments were conducted on real-world datasets to answer the questions. Section 4.1 describes the datasets used in this study. Section 4.2 and Section 4.3 reveal evidence to answer the questions.

4.1 Dataset

Heterogeneous Network

We performed experiments on the *four-area* dataset, provided by [10] and [16]. It is a sub-network of the DBLP[1] dataset on four research areas — information retrieval, artificial intelligence, data mining, and database. Hence, there are four possible labels in this dataset. Figure 4 shows the topology of the network derived from this dataset. It contains four types of vertices: *paper*, *author*, *term*, and *conference*. In total, it consists of 36,915 vertices divided into 13,896 papers, 14,216 authors, 8,785 terms, and 18 conferences: SIGIR, WWW, WSDM, CIKM, ECIR, AAAI, ICML, IJCAI, CVPR, KDD, ICDM, PAKDD, SDM, VLDB, EDBT, PODS, ICDE, and SIGMOD. Three kinds of edges exist in the network: author-paper, conference-paper, and term-paper. An author and a paper are connected if and only if the author writes the paper. An edge between a paper and a conference indicates that the paper is published in the conference or journal. The semantic of a relationship between a term and a paper is that the term appears in the title of the paper. In total, the network contains 165,157 edges in the networks. The dataset contains label information of 99 (0.71%) papers, 4049 (28.48%) authors, and 18 (100.00%) conferences.

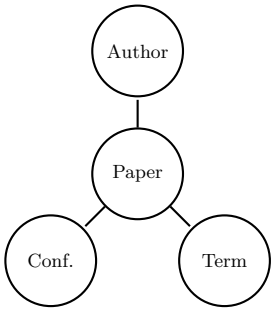


Figure 4: Topology of four-area dataset

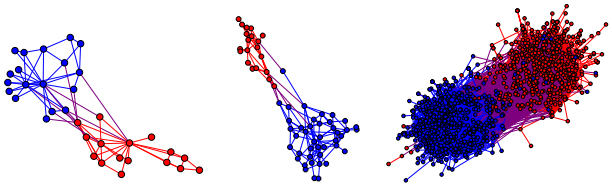
Homogeneous Network

In addition, we also studied the effectiveness of methods on three famous homogeneous networks: *Zachary's karate club* [25], *dolphin* [18], and *political blogs* [1]. The Zachary's karate club network represents friendships of 34 members of a karate club at a US university in the 1970s. The dolphin social network represents frequent associations between 62 dolphins in a community living off Doubtful Sound. The political blogs network represents hyperlinks between weblogs regarding US politics, recorded in 2005. The ground-truth labels of all vertices in datasets are available. It is worth conducting experiments on these datasets because they can confirm the effectiveness of methods when edges between the same type of vertices exist. Table 1 summarizes properties of the networks. Figure 5 visualizes the networks where nodes are colored in accordance with ground-truth labels.

[1]http://www.informatik.uni-trier.de/~ley/db/

Table 1: Properties of political blogs, Zachary's karate club, and dolphin networks.

Measure	Karate	Dolphin	Political blogs
n	34	62	1,222
m	78	159	16,714
$\|\mathcal{C}\|$	2	2	2



(a) Karate (b) Dolphin (c) Political blogs

Figure 5: Visualization of homogeneous networks used in experiments: Zachary's karate club (a), dolphin (b), and political blogs (c) networks. The colors of vertices indicates their ground-truth labels. The purple edges are inter-community edges. The blue and red edges are intra-community edges.

4.2 Comparison of Edge Betweenness Centralities

Section 3.3 proposed two additional refinements of the edge betweenness. Now, we aim to evaluate their effectiveness via two experiments. First, we compare the capabilities of the centralities C_{B0} and C_{B1} to distinguish intra-community and inter-community edges. It is important to note that inter-community edges connect vertices in different classes together. This capability is crucial because it can prevent labels from erroneously flowing beyond its own class. Second, we evaluate how the centralities can support transductive classifiers to gain higher accuracy.

