Semi-Supervised Classification through the Bag-of-Paths Group Betweenness

Bertrand Lebichot, Ilkka Kivimäki, Kevin Françoisse & Marco Saerens, *Member, IEEE*

***Abstract*—This paper introduces a novel, well-founded, betweenness measure, called the Bag-of-Paths (BoP) betweenness, as well as its extension, the BoP group betweenness, to tackle semi-supervised classification problems on weighted directed graphs. The objective of semi-supervised classification is to assign a label to unlabeled nodes using the whole topology of the graph and the labeled nodes at our disposal. The BoP betweenness relies on a bag-of-paths framework [1] assigning a Boltzmann distribution on the set of all possible paths through the network such that long (high-cost) paths have a low probability of being picked from the bag, while short (low-cost) paths have a high probability of being picked. Within that context, the BoP betweenness of node j is defined as the sum of the a posteriori probabilities that node j lies in-between two arbitrary nodes i, k, when picking a path starting in i and ending in k. Intuitively, a node typically receives a high betweenness if it has a large probability of appearing on paths connecting two arbitrary nodes of the network. This quantity can be computed in closed form by inverting a $n \times n$ matrix where n is the number of nodes. For the group betweenness, the paths are constrained to start and end in nodes within the same class, therefore defining a within-class group betweenness for each class. Unlabeled nodes are then classified according to the class showing the highest group betweenness. Experiments on various real-world data sets show that BoP group betweenness performs competitively compared to all the tested state-of-the-art methods [2]–[5]. The benefit of the BoP betweenness is particularly noticeable when only a few labeled nodes are available.**

***Index Terms*—Graph and network analysis, network data, graph mining, betweenness centrality, kernels on graphs, semi-supervised classification.**

I. Introduction

THE goal of a classification task is to automatically assign data to predefined classes. Traditional pattern recognition, machine learning or data mining classification methods require large amounts of labeled training instances, which are often difficult to obtain. The effort required to label the data can be reduced using, for example, semi-supervised learning methods. This name comes from the fact that the used data is a mixture of data used for supervised and unsupervised learning (see, e.g., [6] for an introduction). In short, semi-supervised learning methods learn from both labeled and unlabeled instances. This allows to reduce the amount of labeled instances needed to achieve the same level of classification accuracy.

Graph-based semi-supervised classification has received a growing focus in recent years. The problem can be described as follows: given an input graph with some nodes labeled, the goal is to predict the missing node labels. This problem has numerous applications such as classification of individuals in social networks, categorization of linked documents (e.g. patents or scientific papers), or protein function prediction, to name a few. In this kind of application (as in many others), unlabeled data are usually available in large quantities and are easy to collect: friendship links can be recorded on Facebook, text documents can be crawled from the internet and DNA sequences of proteins are readily available from gene databases. Given a relatively small labeled data set and a large unlabeled data set, semi-supervised algorithms can infer useful information from both sources.

This paper tackles this problem of semi-supervised classification within the **bag-of-paths** (BoP) **framework**. This framework was originaly introduced in the context of distance computation on graphs [1], capturing its global structure with, as building block, network paths. This same framework was previously used in [7] for defining a covariance kernel on a graph. More precisely, we assume a weighted directed graph or network G where a transition cost is associated to each arc. We further consider a bag containing all the possible paths (or walks) between pairs of nodes in G. Then, a Boltzmann distribution, depending on a temperature parameter T, is defined on the set of paths such that long (high-cost) paths have a low probability of being picked from the bag, while short (low-cost) paths have a high probability of being picked. In this framework, the **BoP probabilities**, $\mathrm{P}(s = i, e = j)$, of sampling a path starting in node i and ending in node j can easily be computed in closed form by a simple $n \times n$ matrix inversion, where n is the number of nodes.

Within this context, a betweenness measure quantifying to which extent a node j is in-between two nodes i and k is defined. More precisely, the **BoP betweenness**, $\mathrm{bet}_j = \sum_{i=1}^{n}\sum_{k=1}^{n}\mathrm{P}(int = j|s = i, e = k)$, of a node j of interest is defined quite naturally as the sum of the a posteriori probabilities that node j (intermediate node) lies on a path between the two nodes i and k sampled from the graph bag-of-paths Boltzmann distribution. Intuitively, a node receives a high betweenness if it has a large probability of appearing on paths connecting two arbitrary nodes of the network.

For the **group betweenness**, the paths are constrained to start and end in nodes of the same class, therefore defining a group betweenness between classes, $\mathrm{gbet}_j(\mathcal{C}_i, \mathcal{C}_k) = \mathrm{P}(int = j|s \in \mathcal{C}_i, e \in \mathcal{C}_k)$. Unlabeled nodes are then classified according to the class showing the highest group betweenness when starting and ending within the same class.

In summary, this work has three main contributions:

The authors are with Universite Catholique de Louvain, ICTEAM & LSM (e-mail: bertrand.lebichot@uclouvain.be).

This work was partially supported by the Elis-IT project funded by the "Région wallonne". We thank this institution for giving us the opportunity to conduct both fundamental and applied research.

- It develops both a betweenness measure and a group betweenness measure from a well-founded theoretical framework, the bag-of-paths framework. These two measures can be easily computed in closed form.
- This group betweenness measure provides a new algorithm for graph-based semi-supervised classification.
- It assesses the accuracy of the proposed algorithm on thirteen standard data sets and compares it to state-of-the-art techniques. The obtained performances are competitive with the other graph-based semi-supervised techniques.

The main drawback of the proposed method is that it requires a matrix inversion and therefore does not scale to very large graphs. An approximate method relying on bounded walks (as in [8]) will be investigated and is left for further work.

In this paper, the **BoP classifier** (or just BoP) will refer to the semi-supervised classification algorithm based on the bag-of-paths group betweenness developed in Section V.

The paper is organized as follows. Section II introduces background and notations, mainly the bag-of-paths and the bag-of-hitting-paths models. Then, related work in semi-supervised classification is discussed in Section III. The bag-of-paths betweenness and group betweenness centralities are introduced in Section IV. This enables us to derive the BoP classifier in Section V. Then experiments involving the BoP classifier and classifiers discussed in the related work section will be performed in Section VI. Results and discussions of those experiments can be found in Section VI-C. Finally, Section VII concludes this paper and opens a reflection for further work.

II. Background and notation

This section aims to introduce the theoretical background and notation used in this paper. Furthermore, in order to understand the bag-of-paths betweenness of Section IV, the bag-of-paths and bag-of-hitting-paths frameworks [1] need to be reviewed first. Thus, graph-based semi-supervised classification will be discussed in Subsection II-A, then the bag-of-paths model will be introduced in Subsection II-B and, finally, the bag-of-hitting-paths model will be briefly described in Subsection II-C.

A. Graph-based semi-supervised classification

Consider a weighted directed graph or network, G, strongly connected with a set of n nodes $\mathcal{V}$ (or vertices) and a set of edges $\mathcal{E}$ (or arcs, links). The **adjacency matrix** of the graph, containing non-negative affinities between nodes, is denoted as $\mathbf{A}$, with elements $a_{ij} \geq 0$.

Moreover, to each edge between node i and j is associated a non-negative number $c_{ij} \geq 0$. This number represents the **immediate cost of transition** from node i to j. If there is no link between i and j, the cost is assumed to take a large value, denoted by $c_{ij} = \infty$. The **cost matrix C** is an $n \times n$ matrix containing the c_{ij} as elements. Costs are set independently of the adjacency matrix – they are quantifying the cost of a transition according to the problem at hand. Costs can, e.g., be set in function of some properties, or features, of the nodes or the arcs in order to bias the probability distribution of choosing a path. In the case of a social network, we may, for instance, want to bias the paths in function of the education level of the persons, therefore favoring paths visiting highly educated persons (see [1] for details). Now, if there is no reason to introduce a cost, we simply set $c_{ij} = 1$ (paths are penalized by their length) or $c_{ij} = 1/a_{ij}$ (in this case, a_{ij} is viewed as a conductance and c_{ij} as a resistance) – this last setting will be used in the experimental section.

When tackling the semi-supervised classification problem, we will also consider a set of classes, $\{\mathcal{C}_k\}_{k=1}^m$, with the number of classes equal to m. Each node is assumed to belong to at most one class since the class label can also be unknown. To represent the class memberships, a $n \times m$-dimensional indicator matrix, $\mathbf{Y}$, is used. On each of its rows, it contains, as entries, a 1 in column c when the corresponding node belongs to class c, and 0 otherwise (m zeros on line i if the node i is unlabeled). The c-th column of $\mathbf{Y}$ will be denoted $\mathbf{y}^c$ and contains the binary memberships ot the nodes to class c.

Moreover, a **natural random walk** on G is defined in the standard way. In node i, the random walker chooses the next edge to follow according to reference transition probabilities

$$p_{ij}^{\text{ref}} = \frac{a_{ij}}{\sum_{j'=1}^{n} a_{ij'}} \tag{1}$$

representing the probability of jumping from node i to node $j \in \mathcal{S}ucc(i)$, the set of successor nodes of i. The corresponding transition probabilities matrix will be denoted as $\mathbf{P}^{\text{ref}}$. In other words, the random walker chooses to follow an edge with a probability proportional to the affinity (apart from the sum-to-one normalization), therefore favoring edges associated to a large affinity. The matrix $\mathbf{P}^{\text{ref}}$, containing the p_{ij}^{ref}, is stochastic and is called the **reference transition matrix**.

B. The bag-of-paths framework

This framework was recently introduced in [1] (see also [7]) for computing distances on graphs. In order to make the paper as self-contained as possible, we will briefly review the whole framework in this section (see [1] for details) and then use it in order to define a new betweenness measure, which is the main contribution of the paper. The bag-of-paths (BoP) model can be considered as a motif-based model [9], [10] using, as building block, paths of the network. In the next subsection, hitting paths will be used instead, as motifs.

A path $\wp$ (sometimes called a walk) is a sequence of transitions to adjacent nodes on G (loops are allowed), initiated from a starting node s, and stopping in an ending node e. If we want to emphasise on those starting and ending nodes, we will use $\wp_{se}$ instead of $\wp$.

The BoP framework is based on the probability of picking a path $i \rightsquigarrow j$ starting at a node i and ending in a node j from a virtual bag containing all possible paths in the network [1]. Let us define $\mathcal{P}_{ij}$ as the set of all possible paths connecting node i to node j, including loops. We further define the set of all paths through the graph as $\mathcal{P} = \bigcup_{i,j=1}^{n} \mathcal{P}_{ij}$. The **total cost** of a path $\wp$, $\tilde{c}(\wp)$, is defined as the sum of the individual transition costs c_{ij} along $\wp$.

The potentially infinite set of paths in the graph is enumerated and a probability distribution is assigned to the set of individual paths $\mathcal{P}$. This probability distribution on the set $\mathcal{P}$, represents the probability of picking a path $\wp \in \mathcal{P}$ in the bag, and is defined as the probability distribution P minimizing the total expected cost, $\mathbb{E}\left[\tilde{c}(\wp)\right]$, among all the distributions having a fixed relative entropy J_0 with respect to a reference distribution, for instance a natural random walk on the graph [1]. This choice naturally defines a probability distribution on the set of paths such that long (high cost) paths occur with a low probability while short (low cost) paths occur with a high probability. In other words, we are seeking path probabilities, $\mathrm{P}(\wp), \wp \in \mathcal{P}$, minimizing the total expected cost subject to a constant relative entropy constraint:

$$\left| \begin{array}{ll} \underset{\mathrm{P}(\wp)}{\text{minimize}} & \sum_{\wp \in \mathcal{P}} \mathrm{P}(\wp)\, \tilde{c}(\wp) \\ \text{subject to} & \sum_{\wp \in \mathcal{P}} \mathrm{P}(\wp) \ln(\mathrm{P}(\wp)/\tilde{\pi}^{\mathrm{ref}}(\wp)) = J_0 \\ & \sum_{\wp \in \mathcal{P}} \mathrm{P}(\wp) = 1 \end{array} \right. \tag{2}$$

where $\tilde{\pi}^{\mathrm{ref}}$ represents the probability of following the path $\wp$ when walking according to the natural random walk reference distribution. Thus $\tilde{\pi}^{\mathrm{ref}}$ is the product of the transition probabilities p_{ij}^{ref} along the path $\wp$. Here, $J_0 > 0$ is provided a priori by the user, according to the desired degree of randomness, or exploration, he is willing to concede.

