Neurocomputing 326–327 (2019) 123–131

Contents lists available at ScienceDirect

Neurocomputing

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

Shaping graph pattern mining for financial risk

Bernardete Ribeiro[a,*], Ning Chen[b], Alexander Kovacec[c]

[a] CISUC - Department of Informatics Engineering, University of Coimbra, Pólo 2, Coimbra 3030–290, Portugal
[b] College of Computer Science and Technology (Software College), Henan Polytechnic University, 2001 Century Avenue, Jiaozuo, 454003 Henan, PR China
[c] Department of Mathematics, University of Coimbra, Praa Dom Dinis, Coimbra 3001–501, Portugal

ARTICLE INFO

Article history:
Received 2 June 2016
Revised 13 September 2016
Accepted 29 January 2017
Available online 13 September 2017

Keywords:
Graph mining
Classification
Financial risk

ABSTRACT

In recent years graph pattern mining took a prominent role in knowledge discovery in many scientific fields. From Web advertising to biology and finance, graph data is ubiquitous making pattern-based graph tools increasingly important. When it comes to financial settings, data is very complex and although many successful approaches have been proposed often they neglect the intertwined economic risk factors, which seriously affects the goodness of predictions. In this paper, we posit that financial risk analysis can be leveraged if structure can be taken into account by discovering financial motifs. We look at this problem from a graph-based perspective in two ways, by considering the structure in the inputs, the graphs themselves, and by taking into account the graph embedded structure of the data. In the first, we use gBoost combined with a substructure mining algorithm. In the second, we take a subspace learning graph embedded approach. In our experiments two datasets are used: a qualitative bankruptcy data benchmark and a real-world French database of corporate companies. Furthermore, we propose a graph construction algorithm to extract graph structure from feature vector data. Finally, we empirically show that in both graph-based approaches the financial motifs are crucial for the classification, thereby enhancing the prediction results.

1. Introduction

Nowadays, data is naturally structured in form of trees or graphs, which are structures that may convey important information. A graph is a general and powerful data representation formalism, which found widespread application in many scientific fields. Finding subgraphs capable of compressing data by abstracting instances of the substructures and identifying interesting patterns is thus crucial.

The awareness of big data together with the poor understanding of the processes that generate data has enforced techniques to extract frequent structural patterns from such data [27]. Graph mining techniques are sought for a class of problems lying on the crossroads of several research topics including graph theory, data sensing, data mining and data visualization.

Graphs are very important mathematical structures that can represent information in many real world domains such as chemistry, biology and, web and text processing. Examples are protein interactions, phylogenetic trees, and molecular graphs [5], computer networks [18], hypertextual and XML documents, social networks, mobile call networks, to name a few [32].

Pattern mining takes essentially two approaches: statistical learning and structural. In the statistical learning, patterns are represented by feature vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ of n measurements. It has two main drawbacks: first, the vectors uphold a predefined set of features, despite the size and complexity of the objects they represent; second, the binary relationships among (parts of) objects cannot be captured. The above pitfalls, size constraints and lack of ability to represent relationships, might prevent to expose better models. In the structural approach, patterns are represented by graphs that can overcome above limitations with their inherent structure. Yet the complexity increases, for instance, it takes exponential time for finding the isomorphism between two graphs while linear time is needed for the similarity of two features vectors [6].

In this paper, in the settings of financial risk analysis we take two approaches for graph-based pattern mining. In the first, subgraph mining is employed based on an isomorphism search between two graphs, while, in the second, the goal is to learn a low-dimensional subspace spanned by projected vectors, which are dominant for preserving the intrinsic data structure. With respect to the first approach, we propose a graph construction algorithm on the basis of a qualitative data set of financial statements to

* Corresponding author.
E-mail addresses: bribeiro@dei.uc.pt (B. Ribeiro), nchenyx@outlook.com (N. Chen), kovacec@mat.uc.pt (A. Kovacec).

http://dx.doi.org/10.1016/j.neucom.2017.01.119
0925-2312/

gain further insights on the data structure, and then a graph-based model for pattern mining is generated via gBoost [31], a frequent subgraph discovery technique on the grounds of mathematical programming and gSpan algorithm [36]. This pattern-growth method uses Depth-First Search (DFS) and is able to find financial motifs in the graph data rendering the risk estimation very successful. We empirically show that the performance evaluation is competitive to the statistical learning algorithms such as Artificial Neural Networks (ANN), Decision Trees (DT) and Support Vector Machines (SVM) when unstructured dimensional feature vectors are used. Our case study encompasses a graph-based methodology that enables to unravel structural subtleties otherwise hidden in the data. The last aspect is related to the second approach, which by taking an embedded graph learning technique we successfully take into account the data structure. Thus, we look at how good the models are either by using structural components, such as graphs, as inputs – the graph mining approach – or by using structure embedded on the data – the embedded graph learning approach.

In the next section we will review the related work in financial credit risk and graph-based pattern mining. In Section 3 we present the background for the gBoost classifier, propose an algorithm for graph construction and describe the graph-based embedded learning approach based on spectral data matrix decomposition. In the context of financial credit risk, we present in Section 4 the experiments considering two financial datasets, a qualitative bankruptcy benchmark and a real-world data set of French financial ratios. We describe the research design for both graph-based approaches, pointing out their properties for the financial settings and discuss the results. The paper will end with the conclusions and future work in Section 5.

2. Related work

2.1. Financial credit risk assessment

The financial credit risk indicates the risk associated with financing, in other words, a borrower cannot pay the lenders, or goes into default. Accordingly, financial credit risk assessment aims to predict the probability of default of loans, and the likelihood of a firm's going bankrupt. In our paper our efforts are directed to the latter. The problem can be stated as follows: given a number of companies labeled as bankrupt/healthy, and a set of financial variables that describe the situation of a company over a given period, predict the probability that the company may belong to a high risk group or become bankrupt during the following years. Over the last years, some articles reviewed the literature of financial crisis prediction [8,14] or focused on advanced techniques [2,11].

When dealing with real world financial credit risk problems, they are usually characterized by large scale of data and high-dimensional representation. The key financial ratios comprise financial information (operational performance, financial liquidity, risk return, sustainable growth etc.) and non-financial information (government policy, economic environment marking reports, customers screening etc.) [30]. These performance key indicators are well fit to establish the relationships between nodes of financial companies. Through a linear or nonlinear projection, dimensional reduction and subspace learning demonstrated to be very effective to find a compact representation in a low dimensional subspace of high dimensional data [29].

