# An experimental study of graph-based semi-supervised classification with additional node information

B. Lebichot[a,*], M. Saerens[a],

[a] *Universite catholique de Louvain, ICTEAM & LSM*

**Abstract**

The volume of data generated by internet and social networks is increasing every day, and there is a clear need for efficient ways of extracting useful information from them. As those data can take different forms, it is important to use all the available data representations for prediction. In this paper, we focus our attention on supervised classification using both regular plain, tabular, data and structural information coming from a network structure. 14 techniques are investigated and compared in this study and can be divided in three classes: the first one uses only the plain data to build a classification model, the second uses only the graph structure and the last uses both information sources. The relative performances in these three cases are investigated. Furthermore, the effect of using a graph embedding and well-known indicators in spatial statistics is also studied. Possible applications are automatic classification of web pages or other linked documents, of people in a social network or of proteins in a biological complex system, to name a few. Based on our comparison, we draw some general conclusions and advices to tackle this particular classification task: some datasets can be better explained by their graph structure (graph-driven), or by their feature set (features-driven). The most efficient methods are discussed in both cases.

*Keywords:* Graph and network analysis, semi-supervised classification, network data, graph mining.

## 1. Introduction

Nowadays, with the increasing volume of data generated, for instance by internet and social networks, there is a need for efficient ways to predict useful information from those data. Numerous data mining, machine learning and pattern recognition algorithms were developed for predicting information from a labeled database. These data can take several different forms and, in that case, it would be useful to use these alternative views in the prediction model. In this paper, we focus our attention on supervised classification using both regular tabular data and structural information coming from graphs or networks.

Many different approaches have been developed for information fusion in machine learning, pattern recognition and applied statistics, such as simple weighted averages (see, e.g., [1], [2]), Bayesian fusion (see, e.g., [1], [2]), majority vote (see, e.g., [3], [4], [5]), models coming from uncertainty reasoning: fuzzy logic, possibility theory [6] (see, e.g., [7]), standard multivariate statistical analysis techniques such as correspondence analysis [8], maximum entropy modeling (see, e.g., [9], [10], [11]).

As well-known, the goal of classification is to automatically label data to predefined classes. This is also called supervised learning since it uses known labels (the desired prediction of an instance) to fit the classification model. One alternative is to use semi-supervised learning instead [12, 13, 14, 15].

Indeed, traditional pattern recognition, machine learning or data mining classification methods require large amounts of labeled training instances – which is often difficult to obtain – to fit accurate models. Semi-supervised learning methods can reduce the effort by including unlabeled samples. This name comes from the fact that the used dataset is a mixture of supervised and unsupervised data (it contains training samples that are unlabeled). Then, the classifier takes advantage from both the supervised and unsupervised data. The advantage here is that unlabeled data are often much less costly than labeled data. This technique allows to reduce the amount of labeled instances needed to achieve the same level of classification accuracy [12, 13, 14, 15]. In other words, exploiting the distribution of unlabeled data during the model fitting process can prove helpful.

Semi-supervised classification comes in two different settings: inductive and transductive [12]. The goal of the former setting is to predict the labels of future test data, unknown when fitting the model, while the second is to classify (only) the unlabeled instances of the training sample. Some often-used semi-supervised algorithms include: expectation-maximization with generative mixture models, self-training, co-training, transductive support vector machines, and graph-based methods [16, 17, 18].

The structure of the data can also be of different types. This paper focuses on a particular data structure: we assume that our dataset takes the form of a network with features associated to the nodes. Nodes are the samples of our dataset and links between these nodes represent a given type of relation between

*Corresponding author

*Email addresses:* bertrand.lebichot@uclouvain.be (B. Lebichot), bertrand.lebichot@uclouvain.be (M. Saerens)

these samples. For each node, a number of features or attributes characterizing it is also available (see Figure 1 for an example). Other data structures exist but are not studied in this paper; for instance:

- Different types of nodes can be present, with different types of features sets describing them.
- Different types of relations can link the different nodes.

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.

In this work, we investigate experimentally various models combining information on the nodes of a graph and the graph structure. Indeed, it has been shown that network information improves significantly prediction accuracy in a number of contexts [14, 15]. Indeed, 14 classification algorithms using various combinations of data sources, mainly described in [19], are compared. The different considered algorithms are described in Section 4.1, 4.2 and 4.3 . A standard support vector machine (SVM) classifier is used as a baseline algorithm, but we also investigated the ridge logistic regression classifier. The results and conclusions obtained with this second model were similar to the SVM and are therefore not reported in this paper.

In short, the main questions investigated in this work are:

- Does the combination of features on nodes and network structure works better than using the features only?
- Does the combination of features on nodes and network structure works better than using the graph structure only?
- Which classifier performs best on network structure alone, without considering features on nodes?
- Which classifier performs best when combining information, that is, using network structure with features on nodes?

Finally, this comparison leads to some general conclusions and advices when tackling classification problems on network data with node features.

In summary, this work has four main contributions:

- The paper reviews different algorithms used for learning from both a graph structure and node features. Some algorithms are inductive while some others are transductive.
- An empirical comparison of those algorithms is performed on ten real world datasets.
- It investigates the effect of extracting features from the graph structure (and some well-known indicators in spatial statistics) in a classification context.
- Finally, this comparison is used to draw general conclusions and advices to tackle graph-based classification tasks.

The remaining of this paper is organized as follows. Section 2 provides some background and notation. Section 3 investigates related work. Section 4 introduces the investigated classification methods. Then, Section 5 presents the experimental methodology and the results. Finally, Section 6 concludes the paper.

## 2. Background and notation

This section aims to introduce the necessary theoretical background and notation used in the paper. Consider a weighted, undirected, strongly connected, graph or network $G$ (with no self-loop) containing a set of $n$ vertices $\mathcal{V}$ (or nodes) and a set of edges $\mathcal{E}$ (or arcs, links). The $n \times n$ **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. 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.

We also introduce the **Laplacian matrix L** of the graph, defined in the usual manner:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{1}$$

where $\mathbf{D} = \mathbf{Diag}(\mathbf{Ae})$ is the diagonal (out)degree matrix of the graph $G$ containing the $a_{i\bullet} = \sum_{j=1}^{n} a_{ij}$ on its diagonal and $\mathbf{e}$ is a column vector full of ones. One of the properties of $\mathbf{L}$ is that its eigenvalues provide useful information about the connectivity of the graph [20]. The smallest eigenvalue of $\mathbf{L}$ is always equals to 0, and the second smallest one is equals to 0 only if the graph is composed of at least two connected components. This last value is called the algebraic connectivity.

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 transition probabilities

$$p_{ij} = \frac{a_{ij}}{\sum_{j'=1}^{n} a_{ij'}} \tag{2}$$

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 $n \times n$ **transition probabilities matrix** will be denoted as $\mathbf{P}$ and is stochastic. Thus, the random walker chooses to follow an edge with a likelihood proportional to the affinity (apart from the sum-to-one normalization), therefore favoring edges with a large affinity.

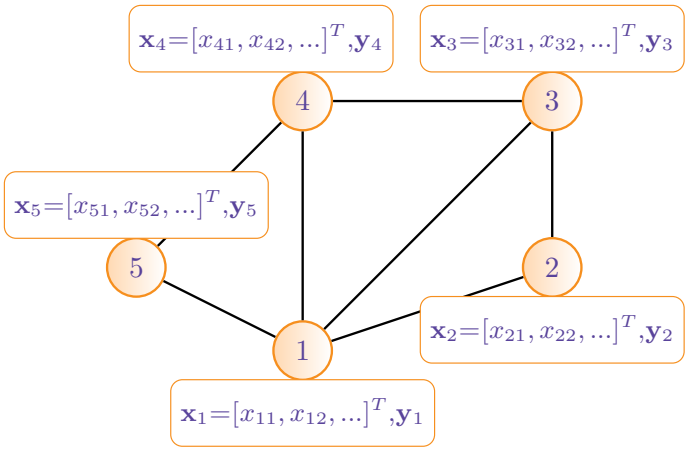


Figure 1: An example of graph with additional node information. Each node is characterized by a feature vector and a class label.

Moreover, we will consider that each of the nodes of $G$ has the same set of $m$ features, or attributes, with no missing values. The column vector $\mathbf{x}_i$ contains the values of the $m$ features of node $i$ and $x_{ij}$ states for the value of feature $j$ taken by node $i$. Moreover, $\mathbf{X}$ will refer to the $n \times m$ data matrix containing the $\mathbf{x}_i^T$ on its rows.

Finally we define $\mathbf{y}$ as the column vector containing the class labels of the nodes. Moreover, $\mathbf{y}^c$ is a binary vector indicating whether or not a node belongs to class number $c$.

Recall that the purpose of the classification tasks will to predict the class of the unlabeled data (in a transductive setting), or to predict new test data (in an inductive setting), while knowing the values of the features $\mathbf{X}$ for all the nodes of $G$ and the class labels $\mathbf{y}^c$ on the *labeled* nodes only for each $c$. Our baseline classifier based on features only will be a linear support vector machines (SVM).

## 3. Some related work

The 14 investigated models are presented in the next Section 4. In addition to those models, other approaches exist.

For example [21, 22] use a standard ridge regression model complemented by a Laplacian regularization term, and has been called the Laplacian regularized least squares. This option was investigated but provided poor results compared to reported models (an is therefore not reported). Note that using a logistic ridge regression as the base classifier was also investigated in this work but results are not reported here for conciseness as it provided performances similar to SVMs.

Laplacian support vector machines (LapSVMs) extend the SVM classifier in order to take the structure of the network into account. It exploits both the information on the nodes and the graph structure in order to categorize the nodes through its Laplacian matrix (see Section 2). To this end, [21] proposed to add a graph Laplacian regularization term to the traditional SVM cost function in order to obtain a semi-supervised version of this model. A matlab toolbox for this model is available but provided poor results in terms of performance and tractability.

Chakrabarti et al. developed, in the context of patents classification [23], a naive Bayes model in the presence of structural autocorrelation. The main idea is to use a naive Bayes classifier (see for example [24, 25, 26]) combining both feature information on the nodes and structural information by making some independence assumptions. More precisely, it is assumed that the label of a node is influenced by two sources: features of the node and labels of the neighboring nodes (and does not depend on other information). It first considers that labels of all neighboring nodes are known and then relax this constrain by using a kind of relaxation labeling (see [23] for details). However, we found out that this procedure is very time consuming, even for small-size networks, and decided to not include it in the present work.

Other semi-supervised classifiers based on network data only (features on nodes are not available) were also developed [15, 18]. The interested reader is invited to read, e.g., [27, 28, 13, 19, 29, 30, 31, 18, 32] (and included references), focused on this topic for a comprehensive description. Finally, an interesting survey and a comparative experiment of related methods can be found in [15].

## 4. Survey of relevant classification methods

The different classification methods compared in this work are briefly presented in this section, which is largely inspired by [19]. For a more thorough presentation, see the provided references to the original work or [19]. The classification models are sorted into different families: graph embedding-based classifiers, extensions of feature-based classifiers, and graph-based classifiers.

### *4.1. Graph embedding-based classifiers*

A first interesting way to combine information from the features on the nodes and from the graph structure is to perform a *graph embedding* projecting the nodes of the graph into a low-dimensional space (an embedding space) preserving as much as possible its structural information, and then use the coordinates of the projected nodes as *additional features* in a standard classification model, such as a logistic regression or a support vector machine.

This procedure has been proposed in the field of spatial statistics for ecological modeling [33, 34, 35], but also more recently in data mining [36, 37, 38, 39, 40]. While many graph embedding techniques could be used, [34] suggests to exploit Moran's or Geary's index of spatial autocorrelation in order to compute the embedding.

Let us briefly develop their approach (see [19]). Moran's $I$ and Geary's $c$ (see, e.g., [41, 42, 43, 44]) are two coefficients commonly used in spatial statistics in order to test the hypothesis of spatial autocorrelation of a continuous measure defined on the nodes. Four possibilities will be investigated to extract features from the graph structure: maximizing Moran's $I$, minimizing Geary's $c$, local principal component analysis and maximizing the Bag-of-Path (BoP) modularity.