The result of the minimization [1] is a **Boltzmann probability distribution**:

$$\mathrm{P}(\wp) = \frac{\tilde{\pi}^{\mathrm{ref}}(\wp) \exp\left[-\theta \tilde{c}(\wp)\right]}{\sum_{\wp' \in \mathcal{P}} \tilde{\pi}^{\mathrm{ref}}(\wp') \exp[-\theta \tilde{c}(\wp')]}, \tag{3}$$

where $\theta = 1/T$ plays the role of an inverse temperature. As expected, short paths $\wp$ (having a low $\tilde{c}(\wp)$) are favored in that they have a larger probability of being chosen. Moreover, from Equation (3), we clearly observe that when $\theta \to 0^+$, the paths probabilities reduce to the probabilities generated by the natural random walk on the graph. In this case, $J_0 \to 0$ and paths are chosen according to their likelihood in a natural random walk. On the other hand, when θ is large, the probability distribution defined by Equation (3) is biased towards short paths (paths shortest ones are more likely). Notice that, in the sequel, it will be assumed that the user provides the value of the parameter θ instead of J_0, with $\theta > 0$.

The **bag-of-paths probability** [1] is now defined as the quantity $\mathrm{P}(s = i, e = j)$ on the set of (starting, ending) nodes of the paths. It corresponds to the probability of drawing a path starting in node i and ending in node j from the virtual bag-of-paths:

$$\mathrm{P}(s = i, e = j) = \frac{\sum_{\wp \in \mathcal{P}_{ij}} \tilde{\pi}^{\mathrm{ref}}(\wp) \exp[-\theta \tilde{c}(\wp)]}{\sum_{\wp' \in \mathcal{P}} \tilde{\pi}^{\mathrm{ref}}(\wp') \exp[-\theta \tilde{c}(\wp')]}, \tag{4}$$

where $\mathcal{P}_{ij}$ is the set of paths connecting the starting node i to the ending node j.

Let us derive the analytical closed form of this expression. To this end, we start from the cost matrix, $\mathbf{C}$, from which we build a new matrix, $\mathbf{W}$, as

$$\mathbf{W} = \mathbf{P}^{\mathrm{ref}} \circ \exp[-\theta \mathbf{C}], \tag{5}$$

where $\mathbf{P}^{\mathrm{ref}}$ is the reference transition matrix containing the p_{ij}^{ref}, the exponential function is taken elementwise and $\circ$ is the elementwise multiplication (Hadamard product). The entries of $\mathbf{W}$ are therefore $w_{ij} = p_{ij}^{\mathrm{ref}} \exp[-\theta c_{ij}]$.

It is shown in [1] that the numerator of Equation (4) is

$$\sum_{\wp \in \mathcal{P}_{ij}} \tilde{\pi}^{\mathrm{ref}}(\wp) \exp\left[-\theta \tilde{c}(\wp)\right] = \sum_{t=0}^{\infty} \left[\mathbf{W}^t\right]_{ij} = \left[\sum_{t=0}^{\infty} \mathbf{W}^t\right]_{ij}, \tag{6}$$

where, by convention, zero-length paths are taken into account and are associated to a zero cost. Computing the series of powers of $\mathbf{W}$ provides

$$\sum_{t=0}^{\infty} \mathbf{W}^t = (\mathbf{I} - \mathbf{W})^{-1} = \mathbf{Z}, \tag{7}$$

which converges if the spectral radius of $\mathbf{W}$ is less than 1, $\rho(\mathbf{W}) < 1$. Since the matrix $\mathbf{W}$ only contains non-negative elements, a sufficient condition for $\rho(\mathbf{W}) < 1$ is that the matrix is sub-stochastic, which is always achieved for $\theta > 0$ and at least one $c_{ij} > 0$ when $a_{ij} > 0$ (see Equation (5)), which is assumed for now. The matrix $\mathbf{Z}$ is called the **fundamental matrix** and z_{ij} is the element i, j of $\mathbf{Z}$.

Hence, following Equations (6-7), we finally obtain, for the numerator of Equation (4),

$$\sum_{\wp \in \mathcal{P}_{ij}} \tilde{\pi}^{\mathrm{ref}}(\wp) \exp\left[-\theta \tilde{c}(\wp)\right] = z_{ij}. \tag{8}$$

On the other hand, for the denominator of Equation (4), we have

$$\sum_{i,j=1}^{n} \sum_{\wp \in \mathcal{P}_{ij}} \tilde{\pi}^{\mathrm{ref}}(\wp) \exp\left[-\theta \tilde{c}(\wp)\right] = \sum_{i,j=1}^{n} z_{ij} \triangleq \mathcal{Z}, \tag{9}$$

where $\mathcal{Z}$ is called the **partition function**.

Therefore, from Equation (4), the probability of picking a path starting in i end ending in j in our bag of paths model is

$$\mathrm{P}(s = i, e = j) = \frac{z_{ij}}{\mathcal{Z}}, \text{ with } \mathbf{Z} = (\mathbf{I} - \mathbf{W})^{-1}. \tag{10}$$

Notice that $\mathrm{P}(s = i, e = j)$ is not symmetric[1]. We now turn to a variant of the bag-of-paths, the **bag-of-hitting-paths**.

C. The bag-of-hitting-paths framework

The idea behind the bag-of-hitting-paths model is the same as the bag-of-paths model but the set of paths is now restricted to paths in which the ending node does not appear more than once. In other words, no intermediate node on the path is allowed to be the ending node j (node j is made *absorbing*) and the motifs are now the hitting paths. Hitting paths will play

[1]For a symmetric variant in the case of undirected graphs [1], we can consider the probability of picking either $i \rightsquigarrow j$ or $j \rightsquigarrow i$, which is $\mathrm{P}(s = i, e = j) + \mathrm{P}(s = j, e = i)$.

an important role in the derivation of the BoP betweenness. Each non-hitting path $\wp_{ij} \in \mathcal{P}_{ij}$ can be split uniquely into two sub-paths, before hitting node j for the first time, $\wp_{ij}^{\text{h}} \in \mathcal{P}_{ij}^{\text{h}}$ (the set of all hitting paths), and after hitting node j, $\wp_{jj} \in \mathcal{P}_{jj}$ (see [1] for details). Notice the usage of the superscript h to refer to *hitting* paths. The composition of the two sub-paths is a valid path, where $\wp_{ij}^{\text{h}} \circ \wp_{jj} \in \mathcal{P}_{ij}$ is the concatenation of the two paths.

In the case of a bag containing hitting paths, the probability of picking a path $i \rightsquigarrow j$ is defined in a similar way as for non-hitting paths (Equation (4)),

$$\text{P}_{\text{h}}(s=i,e=j) = \frac{\sum_{\wp \in \mathcal{P}_{ij}^{\text{h}}} \tilde{\pi}^{\text{ref}}(\wp) \exp[-\theta \tilde{c}(\wp)]}{\sum_{\wp' \in \mathcal{P}^{\text{h}}} \tilde{\pi}^{\text{ref}}(\wp') \exp[-\theta \tilde{c}(\wp')]} \tag{11}$$

and is called the **bag-of-hitting-paths probability** distribution.

Now, since $\tilde{c}(\wp_{ij}) = \tilde{c}(\wp_{ij}^{\text{h}}) + \tilde{c}(\wp_{jj})$ and $\tilde{\pi}^{\text{ref}}(\wp_{ij}) = \tilde{\pi}^{\text{ref}}(\wp_{ij}^{\text{h}})\tilde{\pi}^{\text{ref}}(\wp_{jj})$ for any $\wp_{ij} = \wp_{ij}^{\text{h}} \circ \wp_{jj}$, we easily obtain

$$\begin{aligned} z_{ij} &= \sum_{\wp_{ij} \in \mathcal{P}_{ij}} \tilde{\pi}^{\text{ref}}(\wp_{ij}) \exp[-\theta \tilde{c}(\wp_{ij})] \\ &= \sum_{\wp_{ij}^{\text{h}} \in \mathcal{P}_{ij}^{\text{h}}} \sum_{\wp_{jj} \in \mathcal{P}_{jj}} \tilde{\pi}^{\text{ref}}(\wp_{ij}^{\text{h}}) \exp[-\theta \tilde{c}(\wp_{ij}^{\text{h}})] \\ &\qquad \times \tilde{\pi}^{\text{ref}}(\wp_{jj}) \exp[-\theta \tilde{c}(\wp_{jj})] \\ &= z_{ij}^{\text{h}} z_{jj}. \end{aligned} \tag{12}$$

From which we deduce $z_{ij}^{\text{h}} = \sum_{\wp \in \mathcal{P}_{ij}^{\text{h}}} \tilde{\pi}^{\text{ref}}(\wp) \exp[-\theta \tilde{c}(\wp)] = z_{ij}/z_{jj}$. Now, by analogy with Equation (10), but for hitting paths (Equation (11)),

$$\text{P}_{\text{h}}(s=i,e=j) = \frac{z_{ij}^{\text{h}}}{\sum_{i'=1}^{n}\sum_{j'=1}^{n} z_{i'j'}^{\text{h}}} = \frac{z_{ij}^{\text{h}}}{\mathcal{Z}_{\text{h}}} \tag{13}$$

and the partition function for the bag-of-hitting-paths is therefore

$$\mathcal{Z}_{\text{h}} = \sum_{i,j=1}^{n} z_{ij}^{\text{h}} = \sum_{i,j=1}^{n} \frac{z_{ij}}{z_{jj}}. \tag{14}$$

Finally, let us just mention that another derivation is available in [1], where it is further shown that z_{ij}^{h} can be interpreted as either

- The expected reward endorsed by an agent (the reward along a path $\wp$ being defined as $\exp[-\theta \tilde{c}(\wp)]$) when traveling from i to j along all possible paths $\wp \in \mathcal{P}_{ij}^{h}$ with probability $\tilde{\pi}^{\text{ref}}(\wp)$.
- The expected number of passages through node j for an evaporating, or killed, random walker starting in node i and walking according to the sub-stochastic transition probabilities $p_{ij}^{\text{ref}} \exp[-\theta c_{ij}]$.

Before deriving the BoP betweenness measure, let us consider some related work.

III. Related work

Graph-based semi-supervised classification has been the subject of intensive research in recent years and a wide range of approaches has been developed in order to tackle the problem [6], [11], [12]: Random-walk-based methods [13], [14], spectral methods [15], [16], regularization frameworks [4], [17]–[19], transductive and spectral SVM [20], to name a few. We will compare our method (the BoP) to some of those techniques, namely,

1) A simple alignment with the regularized laplacian kernel (RL) based on a sum of similarities, $\mathbf{K}\mathbf{y}_c$, where $\mathbf{K} = (\mathbf{I} + \lambda \mathbf{L})^{-1}$, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the laplacian matrix, $\mathbf{I}$ is the identity matrix, $\mathbf{D}$ is the generalized outdegree matrix, and $\mathbf{A}$ is the adjacency matrix of G [18], [21]. The similarity is computed for each class c in turn. Then, each node is assigned to the class showing the largest sum of similarities. The (scalar) parameter $\lambda > 0$ is the regularization parameter [8], [22].
2) A simple alignment with the regularized normalized laplacian kernel (RNL) based on a sum of similarities, $\mathbf{K}\mathbf{y}_c$, where $\mathbf{K} = (\mathbf{I} + \lambda \widetilde{\mathbf{L}})^{-1}$, and $\widetilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ is the normalized laplacian matrix [4], [23]. The assignment to the classes is the same as for the previous method. The regularized normalized laplacian approach seems less sensitive to the priors of the different classes than the un-normalized regularized laplacian approach (RL) [23].
3) A simple alignment with the regularized commute time kernel (RCT) based on a sum of similarities, $\mathbf{K}\mathbf{y}_c$, with $\mathbf{K} = (\mathbf{D} - \alpha \mathbf{A})^{-1}$ [4], [22]. The assignment to the classes is the same as for previous methods. Element i, j of this kernel can be interpreted as the discounted cumulated probability of visiting node j when starting from node i. The (scalar) parameter $\alpha \in]0, 1]$ corresponds to an evaporating or killed random walk where the random walker has a $(1 - \alpha)$ probability of disappearing at each step. This method provided the best results in a recent comparative study on semi-supervised classification [22].
4) The harmonic function (HF) approach [5], [11], is closely related to the regularization framework of RL and RNL. Furthermore, it is equivalent and provides the same results as the label propagation algorithm [12] and the wvRN (or pRN) classifier used by the Netkit software as a baseline [24], but Netkit is a more general-purpose toolbox able to tackle more complex situations [25]. As those three algorithms give the same results, we only report HF which appeared first in the litterature and is fastest. It is based on a structural contiguity measure that smoothes the predicted values and leads to a model having interesting interpretations in terms of electrical potential and absorbing probabilities in a Markov chain.
5) The random walk with restart (RWWR) classifier [3], [26] relies on random walks performed on the weighted graph seen as a Markov chain. More precisely, a group betweenness measure is derived for each class, based on the stationary distribution of a random walk restarting

from the labeled nodes belonging to a class of interest. Each unlabeled node is then assigned to the class showing maximal betweenness. In this version [22], the random walker has a probability $(1-\alpha)$ to be teleported – with a uniform probability – to a node belonging to the class of interest c.