In the literature, a wide range of methods have been proposed for financial risk assessment [21]. These methods can be divided into parametric methods, semi-parametric methods, and non-parametric methods from the viewpoint of model specification. Statistical methods are typically parametric and have been widely studied in literature for financial risk assessment even with some limitations. A logit model [37] is developed in the context of Belgian small and medium-sized enterprises, and achieved a satisfactory accuracy of bankruptcy prediction. Most intelligent methods are non-parametric that include Artificial Neural Networks (ANNs), Fuzzy Set Theory (FST), Decision Trees (DTs), Case-Based Reasoning (CBR), Support Vector Machines (SVMs), Rough Set Theory (RST) among others. Semi-parametric methods [9,17] define the modeled process with flexible structure, and have proved to be very successful when the fully parametric and non-parametric do not perform well. Semi-parametric methods have more flexibility in model structure although the modeled process is clearly interpreted. Recently semi-parametric methods have become a future trend of bankruptcy prediction study. Some research results have demonstrated the well-defined semi-parametric methods are possible to improve the prediction accuracy of bankruptcy compared to the conventional parametric and non-parametric methods [3,13,25].

Hybrid techniques aim to improve an individual learner using some heuristics by refining the related instances, significant features, or optimal parameters [24]. Ensemble techniques construct a composite classifier that takes advantage of several learners which have high performance individually and low intercorrelation [33].

2.2. Graph-based pattern mining

Although many successful approaches have been used rarely the structural component has been endorsed in the literature review. It becomes important to provide structural performance data mining techniques in financial domain where a large-scale complex data is produced today. Graph-based pattern mining intend to discover the hidden structures represented by graphs. The main advantage of graph is that not only the nodes (instances) but also the edges (relation) contribute to the representation of data [34].

Graph-based pattern mining took a new breed of approaches since the introduction of frequent pattern mining in [1]. In particular, many subgraph mining algorithms have been developed such as Apriori based methods like AGM [19], FSG [23], or pattern-growth methods like gSpan [36] and Gaston [28]. A major challenge in subgraph mining is the subgraph isomorphism, which is an NP-complete problem [36]. In gSpan, Depth-First Search (DFS) is employed to reduce the search space significantly making possible to check whether between two graphs an isomorphism exists. Its purpose is to enumerate all connected frequent subgraphs from graph representation of patterns. gBoost [31] is an extension of boosting for graphs which uses gSpan. Apart from the mathematical graph theory based approaches, a few other can be considered for graph mining such as: greedy search-based approaches [16], inductive programming logic [4] and inductive database approaches [22].

3. Graph learning

3.1. Graph preliminaries

In this section the basic notation and graph concepts are introduced.

Definition 3.1. Graph. A graph g is defined as a pair of sets (V_g, E_g), where $V_g = \{v_1, v_2, \ldots, v_n\}$ is a set of ordered vertices and $E_g = \{(v_i, v_j), \ldots, (v_k, v_l)\}$ is a set of pairs of vertices, the edges.

Definition 3.2. Graph isomorphism. Two graphs g_1 and g_2 are isomorphic if there is a bijective mapping such that every edge in E_1 is mapped to a single edge in E_2 and vice-versa.

Definition 3.3. Subgraph. A subgraph $g_2 = (V_2, E_2)$ of a graph g_1 is a graph for which $V_2 \subseteq V_1, E_2 = E_1 \cap (V_2 \times V_1)$.

Definition 3.4. Subgraph isomorphism. Given two graphs g_1 and g_2 the problem of subgraph isomorphism is to find a isomorphism between g_2 and a subgraph of g_1 that is to determine if g_2 is included in g_1.

Definition 3.5. Support of a subgraph g. Given a labeled dataset $G_D = \{g_1, g_2, \ldots, g_n\}$, support or frequency of a subgraph g is the percentage (or number of graphs) in G_D where g is a subgraph.

Definition 3.6. Frequent subgraph. A frequent subgraph is a graph whose support is not less than a minimum threshold.

3.2. gBoost classifier

gBoost [31] is an extension of boosting for graphs and comprises a mathematical programming tool [12] that progressively collects "informative" frequent patterns to use as features for classification and regression. Furthermore, this tool uses linear program (LP) approaches to boosting providing an efficient solution using LPBoost, a column generation based simplex method [12]. The problem is formulated as if all possible weak hypotheses had already been generated, where the labels produced by the weak hypotheses become the new feature space of the problem. The boosting consists of constructing a learning function in the label space that minimizes misclassification error and maximizes the soft margin. It is also considered a frequent subgraph mining technique similar to gSpan in frequent subgraph mining [36]. gBoost uses first gSpan method [36] which finds frequent subgraphs and constructs a canonical search space in the form of a Depth-First Search (DFS). This algorithm is used for traversing or searching tree or graph data structures. With the proviso that the tree structure and the DFS code are available an optimal search can be constructed.

Let $\{g_t\}_{t=1}^{T}$ denote a set of frequent subgraphs generated from gSpan. Given the learning graphs $\{(G_n, y_n)\}_{n=1}^{N}$ where G_n is a training graph and $y_n \in \{+1, -1\}$ is the associated class label. Let $\mathcal{T}$ the set of all patterns (subgraphs) included in at least one training graph. Each graph G_n can be encoded as a $|\mathcal{T}|$ dimensional vector $\mathbf{x}_n$ through an indicator function $\mathcal{I}(\cdot)$ as indicated below:

$$\mathbf{x}_{n,t} = \mathcal{I}(t \subseteq G_n) \ \forall t \in \mathcal{T} \tag{1}$$