*4.1.1. Maximizing Moran's I*

Moran's $I$ is given by

$$I(\mathbf{x}) = \frac{n}{a_{\bullet\bullet}} \frac{\sum_{i,j=1}^{n} a_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2} \tag{3}$$

where $x_i$ and $x_j$ are the values observed on nodes $i$ and $j$ respectively, for a considered quantity measured on the nodes. The column vector $\mathbf{x}$ is the vector containing the values $x_i$ and $\bar{x}$ is the average value of $\mathbf{x}$. Then, $a_{\bullet\bullet}$ is simply the sum of all entries of $\mathbf{A}$ – the volume of the graph.

$I(\mathbf{x})$ can be interpreted as a correlation coefficient similar to the Pearson correlation coefficient [41, 42, 43, 44]. The numerator is a measure of covariance among the neighboring $x_i$ in $G$, while the denominator is a measure of variance. $I$ is in the interval $[-1, +1]$. A value close to zero indicates no evidence of autocorrelation, a positive value indicates positive autocorrelation and a negative value indicates negative autocorrelation. Autocorrelation means that close nodes tend to take similar values.

In matrix form, Equation (3) can be rewritten as

$$I(\mathbf{x}) = \frac{n}{a_{\bullet\bullet}} \frac{\mathbf{x}^T\mathbf{HAHx}}{\mathbf{x}^T\mathbf{Hx}} \tag{4}$$

where $\mathbf{H} = (\mathbf{I} - \mathbf{E}/n)$ is the centering matrix [45] and $\mathbf{E}$ is a matrix full of ones. Note that the centering matrix is idempotent, $\mathbf{HH} = \mathbf{H}$.

The objective is now to find the score $\mathbf{x}$ that achieves the largest autocorrelation, as defined by Moran's index. This corresponds to the values that most explains the structure of $G$. It can be obtained by setting the gradient equal to zero; we then obtain the following generalized eigensystem:

$$\mathbf{HAHx}' = \lambda\mathbf{x}', \text{ and then } \mathbf{x} = \mathbf{Hx}' \tag{5}$$

The idea is thus to extract the first eigenvector $\mathbf{x}_1$ of the centered adjacency matrix (5) corresponding to the *largest* eigenvalue $\lambda_1$ and then to compute the second-largest eigenvector, $\mathbf{x}_2$, orthogonal to $\mathbf{x}_1$, etc. The eigenvalues $\lambda_i$ are proportional to the corresponding Moran's $I(\mathbf{x}_i)$.

The $p$ largest centered eigenvectors of (5) are thus extracted and then used as additional $p$ features for a supervised classification model (here an SVM). In other words, $\mathbf{X}_{\text{Moran}} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p]^T$ is a new data matrix, capturing the structural information of $G$, that can be concatenated to the feature-based data matrix $\mathbf{X}_{\text{feature}}$, forming the extended data matrix $[\mathbf{X}_{\text{feature}}, \mathbf{X}_{\text{Moran}}]$.

*4.1.2. Minimizing Geary's c*

On the other hand, Geary's $c$ is another weighted estimate of partial autocorrelation given by

$$c(\mathbf{x}) = \frac{(n-1)}{2\,a_{\bullet\bullet}} \frac{\sum_{i,j=1}^{n} a_{ij}(x_i - x_j)^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2} \tag{6}$$

and is related to Moran's $I$. However, while Moran's $I$ considers a covariance between neighboring nodes, Geary's $c$ considers distances between pairs of neighboring nodes. It ranges from 0 to 2 with 0 indicating perfect positive autocorrelation and 2 indicating perfect negative autocorrelation [35, 42, 44].

In matrix form, Geary's $c$ can be written as

$$c(\mathbf{x}) = \frac{(n-1)}{2a_{\bullet\bullet}} \frac{\mathbf{x}^T\mathbf{Lx}}{\mathbf{x}^T\mathbf{Hx}}. \tag{7}$$

This time, the objective is to find the score vector minimizing Geary's $c$. By proceeding as for Moran's $I$, we find that minimizing $c(\mathbf{x})$ aims to compute the *p lowest* non-trivial eigenvectors of the Laplacian matrix:

$$\mathbf{Lx} = \lambda\mathbf{Hx} \tag{8}$$

and then use these eigenvectors as additional $p$ features in a classification model. We therefore end up with the problem of computing the lowest eigenvectors of the Laplacian matrix, which also appears in spectral clustering (ratio cut, see, e.g., [46, 19, 47]).

Geary's $c$ has a computational advantage over Moran's $I$: the Laplacian matrix is usually sparse, which is not the case for Moran's $I$. Moreover, note that since the Laplacian matrix $\mathbf{L}$ is centered, any solution of $\mathbf{Lx} = \lambda\mathbf{x}$ is also a solution of Equation (8).

*4.1.3. Local principal component analysis*

In [48, 49], the authors propose to use a measure of local, structural, association between nodes. The **contiguity ratio** is defined as

$$cr(\mathbf{x}) = \frac{\sum_{i=1}^{n}(x_i - m_i)^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2}, \text{ with } m_i = \sum_{j \in \mathcal{N}(i)} p_{ij}x_j. \tag{9}$$

and $m_i$ is the average value observed on the neighbors of $i$, $\mathcal{N}(i)$. As for Geary's index, the value is close to zero when there is a strong structural association. However, there are no clear bounds indicating no structural association or negative correlation [49].

The numerator of Equation (9) is the mean squared difference between the value on a node and the average of its neighboring values; it is called the local variance in [49]. The denominator is the standard sample variance. In matrix form,

$$cr(\mathbf{x}) = \frac{\mathbf{x}^T(\mathbf{I} - \mathbf{P})^T(\mathbf{I} - \mathbf{P})\mathbf{x}}{\mathbf{x}^T\mathbf{Hx}}. \tag{10}$$

Proceeding as for Geary and Moran's indexes, minimizing $cr(\mathbf{x})$ aims to solve

$$(\mathbf{I} - \mathbf{P})^T(\mathbf{I} - \mathbf{P})\mathbf{x} = \lambda\mathbf{Hx} \tag{11}$$

Here again, eigenvectors corresponding to the smallest non-trivial eigenvalues of the eigensystem (11) are extracted. This procedure is also referred to as local principal component analysis in [49].

#### 4.1.4. Bag-of-path modularity

For this algorithm, we also compute a number of structural features, but now derived from the modularity measure computed in the bag-of-path (BoP) framework [50], and concatenate them to the node features $[\mathbf{X}_{\text{feature}}, \mathbf{X}_{\text{BoPMod}}]$. Again, a SVM is then used to classify all unlabeled nodes. Indeed, it has been shown that using the dominant eigenvectors of the BoP modularity matrix provides better performances than using the eigenvectors of the standard modularity matrix . The result for the standard modularity matrix are therefore not reported here.

It can be shown (see [50] for details) that the BoP modularity matrix is equal to

$$\mathbf{Q}_{\text{BoP}} = \mathbf{Z} - \frac{(\mathbf{Ze})(\mathbf{e}^T\mathbf{Z})}{\mathbf{e}^T\mathbf{Ze}} \tag{12}$$

where $\mathbf{Z}$ is the fundamental bag-of-path $n \times n$ matrix and $\mathbf{e}$ is a length $n$ column vector full of ones. Then as for Moran's $I$ and Geary's $c$, an eigensystem

$$\mathbf{Q}_{\text{BoP}}\mathbf{x} = \lambda\mathbf{x} \tag{13}$$

must be solved and the largest eigenvectors are used as new, additional, structural, features.

### 4.2. Extensions of standard feature-based classifiers

These techniques rely on extensions of standard feature-based classifiers (for instance a logistic regression model or a support vector machine). The extension is defined in order to take the network structure into account.

#### 4.2.1. The AutoSVM: taking autocovariates into account

This model is also known as the **autologistic** or **autologit** model [51, 52, 53, 54], and is frequently used in the spatial statistics and biostatistics fields.

Note that, as a SVM is used as base classifier in this work (see Section 1), we adapted this model (instead of the logistic regression in [53]) in order to take the graph structure into account. The method is based on the quantity $ac_i^c = \sum_{j\in\mathcal{N}(i)} p_{ij}\hat{y}_j^c$, where $\hat{y}_j^c$ is the predicted membership of node $j$, called the **autocovariate** in [53] (other forms are possible, see [52, 53]). It corresponds to the weighted averaged membership to class $c$ within the neighborhood of $i$: it indicates to which extent neighbors of $i$ belong to class $c$. The assumption is that node $i$ has a higher chance to belong to class $c$ if its neighbors also belong to that class. $\mathbf{Ac}$ will be the matrix containing $ac_i^c$ for all $i$ and $c$.

However, since the predicted value $\hat{y}_j^c$ depends on the occurrence of the predicted value on other nodes, building the model is not straightforward. For the autologistic model, it goes through the maximization of the (pseudo-)likelihood (see for example [55, 51]), but we will consider another alternative [53] which uses a kind of expectation-maximization-like heuristics (EM, see, e.g. [56, 57]), and is easy to adapt to our SVM classifier.

Here is a summary (see [19]) of the estimation procedure proposed in [53]:

1. At $t = 0$, initialize the predicted class memberships $\hat{y}_i^c(t = 0)$ of the unlabeled nodes by a standard SVM depending on the feature vectors only, from which we disregard the structural information (the information about neighbors' labels). For the labeled nodes, the membership values are not modified and are thus set to the true, observed, memberships.
2. Compute the current values of the autocovariates, $ac_i^c = \sum_{j\in\mathcal{N}(i)} p_{ij}\hat{y}_j^c(t)$, for all nodes.
3. Train a so-called **autoSVM** model based on these current autocovariate values as well as the features on nodes. This provides parameter estimates $\hat{\mathbf{w}}^c$.
4. Compute the predicted class memberships $\hat{y}_i^c(t + 1)$ of the set of unlabeled nodes from the fitted autoSVM model. This is done by sequentially selecting each unlabeled node $i$ in turn, and applying the fitted autoSVM model, obtained in step 3 based on the autocovariates of step 2. After having considered all the nodes, we have the new predicted values $\hat{y}_i^c(t + 1)$.
5. Steps 2 to 4 are be iterated until convergence of the predicted membership values $\hat{y}_i^c(t)$.

#### 4.2.2. Double kernel SVM

Here, we describe another simple way of combining the information coming from features on nodes and graph structure. The basic idea ([58, 19]) is to

1. Compute a $n \times n$ kernel matrix based on node features [59, 60], for instance a linear kernel or a gaussian kernel.
2. Compute a $n \times n$ kernel matrix on the graph [61, 19, 62, 60], for instance the regularized commute-time kernel (see Subsection 5.2.2).
3. Fit a SVM based on these two combined kernels.

Then, by using the kernel trick, everything happens as if the new data matrix is

$$\mathbf{X}_{\text{new}} = [\mathbf{K}_{\mathbf{A}}, \mathbf{K}_{\mathbf{X}}] \tag{14}$$

where $\mathbf{K}_G$ is a kernel on a graph and $\mathbf{K}_{\mathbf{X}} = \mathbf{XX}^T$ is the kernel matrix associated to the features on the nodes (see [19] for details). Then, we can fit a SVM classifier based on this new data matrix and the class labels of labeled nodes.

#### 4.2.3. A spatial autoregressive model

This model is a spatial extension of a standard regression model [63] and is well known in spatial econometrics. This extended model assumes that the vector of class memberships $\mathbf{y}^c$ is generated in each class $c$ according to

$$\mathbf{y}^c = \rho\mathbf{Py}^c + \mathbf{Xw}^c + \boldsymbol{\epsilon} \tag{15}$$

where $\mathbf{w}^c$ is the usual parameter vector, $\rho$ is a scalar parameter introduced to account for the structural dependency through $\mathbf{P}$ and $\epsilon$ is an error term. Obviously if $\rho$ is equal to zero, there is no structural dependency and the model reduce to a standard linear regression model. Lesage's Econometrics Matlab toolbox was used for the implementation of this model [63]; see this reference for more information.