6) The discriminative random walks approach ($\mathcal{D}$-walk or DW1; see [2]) also relies on random walks performed on the weighted graph. As for the RWWR, a group betweenness measure, based on passage times during random walks, is derived for each class. More precisely, a $\mathcal{D}$-walk is a random walk starting in a labeled node and ending when any node having the same label (possibly the starting node itself) is reached for the first time. During this random walk, the number of visits to any unlabeled node is recorded and corresponds to a group betweenness measure. As for the previous method, each unlabeled node is then assigned to the class showing maximal betweenness.
7) A modified version of the $\mathcal{D}$-walk (or DW2). The only difference is that all elements of the transition matrix $\mathbf{P}^{\text{ref}}$ (since the random walks is seen as a Markov chain) are multiplied by $\alpha \in]0,1]$ so that α can be seen as a probability of continuing the random walk at each time step (and so $(1-\alpha) \in [0,1[$ is the probability of stopping the random walk at each step. This defines a killed random walk since $\alpha\mathbf{P}^{\text{ref}}$ is now sub-stochastic.

All these methods will be compared to the bag-of-paths (BoP) developed in the next sections. Notice that the random walker of the random-walk-based methods usually follows too long – and thus irrelevant – paths into account: popular entries are therefore intrinsically favored [27], [28]. The bag-of-paths approach tackles this issue by putting a negative exponential term in (5) and part of its success can be imputed to this fact.

Some authors also considered bounded (or truncated) walks [8], [29], [30] and obtained promising results on large graphs. This approach could also be considered in our framework in order to tackle large networks; this will be investigated in further work.

Tong et al. suggested a method avoiding to take the inverse of an $n \times n$ matrix for computing the random walk with restart measure [26]. They reduce the computing time by partitioning the input graph into smaller communities. Then, a sparse approximate of the random walk with restart is obtained by applying a low rank approximation. This approach suffers from the fact that it adds a hyperparameter k (the number of communities) that depends on the network and is still untractable for large graphs with millions of nodes. On the other hand, the computing time is reduced by this same factor k. This is another track to investigate in further work.

Herbster et al. [31] proposed a technique for fast label prediction on graphs through the approximation of the graph with either a minimum spanning tree or a shortest path tree. Once the tree has been extracted, the pseudo inverse of the laplacian matrix can be computed efficiently. The fast computation of the pseudo inverse enables to address prediction problems on large graphs. Finally, Tang and Liu have investigated relational learning via latent social dimensions [32]–[34]. They proposed to extract latent social dimensions based on network information (such as Facebook, Twitter,...) first, then they used these as features for discriminative learning (via an SVM, for example [32]). Their approach tackles very large networks and provides promising results, especially when only a few labeled data are available.

A lot of research has also been done on collective classification of nodes in networks (see [35] for an introduction). Collective classification also uses the graph topology and a proportion of labeled nodes to classify unlabeled nodes using the same assumption as our proposed technique (i.e, local consistency or homophily).

We also experimented a group betweenness using Freeman's, i.e. the shortest path, betweenness [36] and a modified version of Newman's betweenness [37]. For this latter one, the transition probabilities were set to $\mathbf{P}^{\text{ref}}$, and the ending node of the walk was forced to be absorbing. Then, the expected number of visits to each node was recorded and cumulated for each input-output path. However, our BoP group betweenness outperformed these two other class betweenness measures and consequently, results are not reported in this paper.

IV. The bag-of-paths betweennesses

In order to define the BoP classifier, we need to introduce the BoP group betweenness centrality. This concept is itself an extension of the BoP betweenness centrality, which will be developed in the next subsection. The BoP betweenness is related to well-known betweenness measures in some sense: if $\theta \to \infty$ the BoP betweenness tends to be correlated with Freeman's betweenness [36] (only shortest-paths are considered), while if $\theta \to 0^+$, the BoP betweenness tends to be correlated with Newman's betweenness [37] (based on a natural random walk). This section starts with the presentation of the BoP betweenness centrality measure in Subsection IV-A. Then, its extension, the BoP group betweenness centrality, is described in Subsection IV-B.

A. The bag-of-paths betweenness centrality

The BoP betweenness will measure to which extent a node j is likely to lie in-between other pairs of nodes (i,k), and therefore is an important intermediary between nodes. In short, the **bag-of-paths betweenness measure** is defined as

$$\text{bet}_j = \sum_{i=1}^{n}\sum_{k=1}^{n} \text{P}(int = j | s = i, e = k; i \neq j \neq k \neq i) \quad (15)$$

which corresponds to the a posteriori probability of finding intermediate node j on a path $i \rightsquigarrow j$ drawn from the bag of paths, cumulated over all source-destination pairs (i,k).

For computing this quantity from the bag-of-paths framework, we first have to calculate the probability $\text{P}(s = i, int = j, e = k; i \neq j \neq k \neq i)$ – or P_{ijk} in short – that such paths

visit an *intermediate* node $int = j$ with $i \neq j \neq k \neq i$. Indeed, by using Equations (3) and (4),

$$\begin{aligned}\mathrm{P}_{ijk} &= \sum_{\wp \in \mathcal{P}_{ik}} \delta(j \in \wp)\, \mathrm{P}(\wp) \\ &= \frac{\sum_{\wp \in \mathcal{P}_{ik}} \delta(j \in \wp)\, \tilde{\pi}^{\mathrm{ref}}(\wp) \exp\left[-\theta \tilde{c}(\wp)\right]}{\sum_{\wp' \in \mathcal{P}} \tilde{\pi}^{\mathrm{ref}}(\wp') \exp\left[-\theta \tilde{c}(\wp')\right]} \\ &= \frac{1}{\mathcal{Z}} \sum_{\wp \in \mathcal{P}_{ik}} \delta(j \in \wp)\, \tilde{\pi}^{\mathrm{ref}}(\wp) \exp\left[-\theta \tilde{c}(\wp)\right], \end{aligned} \tag{16}$$

where $\delta(j \in \wp) = 1$ when node j is visited on path $\wp$, and 0 otherwise.

We will now use the fact that each path $\wp_{ik}$ between i and k passing through j can be decomposed uniquely into a *hitting* sub-path $\wp_{ij}$ from i to j and a regular sub-path $\wp_{jk}$ from j to k (see Subsection II-C). The sub-path $\wp_{ij}$ is found by following path $\wp_{ik}$ until reaching j for the first time. Therefore, for $i \neq j \neq k \neq i$,

$$\begin{aligned}\mathrm{P}_{ijk} = \frac{1}{\mathcal{Z}} &\sum_{\wp_{ij}^{\mathrm{h}} \in \mathcal{P}_{ij}^{\mathrm{h}}} \sum_{\wp_{jk} \in \mathcal{P}_{jk}} \tilde{\pi}^{\mathrm{ref}}(\wp_{ij}^{\mathrm{h}}) \tilde{\pi}^{\mathrm{ref}}(\wp_{jk}) \\ &\times \exp\left[-\theta \tilde{c}(\wp_{ij}^{\mathrm{h}})\right] \exp\left[-\theta \tilde{c}(\wp_{jk})\right]. \end{aligned} \tag{17}$$

This equation can be reordered to get, for $i \neq j \neq k \neq i$:

$$\begin{aligned}\mathrm{P}_{ijk} = \frac{1}{\mathcal{Z}} &\left[\sum_{\wp_{ij}^{\mathrm{h}} \in \mathcal{P}_{ij}^{\mathrm{h}}} \tilde{\pi}^{\mathrm{ref}}(\wp_{ij}^{\mathrm{h}}) \exp\left[-\theta \tilde{c}(\wp_{ij}^{\mathrm{h}})\right]\right] \\ &\times \left[\sum_{\wp_{jk} \in \mathcal{P}_{jk}} \tilde{\pi}^{\mathrm{ref}}(\wp_{jk}) \exp\left[-\theta \tilde{c}(\wp_{jk})\right]\right]. \end{aligned} \tag{18}$$

Then, after multiplying by $\mathcal{Z}_{\mathrm{h}}/\mathcal{Z}_{\mathrm{h}}$, we obtain

$$\begin{aligned}\mathrm{P}_{ijk} &= \frac{1}{\mathcal{Z}} z_{ij}^{\mathrm{h}} z_{jk} = \mathcal{Z}_{\mathrm{h}} \frac{z_{ij}^{\mathrm{h}}}{\mathcal{Z}_{\mathrm{h}}} \frac{z_{jk}}{\mathcal{Z}} \\ &= \mathcal{Z}_{\mathrm{h}}\, \mathrm{P}_{\mathrm{h}}(s = i, e = j)\, \mathrm{P}(s = j, e = k), \end{aligned} \tag{19}$$

with $i \neq j \neq k \neq i$ and where we used Equations (12) and (13).

Finally, recalling Equations (10), (13),

$$\begin{aligned}\mathrm{P}_{ijk} &= \frac{\left(\frac{z_{ij}}{z_{jj}}\right)(z_{jk})}{\mathcal{Z}} \\ &= \frac{1}{\mathcal{Z}} \frac{z_{ij} z_{jk}}{z_{jj}}, \text{ with } i \neq j \neq k \neq i. \\ &= \frac{1}{\mathcal{Z}} \frac{z_{ij} z_{jk}}{z_{jj}}\, \delta(i \neq j \neq k \neq i) \end{aligned} \tag{20}$$

Now, using the Bayes's rule, the *a posteriori* probabilitiy of finding intermediate node j on a path starting in i and ending in k is

$$\begin{aligned}&\mathrm{P}(int = j | s = i, e = k; i \neq j \neq k \neq i) \\ &\quad = \frac{\mathrm{P}(s = i, int = j, e = k; i \neq j \neq k \neq i)}{\sum_{j'=1}^{n} \mathrm{P}(s = i, int = j', e = k; i \neq j' \neq k \neq i)}. \end{aligned}$$

Using Equation (20), if we assume that node k can be reached from node i, this leads to

$$\begin{aligned}&\mathrm{P}(int = j | s = i, e = k; i \neq j \neq k \neq i) \\ &\quad = \frac{\left(\frac{z_{ij} z_{jk}}{z_{jj}}\right)}{\sum_{\substack{j'=1 \\ j' \notin \{i,k\}}}^{n} \left(\frac{z_{ij'} z_{j'k}}{z_{j'j'}}\right)} \delta(i \neq j \neq k \neq i). \end{aligned} \tag{21}$$

Based on this a posteriori probability distribution, the *bag-of-paths betweenness* of node j is defined as the sum of the a posteriori probabilities of visiting j for all possible starting-ending pairs (i, k):

$$\mathrm{bet}_j = \sum_{i=1}^{n} \sum_{k=1}^{n} \mathrm{P}(int = j | s = i, e = k; i \neq j \neq k \neq i) \tag{22}$$

$$= \frac{1}{z_{jj}} \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{\substack{k=1 \\ k \notin \{i,j\}}}^{n} \frac{z_{ij} z_{jk}}{\sum_{\substack{j'=1 \\ j' \notin \{i,k\}}}^{n} \left(\frac{z_{ij'} z_{j'k}}{z_{j'j'}}\right)}, \tag{23}$$

which allows to compute the betweenness from the fundamental matrix $\mathbf{Z}$ (Equation (7)).