The hypotheses or individual stumps are defined as:

$$h(\mathbf{x}_n, g_t) = \begin{cases} +1 & \text{if } g_t \in \mathbf{x}_n \\ -1 & \text{if } g_t \notin \mathbf{x}_n \end{cases} \tag{2}$$

where we simplified the notation for $\mathbf{x}_n$. Given training data $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ directly solving the optimization problem is intractable. Therefore, the equivalent dual problem below is solved instead which can be expressed as follows:

$$\begin{aligned} \underset{\alpha,\gamma}{\text{minimize}} \quad & \gamma \\ \text{subject to} \quad & \sum_{n=1}^{N} \lambda_n y_n h(\mathbf{x}_n, g_t) \le \gamma \quad s = 1, 2, \ldots, T \\ & \sum_{n=1}^{N} \lambda_n = 1, \ 0 \le \lambda_n \le \Delta \quad s = 1, 2, \ldots, T, \end{aligned} \tag{3}$$

where $\Delta = \frac{1}{\nu N}$, $\nu \in (0, 1)$ is the cost classification parameter controlling the misclassification errors [12,31] which has to be found using model selection techniques such as cross-validation. After solving the dual optimization problem, the primal solution $\boldsymbol{\alpha}$ is obtained from the Lagrange multipliers. It has a limited number of variables and an intractably number of constraints. Therefore, gBoost algorithm uses a methodology based on the column generation [26]. The algorithm sets up a maximum number of columns to add at each iteration. Then rather than considering all the constraints, the subgraph g_s whose corresponding constraint is violated the most is selected. At the k iteration the constraints are formulated as

$$\sum_{n=1}^{N} \lambda_n^{(k)} y_n h(\mathbf{x}_n, g_t) \le \gamma^{(k)}, \ t \in \mathcal{T}^{(k)} \tag{4}$$

As defined above $\mathcal{T}$ gathers the index number of the selected subgraphs. At the start of this procedure, $\mathcal{T}^{(0)}$ is set to empty and $\alpha_n(0) = \frac{1}{N}$. Following, the optimal solutions $\alpha_n^{(k)}$ and $\gamma^{(k)}$ for solving the restricted dual optimization problem are updated iteratively. In the sequel, the subgraph that violates the constraint the most (corresponding to the largest margin) is selected:

$$t^* = \arg\max_{t=1,2,\ldots,T} \sum_{n=1}^{N} \lambda_n^{(k)} y_n h(\mathbf{x}_n, g_t) \tag{5}$$

The set $\mathcal{T}^{(k)}$ is updated by adding the new index number t^*: $\mathcal{T}^{(k+1)} = \mathcal{T}^{(k)} \cup \{t^*\}$. The procedure iterates until the criteria based on the satisfaction of all constraints are met. For a specific test graph $\mathbf{x}$ the prediction rule is a convex combination of simple classification stumps $h(\mathbf{x}, g_t)$:

$$y = \text{sign}\left(\sum_{t \in \mathcal{T}^{(K)}} \alpha_t h(\mathbf{x}, g_t) \right) \tag{6}$$

A test graph is labeled in the positive class if $y = 1$ and in the negative class if $y = -1$.

3.3. Graph construction algorithm

The algorithm to build the graph data takes feature vectors from the data collection and constructs graphs to be used as inputs into the gBoost classifier. The main focus is to set up the nodes and edges for the data, running over all the data samples in the dataset. Depending on the problem the relationships between nodes should be taken into account for setting up the edges to link the nodes among graph 'points'. The proposed Algorithm 1 will be used with the benchmark qualitative data. We have coded the algorithm in Matlab for easiness of use with gBoost package.[1] With the Algorithm 1 we built the graphs to be used as inputs in gBoost. Thus each sample of the data is a graph with a set of nodes corresponding to the features in the feature dimensional space. The edges are assigned during graph construction and represent the relationships between nodes. The graph samples are connected, undirected and labeled graphs. The overall graph data samples were further partitioned to find the training and test graphs for further use in the gBoost classifier. More specifically, the algorithm cycles over the N rows of the feature dimensional vectors matrix data, assigns the nodes of each graph and updates the edges as shown in Algorithm 1.

3.4. Graph embedded learning

In this section we take rather into account the structure in the data by building the graph weight matrix. Given a graph G with n nodes, each node representing a data point, let W be a symmetric $n \times n$ matrix where W_{ij} is the connection weight between node i and j. Each node of the graph is represented as a low-dimensional vector and the similarities between pairs of data (in the original high-dimensional space) are preserved. The corresponding diagonal matrix and the Laplacian matrix [10] are defined as:

$$L = D - W, D_{ii} = \sum_{j \neq i} W_{ij} \quad \forall i \tag{7}$$

[1] The Mex Matlab wrappers from the Graph Boosting Toolbox for Matlab were downloaded from http://www.nowozin.net/sebastian/gboost/.

Algorithm 1: Graph construction from feature vector data.

Input: For each collection of data $\mathcal{D}$, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with labels $\{y_n \in \{+1, -1\} \quad n = 1, 2, \ldots N\}$
/*Cycle over N rows*/
for all $n \leftarrow 1, \ldots, N$ **do**
  Initializations
  /* Cycle over the first row */
  for all $j \leftarrow 1, \ldots, NumNodes - 1$ **do**
    Detects the transition of weights
    Update Edges
  end for
  Makes the connection with last element of index array
  /* Find Dangling Nodes in the Graph */
  if *Dangling Nodes exist* **then**
    Find the node closest Weight Distance
    Adjust Weight Connections
  end if
  /* Find disconnected Components in the graph */
  if *Subgraphs remain to be connected* **then**
    Connect subgraphs
  end if
end for
Output: Get connected learning graphs $\{(G_n, y_n)\}_{n=1}^{N}$ where G_n is a training graph and $y_n \in \{+1, -1\}$ is the associated class label

where D is a diagonal matrix whose entries are sums of columns (or rows) of the matrix W. Let $\mathbf{y} = [y_1 y_2 \ldots y_n]$ be the low-dimensional embedding of the nodes where the column y_i vector is the embedding for the vertex $\mathbf{x}_i$. Direct graph embedding [35] aims to maintain similarities among vertex pairs by following the graph preserving criterion (9):

$$\mathbf{y}^* = \arg \min_{\mathbf{y}^T D \mathbf{y} = 1} \sum_{i \neq j} ||y_i - y_j||^2 W_{ij} \tag{8}$$

$$= \arg \min_{\mathbf{y}^T D \mathbf{y} = 1} (\mathbf{y}^T L \mathbf{y}) = \arg \min \frac{\mathbf{y}^T L \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} \tag{9}$$

Using (7) the above optimization problem has the equivalent form:

$$\mathbf{y}^* = \arg \max \mathbf{y}^T W \mathbf{y} = \arg \max \frac{\mathbf{y}^T W \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} \tag{10}$$

Let $\mathbf{u}$ be the transformation vector and $\mathbf{y}_i = \mathbf{u}^T \mathbf{x}_i$. Linear Graph Embedding (LGE) finds the optimal $\mathbf{u}^*$ which are the eigenvectors corresponding to the maximum eigenvalues of the decomposition problem:

$$XWX^T \mathbf{u} = \lambda XDX^T \mathbf{u} \tag{11}$$

The Spatially Smooth Subspace learning (SSSL) [7] extends the LGE by using the graph structure with the weight matrix W and solves the following optimization problem:

$$XWX^T \mathbf{u} = \lambda((1 - \alpha)XD^T X + \alpha \Delta^T \Delta)\mathbf{u} \tag{12}$$

where Δ is a $m \times m$ matrix giving a discrete approximation for the Laplacian and α is the parameter that controls the smoothness of the approximation.