### 4.3. Graph-based classifiers

We also investigate some semi-supervised methods based on the graph structure only (no node feature exists or features are simply not taken into account). We selected the techniques performing best in a series of experimental comparisons [61, 27, 64]. They rely on some strong assumptions about the distribution of labels: that neighboring nodes (or "close nodes") are likely to share the same class label [13].

#### 4.3.1. The bag-of-paths group betweenness

This model [27] considers a bag containing all the possible paths 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. The **Bag-of-Paths (BoP) probabilities**, $P(s = i, e = j)$, providing the probability of drawing a path starting in $i$ and ending in $j$, can be computed in closed form and a betweenness measure quantifying to which extend a node is in-between two nodes is defined. A node receives a high betweenness if it has a large probability of appearing on paths connecting two arbitrary nodes of the network. A group betweenness between classes is defined as the sum of the contribution of all paths starting and ending in a particular class, and passing through the considered node. Each unlabeled nodes is then classified according to the class showing the highest group betweenness. More information can be found in [27].

#### 4.3.2. A sum-of-similarities based on the regularized commute time kernel

We finally investigate a classification procedure based on a simple alignment with the regularized commute time kernel $\mathbf{K}$, a **sum-of-similarities** defined by $\mathbf{K}\mathbf{y}^c$, with $\mathbf{K} = (\mathbf{D} - \alpha\mathbf{A})^{-1}$ [65, 61, 19]. This expression quantifies to which extend each node is close (in terms of the similarity provided by the regularized commute time kernel) to class $c$. This similarity is computed for each class $c$ in turn. Then, each node is assigned to the class showing the largest sum of similarities. 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 a killed random walk where the random walker has a $(1 - \alpha)$ probability of disappearing at each step. This method provided good results in a comparative study on graph-based semi-supervised classification [61, 64, 66].

## 5. Experiments

In this section, the different classification methods will be compared on semi-supervised classification tasks and several datasets. The goal is to classify unlabeled nodes in partially labeled graphs and to compare the results obtained by the different methods in terms of classification accuracy.

This section is organized as follows. First, the datasets used for semi-supervised classification are described in Subsection 5.1. Then, the compared methods are recalled in Subsection 5.2. The experimental methodology is explained in Subsection 5.3. Finally, results are presented and discussed in Subsection 5.4.

Table 1: Class distribution of the four *WebKB cocite* datasets.

| **Class** | **Cornell** (DB1) | **Texas** (DB2) | **Washington** (DB3) | **Wisconsin** (DB4) |
|---|---|---|---|---|
| Course | 42 | 33 | 59 | 70 |
| Faculty | 32 | 30 | 25 | 32 |
| Student | 83 | 101 | 103 | 118 |
| Project + staff | 38 | 19 | 28 | 31 |
| **Total** | 195 | 183 | 230 | 251 |
| **Majority class (%)** | 42.6 | 55.2 | 44.8 | 47.0 |
| **Number of features** | 1704 | 1704 | 1704 | 1704 |

Table 2: Class distribution of the three *Ego facebook* datasets.

| **Class** | **FB 107** (DB5) | **FB 1684** (DB6) | **FB 1912** (DB7) |
|---|---|---|---|
| Main group | 737 | 568 | 524 |
| Other groups | 308 | 225 | 232 |
| **Total** | 1045 | 793 | 756 |
| **Majority class (%)** | 70.5 | 71.2 | 69.3 |
| **Number of features** | 576 | 319 | 480 |

### 5.1. Datasets

All datasets are described by (i) the adjacency matrix $\mathbf{A}$ of the underlying graph, (ii) a class vector (to predict) and (iii) a number of features on nodes gathered in the data matrix $\mathbf{X}_{\text{features}}$. Using a chi-square test, we kept only the 100 most significant variables for each dataset. The datasets are available at `http://www.isys.ucl.ac.be/staff/lebichot/research.htm`.

For each of these dataset, if more than one graph connected component is present, we only use the biggest connected component, deleting all the others nodes, features and target classes. Also, we choose to work with undirected graphs for all datasets: if a graph is directed, we used $\mathbf{A} = (\mathbf{A}^T + \mathbf{A})/2$ to introduce reciprocal edges.

- *WebKB cocite* (**DB1-DB4**) [67]. These four datasets consist of web pages gathered from computer science departments from four universities (there is four datasets, one for each university), with each page manually labeled into one of four categories: course, faculty, student and project [15]. The pages are linked by citations (if $x$ links to $y$ then it means that $y$ is cited by $x$). Each web page in the dataset is also characterized by a binary word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1703 unique words (words appearing less than 10 times were ignored). Originally, a fifth category, Staff, was present but since it contained only very few instances, it was merged with the Project class. Details on these datasets are shown in Table 1.

Table 3: Class distribution of the *Citeseer, Cora* and *Wikipedia* datasets.

| Class | Citeseer (DB8) | Cora (DB9) | Wikipedia (DB10) |
|---|---|---|---|
| Class 1 | 269 | 285 | 248 |
| Class 2 | 455 | 406 | 509 |
| Class 3 | 300 | 726 | 194 |
| Class 4 | 75 | 379 | 99 |
| Class 5 | 78 | 214 | 152 |
| Class 6 | 188 | 131 | 409 |
| Class 7 | | 344 | 181 |
| Class 8 | | | 128 |
| Class 9 | | | 364 |
| Class 10 | | | 351 |
| Class 11 | | | 194 |
| Class 12 | | | 81 |
| Class 13 | | | 233 |
| Class 14 | | | 111 |
| **Total** | 1392 | 2708 | 3271 |
| **Majority class (%)** | 32.7 | 26.8 | 15.6 |
| **Number of features** | 3703 | 1434 | 4973 |

- The three *Ego Facebook* datasets (**DB5-DB7**) [68] consist of circles (or friends communities) from Facebook. Facebook data were collected from survey participants using a Facebook application. The original dataset includes node features (profiles), circles, and ego networks for 10 networks. Those data are anonymized. We keep the three first networks and we define the classification task as follow: we picked the majority circle (the target circle) and aggregated all the others (non-target circles). Details on these datasets are shown in Table 2. Each dataset has two classes.

- The CiteSeer dataset (**DB8**) [67] consists of 3312 scientific publications classified into six classes. The pages are linked by citation (if $x$ links to $y$ then it means that $y$ is cited by $x$). Each publication in the dataset is described by a binary word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3703 unique words (words appearing less than 10 times were ignored). Target variable is the domain of the publications (six topics, not reported here). Details on this dataset are shown in Table 3.

- The Cora dataset (**DB9**) [67] consists of 2708 scientific publications classified into one of seven classes denoting topics as for previous dataset. Pages are linked by citations (if $x$ links to $y$ then it means that $y$ is cited by $x$). Each publication is also described by a binary word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1434 unique words or features (words appearing less than 10 times were ignored). Target variable is the topic of the publications. Details on this dataset are shown in Table 3.

- The Wikipedia dataset (**DB10**) [67] consists of 3271 Wikipedia articles that appeared in the featured list in the period Oct. 7-21, 2009. Each document belongs to one of 14 distinct categories (such as Science, Sport, Art, ...), which were obtained by using the category under which each article is listed. After stemming and stop-word removal, the content of each document is represented by a tf/idf-weighted feature vector, for a total of 4973 words. Pages are linked by citation (if $x$ links to $y$ then it means that $y$ is cited by $x$). Target variable is the articles field (14 different, not reported here). Details on this dataset are shown in Table 3.

Moreover, in order to study the impact of the relative information provided by the *graph structure* and the *features on nodes*, we created new derived datasets by *weakening gradually* the information provided by the node features. More precisely, for each dataset, the features available on the nodes have been ranked by decreasing association (using a chi-square statistics) with the target classes to be predicted. Then, datasets with *subsets* of the features containing respectively the 5 (5F), 10 (10F), 25 (25F), 50 (50F) and 100 (100F) most informative features were created (sets of features). These datasets are weakened versions of the original datasets, allowing to investigate the respective role of features on nodes and graph structure. We also investigate sets with more features (200 and 400). Conclusions were the same so that they are not reported here for conciseness.

### 5.2. Compared classification models

In this work, a transductive scheme is used, as we need to know the whole graph structure to label unlabeled nodes. 14 different algorithms will be compared and can be sorted in three categories, according to the information they use. Some algorithms use only features to build the model (denoted as X – data matrix with features only), others use only the graph structure (denoted as A – adjacency matrix of the graph only), and the third category uses both the structure of the graph and the features of the nodes (denoted as AX).

#### 5.2.1. Using features on nodes only

This reduces to a standard classification problem and we use a linear Support Vector Machine (SVM) based on the features of the nodes to label these nodes (**SVM-X**). Here, we consider SVMs in the binary classification setting (i.e. $\mathbf{y}_i \in \{-1, +1\}$). For multiclass problems, we used a one-vs-one strategy [69]. This classifier will be used as a baseline. In practical terms, we use the well-known Liblinear library [70]. Notice that SVM follows an inductive scheme, unlike all other methods. Transductive SVMs [71] were also considered, but their implementation was too time-consuming to be included in the present analysis.

#### 5.2.2. Using graph structure only

Three different families of methods using graph structure only are investigated.

- The Bag of Path classifier based on the bag-of-paths group betweenness (**BoP-A**). This betweenness is computed for each class in turn. Then, each unlabeled node is assigned to the class showing the largest value (see Section 4.3.1 for more details).

- The sum-of-similarities method based on the Regularized Commute Time Kernel (**CTK-A**). The classification procedure is the same as BoP-A: the class similarity is computed for each class in turn and each unlabeled node is assigned to the class showing the largest similarity (see Section 4.3.2 for more details).

- The four graph embedding techniques discussed in Section 4.1 are used together with an SVM, without considering any node feature. The SVM is trained using a given number of extracted eigenvectors derived from each measure (this number is a parameter to tune). The SVM model is then used to classify the unlabeled nodes.

  – SVM using Moran's $I$ derived dominant eigenvectors; see Section 4.1.1 (**SVM-M-A**).

  – SVM using Geary's $c$ derived dominant eigenvectors; see Section 4.1.2 (**SVM-G-A**).

  – SVM using the dominant eigenvectors extracted from local principal component analysis; see Section 4.1.3 (**SVM-L-A**).

  – SVM using the dominant eigenvectors extracted from the bag-of-paths modularity; see Section 4.1.4 (**SVM-BoPM-A**).

#### 5.2.3. *Using both information (features on nodes and graph structure)*

Here, we investigate:

- A double kernel SVM (**DK SVM**). In this case, two kernels are computed, one defined on the graph and the second from the node features $\mathbf{X}_{\text{new}} = [\mathbf{K}_{\text{A}}, \mathbf{K}_{\text{X}}]$ (see Section 4.2.2). A SVM is then used to classify the unlabeled nodes.

- Support vector machine using Autocovariates (**ASVM-AX**). In this algorithm, autocovariates are added to the node features $\mathbf{X}_{\text{new}} = [\mathbf{X}_{\text{feat}}, \mathbf{Ac}]$ (see Section 4.2.1).

- Spatial AutoRegressive model (**SAR-AX**). This model is a spatial extension of the standard regression model (see Section 4.2.3), used to classify the unlabeled nodes.

- The dominant eigenvectors (this number is a parameter to tune) provided by the four graph embedding techniques (Section 4.1) are combined with the node features and then injected into a linear SVM classifier:

  – SVM using Moran's $I$-derived features (see 4.1.1) in addition to node features (**SVM-M-AX**): $\mathbf{X}_{\text{new}} = [\mathbf{X}_{\text{feat}}, \mathbf{X}_{\text{Moran}}]$.

  – SVM using Geary's $c$-derived features (see 4.1.2) in addition to node features (**SVM-G-AX**): $\mathbf{X}_{\text{new}} = [\mathbf{X}_{\text{feat}}, \mathbf{X}_{\text{Geary}}]$.