Let us now derive the matrix formula providing the betweenness vector $\mathbf{bet}$, containing the betweennesses for each node. First of all, the normalization factor appearing in the denominator of Equation (23), denoted here by n_{ik}, is computed,

$$n_{ik} = \sum_{j'=1}^{n} (1 - \delta_{ij'})(1 - \delta_{j'k})\ (z_{ij'} z_{j'k})/z_{j'j'}, \tag{24}$$

which can be re-written as

$$n_{ik} = \sum_{j'=1}^{n} \{(1 - \delta_{ij'}) z_{ij'}\} \{1/z_{j'j'}\} \{(1 - \delta_{j'k}) z_{j'k}\}. \tag{25}$$

Therefore, the matrix containing the normalization factors n_{ik} is

$$\mathbf{N} = (\mathbf{Z} - \mathbf{Diag}(\mathbf{Z}))\, (\mathbf{Diag}(\mathbf{Z}))^{-1} (\mathbf{Z} - \mathbf{Diag}(\mathbf{Z})), \tag{26}$$

where for a given matrix $\mathbf{M}$, $\mathbf{diag}(\mathbf{M})$ is a column vector containing the diagonal of $\mathbf{M}$ and $\mathbf{Diag}(\mathbf{M})$ is a diagonal matrix containing the diagonal of $\mathbf{M}$.

Moreover, the inner term appearing in Equation (23) can be rewritten as

$$\begin{aligned}&\sum_{i=1}^{n} \sum_{k=1}^{n} \delta(i \neq j \neq k \neq i) z_{ij} (1/n_{ik}) z_{jk} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{n} \{(1 - \delta_{ji}) z_{ji}^{\mathrm{t}}\} \{(1 - \delta_{ik})(1/n_{ik})\} \{(1 - \delta_{kj}) z_{kj}^{\mathrm{t}}\}, \end{aligned} \tag{27}$$

where z_{ij}^{t} is the element i, j of matrix $\mathbf{Z}^{\mathrm{T}}$ (transpose of $\mathbf{Z}$). In matrix form, $\mathbf{bet}$ (see Equation (21)) is therefore equal to

$$\begin{aligned}\mathbf{bet} = &(\mathbf{Diag}(\mathbf{Z}))^{-1} \mathbf{diag}\big[(\mathbf{Z}^{\mathrm{T}} - \mathbf{Diag}(\mathbf{Z})) \\ &\times (\mathbf{N}^{\div} - \mathbf{Diag}(\mathbf{N}^{\div}))(\mathbf{Z}^{\mathrm{T}} - \mathbf{Diag}(\mathbf{Z}))\big], \end{aligned} \tag{28}$$

with matrix $\mathbf{N}^{\div}$ containing elements $n_{ik}^{\div} = 1/n_{ik}$ (element-wise reciprocal).

B. The bag-of-paths group betweenness centrality

Let us now generalize the bag-of-paths betweenness to a group betweenness measure. Quite naturally, the **bag-of-paths group betweenness** of node j will be defined as

$$\text{gbet}_j(\mathcal{C}_i,\mathcal{C}_k) = \text{P}(int = j | s \in \mathcal{C}_i, e \in \mathcal{C}_k; s \neq int \neq e \neq s) \tag{29}$$

and can be interpreted as the extent to which the node j lies in-between the two sets of nodes $\mathcal{C}_i$ and $\mathcal{C}_k$. It is assumed that the sets $\{\mathcal{C}_i\}_{i=1}^m$ are disjoint. Using Bayes' law provides

$$\begin{aligned}
&\text{P}(int = j | s \in \mathcal{C}_i, e \in \mathcal{C}_k; s \neq int \neq e \neq s) \\
&= \frac{\text{P}(s \in \mathcal{C}_i, int = j, e \in \mathcal{C}_k; s \neq int \neq e \neq s)}{\text{P}(s \in \mathcal{C}_i, e \in \mathcal{C}_k; s \neq int \neq e \neq s)} \\
&= \frac{\displaystyle\sum_{i' \in \mathcal{C}_i} \sum_{k' \in \mathcal{C}_k} \text{P}(s = i', int = j, e = k'; s \neq int \neq e \neq s)}{\displaystyle\sum_{j'=1}^{n} \sum_{i' \in \mathcal{C}_i} \sum_{k' \in \mathcal{C}_k} \text{P}(s = i', int = j', e = k'; s \neq int \neq e \neq s)}.
\end{aligned} \tag{30}$$

Substituting (20) for the joint probabilities in Equation (30) allows to compute the group betweenness measure in terms of the elements of the fundamental matrix $\mathbf{Z}$:

$$\begin{aligned}
&\text{gbet}_j(\mathcal{C}_i,\mathcal{C}_k) \\
&= \frac{\displaystyle\sum_{i' \in \mathcal{C}_i} \sum_{k' \in \mathcal{C}_k} \delta(i' \neq j \neq k' \neq s) \frac{z_{i'j} z_{jk'}}{z_{jj}}}{\displaystyle\sum_{j'=1}^{n} \sum_{i' \in \mathcal{C}_i} \sum_{k' \in \mathcal{C}_k} \delta(i' \neq j' \neq k' \neq s) \frac{z_{i'j'} z_{j'k'}}{z_{j'j'}}},
\end{aligned} \tag{31}$$

where the denominator is simply a normalization factor ensuring that the probability distribution sums to one. It is therefore sufficient to compute the numerator only and then normalize the resulting quantity.

Let us put this expression in matrix form. As before, we denote element i, j of matrix $\mathbf{Z}^{\text{T}}$ as z^{t}_{ij}. It is also assumed that node i' and k' belong to different groups, $\mathcal{C}_i \neq \mathcal{C}_k$, so that i and k are necessarily different (classes are disjoint). The numerator in Equation (31) is

$$\begin{aligned}
&\text{num}(\text{gbet}_j(\mathcal{C}_i,\mathcal{C}_k)) \\
&= \frac{1}{z_{jj}} \sum_{i' \in \mathcal{C}_i} \sum_{k' \in \mathcal{C}_k} (1 - \delta_{ji'})(1 - \delta_{jk'})\, z_{i'j} z_{jk'} \\
&= \frac{1}{z_{jj}} \left(\sum_{i' \in \mathcal{C}_i} (1 - \delta_{ji'}) z^{\text{t}}_{ji'} \right) \left(\sum_{k' \in \mathcal{C}_k} (1 - \delta_{jk'}) z_{jk'} \right).
\end{aligned} \tag{32}$$

If y_i^c is a binary indicator indicating if node i belongs to the class c (as described in Section II-A), the numerator can be rewritten as

$$\begin{aligned}
&\text{num}(\text{gbet}_j(\mathcal{C}_i,\mathcal{C}_k)) \\
&= \frac{1}{z_{jj}} \left(\sum_{i'=1}^{n} (1 - \delta_{ji'}) z^{\text{t}}_{ji'} y^i_{i'} \right) \left(\sum_{k'=1}^{n} (1 - \delta_{jk'}) z_{jk'} y^k_{k'} \right).
\end{aligned} \tag{33}$$

Consequently, in matrix form, the group betweenness vector reads

$$\begin{cases}
\mathbf{gbet}(\mathcal{C}_i,\mathcal{C}_k) \leftarrow (\mathbf{Diag}(\mathbf{Z}))^{-1} \left((\mathbf{Z}_0^{\text{T}} \mathbf{y}^i) \circ (\mathbf{Z}_0 \mathbf{y}^k) \right) \\
\qquad\qquad \text{with } \mathbf{Z}_0 = \mathbf{Z} - \mathbf{Diag}(\mathbf{Z}), \\
\mathbf{gbet}(\mathcal{C}_i,\mathcal{C}_k) \leftarrow \dfrac{\mathbf{gbet}(\mathcal{C}_i,\mathcal{C}_k)}{\|\mathbf{gbet}(\mathcal{C}_i,\mathcal{C}_k)\|_1} \text{ (normalization)}
\end{cases} \tag{34}$$

where we assume $i \neq k$. In this equation, the vector $\mathbf{gbet}(\mathcal{C}_i,\mathcal{C}_k)$ must be normalized by dividing it by its L_1 norm. Notice that $\mathbf{Z}_0 = \mathbf{Z} - \mathbf{Diag}(\mathbf{Z})$ is simply the fundamental matrix whose diagonal is set to zero.

V. Semi-supervised classification through the bag-of-paths group betweenness

In this section, the bag-of-paths model, and more precisely the bag-of-paths group betweenness measure, will be used for *classification purposes*. Notice, however, that in the derivation of the group betweenness measure (see Equation (34)), it was assumed that the starting and ending classes are different ($\mathcal{C}_i \neq \mathcal{C}_k$). We will now recompute this quantity when starting and ending in the same class c, i.e. calculating $\text{gbet}_j(\mathcal{C}_c,\mathcal{C}_c)$, which provides a **within-class betweenness**. Indeed, this quantity measures the extent to which nodes of G are in-between – and therefore in the neighborhood of – the nodes of class c.

A within-class betweenness is thus computed for each class c and each node will define assigned to the class showing the highest betweenness. This will be our simple classification rule based on the within-class betweenness. The main hypothesis underlying this classification technique is that a node is likely to belong to the same class as its "neighboring nodes". This is usually called the local consistency assumption (also called smoothness, homophily or cluster assumption [5], [12], [38]).

The same reasoning as for deriving Equation (34) is applied in order to compute the numerator of (31) in this new case. We start with Equation (31), considering now the same starting and ending class c but multiplying the term inside the double sum by $(1 - \delta_{i'k'})$. This new term will ensure that the starting node is different from the ending node (this was always the case with different starting and ending classes, but now this must be forced). From Equation (32), this can be rewritten as

$$\begin{aligned}
\text{num}(\text{gbet}_j(\mathcal{C}_c,\mathcal{C}_c)) = \frac{1}{z_{jj}} &\sum_{i',k' \in \mathcal{C}_c} (1 - \delta_{ji'})(1 - \delta_{i'k'}) \\
&\times (1 - \delta_{jk'})\, z_{i'j} z_{jk'}.
\end{aligned} \tag{35}$$

$\text{num}(\text{gbet}_j(\mathcal{C}_c,\mathcal{C}_c))$ is thus the same as $\text{num}(\text{gbet}_j(\mathcal{C}_i,\mathcal{C}_k))$ with $\mathcal{C}_i = \mathcal{C}_c$ of Equation (32) and $\mathcal{C}_k = \mathcal{C}_c$, plus an extra term:

$$\begin{aligned}
&\text{num}(\text{gbet}_j(\mathcal{C}_c,\mathcal{C}_c)) = \\
&\quad \frac{1}{z_{jj}} \sum_{i' \in \mathcal{C}_c} \sum_{k' \in \mathcal{C}_c} (1 - \delta_{ji'})(1 - \delta_{jk'})\, z_{i'j} z_{jk'} \\
&\quad - \frac{1}{z_{jj}} \sum_{i' \in \mathcal{C}_c} \sum_{k' \in \mathcal{C}_c} (1 - \delta_{ji'}) \delta_{i'k'} (1 - \delta_{jk'})\, z_{i'j} z_{jk'}.
\end{aligned} \tag{36}$$

TABLE I
CLASS DISTRIBUTION OF THE NINE *Newsgroups* DATA SETS. NG 1-3 CONTAIN TWO CLASSES, NG 4-6 CONTAIN THREE CLASSES AND NG 7-9 CONTAIN FIVE CLASSES.