The first experiment evaluates the proposed centrality which excludes flows from endpoints, C_{B1}, and the conventional centrality, C_{B0}, by using them to rank edges of a network. The inter-community edges are expected to be ranked at the beginning positions of the output. The evaluation metric used in this experiment was the *normalized discounted cumulative gain* (NDCG), proposed by [14]. It is a well-known evaluation metric for information retrieval. It is defined as

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}}, \tag{15}$$

$$\mathrm{DCG} = \sum_{i=1}^{|Q|} \frac{2^{\mathrm{r}(i)} - 1}{\log_2(i+1)},$$

$$\mathrm{IDCG} = \sum_{i=1}^{|R|} \frac{1}{\log_2(i+1)},$$

where $|Q|$ is the size of a ranking result, $|R|$ is the number of inter-community edges, and $\mathrm{r}(i)$ is the relevance score of the i^{th} edge of the result. The relevance score is binary, $\mathrm{r}(i) \in \{0, 1\}$. A relevance score $\mathrm{r}(i)$ is 1 if and only if the i^{th} edge is an inter-community edge, otherwise zero. NDCG ranges from 0.0 to 1.0. NDCG assumes that it is less useful when an inter-community edge is ranked at a lower position of the result. Thus, NDCG penalizes the relevance score logarithmically proportional to the position of the edge. A higher NDCG indicates much better performance. The Zachary's karate club, dolphins, and political blogs networks were used in this experiment because they contain complete label information of vertices. Since the datasets are homogeneous networks, this means that only the property regarding edges' endpoints of C_{B1} was tested. The results are shown in Table 2. They indicate that, by excluding endpoints, the edge betweenness is more likely to assign a high value to inter-community edges that help to prevent erroneous flows. With this property and Eq.(4), normalized weights of inter-community edges tend to be lower than those of intra-community edges.

Table 2: Comparison of NDCG calculated from lists of edges that were sorted according to the proposed centrality C_{B1} which excludes flows from/to endpoints and the conventional homogeneous edge betweenness centralities C_{B0}.

Network	NDCG	
	Excl. endpoints (C_{B1})	Incl. endpoints (C_{B0})
Karate	0.812	0.740
Dolphin	0.804	0.801
Political blogs	0.847	0.820

Next, four variants of edge betweenness were evaluated on the four-area network. The normalizing function of Eq.(4), C(.), is substituted with the following four measures:

Heterogeneous (C_{B1}): $1 + \sum_{s \in V_i \setminus \mathrm{EP}(e)} \sum_{t \in V_j \setminus \mathrm{EP}(e)} \frac{\sigma(s,t|e)}{\sigma(s,t)}$

Heterogeneous incl. endpoints: $1 + \sum_{s \in V_i} \sum_{t \in V_j} \frac{\sigma(s,t|e)}{\sigma(s,t)}$

Homogeneous (C_{B0}): $\sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}$

Without normalization: 1.

This experiment evaluated the classification ability of the proposed method with the four normalizing functions on the four-area network. Labeled authors were randomly chosen and added to L. The sizes of L were set to 5%, 10%, 15%, and 20% of authors in the network. The parameters λ_{ij} and α_i were set to 0.2 and 0.1, for $i, j \in \{1, \cdots, 4\}$, respectively, as previous studies [15, 16, 17]. Even though these are not the best parameter settings, but they are effective enough to demonstrate capabilities of centralities. Five independent iterations of the experiment were conducted. We allowed optimization methods to run until L^2 norm of the difference between soft label vectors at the present step and the previous one become less than 0.001.

The accuracies of methods are plotted in Figure 6. The results suggest that the centrality C_{B1} employing our proposed refinements can help the method to achieve a relatively higher accuracy. The method without normalization yields slightly lower accuracy, especially when seed vertices are limited. Thus, from now on, C_{B1} is used as the normalizing function, by plugging it into Eq.(4).

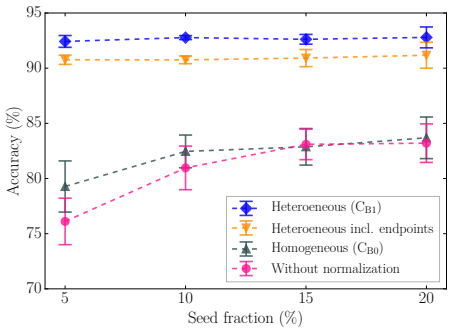


Figure 6: Mean accuracy (± 1 SD) of the proposed method, with four normalizing functions, evaluated on the four-area network where seed vertices are randomly 5%, 10%, 15%, and 20% of authors in the network.