  – SVM using graph local principal component analysis (see 4.1.3) in addition to node features (**SVM-L-AX**): $\mathbf{X}_{\text{new}} = [\mathbf{X}_{\text{feat}}, \mathbf{X}_{\text{LPC}}]$.

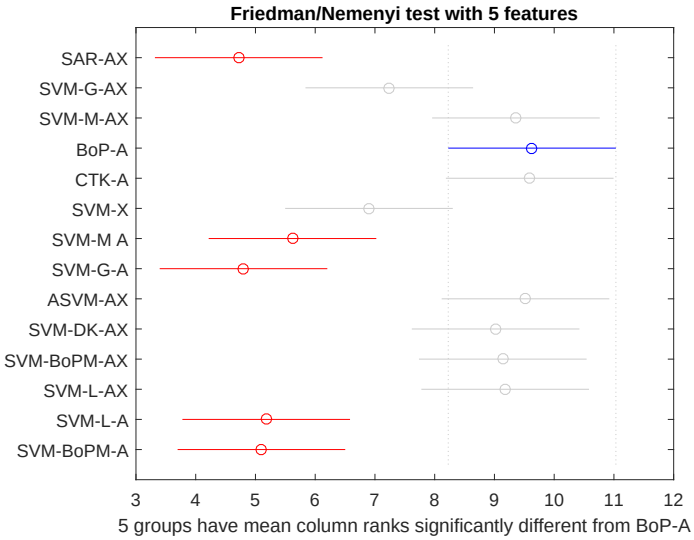


Figure 2: Mean rank (circles) and critical difference (plain line) of the Friedman/Nemenyi test, over 5 runs and all datasets, obtained on partially labeled graphs. The blue method has the best mean rank and is statistically better than red methods. Labeling rate is 20% and the critical difference is 2.81. This figure shows the results when only 5 node features are considered (5F datasets).

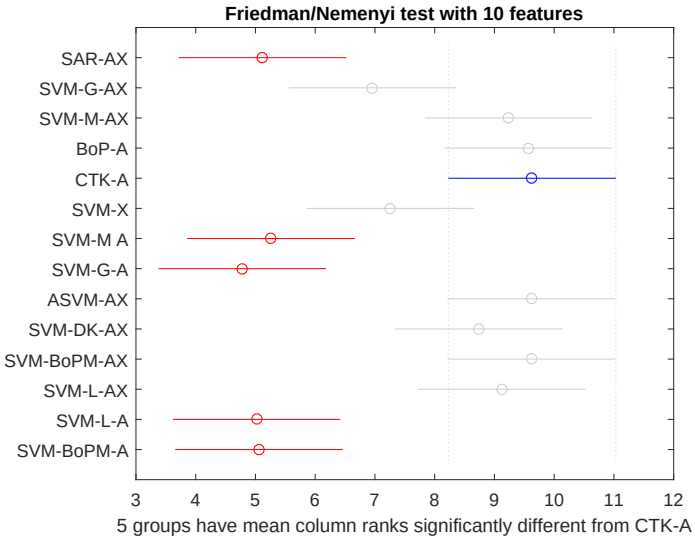


Figure 3: The Friedman/Nemenyi test considering 10 node features (10F datasets); see Figure 2 for details. The critical difference is 2.81.

  – SVM using bag-of-path modularity (see 4.1.4) in addition to node features (**SVM-BoPM-AX**): $\mathbf{X}_{\text{new}} = [\mathbf{X}_{\text{feat}}, \mathbf{X}_{\text{BoPMod}}]$.

The considered classifiers, together with their parameters to be tuned, are listed in Table 4.

### 5.3. *Experimental methodology*

The classification accuracy will be reported for a 20% labeling rate i.e. proportion of nodes for which labels are known. Labels of remaining nodes are deleted during model fitting phase

Table 4: The 14 classifiers, the value range tested for tuning their parameters and the most frequently selected values: Mode is the most selected value across all datasets. Note that $p$, the number of extracted eigenvector, is given in %: this is relative number of created features with respect to the number of node of the graph (different for each dataset).

| **Classification model** | **Use A** | **Use X** | **Acronym** | **Param.** | **Tested values** | **Mode** |
|---|---|---|---|---|---|---|
| Bag of paths betweenness (4.3.1) | yes | no | BoP-A | $\theta > 0$ | $10^{[-9,-6,-3,0]}$ | $10^{-6}$(40.2%) |
| Sum of similarities based on the RCT kernel (4.3.2) | yes | no | CTK-A | $\lambda > 0$ | 0.2, 0.4, 0.6, 0.8, 1 | 0.8(39.4%) |
| SVM based on Moran's extracted features only (4.1.1) | yes | no | SVM-M-A | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{-2}$(63.0%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(74.0%) |
| SVM based on Geary's extracted features only (4.1.2) | yes | no | SVM-G-A | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{-2}$(34.8%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(39.6%) |
| SVM based on LPCA's extracted features only (4.1.3) | yes | no | SVM-L-A | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{2}$(47.3%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(69.5%) |
| SVM based on BoP modularity's extracted features (4.1.4) | yes | no | SVM-BoPM-A | $\theta > 0$ | $10^{[-9,-6,-3,0]}$ | $10^{0}$(35.2%) |
| | | | | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{3}$(44.4%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(72.0%) |
| SVM based on node features only (**baseline**) | no | yes | SVM-X | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{-2}$(27.2%) |
| Spatial autoregressive model (4.2.3) | yes | yes | SAR-AX | *none* | – | - |
| SVM on Moran and features on nodes (4.1.1) | yes | yes | SVM-M-AX | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{2}$(26.9%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(33.3%) |
| SVM on Geary and features on nodes (4.1.2) | yes | yes | SVM-G-AX | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{2}$(21.2%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(31.8%) |
| SVM on LPCA and features on nodes (4.1.3) | yes | yes | SVM-L-AX | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{2}$(28.4%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(41.8%) |
| SVM on BoP modularity and features on nodes (4.1.4) | yes | yes | SVM-BoPM-AX | $\theta > 0$ | $10^{[-9,-6,-3,0]}$ | $10^{0}$(27.4%) |
| | | | | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{3}$(41.0%) |
| | | | | $p > 0$ | [5, 10, 20, 35, 50%] | 5%(48.9%) |
| SVM on autocovatiates and features on nodes (4.2.1) | yes | yes | ASVM-AX | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{0}$(28.1%) |
| SVM on a double kernel (4.2.2) | yes | yes | SVM-DK-AX | $C > 0$ | $10^{[-6,-4,-2,0,2,4,6]}$ | $10^{-4}$(31.7%) |

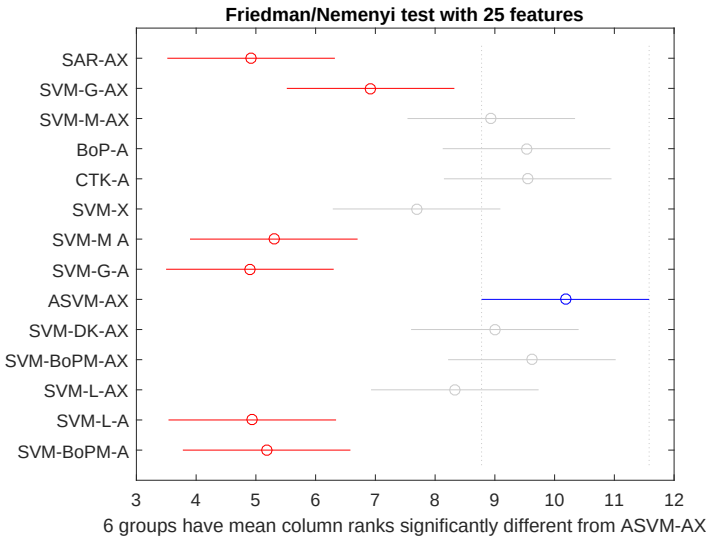


Figure 4: The Friedman/Nemenyi test considering 25 node features (25F datasets); see Figure 2 for details. The critical difference is 2.81.

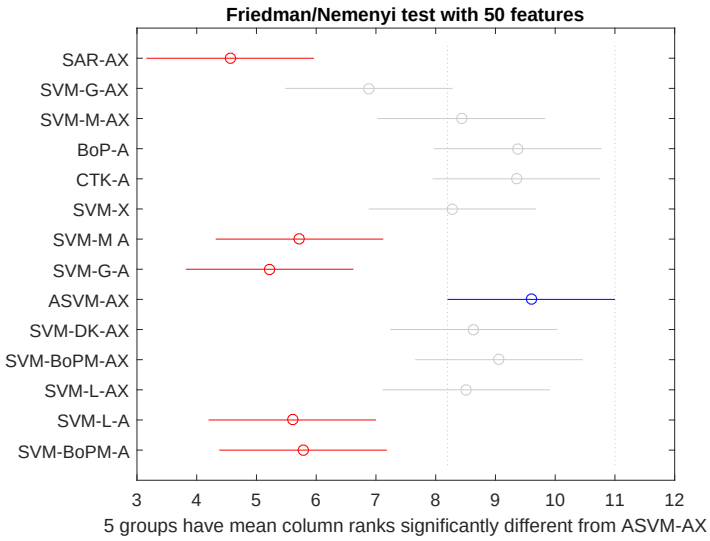


Figure 5: The Friedman/Nemenyi test considering 50 node features (50F datasets); see Figure 2 for details. The critical difference is 2.81.

and are used as test data during the assessment phase, where the various classification models predict the most suitable category of each unlabeled node in the test set.

For each considered feature sets and for each dataset, samples are randomly assigned into 25 folds: 5 external folds are defined and, for each of them, 5 nested folds are also defined. This procedure is performed 5 time to get 5 *runs* for each feature sets and dataset. The performances for one specific run are then computed by taking the average over the 5 external folds and, for each of them, a 5-fold nested cross-validation is performed to tune the parameters of the models on the inner folds (see Table 4).

Table 5: Classification accuracy in percent ± standard deviation, averaged over the 5 runs, obtained for the 14 methods and the ten datasets. Results are reported for the five feature sets (100F stands for the set of 100 features, and so on). The standard deviation is computed over the 5 folds of the external cross-validation and the 5 independent runs. Bold results are the best obtained performance for each different dataset and feature set.