3.4.1. Building the affinity graph matrix

The affinity graph weight matrix W is built by assuming that each ith node corresponds to a given firm $\mathbf{x}_i$. It can be specified by means of weight schemes as follows:

1. Binary weighting. $W_{ij} = 1$ if and only if nodes i and j are connected by an edge, otherwise $W_{ij} = 0$.

Table 1
Attributes (Positive, Average, Negative) and class is (Bankrupt, Healthy).

	Financial indicators	Qualitative attributes
1.	Industrial Risk	{Positive,Average,Negative}
2.	Management Risk	{Positive,Average,Negative}
3.	Financial Flexibility	{Positive,Average,Negative}
4.	Credibility	{Positive,Average,Negative}
5.	Competitiveness	{Positive,Average,Negative}
6.	Operating Risk	{Positive,Average,Negative}
7.	Class	{Bankrupt,Healthy}

2. Heat kernel weighting (with σ the kernel width). The scheme for assigning weights between nodes i and j is:

$$W_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ share the same class;} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

3. Dot-product weighting.

$$W_{ij} = \begin{cases} \frac{\mathbf{x}_j^T \mathbf{x}_i}{||\mathbf{x}_i|| \cdot ||\mathbf{x}_j||}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ share the same class;} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

4. Research design

4.1. Datasets

Qualitatitive bankruptcy. The QB dataset[2] attributes and samples are described in [20]. The sample size is 250 and has 6 attributes each corresponding to qualitative parameters in bankruptcy: Industrial Risk, Management Risk, Financial Flexibility, Credibility, Competitiveness, and Operating Risk. The attribute value is nominal (Positive, Average, Negative) and there are two classes (Bankrupt, Healthy) as described in Table 1. The dataset is unbalanced consisting of 143 samples in the Healthy class and 107 samples in Bankrupt class. We assigned Bankrupt to the positive class and Healthy to the negative class. In short, sample #1 = (Positive, Positive, Average, Average, Average, Positive) is assigned to class Bankrupt while sample #250 = (Positive, Negative, Negative, Negative, Average, Average) is assigned to the other. After running the Algorithm 1 we built the training data $train_G$ with 143 graphs for training and the test data $test_G$ with 107 graphs for test with identical distribution of positive and negative samples as in the whole original dataset. For easiness of handling the data we decided to assign a weight corresponding to each qualitative value (for example, we assigned 2 to Positive, 1 to Average and 3 to Negative). We assigned the label $(+1, -1)$ to the positive (Bankrupt) class and negative (Healthy) class, respectively, for use in the gBoost algorithm. According to the train and test partitions mentioned above, we built the vectors $train_Y$ and $test_Y$ containing the graph labels.

In Fig. 1 examples of data samples found in the qualitative data are represented for better illustration of the financial motifs built by the graph construction algorithm. These motifs are 6-node graphs (each node is an attribute of the qualitative data illustrated in Table 1) that play a decision role on the overall classification procedure influencing the classifier prediction.

DIANE database. The database is composed of 107,389 French companies and their foreign subsidiaries spanned over the years from 2002 to 2006. It contains complete information about the financial ratios including financial strength, liquidity, solvability, productivity of labor and capital, margins, net profitability and return on investment (the financial ratios are described in Table 2. In In the

[2] The Qualitative Bankruptcy (QB) dataset can be download from https://archive.ics.uci.edu/ml/datasets/Qualitative_Bankruptcy.

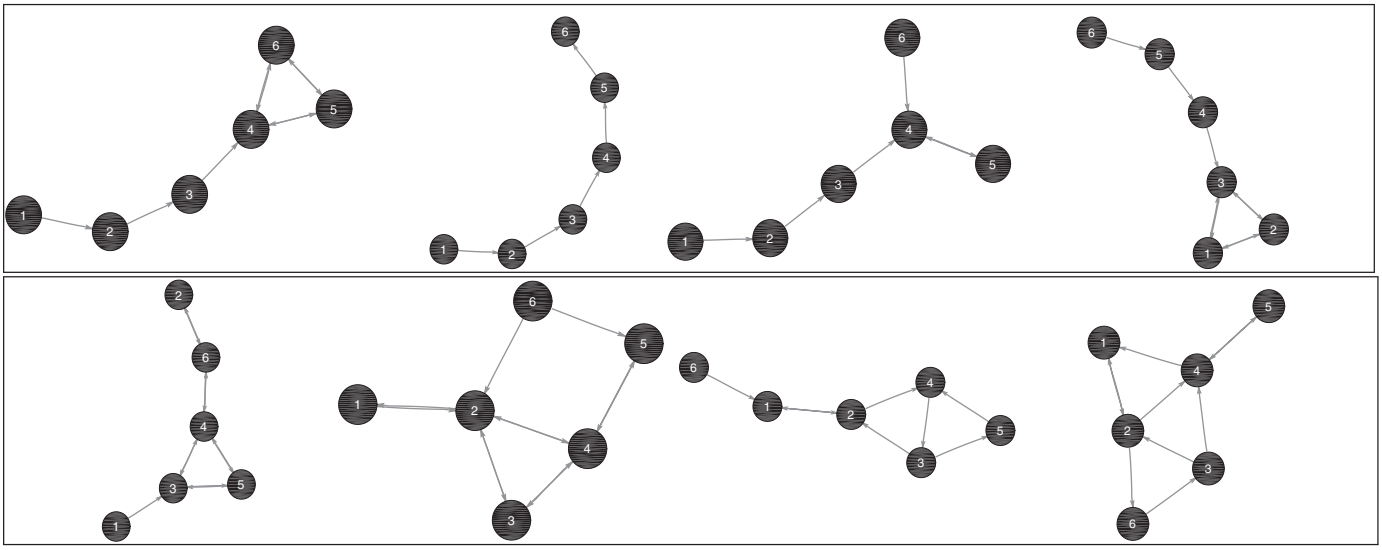

Fig. 1. Motifs for 6-node graphs financial samples.