Class	NG1	NG2	NG3	NG4	NG5	NG6	NG7	NG8	NG9
1	200	198	200	200	200	197	200	200	200
2	200	200	199	200	198	200	200	200	200
3				200	200	198	200	198	197
4							200	200	200
5							198	200	200
Total	400	398	399	600	598	595	998	998	997

It is easy to show that with this additional term, the matrix equation for $\text{num}(\mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c))$ (Equation (34)) becomes

$$\text{num}(\mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c)) = (\mathbf{Diag}(\mathbf{Z}))^{-1}\left[(\mathbf{Z}_0^{\mathrm{T}}\mathbf{y}^c) \circ (\mathbf{Z}_0\mathbf{y}^c) - (\mathbf{Z}_0^{\mathrm{T}} \circ \mathbf{Z}_0)\mathbf{y}^c\right]. \quad (37)$$

Once again, this is the same result as in Equation (34) with one more term that ensure that the starting node is different from the ending node. After having computed this equation, the numerator must be normalized in order to obtain $\mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c)$ (as for Equation (34)).

Finally, if we want to classify a node, $\mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c)$ is computed for each class c in turn and then, for each node, the class label showing the maximal betweenness is chosen,

$$\hat{\ell} = \arg\max_{c \in \mathcal{L}} \left(\mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c)\right), \text{ with}$$
$$\begin{cases} \mathbf{D}_z = \mathbf{Diag}(\mathbf{Z}); \mathbf{Z}_0 = \mathbf{Z} - \mathbf{D}_z \text{ (set diagonal to 0)} \\ \mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c) \leftarrow \mathbf{D}_z^{-1}\left[(\mathbf{Z}_0^{\mathrm{T}}\mathbf{y}^c) \circ (\mathbf{Z}_0\mathbf{y}^c) - (\mathbf{Z}_0^{\mathrm{T}} \circ \mathbf{Z}_0)\mathbf{y}^c\right] \\ \mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c) \leftarrow \dfrac{\mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c)}{\|\mathbf{gbet}(\mathcal{C}_c, \mathcal{C}_c)\|_1} \text{ (normalization)} \end{cases} \quad (38)$$

where $\mathcal{L}$ is the set of class labels. The pseudo-code for the BoP classifier can be found in Algorithm 1. Of course, once computed, the group betweenness is only used to classify the unlabeled nodes.

VI. EXPERIMENTAL COMPARISONS

In this section, the bag-of-paths group betweenness approach for semi-supervised classification (referred to as the BoP classifier for simplicity) will be compared to other semi-supervised classification techniques on multiple data sets. The different classifiers to which the BoP classifier will be compared were already introduced in Section III and are recalled in Table IV.

TABLE II
CLASS DISTRIBUTION OF THE *IMDb-proco* DATA SET.

Class	IMDb
High-revenue	572
Low-revenue	597
Total	1169

The goal of the experiments of this section is to classify unlabeled nodes in partially labeled graphs and to compare the different methods in terms of classification accuracy. This comparison is performed on medium-size networks only since kernel approaches are difficult to compute on large networks. The computational tractability of the methods used in this experimental section will also be analyzed.

This section is organized as follows. First, the data sets used for the semi-supervised classification will be described

Algorithm 1 Classification through the bag-of-paths group betweenness algorithm.

Input:
– A weighted directed graph G containing n nodes, represented by its $n \times n$ adjacency matrix $\mathbf{A}$, containing affinities.
– The $n \times n$ transition cost matrix $\mathbf{C}$ associated to G.
– m binary indicator vectors $\mathbf{y}_c$ containing as entries 1 for nodes belonging to the class $\mathcal{C}_c$, and 0 otherwise. Classes are mutually exclusive.
– The inverse temperature parameter θ.
Output:
– The $n \times 1$ vector $\hat{\ell}$ containing the predicted class labels of each node.

$\mathbf{D} \leftarrow \mathbf{Diag}(\mathbf{Ae})$ {the row-normalization matrix}
$\mathbf{P}^{\text{ref}} \leftarrow \mathbf{D}^{-1}\mathbf{A}$ {the reference transition probabilities matrix}
$\mathbf{W} \leftarrow \mathbf{P}^{\text{ref}} \circ \exp[-\theta\mathbf{C}]$ {elementwise exponential and multiplication $\circ$}
$\mathbf{Z} \leftarrow (\mathbf{I} - \mathbf{W})^{-1}$ {the fundamental matrix}
$\mathbf{Z}_0 \leftarrow \mathbf{Z} - \mathbf{Diag}(\mathbf{Z})$ {set diagonal to zero}
$\mathbf{D}_z \leftarrow \mathbf{Diag}(\mathbf{Z})$
for $c = 1$ to m **do**
  $\hat{\mathbf{y}}_c^* \leftarrow \mathbf{D}_z^{-1}\left[(\mathbf{Z}_0^{\mathrm{T}}\mathbf{y}^c) \circ (\mathbf{Z}^0\mathbf{y}^c) - (\mathbf{Z}_0^{\mathrm{T}} \circ \mathbf{Z}_0)\mathbf{y}^c\right]$ {compute the group betweenness for class c; $\circ$ is the elementwise multiplication (Hadamard product)}
  $\hat{\mathbf{y}}_c^* \leftarrow \dfrac{\hat{\mathbf{y}}_c^*}{\|\hat{\mathbf{y}}_c^*\|_1}$ {normalize the betweenness scores}
end for
$\hat{\ell} \leftarrow \arg\max_{c \in \mathcal{L}}(\hat{\mathbf{y}}_c^*)$ {each node is assigned to the class showing the largest class betweenness}
return $\hat{\ell}$

TABLE III
CLASS DISTRIBUTION OF THE FOUR *WebKB cocite* DATA SETS.

Class	Cornell	Texas	Washington	Wisconsin
Course	54	51	170	83
Department	25	36	20	37
Faculty	62	50	44	37
Project	54	28	39	25
Staff	6	6	10	11
Student	145	163	151	155
Total	346	334	434	348
Majority class (%)	41.9	48.8	39.2	44.5

TABLE IV
THE EIGHT CLASSIFIERS, THE VALUE RANGE TESTED FOR TUNING THEIR PARAMETERS AND THE MOST SELECTED VALUES. MODE1 IS THE MOST SELECTED VALUE, MODE2 IS THE SECOND MOST SELECTED VALUE AND MODE3 IS THE THIRD MOST SELECTED VALUE. NOTICE THAT DW2 WITH $\alpha = 1.0$ IS THE SAME AS DW1.

Classifier name	Acronym	Parameter	Tested values	Mode1	Mode2	Mode3
Regularized laplacian kernel	RL	$\lambda > 0$	$10^{-6}, 10^{-5}, ..., 10^{6}$	$10^{-9}(12.3\%)$	$10^{-1}(11.7\%)$	$10^{-2}(11.5\%)$
Regularized normalised laplacian kernel	RNL	$\lambda > 0$	$10^{-6}, 10^{-5}, ..., 10^{6}$	$10^{-1}(42.1\%)$	$10^{-2}(13.9\%)$	$10^{-3}(09.3\%)$
Label Propagation	LP	none	/	/	/	/
Regularized commute-time kernel	RCT	$\alpha \in]0, 1]$	$0.1, 0.2, ..., 1$	$0.9(28.0\%)$	$0.8(16.2\%)$	$0.7(12.2\%)$
Harmonic function	HF	none	/	/	/	/
Random walk with restart	RWWR	$\alpha \in]0, 1]$	$0.1, 0.2, ..., 1$	$0.9(45.8\%)$	$0.8(16.8\%)$	$0.7(10.1\%)$
Discriminative random walks	DW1	none	/	/	/	/
Killed discriminative random walks	DW2	$\alpha \in]0, 1]$	$0.1, 0.2, ..., 1$	$1.0(19.5\%)$	$0.1(11.8\%)$	$0.9(11.2\%)$
BoP classifier	BoP	$\theta > 0$	$10^{-6}, 10^{-5}, ..., 10^{2}$	$10^{-4}(28.3\%)$	$10^{-3}(25.9\%)$	$10^{-2}(12.3\%)$

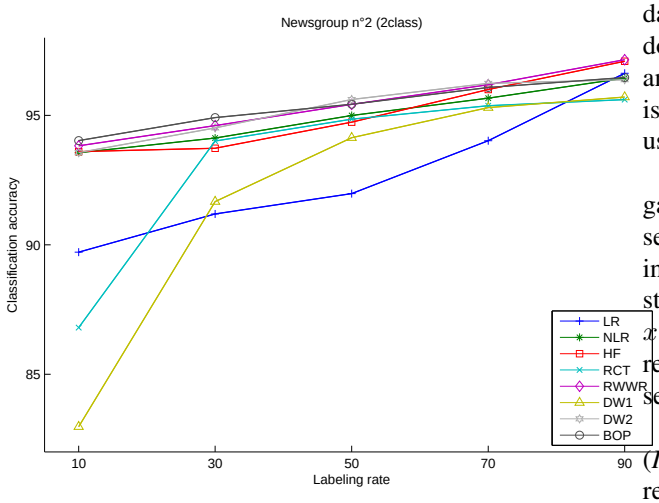

Fig. 1. Classification accuracies in percents, averaged over 20 runs, obtained on partially labeled graphs. Results are reported for the eight methods (RL, RNL, RCT, HF, RWWR, DW1, DW2, BoP) and for five labeling rates (10%, 30%, 50%, 70%, 90%). This graphs shows the results obtained on the *NewsGroups* (NG2) data set.

in Subsection VI-A. Second, the experimental methodology is detailed in Subsection VI-B. Third, the results will be discussed in Subsection VI-C. Fourth, the computation time will be investigated in Subsection VI-D. Finally, extreme cases are studied in Subsection VI-E.

A. Datasets

The different classifiers are compared on 14 data sets that have been used previously for semi-supervised classification: nine *Newsgroups* data sets [39], the four universities *WebKB cocite* data sets [24], [17] and the *IMDb prodco* data set [24][2].

Newsgroups: The *Newsgroups* data set is composed of about 20,000 unstructured documents, taken from 20 discussion groups (newsgroups) of the Usenet diffusion list. 20 Classes (or topics) were originally present in the data set. For our experiments, nine subsets related to different topics are extracted from the original data set, resulting in a total of nine different data sets. The data sets were built by sampling about 200 documents at random in each topic (three samples of two, three and five classes, thus nine samples in total). The repartition is listed in Table I. The extraction process and the procedure used for building the graph are detailed in [40].

WebKB cocite: These data sets consist of sets of web pages gathered from four computer science departments (four data sets, one for each university), with each page manually labeled into one of six categories: course, department, faculty, project, staff, and student [24]. The pages are linked by co-citation (if x links to z and y links to z, then x and y are co-citing z), resulting in an undirected graph. The composition of the data sets is shown in Table III.

IMDb-prodco: The collaborative Internet Movie Database (*IMDb*, [24]) has several applications such as making movie recommendations or movie category classification. The classification problem focuses on the prediction of whether the movie is a box-office hit or not. It contains a graph of movies linked together whenever they share the same production company and weight of an edge in the graph is the number of production companies that two movies have in common. The *IMDb-proco* class distribution is shown in Table II.

B. Experimental methodology

The classification accuracy will be reported for several labeling rates (10%, 30%, 50%, 70%, 90%), i.e. proportions of nodes for which the label is known. The labels of remaining nodes are deleted during the modeling phase and are used as test data during the assessment phase. For each considered labeling rate, 20 random node label deletions were performed (20 runs) and performances are averaged on these 20 runs. For each unlabeled node, the various classifiers predict the most suitable category. For each run, a 10-fold nested cross-validation is performed for tuning the parameters of the models. The external folds are obtained by 10 successive rotations of the nodes and the performance of one specific run is the average over these 10 folds. Moreover, for each fold of the external cross-validation, a 10-fold internal cross-validation is performed on the remaining labeled nodes in order to tune the hyperparameters of the classifiers (i.e. parameters α, λ and θ (see Table IV) – methods HF and DW1 do not have any hyperparameter). Thus, for each method and each labeling rate, the mean classification accuracy averaged on the 20 runs will be reported.