| | $l$ | DB1 | DB2 | DB3 | DB4 | DB5 | DB6 | DB7 | DB8 | DB9 | DB10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SAR AX | 100F | 53.18±12.66 | 62.90±5.20 | 64.16±12.41 | 70.55±7.87 | 50.68±25.83 | 72.71±23.86 | 56.36±24.09 | 61.18±9.57 | 77.50±1.34 | 32.07±10.75 |
| | 50F | 66.94±9.75 | 66.54±5.19 | 65.86±8.12 | 71.98±7.00 | 66.15±28.63 | 78.89±17.94 | 61.48±22.45 | 64.52±3.16 | 64.28±5.23 | 35.64±7.59 |
| | 25F | 65.43±10.55 | 66.90±5.20 | 69.07±4.48 | 68.08±6.99 | 80.29±20.51 | 86.28±10.30 | 69.68±12.28 | 56.04±7.31 | 53.60±9.30 | 35.44±5.74 |
| | 10F | 64.37±5.13 | 64.84±5.62 | 63.56±4.99 | 67.70±6.16 | 76.14±9.22 | 78.32±9.27 | 66.02±13.80 | 44.18±10.32 | 42.93±9.14 | 29.03±2.48 |
| | 5F | 62.03±4.93 | 55.70±8.39 | 60.47±6.69 | 61.64±8.26 | 74.70±7.70 | 78.12±9.39 | 66.17±13.90 | 42.74±11.03 | 37.07±6.25 | 21.20±1.84 |
| SVM-G AX | 100F | 83.99±3.19 | **80.98±2.36** | 80.77±3.84 | 83.17±2.00 | 87.82±1.76 | 91.63±1.72 | 74.94±2.05 | 70.46±1.06 | 71.37±1.54 | 54.62±0.91 |
| | 50F | 79.51±3.81 | 76.64±2.64 | 77.21±3.27 | 78.47±3.81 | 87.59±1.35 | 89.65±1.10 | 77.45±1.78 | 62.92±1.21 | 65.98±1.27 | 45.94±2.42 |
| | 25F | 69.77±4.36 | 70.60±4.48 | 73.57±3.18 | 74.32±3.69 | 90.36±1.24 | 91.65±1.61 | 79.31±1.05 | 59.92±1.46 | 67.51±2.18 | 43.40±1.70 |
| | 10F | 63.03±2.62 | 58.89±5.68 | 64.00±4.36 | 64.76±4.32 | 93.12±1.61 | 93.47±1.27 | 78.70±3.10 | 57.11±2.15 | 72.06±1.77 | 39.23±1.22 |
| | 5F | 57.31±4.58 | 56.83±6.35 | 64.28±5.44 | 59.56±3.16 | 93.21±2.14 | 93.34±1.53 | 79.86±2.68 | 57.14±2.57 | 73.19±2.70 | 35.90±1.29 |
| SVM-M AX | 100F | 83.67±3.50 | 80.62±3.32 | 80.70±3.97 | 83.25±2.17 | 87.68±2.06 | 93.17±3.56 | 74.91±2.62 | 70.47±1.19 | 71.28±1.24 | 54.41±0.87 |
| | 50F | 79.46±3.72 | 76.45±3.08 | 77.31±2.97 | 78.51±4.85 | 89.93±1.19 | 94.97±2.92 | 76.03±1.78 | 64.10±1.57 | 72.07±2.60 | 46.67±2.00 |
| | 25F | 68.10±5.00 | 71.01±4.75 | 73.05±3.34 | 74.42±3.99 | 93.73±2.60 | 96.56±2.01 | 79.94±2.65 | 65.41±1.08 | 73.83±1.63 | 45.50±1.21 |
| | 10F | 62.21±4.30 | 57.29±6.67 | 63.67±3.94 | 64.23±4.05 | 94.91±1.73 | 97.59±1.43 | 80.66±2.55 | 66.33±2.15 | 76.36±1.19 | 41.91±1.74 |
| | 5F | 58.11±4.65 | 56.37±5.53 | 65.35±2.69 | 59.38±3.04 | 94.14±2.01 | 97.55±1.57 | 79.78±2.72 | 67.51±3.51 | 76.06±1.01 | 38.74±1.77 |
| BoP A | 100F | 54.45±3.65 | 41.84±4.25 | 48.42±0.85 | 45.70±0.21 | **96.76±10.34** | **98.63±11.08** | 82.48±6.06 | 69.91±19.19 | 78.11±19.89 | 35.34±4.16 |
| | 50F | 54.45±3.65 | 41.84±4.25 | 48.51±0.90 | 45.72±0.21 | **96.74±12.06** | **98.62±11.82** | 82.50±6.02 | 69.91±19.19 | 78.11±19.89 | 35.34±4.16 |
| | 25F | 54.45±3.65 | 41.84±4.25 | 48.33±0.88 | 45.64±0.15 | **96.80±11.26** | **98.62±10.15** | 82.50±6.04 | 69.91±19.19 | 78.11±19.89 | 35.34±4.16 |
| | 10F | 54.45±3.65 | 41.84±4.25 | 48.33±0.90 | 45.70±0.32 | **96.76±12.06** | **98.63±11.08** | 82.49±6.00 | 69.91±19.19 | 78.11±19.89 | 35.34±4.16 |
| | 5F | 54.45±3.65 | 41.84±4.25 | 48.30±0.86 | 45.70±0.25 | **96.77±10.33** | **98.63±11.82** | 82.50±6.07 | 69.91±19.19 | 78.11±19.89 | 35.34±4.16 |
| CTK A | 100F | 54.21±2.66 | 42.43±4.06 | 46.70±3.13 | 42.94±7.17 | 96.30±0.33 | 98.44±0.40 | **82.87±1.21** | 70.46±1.27 | **81.69±0.78** | 36.38±1.00 |
| | 50F | 54.21±2.66 | 42.43±4.06 | 47.25±2.80 | 42.88±7.14 | 96.31±0.34 | 98.44±0.38 | **82.92±1.17** | **70.46±1.27** | **81.69±0.78** | 36.38±1.00 |
| | 25F | 54.21±2.66 | 42.41±4.06 | 46.86±2.95 | 42.92±7.17 | 96.27±0.31 | 98.44±0.40 | **82.91±1.18** | **70.46±1.27** | **81.69±0.78** | 36.38±1.00 |
| | 10F | 54.21±2.66 | 42.49±3.95 | 46.87±3.02 | 42.98±7.17 | 96.32±0.34 | 98.41±0.41 | **82.92±1.19** | 70.46±1.27 | **81.69±0.78** | 36.38±1.00 |
| | 5F | 53.86±3.13 | 42.46±3.95 | 47.14±2.85 | 42.94±7.17 | 96.30±0.29 | 98.42±0.38 | **82.90±1.17** | 70.46±1.27 | **81.69±0.78** | 36.38±1.00 |
| SVM X | 100F | 83.72±3.70 | 80.76±2.46 | 80.93±3.88 | 83.11±2.54 | 88.45±1.10 | 91.41±1.15 | 74.66±2.65 | 70.46±1.12 | 71.41±0.89 | 54.59±1.06 |
| | 50F | 80.60±2.34 | **76.91±2.78** | **78.33±3.24** | 81.12±4.07 | 89.10±1.35 | 91.19±1.11 | 77.98±2.24 | 68.81±1.00 | 68.58±0.85 | 40.21±1.51 |
| | 25F | 74.56±3.29 | **75.49±4.25** | 77.52±2.89 | 79.42±3.19 | 89.45±1.68 | 91.76±1.12 | 78.89±5.87 | 66.68±1.07 | 64.18±0.63 | 34.23±0.96 |
| | 10F | 70.85±3.36 | **74.35±3.78** | 75.64±4.13 | 71.71±3.27 | 89.52±1.54 | 91.97±0.67 | 80.52±0.93 | 59.45±1.33 | 56.27±1.86 | 30.82±0.51 |
| | 5F | 65.55±3.13 | 66.17±7.07 | 71.02±2.88 | 75.02±2.44 | 87.18±0.92 | 92.19±0.49 | 78.48±5.79 | 53.90±1.60 | 42.26±3.22 | 25.23±0.46 |
| SVM-M A | 100F | 46.26±6.78 | 33.13±5.08 | 47.03±7.53 | 40.03±4.52 | 95.43±1.10 | 95.07±1.40 | 81.60±1.59 | 55.93±1.63 | 74.51±1.62 | 30.82±1.10 |
| | 50F | 46.26±6.78 | 33.13±5.08 | 47.03±7.36 | 40.09±4.52 | 95.43±1.10 | 95.06±1.53 | 81.61±1.59 | 55.93±1.63 | 74.51±1.62 | 30.82±1.10 |
| | 25F | 46.26±6.78 | 32.96±5.08 | 47.10±7.51 | 40.09±4.52 | 95.43±1.09 | 95.15±1.53 | 81.61±1.59 | 55.93±1.63 | 74.51±1.62 | 30.82±1.10 |
| | 10F | 46.29±6.82 | 33.13±5.08 | 47.17±7.46 | 40.03±4.50 | 95.43±1.09 | 95.08±1.53 | 81.60±1.59 | 55.93±1.63 | 74.51±1.62 | 30.82±1.10 |
| | 5F | 46.39±6.78 | 33.04±5.08 | 47.14±7.51 | 40.03±4.50 | 95.43±1.09 | 95.07±1.53 | 81.61±1.59 | 55.93±1.63 | 74.51±1.62 | 30.82±1.10 |
| SVM-G A | 100F | 43.45±8.24 | 33.32±6.54 | 44.10±4.30 | 39.97±4.93 | 89.82±12.23 | 93.79±1.53 | 78.53±1.25 | 68.13±1.93 | 75.58±1.16 | 35.03±1.49 |
| | 50F | 43.45±8.54 | 33.13±6.55 | 44.21±4.03 | 39.79±4.93 | 89.82±12.23 | 93.67±1.55 | 78.53±1.24 | 68.13±1.93 | 75.58±1.16 | 35.03±1.49 |
| | 25F | 43.45±7.89 | 33.29±6.54 | 44.21±4.10 | 39.81±4.48 | 89.98±12.20 | 93.80±1.54 | 78.52±1.24 | 68.14±1.93 | 75.58±1.16 | 35.03±1.49 |
| | 10F | 43.45±7.89 | 33.15±6.60 | 44.21±4.23 | 39.97±4.93 | 89.95±12.23 | 93.54±1.44 | 78.52±1.25 | 68.13±1.93 | 75.58±1.16 | 35.03±1.49 |
| | 5F | 43.45±8.54 | 33.15±6.57 | 44.17±4.03 | 39.81±4.93 | 89.83±12.23 | 93.66±1.53 | 78.52±1.24 | 68.13±1.93 | 75.58±1.16 | 35.03±1.49 |
| ASVM AX | 100F | 79.38±3.71 | 75.08±3.78 | 80.10±3.29 | 81.12±3.69 | 92.07±1.74 | 92.65±2.27 | 79.80±1.17 | 66.14±1.97 | 70.58±3.58 | 44.55±4.77 |
| | 50F | 79.21±4.09 | 74.40±2.84 | 77.66±3.51 | 79.41±4.43 | 92.32±2.16 | 94.76±1.90 | 80.50±2.41 | 66.08±1.63 | 76.28±2.43 | 37.13±7.96 |
| | 25F | **74.64±4.89** | 73.27±3.50 | 75.74±3.76 | **79.80±3.48** | 95.34±1.80 | 95.98±1.52 | 81.59±2.75 | 65.96±2.59 | 76.95±2.46 | 32.66±8.29 |
| | 10F | **71.41±3.45** | 72.29±6.41 | **75.74±7.78** | **76.15±4.31** | 95.48±1.76 | 96.65±1.34 | 81.16±2.99 | 62.93±3.41 | 74.93±2.85 | 28.24±6.85 |
| | 5F | **65.82±7.35** | **70.92±6.94** | 69.39±6.70 | **75.46±7.53** | 95.58±1.05 | 96.60±1.93 | 80.85±5.92 | 61.89±3.59 | 71.80±4.94 | 25.10±6.08 |
| SVM-DK AX | 100F | **84.89±2.59** | 79.89±3.40 | **81.31±3.31** | **84.34±2.50** | 88.75±2.04 | 92.14±0.91 | 76.09±1.95 | 70.38±0.76 | 71.32±1.20 | 54.23±0.88 |
| | 50F | **80.88±4.78** | 76.79±2.72 | 77.74±5.01 | 80.05±4.56 | 88.88±3.96 | 91.54±3.52 | 79.62±3.46 | 68.50±2.06 | 69.43±3.88 | 41.90±2.39 |
| | 25F | 73.59±4.12 | 74.24±3.09 | 76.89±4.23 | 78.65±3.88 | 89.08±2.64 | 91.95±5.71 | 80.32±3.95 | 70.25±2.10 | 74.04±10.60 | 36.74±3.48 |
| | 10F | 69.68±3.54 | 72.36±5.40 | 74.91±7.21 | 72.23±4.44 | 89.40±7.05 | 89.70±9.53 | 80.22±4.34 | **72.71±1.65** | 76.62±17.59 | 31.83±3.17 |
| | 5F | 65.17±5.68 | 68.11±8.81 | **71.12±7.52** | 74.42±9.03 | 87.25±8.24 | 88.54±17.01 | 80.48±4.20 | **71.97±2.90** | 78.01±19.64 | 25.82±2.82 |
| SVM-BoPM AX | 100F | 84.07±4.00 | 80.43±3.02 | 81.19±2.98 | 83.39±2.48 | 88.15±1.54 | 92.54±3.27 | 74.10±2.15 | **70.60±0.86** | 71.70±2.17 | **56.24±2.61** |
| | 50F | 79.05±4.53 | 76.58±2.37 | 78.31±6.22 | **81.64±4.71** | 88.94±2.53 | 92.92±2.71 | 77.75±3.44 | 68.67±3.13 | 71.94±2.99 | 44.97±2.41 |
| | 25F | 73.89±5.29 | 73.38±6.51 | **78.19±6.81** | 79.19±5.48 | 91.08±3.46 | 94.67±2.02 | 79.13±3.19 | 66.05±3.69 | 76.21±3.24 | 41.60±3.04 |
| | 10F | 69.84±6.26 | 72.09±8.59 | 74.86±7.14 | 70.15±5.75 | 92.92±2.05 | 95.60±1.81 | 79.33±3.21 | 63.64±2.17 | 77.26±1.85 | 42.20±2.55 |
| | 5F | 60.38±7.24 | 65.70±9.05 | 68.45±9.61 | 73.70±9.29 | 90.13±2.62 | 95.00±2.61 | 77.12±7.46 | 63.05±3.33 | 80.35±1.67 | **41.99±2.23** |
| SVM-L AX | 100F | 83.67±3.72 | 80.41±3.00 | 80.79±4.02 | 83.15±2.50 | 87.90±1.80 | 91.82±2.44 | 75.26±2.11 | 70.48±1.11 | 71.18±1.11 | 54.87±0.87 |
| | 50F | 79.46±3.62 | 76.70±3.02 | 77.26±2.97 | 78.75±3.75 | 88.78±3.47 | 93.09±2.43 | 78.40±1.96 | 62.81±4.09 | 73.04±3.04 | **48.56±1.62** |
| | 25F | 68.57±6.27 | 70.99±4.87 | 73.45±4.07 | 74.11±5.22 | 90.77±2.29 | 95.20±3.24 | 79.95±1.57 | 62.33±2.68 | 73.92±3.37 | **46.22±2.66** |
| | 10F | 62.71±5.48 | 58.59±9.74 | 64.61±8.36 | 65.69±7.10 | 93.68±1.95 | 96.65±1.56 | 81.18±1.90 | 61.35±3.82 | 76.79±2.00 | **42.31±1.98** |
| | 5F | 57.95±5.99 | 57.27±7.30 | 64.12±8.45 | 58.83±8.49 | 94.19±2.30 | 96.47±2.80 | 80.94±1.80 | 61.91±4.95 | 76.33±1.99 | 39.62±2.44 |
| SVM-L A | 100F | 40.85±4.08 | 33.07±5.31 | 40.43±4.70 | 42.01±3.00 | 90.93±1.92 | 95.85±1.75 | 79.07±2.33 | 62.40±1.67 | 76.58±0.93 | 34.42±1.00 |
| | 50F | 40.85±4.10 | 33.59±4.88 | 39.66±5.11 | 41.99±3.00 | 90.93±1.92 | 95.83±1.75 | 79.07±2.33 | 62.40±1.67 | 76.58±0.93 | 34.42±1.00 |
| | 25F | 40.74±4.08 | 33.96±5.57 | 39.56±5.13 | 42.01±3.00 | 90.96±1.87 | 95.80±1.78 | 79.07±2.33 | 62.40±1.67 | 76.58±0.93 | 34.42±1.00 |
| | 10F | 40.79±4.10 | 33.67±4.90 | 39.73±4.94 | 42.01±3.00 | 90.93±1.92 | 95.72±1.80 | 79.07±2.33 | 62.40±1.67 | 76.58±0.93 | 34.42±1.00 |
| | 5F | 40.79±4.13 | 33.96±5.57 | 39.68±5.11 | 42.01±3.00 | 90.96±1.87 | 95.71±1.80 | 79.07±2.33 | 62.40±1.67 | 76.58±0.93 | 34.42±1.00 |
| SVM-BoPM A | 100F | 43.54±0.78 | 32.32±3.38 | 39.35±2.34 | 42.61±3.32 | 91.75±0.63 | 94.41±1.12 | 79.13±0.87 | 67.50±1.16 | 80.28±0.31 | 14.87±0.34 |
| | 50F | 43.54±0.78 | 32.32±3.38 | 39.35±2.34 | 42.61±3.32 | 91.75±0.63 | 94.41±1.12 | 79.13±0.87 | 67.50±1.16 | 80.28±0.31 | 14.87±0.34 |
| | 25F | 43.54±0.78 | 32.32±3.38 | 39.35±2.34 | 42.61±3.32 | 91.75±0.63 | 94.41±1.12 | 79.13±0.87 | 67.50±1.16 | 80.28±0.31 | 14.87±0.34 |
| | 10F | 43.54±0.78 | 32.32±3.38 | 39.35±2.34 | 42.61±3.32 | 91.75±0.63 | 94.41±1.12 | 79.13±0.87 | 67.50±1.16 | 80.28±0.31 | 14.87±0.34 |
| | 5F | 43.54±0.78 | 32.32±3.38 | 39.35±2.34 | 42.61±3.32 | 91.75±0.63 | 94.41±1.12 | 79.13±0.87 | 67.50±1.16 | 80.28±0.31 | 14.87±0.34 |