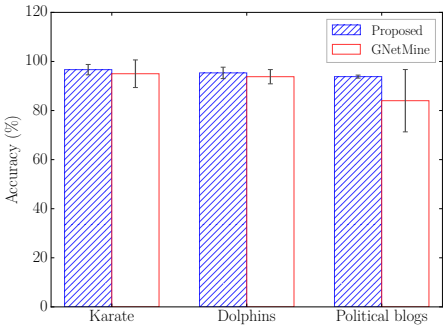
4.3 Comparison of Methods

We compared the accuracies of the proposed method, using C_{B1}, and two state-of-the-art transductive classifiers on various datasets and settings. The proposed method shares several common hypothesis with GNetMine [16], except the edge weight normalization. Therefore, GNetMine was treated as one of the baselines. Some readers doubt the advantage of generalizing transductive classification to heterogeneous networks. This experiment aims to clarify this point by adding LGC [26], a transductive classifier designed for homogeneous networks, as another baseline system. As in the previous experiment, the parameters λ_i and α_{ij} were set to 0.2 and 0.1 respectively, for all i, j and methods. In all experiments, the label information was the same for every method.

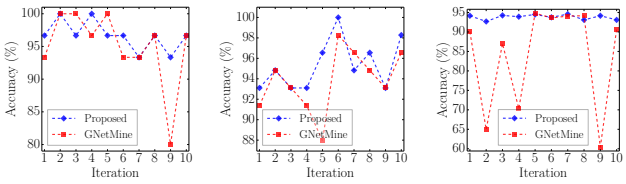
Next, the proposed method and baselines were evaluated on the Zachary's karate club, dolphin, and political blogs networks. Four random vertices were selected and added to L. At least one labeled vertex was guaranteed to be present in each class. The experiment was repeated ten times independently. The networks are homogeneous, hence GNetMine reduces to LGC.

Figure 7(a) compares the accuracies and standard deviations of the proposed method and GNetMine. On average, the proposed method achieves better accuracy. According to the paired one tailed t-test, the differences are statistically significant (p=0.022) in the political blogs network but not in the Zachary's karate club (p=0.150) and dolphin (p=0.054) networks. It is possible to hypothesize that the insignificance is likely to occur in a small network. Figure 7(b), 7(c), and 7(d) present the accuracy obtained in each repetition of experiments on Zachary's karate, dolphin, and political blogs networks, respectively. The detailed results reveal that one important benefit of the proposed method is its stability. The proposed method gains an equivalent level of accuracy in every trial of the experiments but the degree-based methods fails in some repetitions.

Figure 8(a) presents the four seed vertices used in the fifth iteration of the dolphin network experiment reported in Figure 7(c). Figure 8(b) illustrates edge betweenness centralities of edges. A darker line means a higher edge betweenness. Vertex 36 is one of the seed vertices. The blue label could flow into vertex 39. Since vertex 39 is a low degree vertex, based on degree-based normalization, it is highly likely

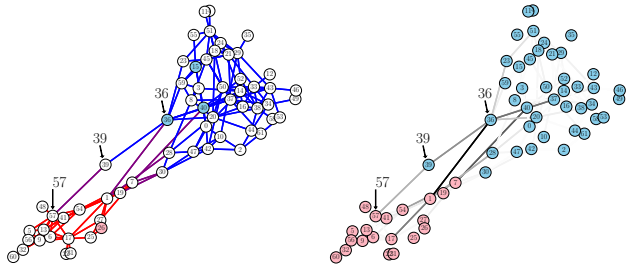


(a) Mean accuracy (± 1 SD)



(b) Karate (c) Dolphin (d) Political blogs

Figure 7: Mean accuracy (± 1 SD) of the proposed method and GNetMine with four seed vertices on three homogeneous networks (a) and the raw accuracy evaluated in each independent iteration of experiments on Zachary's karate club (b) , dolphin (c), and political blogs (d) networks.