[2]The different data sets used for these comparisons are described in Subsection VI-A. Implementations and datasets are available at http://www.isys.ucl.ac.be/staff/lebichot/research.htm.

C. Results & discussion

Comparative results for each method on the fourteen data sets are reported as follows: the results on the nine *News-Groups* data sets, on the four *WebKB Cocite* data sets and on the *IMBd-prodco* data set are shown in Table V. The results of the second *NewsGroups* data set are also reported as a plot on Fig. 1 to visualize the typical relation between classification accuracy and labeling rate.

Statistical significance tests for each labeling rate are detailed in Table VI. One-sided t-tests were performed to determine whether or not the performance of a method is significantly superior (p-value less than 0.05 on the 20 runs) to another. Table VI can be read as follows. At the bottom of each table, the Win/Tie/Lose frequency summarizes how many times the BoP classifier was significantly better (Win), equivalent (Tie), or significantly worse (Lose) than each other method.

The most selected values of the parameters for each method are reported in Table IV. With each methods, all the tested values were selected at least once. RNL, RCT, RWWR and BoP selected a short range of parameters, which is of course desirable. Some methods, such as RL and DW2 selected a very wide range of parameters, and the different most represented values are not grouped.

Moreover, for each labeling rate, the different classifiers have been ordered according to a Borda score ranking. For each data set, each method is granted with a certain number of points, or rating. This number of points is equal to eight if the classifier is the best classifier (i.e., has the best mean classification accuracy on this data set), seven if the classifier is the second best and so on, so that the worst classifier is granted with only one point. The ratings are then summed across all the considered data sets and the classifiers are sorted by descending total rating. The final ranking, together with the total ratings, are reported in Table VII.

We observe that the BoP classifier always achieved competitive results since it ranges among the top methods on all data sets. The BoP classifier actually tends to be the best algorithm for all labeling rates except for the 90% labeling rate, where it comes second as observed from Table VII and from Table VI. The RWWR method is often second. The RCT kernel also achieves good performance and is the best of the kernel-based classifiers (as suggested in [22]). It is also the best algorithm when the labeling rate is very high (90%).

Notice that RWWR, RCT and DW2 largely outperform the other algorithms (besides BoP). However, it is difficult to figure out which of those three methods is the best, after BoP. It can be noticed that the DW2 version of the $\mathcal{D}$-walk is more competitive when the labeling rate is low and that it performs much better than the DW1 version, especially for low labeling rates: the Win/Tie/Lose scores for DW2 against DW1 are 7/1/6, 5/2/7, 8/1/5, 13/1/0 and 14/0/0 respectively for 90%, 70%, 50%, 30%, 10% labeling rate.

From the fifth to the eighth position, the ranking is less clear since none of the methods is really better than the other. However, all of these methods (NR and RNL as well as HF and DW1) are significantly worse than BoP, RCT, RWWR and DW2. Notice also that the performance of DW1 and HF drops significantly when labeling rate decreases.

D. Computation time

The computational tractability of a method is an important consideration to take into account. Table VIII provides a comparison of the running time of all methods. To explore computation time with respect to the number of nodes and the number of classes, artificial graphs with a certain number of classes have been created. For each method, 20 runs on each of the data sets are performed and the running time is recorded for each run. The 20 running times are averaged and results are reported in Table VIII.

We observe that HF is one of the quickest methods, but sadly it is not competitive in terms of accuracy, as reported in Subsection (VI-C). Notice that the two kernel methods, RL and RCT, have more or less the same computation time since the alignment is done once for all the classes. RNL, the last kernel method, is slower than RL, HF and RCT (because of the time-consuming normalisation). After the HF and the kernel methods, the BoP classifier achieves competitive results compared with the remaining classifiers. The time augmentation when the graph size increases is similar for all methods (except for RL and RC for which the augmentation is smaller). The cause is that all those methods require a matrix inversion: the complexity of such an operation is $\mathcal{O}(n^3)$ (where n is the number of nodes) and this is what can be observed from Table VIII (when the number of nodes doubles, the time is more or less multiplied by eight). But the BoP classifier has the same advantage as the kernel methods: its computation time does not increase strongly when the number of classes increases. This comes from the algorithm structure: to contrary to RWWR, DW1 and DW2, the BoP classifier does not require a matrix inversion for each class. Furthermore, the matrix inversions (or linear systems of equations to solve) required for the BoP can be computed as far as the graph (through its adjacency matrix) is known, which is not the case with kernel methods. This is a good property for BoP, since it means that rows 1 to 6 of Algorithm 1 can be pre-computed once for all folds in the cross-validation. Finally, the space complexity is $\mathcal{O}(n^2)$ for all the methods.

E. Extreme cases

In this subsection, two extreme classification cases will be studied. First, what happens if only one or two labeled data points are available? As described in Section V, the BoP classifier requires at least two nodes for computing the BoP group betweenness. We performed a small experiment on the first *NewsGroups* NG1 dataset. The parameters were tuned by a 10-fold cross-validation and 20 runs were averaged. The classification accuracies are reported in Table IX. Only the RCT, RWWR and BoP classifiers remain competitive for this first extreme case.

Secondly, let us consider the case where the classes are imbalanced. The following experiment was designed to study this other extreme case. The classes of the well-known *industry-yh* dataset were merged to get two classes: the

TABLE V

CLASSIFICATION ACCURACIES IN PERCENTS ± THE STANDARD DEVIATION, AVERAGED OVER 20 RUNS, OBTAINED ON PARTIALLY LABELED GRAPHS. RESULTS ARE REPORTED FOR THE EIGHT METHODS (RL, RNL, LP, RCT, HF, RWWR, DW1, DW2, BOP) AND FOR FIVE LABELING RATES (10%, 30%, 50%, 70%, 90%). THE TABLE SHOWS THE RESULTS FOR ALL THE TESTED DATA SETS. THE STANDARD DEVIATION IS CALCULATED OVER THE 10 FOLDS OF THE EXTERNAL CROSS-VALIDATION OF THE 20 INDEPENDENT RUNS.