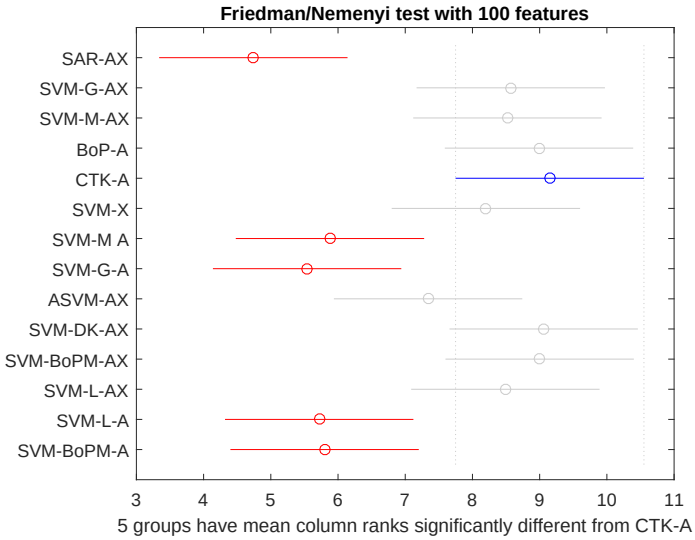


Figure 6: The Friedman/Nemenyi test considering 100 node features (100F datasets); see Figure 2 for details. The critical difference is 2.81.

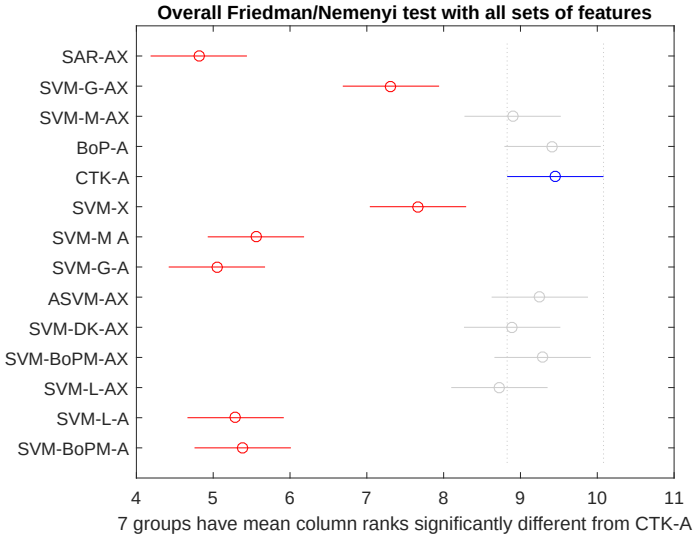


Figure 7: Friedman/Nemenyi test considering all feature sets (5F, 10F, 25F , 50F, 100F). The critical difference is 1.25; see Figure 2 for details.

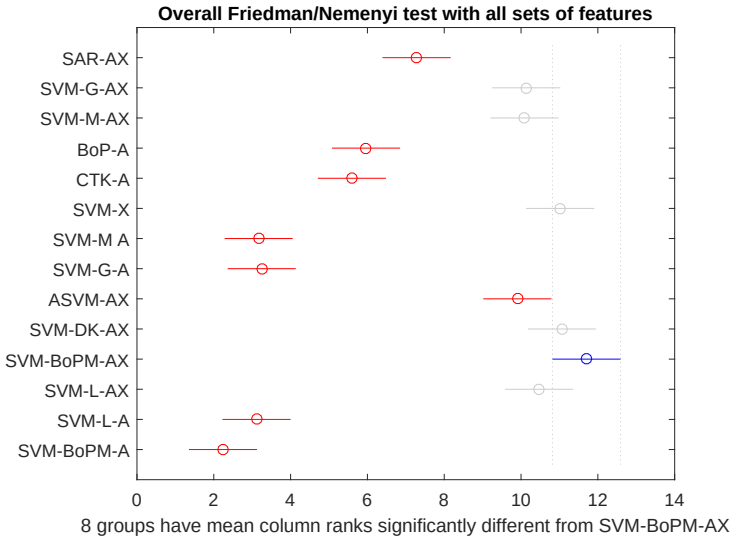


Figure 8: Friedman/Nemenyi test considering all feature sets (5F, 10F, 25F, 50F, 100F), but only on databases DB1 to DB4 and DB10 (driven by node features, **X**); see Figure 2 for details. The critical difference is 1.77.

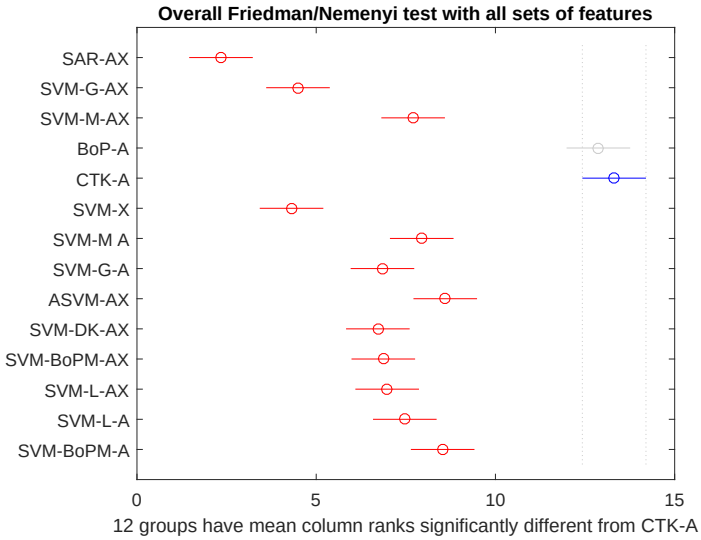


Figure 9: Friedman/Nemenyi test considering all feature sets (5F, 10F, 25F, 50F, 100F), but only on databases DB5 to DB9 (driven by graph structure, **A**); see Figure 2 for details. The critical difference is 1.77.

### *5.4. Results and discussion*

First of all, most frequently selected parameters values are indicated on Table 4. We observe that the most selected value for $p$ (the number of eigenvectors extracted for representing the graph structure; see Section 4.1) is actually low. This is a good news since efficient eigensystem solvers can be used to compute the first eigenvectors corresponding to the largest (or smallest) eigenvalues.

The classification accuracy and standard deviation, averaged on the 5 runs, are reported on Table 5, for the 10 different datasets and the 5 sets of features. Bold values indicate the best performance on those 50 combinations. Recall that the BoP-A, CTK-A, SVM-M-A, SVM-G-A, and SVM-L-A methods do not depend on the node features as they are based on the graph structure only. It explains why results do not depend on feature set for those five methods.