Table 2
Financial ratios of DIANE database.

x_1-	Number of Employees	x_{16}-	Cashflow/Turnover
x_2-	Capital Employed/Fixed Assets	x_{17}-	Working Capital/Turnover days
x_3-	Financial Debt/Capital Employed	x_{18}-	Net Current Assets/Turnover days
x_4-	Depreciation of Tangible Assets	x_{19}-	Working Capital Needs/Turnover
x_5-	Working Capital/Current Assets	x_{20}-	Export
x_6-	Current ratio	x_{21}-	Added Value/Employee k EUR)
x_7-	Liquidity ratio	x_{22}-	Total Assets Turnover
x_8-	Stock Turnover days	x_{23}-	Operating Profit Margin
x_9-	Collection Period days	x_{24}-	Net Profit Margin
x_{10}-	Credit Period days	x_{25}-	Added Value Margin
x_{11}-	Turnover/Employee k EUR	x_{26}-	Part of Employees
x_{12}-	Interest/Turnover	x_{27}-	Return on Capital Employed
x_{13}-	Debt Period days	x_{28}-	Return on Total Assets
x_{14}-	Financial Debt/Equity	x_{29}-	EBIT Margin
x_{15}-	Financial Debt/Cashflow	x_{30} -	EBITA Margin

original database, 973 companies are labeled as distressed in 2007 and the others are labeled as healthy. Due to the large number of missing values existed in the companies (particularly in bankrupt companies), we select 600 companies with at most 10 missing values from the bankrupt group. It was known that the classification tends to favor the majority class (non-default companies) under the highly skewed distribution of the original database. We then sampled randomly 600 non-default companies in order to generate a balanced data set for experiments. The outcome is a balanced data set made up of 1200 French companies, 600 examples distressed in 2007, and the remainder are healthy. We consider the financial ratios from both one year and up to three fiscal years before bankruptcy for constructing the prediction models. Fig. 2 illustrates the graph financial motifs corresponding to the bankrupt (and healthy) companies constructed with the heat kernel ($\sigma = 0.5$) and (p-neighbors =5) in the supervised mode, i.e., with class label information.

4.2. Evaluation metrics

In order to evaluate a binary decision task we first define a contingency matrix representing the possible outcomes of the classification. Evaluation measures are defined based on the contingency Table 3, such as, error rate ($\frac{fp+fn}{tp+fp+tn+fn}$), and accuracy $\frac{tp+tn}{tp+fn+fp+tn}$ which measures the overall effectiveness of a classifier and AUC (Area Under the Curve)$\frac{1}{2}\left(\frac{tp}{tp+fn}+\frac{tn}{tn+fp}\right)$ which captures the classifier's capability to avoid false classification.

Table 3
Contingency matrix of prediction results.

Real class	Predicted class	
	Positive	Negative
Positive	*tp*	*fn*
Negative	*fp*	*tn*

Positive: bad credit or bankrupt, Negative: good credit or healthy.

4.3. Empirical analysis

4.3.1. Qualitative bankruptcy data: gBoost classifier

In this section for the sake of comparison with gBoost we present several algorithms spanning over machine learning and data mining methods using the open source WEKA Toolbox.[3]

Support Vector Machines (SVM) belong to the maximum margin classifiers aiming to find an optimal separating hyperplane, which maximizes the margin between two classes of data in kernel, induced feature space. SVM use the structural risk minimization principle to avoid overfitting. Since the introduction to the area of financial risk analysis, SVM have gained wide popularity owing to the good generalization on a small amount of high-dimensional data. Apart from SVM, we also used Neural Networks,

[3] http://www.cs.waikato.ac.nz/ml/weka/ [15].

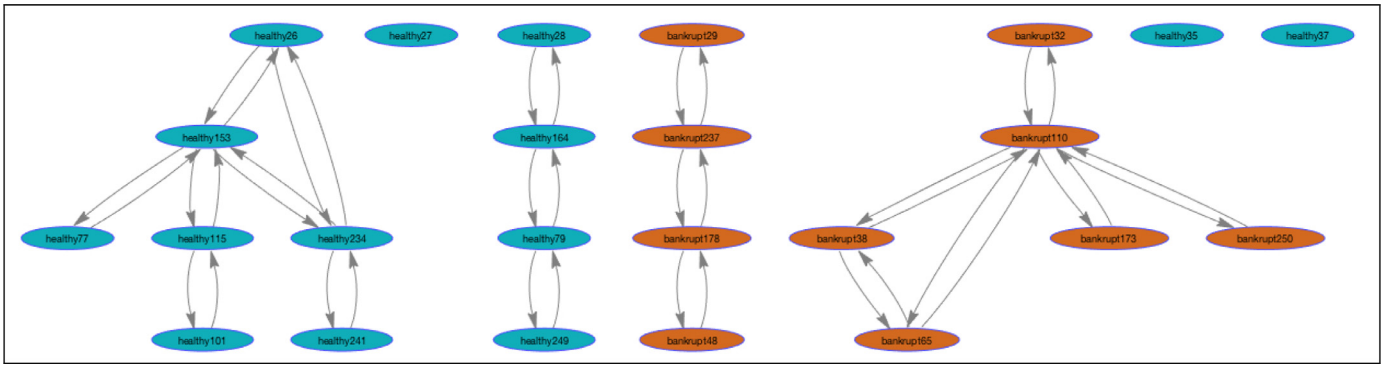

Fig. 2. Motifs for a real world data set Diane of French companies.

Table 4
Classifier results: performance measures (%).