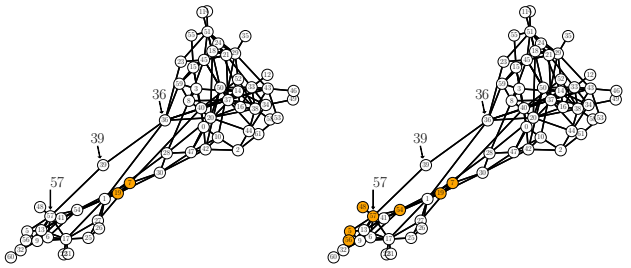


(a) Seed vertices (b) Edge betweenness

Figure 8: Setting of the fifth iteration of experiments on dolphin network, reported in Figure 7(c). (a) The blue and red vertices are seed vertices colored according to their initial labels. The white vertices are unlabeled. The blue and red edges are intra-community edges. The purple edges are inter-community edges. (b) A darker color of an edge shows its higher edge betweenness centrality. The colors of vertices indicate their ground-truth labels.

that the blue label can flow further into vertex 57 and its neighbors. In contrast, the proposed method can limit the flow from vertices 39 to 57 because their connection has a relatively high edge betweenness centrality and only a small amount of information flows from vertex 36 to vertex 39. As shown in Figure 9, vertex 57 and its neighbors were incorrectly classified by GNetMine. In contrast, the proposed method can predict their true labels.

The last experiment for this question was done on the four-area dataset. Again, to create the label information,



(a) Proposed method (b) GNetMine

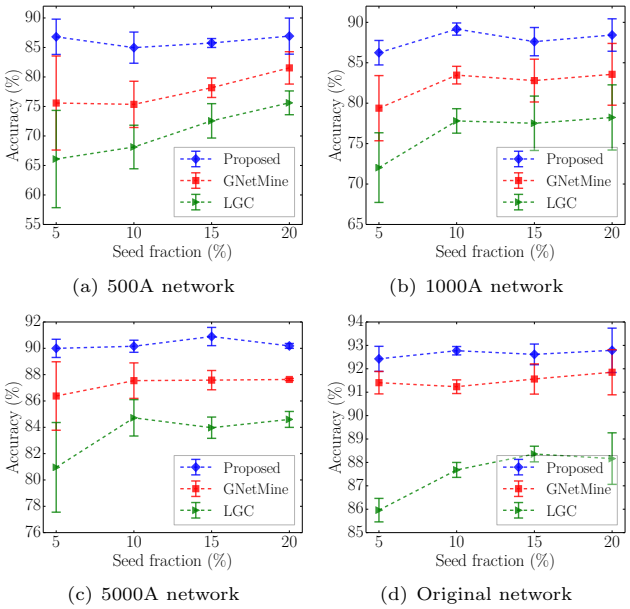
Figure 9: Incorrectly predicted vertices, which are orange, of the proposed method (a) and GNetMine (b) in the fifth iteration of the experiment on the dolphin network, reported in Figure 7(c).

Table 3: Properties of four variants of the four-area network

Measure	500A	1000A	5000A	Original
Vertices				
Authors	500	1,000	5,000	14,216
Papers	7,598	9,589	12,998	13,896
Conferences	18	18	18	18
Terms	6,067	7,004	8,453	8,785
Total	14,183	17,611	26,469	36,915
Edges				
Author-Paper	11,530	16,136	30,030	40,491
Term-Paper	59,925	75,798	103,440	110,770
Conf.-Paper	7,598	9,589	12,998	13,896
Total	79,053	101,523	146,468	165,157
Avg. Degree	11.148	11.529	11.067	8.948
Diameter	8	8	8	8

5%, 10%, 15%, and 20% of author vertices were randomly chosen and put into L. In this experiment, five independent repetitions of experiments were conducted as well. Furthermore, the methods were tested on three sub-networks derived from the four-area dataset. The top 500, 1000, and 5000 authors with their corresponding papers, terms, and conferences were extracted. The term top authors means the number of their papers are larger than the others. In the following, the sub-networks will be referred to as 500A, 1000A, and 5000A networks, respectively. Table 3 summarizes these networks and the original network.

The results are shown in Figure 10. They demonstrate that the proposed method significantly outperforms baseline systems on all variants of the four-area network. Figure 10(a), 10(b), and 10(c) shows that the proposed method achieves more than 5 percentage points increase in accuracy from GNetMine when 5% of authors are initially labeled. Table 3 shows the average degrees of vertices in 500A, 1000A, and 5000A networks are higher than eleven. These mean our proposed method is strongly better than GNetMine in dense networks with a few seed vertices. GNetMine yields higher accuracies than LGC. The generalization to heterogeneous networks have a significant advantage in the real-world dataset. The proposed method shows it has great stability as the variance of its accuracy is low, which conforms to the previous results on homogeneous networks.