Moreover, the different classifiers have been compared across datasets through a Friedman test and a Nemenyi post-hoc test [72]. The Friedman test is a non-parametric equivalent of the repeated-measures ANOVA. It ranks the methods for each dataset separately, the best algorithm getting the rank 1, the second best rank 2, etc. Once the null hypothesis (the mean ranking of all methods is equal, meaning all classifiers are equiva-

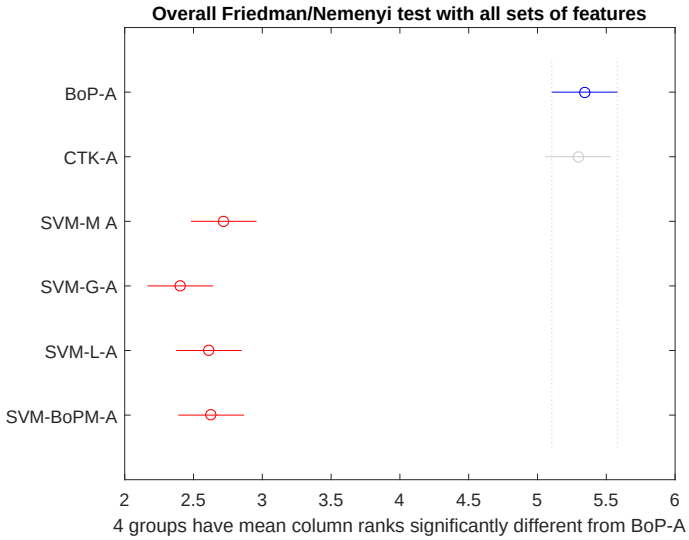


Figure 10: Friedman/Nemenyi test considering all feature sets (5F, 10F, 25F, 50F, 100F), for methods based on graph information alone (A); see Figure 2 for details. The critical difference is 0.48.

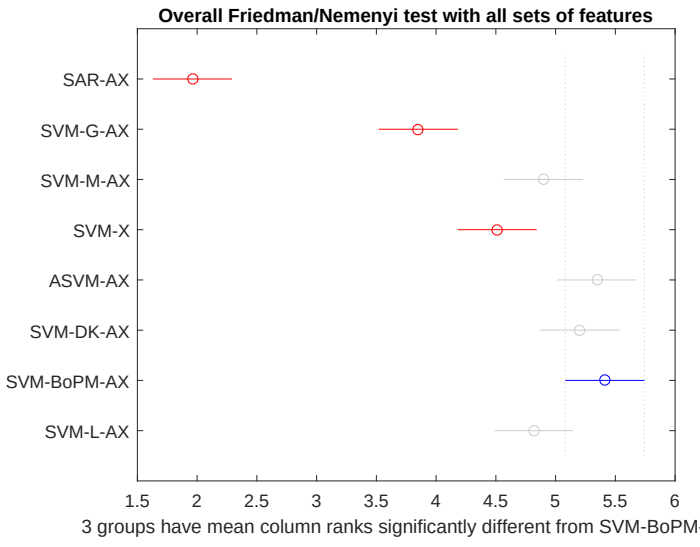


Figure 11: Friedman/Nemenyi test considering all feature sets (5F, 10F, 25F, 50F, 100F), performed only on methods combining graph information and node information (AX, plus simple SVM-X as baseline). See Figure 2 for details. The critical difference is 0.66.

lent) is rejected with $p$-value $< 0.05$, the (non parametric) post-hoc Nemenyi test is then computed. Notice that all Friedman tests were found to be positive. The Nemenyi test determines whether or not each method is significantly better ($p$-value less than 0.05 based on the 5 runs) to another. This is reported, for each feature set in turn (5F, 10F, ..., 100F), and thus increasing information available on the nodes, in Figures 2 to 6, and an overall test based on all the features sets and datasets is reported in Figure 7.

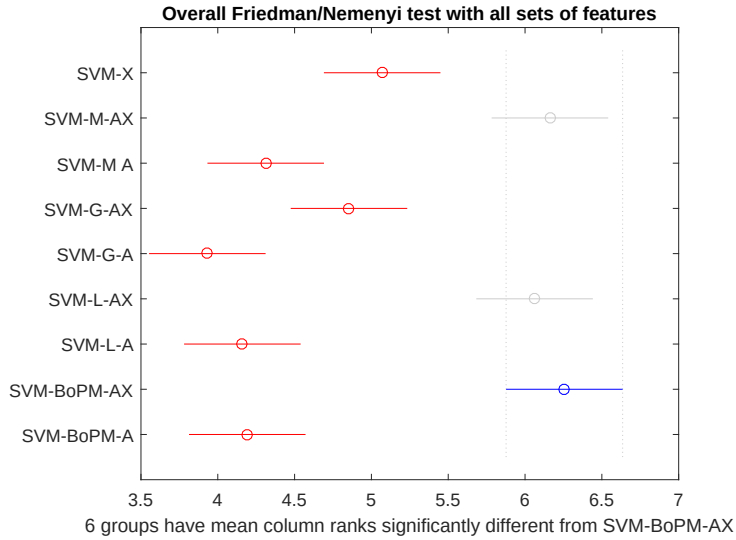


Figure 12: Friedman/Nemenyi test considering all feature sets (5F, 10F, 25F, 50F, 100F), for methods based on a graph embedding (plus regular linear SVM for comparison). See Figure 2 for details. The critical difference is 0.76.

#### 5.4.1. *Overall performances on all datasets and all node feature sets*

From Table 5 and Figure 7, overall best performances on all dataset and all node feature sets are often obtained either by a SVM based on node features combined with new features derived from the graph structure (Subsection 4.1), or, unexpectedly, by the CTK-A sum-of-similarities method (using graph structure only; see Subsection 4.3.2), which performs quite well on datasets five to nine. The BoP-A node betweenness (using graph structure only, see Subsection 4.3.1) is also competitive and achieves results similar to the sum-of-similarities CTK-A method (as already observed in [27]).

On the contrary, the best method among the graph structure plus node features SVM is not straightforward do determine (see Figure 7). From Figures 2 to 6, the main trend is that the performance decreases when the number of features decreases, which seems normal.

However, this trend is not always observed; for example, with the SVM-M-AX method (SVM with features extracted from Moran's index and features on nodes, see Subsection 4.1.1) and database DB5, the performances rise when the number of features decreases. This can be explained if we realize that each dataset can be better described in terms of its graph structure (graph-driven dataset, databases DB5 to DB9), or by its node features (features-driven dataset, databases DB1 to DB4 and DB10).

To highlight this behavior, the network autocorrelation for each class (i.e., for each $\mathbf{y}^c$) was computed and the average is reported for each dataset. This measure quantifies to which extent the target variable is correlated through on neighboring nodes. The values are reported on Table 6 for Moran's $I$ , Geary's $c$ and LPCA contiguity ratio (see Subsection 4.1.1). For Moran's $I$, high values (large autocorrelation) indicates that the graph

Table 6: Mean autocorrelation of class memberships. For Moran's $I$, high value implies large autocorrelation. For Geary's $c$ and LPCA, small value implies large autocorrelation. For each measure, **-** indicates the bound of maximum negative autocorrelation, a value close to **0** indicates a lack of structural association and **+** indicates the bound of maximum positive autocorrelation. See Subsection 4.1 for details. Datasets can be divided into two groups (more driven by **A** or by **X**) according to these measures.

| **A-driven** | **+** | **0** | **-** | DB 1 | DB 2 | DB 3 | DB 4 | DB 10 |
|---|---|---|---|---|---|---|---|---|
| Moran's $I$ | -1 | 0 | 1 | 1.27 | 1.09 | 0.66 | 0.53 | 0.79 |
| Geary's $c$ | 0 | 1 | 2 | 0.09 | 0.09 | 0.33 | 0.19 | 0.12 |
| LPCA c. ratio | 0[1] | $> 1$ [1] | | 0.20 | 0.13 | 0.58 | 0.67 | 0.26 |
| **X-driven** | **+** | **0** | **-** | DB 5 | DB 6 | DB 7 | DB 8 | DB 9 |
| Moran's $I$ | -1 | 0 | 1 | -0.22 | -0.12 | -0.15 | -0.06 | 0.15 |
| Geary's $c$ | 0 | 1 | 2 | 0.78 | 0.59 | 0.63 | 0.57 | 0.43 |
| LPCA c. ratio | 0[1] | $> 1$ [1] | | 2.54 | 2.10 | 1.86 | 1.90 | 0.82 |

structure is highly informative. This is the opposite for Geary and LPCA, as small values correspond to large autocorrelation.

Nevertheless, from Table 5 and Figure 7, the best overall performing methods combining node features and graph structure are (excluding the methods based on the graph alone, BoP-A and CTK-A), SVM-BoPM-AX (SVM with bag-of-paths modularity, see Subsection 4.1.4) and ASVM-AX (SVM based on autocovariates, see Subsection 4.2.1). They are not statistically different from SVM-M-AX, SVM-L-AX and SVM-DK-AX, but their mean rank is slightly higher.

Notice also that, from Figure 7, if we look at the performances obtained by a baseline linear SVM based on node features only (SVM-X), we clearly observe that integrating the information extracted from the graph structure significantly improves the results. Therefore, it seems to be be a good idea to consider collecting link information which could improve the predictions.

#### *5.4.2. Exploiting either the structure of the graph or the node features alone*

Obviously, as already mentioned, databases DB5 to DB9 are graph-driven, which explains the good performances of the sum-of-similarities CTK-A and BoP-A on these data. For these datasets, the features on the nodes do not help much for predicting the class label, as observed when looking to Figure 9 where results are displayed only on these datasets. It also explains the behavior of method SVM-M-AX on database DB5, among others.

In this case, the best performing methods are the sum-of-similarities CTK-A and the bag-of-paths betweenness BoP-A (see Subsection 4.3). This is clearly confirmed by displaying the results of the methods based on the graph structure only in Figure 10 and the results obtained on the graph-driven datasets in Figure 9. Interestingly, in this setting, these two methods ignoring the node features (CTK-A and BoP-A) are outperforming the SVM-based methods.

Conversely, on the node features-driven datasets (DB1 to DB4 and DB10; results displayed in Figure 8 and Table 5), all SVM methods based on node features and graph structure perform well while the methods based on the graph structure only obtain much worse results, as expected. In this setting, the two best performing methods are the same as for the overall results, that is, SVM-BoPM-AX (SVM with bag-of-paths modularity, see Subsection 4.1.4) and SVM-DK-AX (SVM based on a double kernel, see Subsection 4.2.2). However, these methods are not significantly better than a simple linear SVM based on features only (SVM-X), as shown in Figure 8.

From another point of view, Figure 11 takes into account all datasets and compares only the methods combining node features and graph structure. In this setting, the two best methods are also SVM-BoPM-AX, ASVM-AX and SVM-DK-AX which are now significantly better than the baseline SVM-X (less methods and more datasets are compared).

#### *5.4.3. Comparison of graph embedding methods*

Concerning the embedding methods described in Subsection 4.1, we can conclude that Geary's index (SVM-G-A and SVM-G-AX) should be avoided by preferring the bag-of-paths modularity (SVM-BoP-AX), Moran's index (SVM-M-AX) or graph Local Principal Component Analysis (SVM-L-AX). This is clearly observable when displaying only the results of the methods combining node features and graph structure in Figure 12. This result is confirmed when comparing only the methods based on graph embedding in Figure 10.

#### *5.4.4. Main findings*

To summarize, the experiments lead to the following conclusions:

- The best performing methods are highly dependent on the dataset. We observed (see Table 6) that, quite naturally, some datasets are more graph-driven in the sense that the network structure conveys important information for predicting the class labels, while other datasets are node features-driven and, in this case, the graph structure does not help much. However, it is always a good idea to consider information about the graph structure as this additional information can improve significantly the results (see Figures 11 and 12).

- If we consider the graph structure alone, the two best investigated methods are the sum-of-similarities CTK-A and the bag-of-paths betweenness BoP-A (see Subsection 4.3). They clearly outperform the graph embedding methods, but also the SVMs on some datasets.

- When informative features on nodes are available, it is always a good idea to combine the information, and we found that the best performing methods are SVM-BoPM-AX (SVM with bag-of-paths modularity, see Subsection 4.1.4), ASVM-AX (SVM based on autocovariates, see Subsection 4.2.1) and SVM-DK-AX (SVM based on a double kernel, see Subsection 4.2.2) (see Figure 11). Taking the graph structure into account clearly improves the results over a baseline SVM based on node features only.

[1] LPCA continuity ration is positive and lower-bounded by $1-\sqrt{\lambda_{max}}$ (which tends to be close to zero) where $\lambda_{max}$ is the largest eigenvalue of $\mathbf{A}$, the upper bound is unknown [49].