QB data	(1)	(2)	(3)	(4)	(5)
Acc	96.52 ± 10.25	94.88 ± 3.29	97.72 ± 2.77	98.08 ± 1.85	88.48 ± 1.95
Precision	93.28 ± 23.86	95.72 ± 5.32	100.00 ± 0.00	98.93 ± 2.29	97.10 ± 1.68
Recall	92.68 ± 23.81	92.52 ± 5.43	94.68 ± 6.44	96.61 ± 4.58	97.10 ± 1.68
F1	92.95 ± 23.77	93.92 ± 3.79	97.16 ± 3.49	97.67 ± .2.31	97.10 ± 1.68
AUC	97.04 ± 13.75	94.58 ± 3.35	99.62 ± 0.27	98.62 ± 2.32	98.53 ± 0.84
Spec.	99.44 ± 1.30	96.64 ± 4.47	100.00 ± 0.00	99.17 ± 1.79	89.20 ± 3.10
QB data	(6)	(7)	(8)	(9)	(10)
Acc	98.12 ± 1.91	**98.13 ± 0.05**	97.72 ± 2.67	86.92 ± 4.16	97.68 ± 3.01
Precision	98.58 ± 2.58	97.53 ± 1.14	**100.00 ± 0.00**	77.15 ± 5.20	100.00 ± 0.00
Recall	97.87 ± 4.70	98.33 ± 0.01	94.67 ± 6.25	**99.44 ± 2.41**	94.59 ± 6.97
F1	97.73 ± 2.38	**97.93 ± 0.57**	97.15 ± 3.47	86.80 ± 3.70	97.08 ± 3.90
AUC	97.98 ± 2.00	**99.81 ± 0.05**	97.33 ± 3.12	88.48 ± 3.84	97.30 ± 3.49
Spec.	98.89 ± 2.04	96.81 ± 1.50	**100.00 ± 0.00**	77.52 ± 6.75	100.00 ± 0.00

Tested Classifiers (Weka) (1) Multilayer Perceptron (2) SMO supportVector.PolyKernel (3) RBF Network (4) Random Committee (5) Simple Fuzzy Grid (6) SimpleCART (7) gBoost (8) LibSVM RBF Kernel (9) LibSVM Sigmoid Kernel (10) SMO supportVector.RBFKernel

Decision Trees, fuzzy grid, and random committee for comparison. SimpleCART constructs a decision tree well adapted to the training data and implementing minimal cost-complexity pruning to the tree structure to avoid over-fitting. Multi-Level Perceptron (MLP) and Radial Basis Function (RBF) network are artificial neural networks for machine learning. The former is a multi-layer, feed-forward neural network, trained iteratively to adjust the connection weights via back-propagation algorithm. The latter has only one hidden layer, each node of which implements a normalized Gaussian radial basis function with the center and width as parameters. Fuzzy grid method partitions the input and output data into grids and extracts the fuzzy rules for data classification. Random committee builds an ensemble of randomized base decision tree classifiers to improve the classification accuracy. In Table 4 the performance measures of 30 runs with 5-fold cross validation of machine learning methods including SVM, neural networks (Multi-Layer Perceptron (MLP) and Radial Basis Functions (RBF)), fuzzy grid, and random committee are illustrated. The performance metrics used to compare the algorithms are often used in machine learning and easily deduced from the confusion matrix illustrated in Table 3. We also added the specificity or the true negative rate which gives an understanding of how good the model is in complying with the negative examples.

The gBoost classifier results are in bold-type when it outperforms the other algorithms in the performance measure of the respective row in Table 4. The experimental results in WEKA show that all the variants of SVM are statistically significant with paired t test and 0.05 confidence in terms of Accuracy, F1, and AUC compared with SimpleFuzzyGrid, MLP, RBF, Simple CART for the QB dataset. As it is shown, gBoost shows improvement over SVM in the mean of Accuracy, F1 and AUC. Overall gBoost is better than neural networks and rule-based algorithms while showing competitive performance with random committee, decision trees and SVM with Gaussian kernel. The test AUC obtained with gBoost was found 99.81% while for the best SVM the test AUC was 97.33%.

In the prediction phase the algorithm takes a test graph $\mathbf{x}$ and outputs a classification result as indicated in Eq. (6) by the convex combination of simple classification stumps $h(\mathbf{x}, g_t)$. For gBoost, the maximum pattern size which in our case corresponds to the maximum number of nodes in a subgraph was constrained to 6, since this is the number of attributes defining the financial indicators. In order to characterize the influence of the regularization parameter ν, gBoost is applied to QB dataset with {0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4}. The results from running gBoost are illustrated in Fig. 3. The left plot graph (a) represents an array of classifier responses for $test_G$ where (i, j) element encodes the response $(+1, -1)$ of a subclassifier j on a sample i. In the right graph plot (b) the performance of gBoost is examined by varying ν parameter controlling training accuracy. This parameter is used in the graph optimization process LPBoost for finding frequent subgraphs [12]. When ν is low gBoost creates a complex classification rule so that it can classify the training patterns completely. As ν is decreased, the regularization works and the rule gets simpler thus controlling overfitting. The best testing result was obtained with $\nu = 0.15$ and $\epsilon = 0.001$.

The convergence tolerance ϵ used in the runs was varied from 0.001 to 0.1. The bars in the plot indicate the train and test