	l	RL	RNL	RCT	HF	RWWR	DW1	DW2	BoP
NG1	**90%**	97.56 ± 2.28	98.42 ± 2.03	97.16 ± 2.46	97.39 ± 2.55	97.17 ± 2.42	97.16 ± 2.71	97.56 ± 2.50	97.58 ± 2.08
	70%	94.64 ± 10.11	97.31 ± 1.74	96.66 ± 1.26	97.25 ± 1.35	**96.76 ± 1.32**	96.65 ± 1.50	96.59 ± 1.51	**97.36 ± 1.04**
	50%	92.87 ± 12.97	96.77 ± 1.27	96.53 ± 1.18	96.80 ± 1.04	96.70 ± 1.32	95.73 ± 1.30	96.41 ± 1.17	**97.27 ± 0.98**
	30%	93.79 ± 10.52	96.31 ± 1.35	95.99 ± 1.30	95.92 ± 1.06	96.14 ± 1.10	94.34 ± 1.24	96.03 ± 0.94	**96.94 ± 0.70**
	10%	87.27 ± 18.01	95.36 ± 2.17	96.15 ± 1.09	88.46 ± 7.32	96.19 ± 0.63	88.55 ± 1.94	95.27 ± 0.84	**96.50 ± 1.27**
NG2	**90%**	96.63 ± 2.78	96.44 ± 2.54	97.09 ± 2.67	95.61 ± 3.14	**97.15 ± 2.53**	95.71 ± 3.00	96.38 ± 2.54	96.46 ± 2.75
	70%	94.02 ± 9.97	95.66 ± 1.86	96.00 ± 1.63	95.37 ± 1.65	96.18 ± 1.46	95.30 ± 1.74	**96.23 ± 1.51**	96.07 ± 1.81
	50%	91.98 ± 11.63	94.99 ± 2.22	94.73 ± 1.43	94.85 ± 1.30	95.42 ± 1.37	94.13 ± 1.54	**95.61 ± 1.50**	95.43 ± 1.71
	30%	91.19 ± 11.90	94.12 ± 3.46	93.73 ± 1.08	94.01 ± 1.09	94.60 ± 1.18	91.67 ± 1.60	94.51 ± 1.42	**94.91 ± 1.43**
	10%	89.72 ± 13.38	93.55 ± 3.25	93.60 ± 0.80	86.80 ± 6.11	93.82 ± 0.63	82.97 ± 2.11	93.56 ± 1.29	**94.02 ± 1.45**
NG3	**90%**	96.80 ± 2.62	96.59 ± 2.16	98.05 ± 2.58	96.81 ± 2.76	98.05 ± 2.62	96.91 ± 2.83	96.94 ± 2.83	**98.06 ± 2.34**
	70%	95.66 ± 6.75	96.63 ± 0.95	97.70 ± 1.52	96.59 ± 1.41	**97.77 ± 1.59**	96.89 ± 1.37	97.19 ± 1.18	**97.77 ± 1.57**
	50%	93.00 ± 12.04	96.43 ± 0.94	97.13 ± 1.08	96.54 ± 0.92	97.21 ± 1.24	96.70 ± 0.98	96.43 ± 0.90	**97.36 ± 1.26**
	30%	93.00 ± 10.84	95.98 ± 0.84	96.59 ± 0.81	95.80 ± 0.93	96.64 ± 0.81	95.78 ± 1.06	96.27 ± 0.91	**96.84 ± 1.00**
	10%	87.73 ± 16.95	95.29 ± 0.97	95.45 ± 1.04	83.44 ± 9.70	95.21 ± 1.24	91.81 ± 1.47	95.46 ± 0.73	**96.05 ± 0.81**
NG4	**90%**	95.14 ± 2.18	**95.99 ± 2.01**	95.58 ± 2.07	95.03 ± 2.68	95.55 ± 2.05	94.99 ± 2.74	94.73 ± 2.17	95.19 ± 2.27
	70%	89.17 ± 16.52	94.76 ± 0.96	94.41 ± 1.01	**94.79 ± 1.44**	94.29 ± 1.02	95.06 ± 1.45	94.23 ± 1.00	94.32 ± 0.92
	50%	87.59 ± 17.33	93.60 ± 0.94	93.27 ± 0.88	**94.38 ± 1.01**	93.27 ± 0.78	94.35 ± 1.21	93.82 ± 0.58	93.85 ± 0.77
	30%	87.22 ± 18.23	93.26 ± 0.88	92.97 ± 0.82	93.00 ± 1.02	93.16 ± 0.90	92.65 ± 1.13	93.38 ± 0.58	**93.51 ± 0.75**
	10%	84.41 ± 22.69	91.05 ± 6.17	94.53 ± 1.38	74.92 ± 9.69	**95.60 ± 0.52**	85.99 ± 1.63	94.86 ± 0.52	95.14 ± 1.06
NG5	**90%**	95.35 ± 2.41	95.65 ± 2.84	95.94 ± 2.46	95.25 ± 2.54	**95.97 ± 2.52**	94.97 ± 2.57	95.84 ± 2.33	95.80 ± 2.98
	70%	93.83 ± 7.58	94.40 ± 1.47	94.78 ± 1.13	94.83 ± 1.36	**95.05 ± 1.07**	94.67 ± 1.48	94.12 ± 1.55	94.74 ± 1.22
	50%	91.26 ± 9.54	92.59 ± 1.87	93.65 ± 1.36	93.94 ± 1.02	**94.48 ± 0.80**	93.69 ± 1.17	93.64 ± 1.38	94.41 ± 1.00
	30%	87.89 ± 14.24	90.41 ± 1.92	92.41 ± 1.18	91.51 ± 1.30	93.83 ± 0.69	91.65 ± 1.20	92.84 ± 0.96	**94.21 ± 1.19**
	10%	88.76 ± 11.94	90.64 ± 2.22	93.58 ± 1.01	77.94 ± 6.73	**94.84 ± 0.65**	84.42 ± 1.89	92.35 ± 0.60	94.27 ± 1.07
NG6	**90%**	93.89 ± 2.91	92.49 ± 2.50	96.25 ± 2.84	94.19 ± 2.81	**96.27 ± 2.84**	93.99 ± 2.93	95.21 ± 2.63	96.17 ± 2.77
	70%	92.43 ± 1.47	91.05 ± 1.21	95.93 ± 1.59	93.07 ± 1.55	**96.07 ± 1.57**	93.86 ± 1.76	94.00 ± 1.37	95.96 ± 1.75
	50%	91.44 ± 1.00	90.19 ± 1.76	94.80 ± 1.10	91.42 ± 1.34	**95.41 ± 0.71**	92.90 ± 1.39	93.43 ± 1.27	95.36 ± 0.77
	30%	89.58 ± 4.11	88.42 ± 1.69	93.33 ± 1.21	88.04 ± 1.83	**94.52 ± 0.85**	90.75 ± 1.49	92.49 ± 0.82	94.51 ± 1.19
	10%	89.19 ± 10.35	89.20 ± 3.13	94.18 ± 0.65	73.74 ± 5.37	**94.93 ± 0.56**	83.66 ± 1.84	92.82 ± 0.88	94.21 ± 1.44
NG7	**90%**	92.48 ± 2.31	91.18 ± 2.90	91.98 ± 2.52	92.22 ± 2.46	91.99 ± 2.51	92.46 ± 2.52	92.61 ± 2.25	**93.06 ± 2.02**
	70%	91.73 ± 1.51	91.04 ± 1.46	91.57 ± 1.19	91.34 ± 1.58	91.56 ± 1.14	91.87 ± 1.44	91.97 ± 1.25	**92.04 ± 1.15**
	50%	89.97 ± 8.93	90.40 ± 1.10	91.42 ± 0.79	90.03 ± 1.08	91.61 ± 0.51	90.69 ± 1.10	**91.88 ± 0.80**	91.61 ± 0.87
	30%	90.65 ± 1.33	89.86 ± 0.71	90.65 ± 1.16	87.05 ± 1.32	91.06 ± 0.68	88.45 ± 1.09	**91.09 ± 1.12**	90.97 ± 0.75
	10%	85.52 ± 16.28	88.66 ± 0.76	88.89 ± 1.10	68.85 ± 6.90	**91.08 ± 0.57**	80.31 ± 1.67	90.33 ± 0.57	90.23 ± 1.05
NG8	**90%**	90.34 ± 2.70	90.16 ± 2.12	**90.81 ± 2.42**	88.73 ± 2.79	90.80 ± 2.43	88.73 ± 2.87	90.54 ± 2.19	90.48 ± 2.81
	70%	89.30 ± 7.26	89.45 ± 1.07	90.44 ± 1.08	88.01 ± 1.60	90.46 ± 1.04	87.91 ± 1.60	**90.68 ± 1.46**	90.19 ± 1.64
	50%	89.06 ± 5.17	88.14 ± 1.16	89.74 ± 0.74	86.67 ± 1.37	89.94 ± 0.59	86.66 ± 1.34	89.87 ± 0.68	**90.07 ± 0.96**
	30%	87.34 ± 5.24	84.47 ± 2.06	88.05 ± 0.89	83.48 ± 1.32	89.35 ± 0.41	83.72 ± 1.23	88.52 ± 0.66	**89.54 ± 0.68**
	10%	81.75 ± 13.35	81.38 ± 3.24	85.82 ± 1.87	62.52 ± 6.48	**88.56 ± 0.51**	73.57 ± 1.59	87.28 ± 0.61	88.26 ± 1.71
NG9	**90%**	88.60 ± 2.88	**89.20 ± 2.97**	89.09 ± 2.86	88.75 ± 2.71	89.11 ± 2.92	87.96 ± 2.64	88.50 ± 2.58	88.62 ± 3.14
	70%	87.24 ± 1.61	**88.27 ± 1.70**	88.03 ± 1.71	87.62 ± 1.67	88.09 ± 1.67	87.42 ± 1.62	87.74 ± 1.75	88.09 ± 1.48
	50%	85.68 ± 3.03	86.11 ± 1.41	87.21 ± 1.65	85.94 ± 1.44	87.37 ± 1.60	86.06 ± 1.38	87.22 ± 1.11	**87.56 ± 1.47**
	30%	83.50 ± 6.71	82.11 ± 2.54	85.71 ± 1.54	82.32 ± 1.46	**87.07 ± 0.88**	83.03 ± 1.30	86.25 ± 0.81	86.97 ± 0.97
	10%	80.60 ± 9.47	80.16 ± 2.09	84.55 ± 1.10	68.64 ± 4.43	86.17 ± 0.66	72.78 ± 1.81	85.56 ± 0.69	**86.28 ± 1.14**
Cornell	**90%**	58.91 ± 6.25	52.86 ± 4.17	**65.22 ± 5.56**	62.67 ± 7.04	56.50 ± 8.58	60.64 ± 6.65	60.38 ± 5.78	59.78 ± 7.22
	70%	59.00 ± 3.22	51.69 ± 4.21	**63.66 ± 3.47**	60.26 ± 3.87	57.82 ± 3.90	57.97 ± 4.60	59.25 ± 2.03	61.34 ± 4.80
	50%	55.40 ± 5.04	46.74 ± 4.41	**61.70 ± 2.98**	**56.88 ± 3.37**	56.05 ± 2.73	54.53 ± 5.04	57.09 ± 3.25	61.25 ± 4.41
	30%	49.33 ± 5.91	42.95 ± 3.30	60.55 ± 3.78	50.35 ± 3.25	54.84 ± 4.37	51.18 ± 5.44	56.27 ± 4.12	**61.19 ± 6.19**
	10%	44.97 ± 3.99	41.91 ± 1.90	58.07 ± 5.31	42.71 ± 1.49	57.51 ± 3.78	47.11 ± 7.05	**58.58 ± 4.65**	58.17 ± 8.77
Texas	**90%**	72.11 ± 4.83	69.11 ± 4.73	78.29 ± 4.96	74.04 ± 6.46	78.99 ± 4.78	81.34 ± 6.48	81.72 ± 4.67	**82.29 ± 4.27**
	70%	71.16 ± 2.09	68.01 ± 1.90	78.21 ± 2.46	71.96 ± 3.94	78.52 ± 1.83	80.30 ± 3.62	79.56 ± 3.24	**80.74 ± 3.07**
	50%	68.98 ± 2.18	65.04 ± 3.26	77.29 ± 2.69	69.16 ± 3.07	76.73 ± 1.70	78.95 ± 2.86	77.86 ± 1.83	**79.60 ± 2.81**
	30%	66.70 ± 2.84	61.27 ± 4.18	76.11 ± 2.73	65.56 ± 2.89	73.79 ± 2.16	76.50 ± 2.32	76.63 ± 1.02	**78.16 ± 1.86**
	10%	58.99 ± 8.20	54.60 ± 5.22	74.12 ± 3.88	51.16 ± 2.46	71.24 ± 1.78	69.42 ± 4.04	75.04 ± 1.53	**76.12 ± 3.36**
Washington	**90%**	**68.18 ± 2.98**	63.56 ± 4.59	70.27 ± 3.71	66.51 ± 5.74	61.87 ± 5.84	61.11 ± 6.81	58.22 ± 5.31	64.49 ± 5.44
	70%	62.28 ± 6.81	63.65 ± 1.41	**69.29 ± 2.46**	65.37 ± 3.26	59.78 ± 2.80	59.74 ± 3.73	55.45 ± 4.78	64.05 ± 3.66
	50%	61.91 ± 5.08	64.38 ± 1.15	**67.75 ± 1.97**	63.78 ± 2.36	59.01 ± 2.21	57.01 ± 3.97	55.56 ± 4.18	60.99 ± 3.70
	30%	60.57 ± 4.26	64.81 ± 1.62	**65.80 ± 3.01**	59.88 ± 2.28	57.85 ± 2.80	53.29 ± 5.88	55.79 ± 3.30	60.88 ± 4.37
	10%	53.52 ± 12.15	57.73 ± 10.50	**68.04 ± 1.40**	42.18 ± 4.37	52.11 ± 5.61	46.30 ± 8.02	57.30 ± 3.30	61.57 ± 5.84
Wisconsin	**90%**	71.39 ± 4.90	67.92 ± 3.40	74.18 ± 4.71	74.95 ± 5.03	73.42 ± 4.10	**80.43 ± 4.82**	77.47 ± 4.95	73.46 ± 3.62
	70%	70.41 ± 1.89	67.27 ± 2.54	75.21 ± 1.77	73.64 ± 2.95	71.75 ± 2.93	**79.10 ± 3.00**	75.49 ± 2.89	74.18 ± 3.16
	50%	69.23 ± 1.86	67.10 ± 1.44	75.43 ± 1.37	71.91 ± 2.22	69.18 ± 2.94	**77.22 ± 2.28**	75.64 ± 2.11	75.05 ± 2.31
	30%	68.10 ± 1.56	66.25 ± 2.60	**75.53 ± 1.77**	68.60 ± 1.97	69.14 ± 4.54	73.84 ± 2.54	75.21 ± 1.76	75.32 ± 2.13
	10%	64.90 ± 7.20	62.91 ± 5.66	**74.71 ± 3.69**	54.12 ± 3.95	63.36 ± 10.32	65.34 ± 4.23	73.72 ± 1.48	73.65 ± 3.68
IMDb	**90%**	81.10 ± 2.99	80.91 ± 3.25	**82.14 ± 2.99**	76.53 ± 12.06	81.96 ± 3.15	50.64 ± 1.27	81.36 ± 3.06	80.74 ± 2.20
	70%	77.51 ± 9.70	70.75 ± 13.93	**81.41 ± 1.74**	58.00 ± 12.35	81.34 ± 1.68	50.64 ± 1.22	80.73 ± 1.69	80.20 ± 1.94
	50%	69.25 ± 14.41	57.41 ± 12.30	**80.82 ± 1.26**	51.33 ± 2.14	80.68 ± 1.28	51.68 ± 3.52	80.40 ± 1.03	79.87 ± 1.42
	30%	62.65 ± 14.24	51.60 ± 5.68	**80.32 ± 0.80**	51.01 ± 0.17	80.17 ± 0.65	60.64 ± 12.39	77.65 ± 7.19	79.47 ± 1.11
	10%	54.61 ± 9.86	50.30 ± 0.40	79.60 ± 0.61	50.97 ± 0.10	**79.67 ± 0.68**	73.20 ± 8.60	76.33 ± 8.08	78.79 ± 1.40

TABLE VI
ONE-SIDE t-TEST FOR ALL LABELING RATES. THE WIN/TIE/LOSE FREQUENCY SUMMARIZES HOW MANY TIMES THE BOP CLASSIFIER WAS SIGNIFICANTLY BETTER (WIN), EQUIVALENT (TIE) OR SIGNIFICANTLY WORSE (LOSE) THAN EACH OTHER METHOD.

	RL	**RNL**	**LP**	**RCT**	**HF**	**RWWR**	**DW1**	**DW2**
l **= 90%**	6/5/3	9/1/4	8/2/4	3/2/9	9/2/3	5/3/6	12/1/1	6/6/2
l **= 70%**	14/0/0	11/1/2	10/1/3	3/5/6	11/1/2	6/3/5	10/2/2	9/1/4
l **= 50%**	13/0/1	13/0/1	11/0/3	10/1/3	12/0/2	9/4/1	12/0/2	9/1/4
l **= 30%**	13/1/0	13/0/1	13/0/1	10/2/2	14/0/0	10/1/3	14/0/0	12/1/1
l **= 10%**	14/0/0	14/0/0	14/0/0	9/2/3	14/0/0	7/1/6	14/0/0	11/3/0

TABLE VII
FOR EACH LABELING RATE, THE DIFFERENT CLASSIFIERS ARE RANKED THROUGH A BORDA RATING (SEE THE TEXT FOR DETAILS). THE CLASSIFIERS ARE THEN RANKED ACCORDING TO THE TOTAL RATING OBTAINED ACROSS ALL DATA SETS (THE LARGER THE BETTER). l STANDS FOR LABELING RATE AND THE NUMBERS BETWEEN PARENTHESES ARE THE TOTAL RATINGS.