(a) 500A network (b) 1000A network

(c) 5000A network (d) Original network

Figure 10: Mean accuracy (± 1 SD) of the proposed method, GNetMine, and LC on four variants of the four-area network, where seed vertices are randomly 5%, 10%, 15%, and 20% of authors in the network.

5. CONCLUSION

We proposed a novel method for transductive classification on heterogeneous information networks. We have argued that the degree-based edge weight normalization is unsatisfactory when a degree of a vertex cannot be used to imply the reliability of its label. Instead, we proposed a normalization on the basis of edge-betweenness centrality, under the assumption that edges bridging across communities should be considered less reliable. This paper further refined the centrality in two ways. Additional refinements were proposed to make the centrality suitable for heterogeneous networks.

Experimental results have shown that the proposed centrality can distinguish inter-community and intra-community edges effectively. The results revealed that the two refinements have a true benefit on real-world networks. This has played a crucial role in helping the proposed classification method to outperform state-of-the-art methods, GNetMine and LGC. The proposed method gained a lower variance of accuracy over multiple choices of training data.

We studied the betweenness defined under the assumption that flows of information in networks follow the shortest paths. However, another possible assumption is that information flows across random paths rather than the shortest paths. Hence, *random-walk edge betweenness*, proposed by [19], would be worth studying. The main weakness of this study was the computational cost of calculating edge betweenness centrality. Even though it can be computed in $\mathcal{O}(nm)$ with the algorithm introduced by Brandes' algorithm [4], the complexity can be considered impractical in many circumstances. The algorithm needs to be further researched to improve its computational cost, for example, by unitizing a sophisticated approximation method.

6. REFERENCES

[1] L. A. Adamic and N. Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 36–43, New York, NY, USA, 2005. ACM.

[2] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th International Conference on World Wide Web*, pages 895–904. ACM, 2008.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[4] U. Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145, May 2008.

[5] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2010.

[6] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[7] A. Cuzzocrea, A. Papadimitriou, D. Katsaros, and Y. Manolopoulos. Edge betweenness centrality: A novel algorithm for qos-based topology control over wireless sensor networks. *Journal of Network and Computer Applications*, 35(4):1210–1217, July 2012.

[8] R. Dunn, F. Dudbridge, and C. M. Sanderson. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. *BMC bioinformatics*, 6:39, 2005.

[9] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[10] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Advances in Neural Information Processing Systems*, pages 585–593, 2009.

[11] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[12] C. Gong, D. Tao, K. Fu, and J. Yang. Relish: Reliable label inference via smoothness hypothesis. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.

[13] Y. Jacob, L. Denoyer, and P. Gallinari. Learning latent representations of nodes for classifying in heterogeneous social networks. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 373–382, New York, NY, USA, 2014. ACM.

[14] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 41–48, New York, NY, USA, 2000. ACM.

[15] M. Ji, J. Han, and 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 '11*, page 1298, New York, New York, USA, Aug. 2011. ACM Press.

[16] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao. Graph regularized transductive classification on heterogeneous information networks. In J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Machine Learning and Knowledge Discovery in Databases SE - 42*, volume 6321 of *Lecture Notes in Computer Science*, pages 570–586. Springer Berlin Heidelberg, 2010.

[17] C. Luo, R. Guan, Z. Wang, and C. Lin. Hetpathmine: A novel transductive classification algorithm on heterogeneous information networks. In M. de Rijke, T. Kenter, A. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, editors, *Advances in Information Retrieval SE - 18*, volume 8416 of *Lecture Notes in Computer Science*, pages 210–221. Springer International Publishing, 2014.

[18] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[19] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2002.

[21] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1st edition, 1996.

[22] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *Proceedings of the VLDB Endowment 2011*, 2011.

[23] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 945–952. MIT Press, 2002.

[24] P. P. Talukdar and K. Crammer. New regularized algorithms for transductive learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 442–457, Berlin, Heidelberg, 2009. Springer-Verlag.

[25] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[26] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16(16):321–328, 2004.

[27] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003.