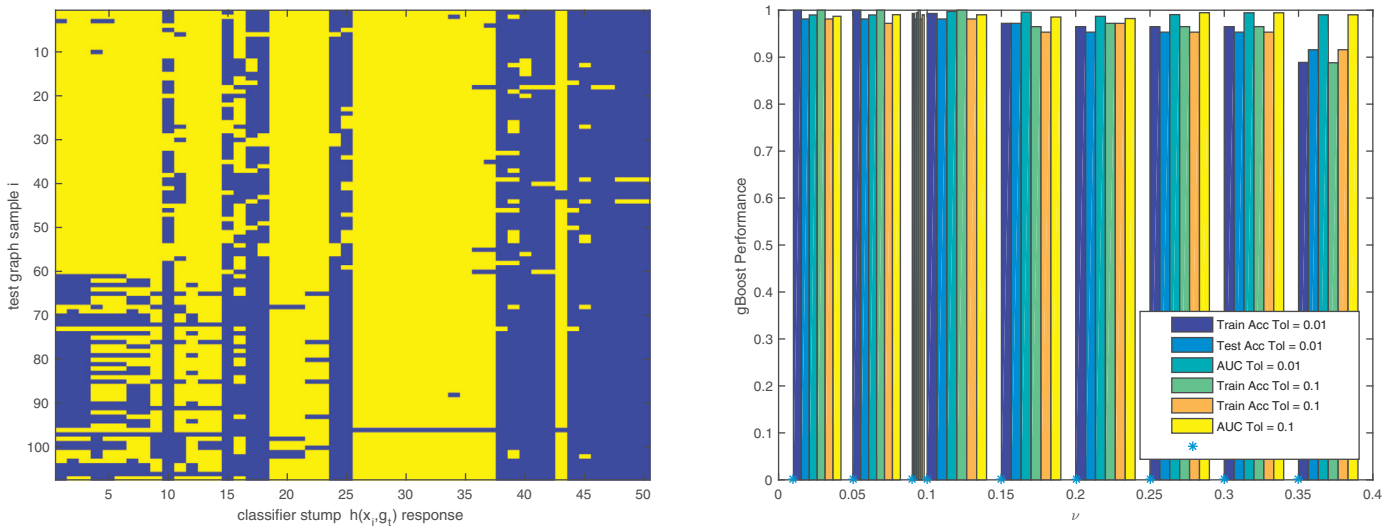

Fig. 3. (a) Graph test data $test_G$: subclassifier responses $+1$ (yellow); -1 (dark blue); (b) ν parameter controlling training accuracy. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

accuracies as well as the AUC for solving the optimization problem with the two used tolerances. For a short range of the parameter ν the results are encouraging and outperform well-known classifier methods as illustrated in Table 4.

The classification algorithm gBoost takes into account the structure embedded in the graphs proving that such information is advantageous as compared to the traditional two-dimensional feature vectors framework. The rationale is that it incorporates relations among the nodes. As a consequence extra knowledge allows better models, which fosters the goodness of predictions. Overall, gBoost is competitive among the state-of-the-art methods considered in the study.

4.3.2. Diane database: graph embedded learning

The results are obtained with the 30 financial ratios described in Table 2 and considering historical data three years before bankruptcy, therefore 90 financial attributes overall. The first step is to build the affinity weight graph matrix W by incorporating geometric neighbourhood information of the bankruptcy data set a described in Section 3.4.1. While solving the problem with Kernel Locality Preserving Projections (KLPP) [7], the Laplacian penalising functional together with the α regularization parameter control the smoothness of the basis vectors approximation. Then the transformation matrix is built mapping the data points to the data subspace. Once the compact representations are obtained, we seek for a learning model where classification can effectively be performed.

The procedure is as follows. First, the Euclidean distance which evaluates the "closeness" between any two data points was chosen. Second, the NeighborMode was set to construct the graph in two modes: K-Nearest Neighbor (KNN) or Supervised mode (SUP). In KNN the number of p-nearest neighbors is set to build a complete graph ($p = 0$) or if and only if two nodes are among the nearest neighbors of each other ($p > 0$) we put an edge between them. In SUP mode, an edge between two nodes is added if and only if they belong to the same class ($p = 0$), or if they belong to same class and they are among the ($p > 0$) nearest neighbors of each other. The Supervised mode was selected ($p = 5$) because we have three years of historical financial data and KNN can hardly handle this information. Third, to build the graph weight matrix we have chosen the HeatKernel by setting up the kernel width parameter. Finally, after projecting the data nicely into the data subspace by the embedding graph learning, an SVM is used to perform classification.

In Fig. 4 on top two representations of non-projected and projected data with embedding graph structure are represented. In the bottom plot the visualization of the 10-fold cross-validation (CV) accuracy of the SVM classification by changing the the σ Gaussian kernel width and the α regularization parameter is depicted. In all the experiments we decided to use RBF kernel since it was shown to be the best in previous empirical results running in the same data set [29]. The results are very good in particular for certain values of σ and α, in fact, those corresponding to the yellow-orange area depicted in the surface (e.g. for the pair (σ, α) with values (0.6, 0.35), the CV accuracy attains 99.59%). The choice of Laplacian penalty in SSSL allows to incorporate the prior information that relate neighboring points across the firms historical data.

5. Conclusion and future work

The combination of the formalism of graphs with a powerful frequent pattern mining algorithm such as gBoost evidenced that the structure is able to effectively capture knowledge essential to attain good predictions in financial settings. In this work we developed an algorithm for graph construction on the grounds of the binary relationships found on qualitative data from the financial credit risk problem. We used gBoost classifier to mine specific sampled graphs that are able to predict the samples category in either bankrupt or non-bankrupt. Furthermore, when large-scale historical data is available the data can be cast into an embedded graph. Once we obtain compact representations of the firms behavior, the subspace learning procedure can effectively be performed in the lower dimensional subspace with an SVM. Both methodologies can find the graph motifs in data which are able to foster better predictions using as the experimental datasets, respectively, a qualitative data benchmark from UCI Machine Learning Repository and a real-world French database of corporate companies. The experimental results empirically demonstrated by using structural approaches the performance results can be enhanced in terms of prediction accuracy in particular if graphs to cast data are carefully built. While most of the approaches consider supplying known values of required input variables, in the pattern mining approach the structural component is also taken into account making the model effective and robust. It can partly be used herein to shed light on how both approaches incorporate more knowledge through the graph component for better exposure of credit risk financial problems. Another reason for using graph pattern mining is the