Ranking	**First**	**Second**	**Third**	**Fourth**	**Fifth**	**Sixth**	**Seventh**	**Last**
l **= 90%**	RCT (86)	BoP (77)	RWWR (76)	DW2 (66)	RL (55)	HF (52)	RNL (49)	DW1 (43)
l **= 70%**	BoP (87)	RCT (81)	RWWR (76)	DW2 (68)	HF (57)	DW1 (55)	RNL (48)	RL (32)
l **= 50%**	BoP (94)	RWWR (78)	RCT (75)	DW2 (72)	HF (56)	DW1 (54)	RNL (44)	RL (31)
l **= 30%**	BoP (103)	RWWR (85)	DW2 (80)	RCT (78)	RNL (45)	DW1 (42)	RL (36)	HF (35)
l **= 10%**	BoP (100)	RWWR (89)	RCT (85)	DW2 (83)	RNL (48)	RL (42)	DW1 (39)	HF (18)

TABLE VIII
OVERVIEW OF CPU TIME IN SECONDS NEEDED TO CLASSIFY ALL THE UNLABELED NODES. RESULTS ARE AVERAGED ON 20 RUNS. THE CPU USED WAS AN INTEL(R)CORE(TM)I3 AT 2.13 GHZ WITH 3072 KO OF CACHE SIZE AND 6 GB OF RAM AND THE PROGRAMMING LANGUAGE IS MATLAB.

	RL	**RNL**	**RCT**	**HF**	**RWWR**	**DW1**	**DW2**	**BoP**
Exp1: 1000 nodes, 2 classes	0.0872	0.4937	0.0751	0.1680	0.3480	0.6826	0.6772	0.4997
Exp2: 2000 nodes, 2 classes	0.4616	3.4961	0.4225	0.9618	2.0441	4.5561	4.5858	3.0574
Exp3: 4000 nodes, 2 classes	2.8274	27.0695	2.5949	7.1481	14.1116	35.7161	36.0207	22.393
Ratio Exp2/Exp1	5.2935	7.0814	5.6258	5.7250	5.8739	6.6746	6.7717	6.1185
Ratio Exp3/Exp2	6.1252	7.7428	6.1418	7.4320	6.9036	7.8392	7.8548	7.3242
Exp2: 2000 nodes, 2 classes	0.4616	3.4961	0.4225	0.9618	2.0441	4.5561	4.5858	3.0574
Exp4: 2000 nodes, 4 classes	0.5011	3.4563	0.4036	1.2064	3.4249	8.5048	8.4003	3.2535
Exp5: 2000 nodes, 8 classes	0.4813	3.8449	0.4482	1.5748	6.0697	16.0031	16.3956	3.5868
Ratio Exp4/Exp2	1.0856	0.9886	0.9553	1.2543	1.6755	1.8667	1.8318	1.0641
Ratio Exp5/Exp4	0.9605	1.1124	1.1105	1.3054	1.7722	1.8817	1.9518	1.1024

majority class with 1768 nodes and the minority class with only 30 nodes (this represents 1.67% for the minority class). Once again, the parameters were tuned by a 10-fold cross-validation and 20 runs were averaged. The classification accuracies for the two classes are reported in Table IX. The best methods to identify the minority class are RL and NRL, followed by RWWR and RCT. In this particular case, the BoP classifier does not perform very well.

VII. CONCLUSION

This paper investigates an application of the bag-of-paths framework viewing the graph as a virtual bag from which paths are drawn according to a Boltzmann sampling distribution.

In particular, it introduces a novel algorithm for graph-based semi-supervised classification through the bag-of-paths group betweenness, or BoP for short (described in Section V). The algorithm sums the a posteriori probabilities of drawing a path visiting a given node of interest according to a biased sampling distribution, and this sum defines our BoP betweenness measure. The Boltzmann sampling distribution depends on a parameter, θ, gradually biasing the distribution towards shorter paths: when θ is large, only little exploration is performed and only the shortest paths are considered, while when θ is small (close to 0^+), longer paths are considered and are sampled according to the product of the transition probabilities p_{ij}^{ref} along the path (a natural random walk).

Experiments on real-world data sets show that the BoP method outperforms the other considered approaches when only a few labeled nodes are available. When more nodes are labeled, the BoP method is still competitive. Its computation time is also substantially lower in most of the cases.

Our future work will include several extensions of the proposed approach. Another interesting issue is how to combine the information provided by the graph and the features of the nodes in a clever, preferably optimal, way. The interest of including node features should be assessed experimentally. A typical case study could be the labeling of protein-protein interaction networks. The node features could involve gene expression measurements for the corresponding proteins.

Yet another application of the bag-of-paths framework could be the definition of a robustness measure or criticality measure of the nodes. The idea would be to compute the change in reachability between nodes when deleting one node within the BoP framework. Nodes having a large impact on reachability would be then considered as highly critical.

TABLE IX
CLASSIFICATION ACCURACIES IN PERCENTS, AVERAGED OVER 20 RUNS, OBTAINED ON PARTIALLY LABELED ARTIFICIAL GRAPHS. RESULTS ARE REPORTED FOR THE EIGHT METHODS (RL, RNL, RCT, HF, RWWR, DW1, DW2, BOP) AND FOR A 50% LABELING RATE. RARE IS THE CASE WHERE ONLY TWO LABELED NODES PER CLASS ARE KNOWN AND IMBALANCED IS THE CASE WHERE ONE OF THE CLASSES IS MUCH MORE REPRESENTED THAN THE OTHER.

	RL	**NRL**	**RCT**	**HF**	**RWWR**	**DW1**	**DW2**	**BoP**
Rare	51.9 $\pm$ 1.4	52.6 $\pm$ 2.6	84.0 $\pm$ 2.5	50.0 $\pm$ 0.05	84.0 $\pm$ 2.5	51.9 $\pm$ 0.5	67.3 $\pm$ 2.8	83.0 $\pm$ 3.1
Imbalanced: Major	98.2 $\pm$ 0.04	98.2 $\pm$ 0.04	98.1 $\pm$ 0.08	99.9 $\pm$ 0.04	95.5 $\pm$ 0.03	88.7 $\pm$ 7.0	93.8 $\pm$ 2.5	96.8 $\pm$ 0.4
Imbalanced: Minor	42.4 $\pm$ 4.2	43.8 $\pm$ 3.3	32.2 $\pm$ 2.0	1.9 $\pm$ 2.4	11.6 $\pm$ 10.5	38.5 $\pm$ 0.0	17.2 $\pm$ 2.7	11.9 $\pm$ 2.5

Finally, the biggest drawback of the BoP classifier is that it is not applicable as-is for large graphs. It would be interesting to investigate if it is possible to modify the classifier to be computationally more tracktable on large graphs. A starting clue would be to use the same trick as in [8].

REFERENCES

[1] K. Francoisse, I. Kivimaki, A. Mantrach, F. Rossi, and M. Saerens, "A bag-of-paths framework for network data analysis," *Submitted for publication and available on ArXiv at http://arxiv.org/abs/1302.6766*.

[2] J. Callut and P. Dupont, "Learning hidden markov models from first passage times," *Proceedings of the European Machine Learning conference (ECML). Lecture notes in Artificial Intelligence, Springer*, 2007.

[3] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery," *Proceedings of the 10th ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD 2004)*, pp. 653–658, 2004.

[4] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Proceedings of the Neural Information Processing Systems Conference (NIPS 2003)*, 2003, pp. 237–244.

[5] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, 2003, pp. 912–919.

[6] X. Zhu and A. Goldberg, *Introduction to semi-supervised learning*. Morgan & Claypool Publishers, 2009.

[7] A. Mantrach, L. Yen, J. Callut, K. Francoise, M. Shimbo, and M. Saerens, "The sum-over-paths covariance kernel: a novel covariance between nodes of a directed graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1112–1126, 2010.

[8] A. Mantrach, N. van Zeebroeck, P. Francq, M. Shimbo, H. Bersini, and M. Saerens, "Semi-supervised classification and betweenness computation on large, sparse, directed graphs," *Pattern Recognition*, vol. 44, no. 6, pp. 1212 – 1224, 2011.

[9] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, p. p.824, 2002.

[10] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez, "Motif-based communities in complex networks," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, p. 224001, 2008.

[11] S. Abney, *Semisupervised learning for computational linguistics*. Chapman and Hall/CRC, 2008.

[12] O. Chapelle, B. Scholkopf, and A. Zien (editors), *Semi-supervised learning*. MIT Press, 2006.

[13] D. Zhou and B. Scholkopf, "Learning from labeled and unlabeled data using random walks," in *Proceedings of the 26th DAGM Symposium*, 2004, pp. 237–244.

[14] M. Szummer and T. Jaakkola, "Partially labeled classification with markov random walks," in *Advances in Neural Information Processiong Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. Vancouver, Canada: MIT Press, 2001.

[15] O. Chapelle, J. Weston, and B. Scholkopf, "Cluster kernels for semi-supervised learning," *in: NIPS, 2002*, pp. 585–592, 2002.

[16] A. Kapoor, Y. Qi, H. Ahn, and R. Picard, "Hyperparameter annd kernel learning for graph based semi-supervised classification," *in: NIPS, 2005*, pp. 627–634, 2002.

[17] D. Zhou, J. Huang, and B. Scholkopf, "Learning from labeled and unlabeled data on a directed graph," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 1041–1048.

[18] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proceedings of the International Conference on Learning Theory (COLT 2004)*, 2004, pp. 624–638.

[19] J. Wang, F. Wang, C. Zhang, H. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1600–1615, 2009.

[20] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceedings of the 20^{th} International Conference on Machine Learning (ICDM 2003)*, Washington DC, 2003, p. 290 297.

[21] T. Kato, H. Kashima, and M. Sugiyama, "Robust label propagation on multiple networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 35–44, 2009.

[22] F. Fouss, K. Francoisse, L. Yen, A. Pirotte, and M. Saerens, "An experimental investigation of kernels on a graph on collaborative recommendation and semisupervised classification," *Neural Networks*, vol. 31, pp. 53–72, 2012.

[23] D. Zhou and B. Scholkopf, "Discrete regularization," in *Semi-supervised learning, O. Chapelle, B. Scholkopf and A. Zien (editors)*. MIT Press, 2006, pp. 237–249.

[24] S. A. Macskassy and F. Provost, "Classification in networked data: a toolkit and a univariate case study," *Journal of Machine Learning Research*, vol. 8, pp. 935–983, 2007.

[25] L. Getoor and B. Taskar (editors), *Introduction to statistical relational learning*, 2007.

[26] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[27] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[28] M. Brand and K. Huang, "A unifying theorem for spectral embedding and clustering," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, January 2003.

[29] P. Sarkar and A. Moore, "A tractable approach to finding closest truncated-commute-time neighbors in large graphs," *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

[30] J. Callut, K. Francoisse, M. Saerens, and P. Dupont, "Semi-supervised classification from discriminative randow walks," *Proceedings of the European Machine Learning conference (ECML 2008). Lecture notes in Artificial Intelligence, Springer*, vol. 5211, pp. 162–177, 2008.

[31] M. Herbster, M. Pontil, and S. Rojas-Galeano, "Fast prediction on tree," *Proceedings of the 22th Neural Information Processing Conference NIPS 2008*, pp. 657–664, 2008.

[32] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proceedings of the ACM conference on Knowledge Discovery and Data Mining (KDD 2009)*, 2009, pp. 817–826.

[33] ——, "Scalable learning of collective behavior based on sparse social dimensions," in *Proceedings of the ACM conference on Information and Knowledge Management (CIKM 2009)*, 2009, pp. 1107–1116.

[34] ——, "Toward predicting collective behavior via social dimension extraction," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 19–25, 2010.

[35] L. Gettor and B. Taskar, Eds., *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[36] L. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[37] M. Newman, "A measure of betweenness centrality based on random walks," *Social Networks*, vol. 27, no. 1, pp. 39–54, 2005.

[38] E. D. Kolaczyk, *Statistical analysis of network data: methods and models*. Springer, 2009.

[39] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 331–339.

[40] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens, "Graph nodes clustering with sigmoid commute-time kernel : A comparative study," *Data & Knowledge Engineering*, no. 68, pp. 338–361, 2009.