## 6. Conclusion

This work considered a data structure made of a graph and plain features on nodes of the graph. 14 semi-supervised classification methods were investigated to compare the feature-based approach, the graph structure-based approach, and the dual approach combining both information sources. It appears that the best performance are often obtained either by a SVM method (the considered baseline classifier) based on plain node features combined to a given number of new features derived from the graph structure (namely from the BoP modularity or autocovariates), or by the sum-of-similarities and the bag-of-paths modularity method, based on the graph structure only, which perform well on some datasets for which the graph structure carries important class information.

Indeed, we observed empirically that some datasets can be better explained by their graph structure (graph-driven datasets), or by their node features (features-driven datasets). Consequently, neither the graph-derived features alone or the plain features alone is sufficient to obtain optimal performances. In other words, standard feature-based classification results can be improved significantly by integrating information from the graph structure. In particular, the most effective methods were based on bag-of-paths modularity (SVM-BoPM-AX), autocovariates (ASVM-AX) or a double kernel (SVM-DK-AX).

The take-away message can be summarize as follow: if the dataset is graph-driven, a simple sum-of-similarities or a bag-of-paths betweenness are sufficient but it is not the case if the features on the nodes are more informative. In both cases SVM-BoPM-AX, ASVM-AX, SVM-DK-AX still ensure good overall performances, as shown on the investigated datasets.

A key point is therefore to determine *a-priori* if a given dataset is graph-driven or features-driven. In this paper we proposed spatial autocorrelation indexes to tackle this issue. Further investigations will be carried in that direction. In particular, how can we automatically infer properties of a new dataset (graph-driven or features-driven) if all class labels are not known (during cross-validation for example)?

Finally, the present work does not take into account the scalability and the complexity point of view. This analysis is left for further work.

## Acknowledgement

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

[1] R. M. Cooke, Experts in uncertainty, Oxford University Press, 1991.
[2] R. A. Jacobs, Methods for combining experts' probability assessments, Neural Computation 7 (1995) 867–888.
[3] D. Chen, X. Cheng, An asymptotic analysis of some expert fusion methods, Pattern Recognition Letters 22 (2001) 901–904.
[4] J. Kittler, F. M. Alkoot, Sum versus vote fusion in multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1) (2003) 110–115.
[5] F. Lad, Operational subjective statistical methods, John Wiley & Sons, 1996.
[6] G. J. Klir, T. A. Folger, Fuzzy sets, uncertainty, and information, Prentice-Hall, 1988.
[7] D. Dubois, M. Grabisch, H. Prade, P. Smets, Assessing the value of a candidate: Comparing belief function and possibility theories, Proceedings of the Fifteenth international conference on Uncertainty in Artificial Intelligence (1999) 170–177.
[8] C. Merz, Using correspondence analysis to combine classifiers, Machine Learning 36 (1999) 226–239.
[9] W. B. Levy, H. Delic, Maximum entropy aggregation of individual opinions, IEEE Transactions on Systems, Man and Cybernetics 24 (4) (1994) 606–613.
[10] I. J. Myung, S. Ramamoorti, J. Andrew D. Bailey, Maximum entropy aggregation of expert predictions, Management Science 42 (10) (1996) 1420–1436.
[11] F. Fouss, M. Saerens, Yet another method for combining classifiers outputs: A maximum entropy approach, Proceedings of the 5th International Workshop on Multiple Classifier Systems (MCS 2004), Lecture Notes in Computer Science, Vol. 3077, Springer-Verlag (2004) 82–91.
[12] X. Zhu, A. Goldberg, Introduction to semi-supervised learning, Morgan & Claypool Publishers, 2009.
[13] O. Chapelle, B. Scholkopf, A. Zien (editors), Semi-supervised learning, MIT Press, 2006.
[14] S. Hill, F. Provost, C. Volinsky, Network-based marketing: Identifying likely adopters via consumer networks, Statistical Science 21 (2) (2006) 256–276.
[15] S. A. Macskassy, F. Provost, Classification in networked data: a toolkit and a univariate case study, Journal of Machine Learning Research 8 (2007) 935–983.
[16] J.-J. Daudin, F. Picard, S. Robin, A mixture model for random graphs., Statistics and Computing 18 (2) (2008) 173–183.
[17] T. Joachims, Transductive inference for text classification using support vector machines, International Conference on Machine Learning (ICML) (1999) 200–209.
[18] X. Zhu, Semi-supervised learning literature survey, unpublished manuscript (available at http://pages.cs.wisc.edu/ jerryzhu/research/ssl/semireview.html) (2008).
[19] F. Fouss, M. Saerens, M. Shimbo, Algorithms and models for network data and link analysis, Cambridge University Press, 2016.
[20] F. R. Chung, Spectral graph theory, American Mathematical Society, 1997.
[21] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from examples, Journal of Machine Learning Research 7 (2006) 2399–2434.
[22] X. He, Laplacian regularized d-optimal design for active learning and its application to image retrieval, IEEE Transactions on Image Processing 19 (1) (2010) 254–263.
[23] S. Chakrabarti, B. Dom, P. Indyk, Enhanced hypertext categorization using hyperlinks, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD 1998), 1998, pp. 307–318.
[24] C. Bishop, Neural networks for pattern recognition, Oxford University Press, 1995.
[25] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, 2th ed., Springer, 2009.
[26] S. Theodoridis, K. Koutroumbas, Pattern recognition, 2nd ed., Academic Press, 2003.
[27] B. Lebichot, I. Kivimaki, K. Françoisse, M. Saerens, Semi-supervised classification through the bag-of-paths group betweenness, IEEE Transactions on Neural Networks and Learning Systems 25 (2014) 1173–1186.
[28] S. Abney, Semisupervised learning for computational linguistics, Chapman and Hall/CRC, 2008.
[29] T. Hofmann, B. Schölkopf, A. J. Smola, Kernel methods in machine learning, The Annals of Statistics 36 (3) (2088) 1171–1220.
[30] T. Silva, L. Zhao, Machine learning in complex networks, Springer, 2016.
[31] A. Subramanya, P. Pratim Talukdar, Graph-based semi-supervised learning, Morgan & Claypool Publishers, 2014.
[32] X. Zhu, A. Goldberg, Introduction to semi-supervised learning, Morgan & Claypool Publishers, 2009.
[33] D. Borcard, P. Legendre, All-scale spatial analysis of ecological data by means of principal coordinates of neighbour matrices, Ecological Mod-

elling 153 (1-2) (2002) 51 – 68.
[34] S. Dray, P. Legendre, P. Peres-Neto, Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices, Ecological Modelling 196 (3-4) (2006) 483 – 493.
[35] A. Meot, D. Chessel, R. Sabatier, Operateurs de voisinage et analyse des donnees spatio-temporelles (in french), in: D. Lebreton, B. Asselain (Eds.), Biometrie et environnement, Masson, 1993, pp. 45–72.
[36] L. Tang, 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.
[37] L. Tang, H. Liu, 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.
[38] L. Tang, H. Liu, Toward predicting collective behavior via social dimension extraction, IEEE Intelligent Systems 25 (4) (2010) 19–25.
[39] D. Zhang, R. Mao, A new kernel for classification of networked entities, in: Proceedings of 6th International Workshop on Mining and Learning with Graphs, Helsinki, Finland, 2008.
[40] D. Zhang, R. Mao, Classifying networked entities with modularity kernels, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008), ACM, 2008, pp. 113–122.
[41] R. Haining, Spatial data analysis, Cambridge University Press, 2003.
[42] D. Pfeiffer, T. Robinson, M. Stevenson, K. Stevens, D. Rogers, A. Clements, Spatial analysis in epidemiology, Oxford University Press, 2008.
[43] T. Waldhor, Moran's spatial autocorrelation coefficient, in: Encyclopedia of Statistical Sciences, 2nd ed. (S. Kotz, N. Balakrishnana, C. Read, B. Vidakovic and N. Johnson, editors), Vol. 12, Wiley, 2006, pp. 7875–7878.
[44] L. Waller, C. Gotway, Applied spatial statistics for public health data, Wiley, 2004.
[45] K. V. Mardia, Some properties of classical multidimensional scaling, Communications in Statistics - Theory and Methods 7 (13) (1978) 1233–1241.
[46] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.
[47] M. Newman, Networks: an introduction, Oxford University Press, 2010.
[48] H. Benali, B. Escofier, Analyse factorielle lissee et analyse des differences locales, Revue de Statistique Appliquee 38 (2) (1990) 55–76.
[49] L. Lebart, Contiguity analysis and classification, in: W. Gaul, O. Opitz, M. Schader (Eds.), Data Analysis, Studies in classification, data analysis, and knowledge organization, Springer, 2000, pp. 233–243.
[50] R. Devooght, A. Mantrach, I. Kivimaki, H. Bersini, A. Jaimes, M. Saerens, Random walks based modularity: Application to semi-supervised learning, Proceedings of the 23rd International Conference on World Wide Web (2014) 213–224.
[51] J. E. Besag, Nearest-neighbour systems and the auto-logistic model for binary data, Journal of the Royal Statistical Society. Series B (Methodological) 34 (1) (1972) 75–83.
[52] N. H. Augustin, M. A. Mugglestone, S. T. Buckland, An autologistic model for the spatial distribution of wildlife, Journal of Applied Ecology 33 (2) (1996) pp. 339–347.
[53] N. H. Augustin, M. A. Mugglestone, S. T. Buckland, The role of simulation in modelling spatially correlated data, Environmetrics 9 (2) (1998) 175–196.
[54] Q. Lu, L. Getoor, Link-based classification, in: Proceedings of the 20th International Conference on Machine Learning (ICML 2003), 2001, pp. 496–503.
[55] Y. Pawitan, In all likelihood: statistical modelling and inference using likelihood, Oxford University Press, 2001.
[56] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the em algorithm (with discussion), Journal of the Royal Statistical Society B 39 (1) (1977) 1–38.
[57] G. McLachlan, T. Krishnan, The EM algorithm and extensions, 2nd ed, Wiley, 2008.
[58] V. Roth, Probabilistic discriminative kernel classifiers for multi-class problems, in: B. Radig, S. Florczyk (Eds.), Pattern Recognition: Proceedings of the 23rd DAGM Symposium, Vol. 2191 of Lecture Notes in Computer Science, Springer, 2001, pp. 246–253.
[59] B. Scholkopf, A. Smola, Learning with kernels, The MIT Press, 2002.
[60] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis, Cambridge University Press, 2004.
[61] F. Fouss, K. Francoisse, L. Yen, A. Pirotte, M. Saerens, An experimental investigation of kernels on a graph on collaborative recommendation and semisupervised classification, Neural Networks 31 (2012) 53–72.
[62] T. Gartner, Kernels for structured data, World Scientific Publishing, 2008.
[63] J. LeSage, R. K. Pace, Introduction to spatial econometrics, Chapman & Hall, 2009.
[64] A. Mantrach, N. van Zeebroeck, P. Francq, M. Shimbo, H. Bersini, M. Saerens, Semi-supervised classification and betweenness computation on large, sparse, directed graphs, Pattern Recognition 44 (6) (2011) 1212 – 1224.
[65] D. Zhou, O. Bousquet, T. Lal, J. Weston, B. Scholkopf, Learning with local and global consistency, in: Proceedings of the Neural Information Processing Systems Conference (NIPS 2003), 2003, pp. 237–244.
[66] K. Francoisse, I. Kivimaki, A. Mantrach, F. Rossi, M. Saerens, A bag-of-paths framework for network data analysis, To appear in: Neural Networks.
[67] S. Prithviraj, G. Galileo, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad, Collective classification in network data, AI Magazine 29 (3) (2008) 93–106.
[68] J. McAuley, J. Leskovec, Learning to discover social circles in ego networks, Advances in Neural Information Processing Systems (NIPS).
[69] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, Transaction on Neural Network 13 (2) (2002) 415–425.
[70] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research 9 (2008) 1871–1874.
[71] V. Vapnik, Statistical Learning Theory, Wiley, 1998.
[72] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.