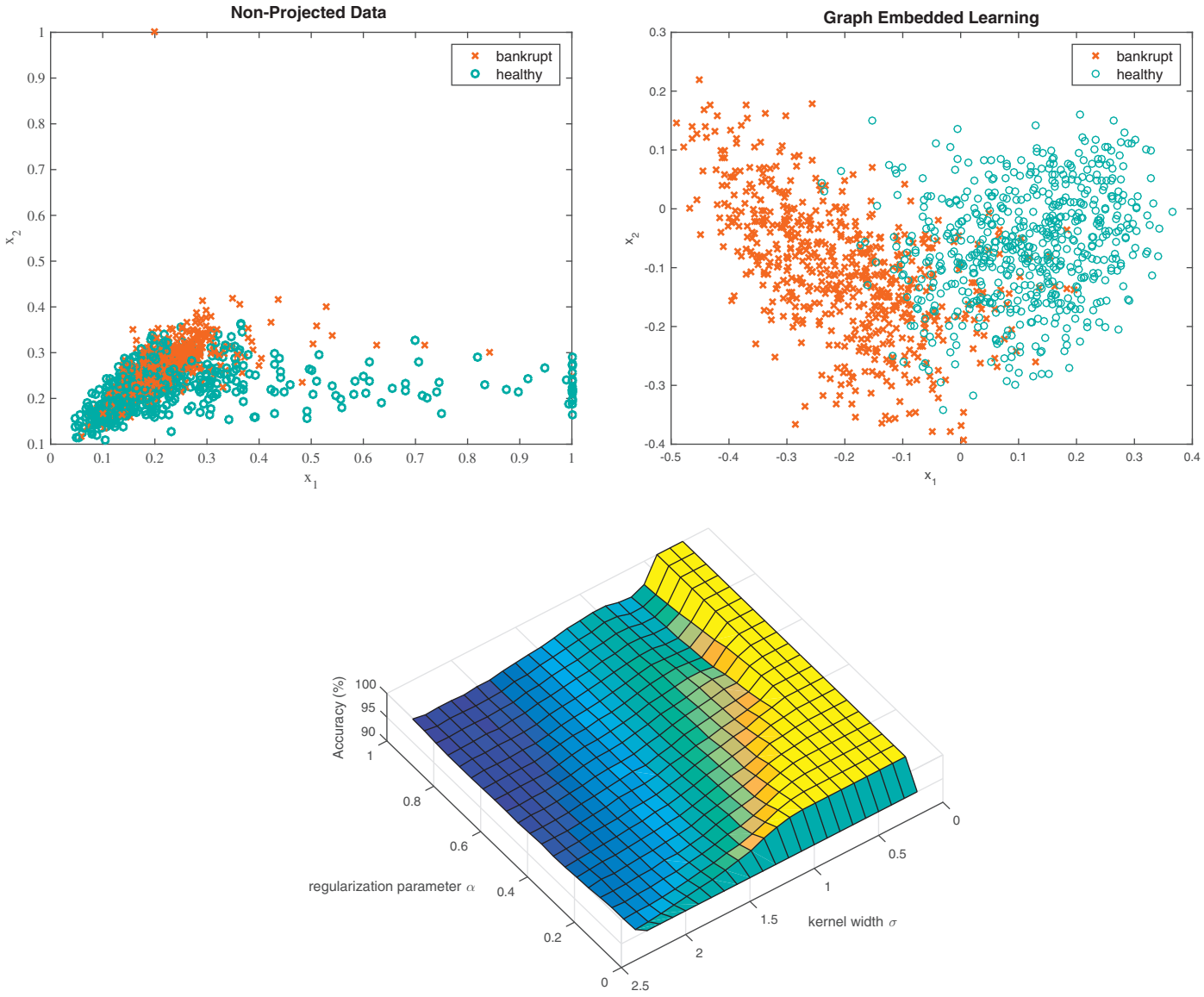

Fig. 4. Top: Non-projected and Projected data with features x_1 and x_2 of DIANE dataset; Bottom: Cross-Validation Performance Accuracy (%) of Diane database using an SVM classifier with σ kernel width and α regularization parameter in KLPP.

easiness of visualizing financial data in the big data era. In summary, graph mining has promising advantages on distributed graph algorithms, graph data visualization and easiness to deal with big data.

Future work will study the scalability to large graph data possibly with distributed approaches. Additionally the presented study is not limited to financial credit risk assessment problems. The structure pattern mining methodology can be simply extended to other kind of data.

Acknowledgments

This work was supported by national funds through the Portuguese Foundation for Science and Technology (FCT), National Natural Science Foundation of China (Contact No. 11601129), and the European Regional Development Fund (FEDER) through COMPETE 2020 Operational Program for Competitiveness and Internationalization (POCI).

We also gratefully acknowledge the useful comments and suggestions of the editor and the anonymous reviewers that helped to improve the paper.

References

[1] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, NY, USA, 1993, pp. 207–216.
[2] E. Altman, M. Iwanicz-Drozdowska, E.K. Laitinen, A. Suvas, Distressed Firm and Bankruptcy Prediction in an International Context: A Review and empirical Analysis of Altman's z-score Model, Social Science Electronic Publishing, 2014.
[3] E. Bartoloni, M. Baussola, Financial performance in manufacturing firms: a comparison between parametric and non-parametric approaches, Bus. Econ. 49 (1) (2012) 32–45.
[4] H. Blockeel, T. Witsenburg, J. Kok, Graphs, hypergraphs and inductive logic programming, in: P. Frasconi, K. Kersting, K. Tsuda (Eds.), Proceedings of the 5th International Workshop on Mining and Learning with Graphs, August 1–3, Florence, Italy, 2007, pp. 93–96.
[5] C. Borgelt, M. Berthold, Mining molecular fragments: finding relevant substructures of molecules, in: Proceedings of the IEEE International Conference on Data Mining, 2002.

[6] H. Bunke, K. Riesen, Recent advances in graph-based pattern recognition with applications in document analysis, Pattern Recognit. 44 (5) (2011) 1057–1067. http://dx.doi.org/10.1016/j.patcog.2010.11.015.
[7] D. Cai, X. He, Y. Hu, J. Han, T. Huang, Learning a spatially smooth subspace for face recognition, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition Machine Learning (CVPR'07), 2007, pp. 1–7.
[8] N. Chen, B. Ribeiro, A. Chen, Financial credit risk assessment: a recent review, Art. Intell. Rev. 45 (1) (2016) 1–23.
[9] K.F. Cheng, C.K. Chu, R. Hwang, Predicting bankruptcy using the discrete-time semi-parametric hazard model, Quant. Financ. 10 (9) (2010) 1055–1066.
[10] F. Chung, Spectral Graph Theory, first ed., American Mathematical Socitey, Providence, 1997.
[11] K.G. Coleman, T.J. Graettinger, W.F. Lawrence, Neural networks for bankruptcy prediction: The power to solve financial problems, Artif. Intell. Rev. 4 (4) (1991) 48–50.
[12] A. Demiriz, K.P. Bennett, J. Shawe-Taylor, Linear programming boosting via column generation, Mach. Learn. 46 (1–3) (2002) 225–254, doi:10.1023/A:1012470815092.
[13] G. Fejerkiraly, Bankruptcy prediction: a survey on evolution, critiques, and solutions, Acta Univ. Sapientiae Econ. Bus. 3 (1) (2015) 93–108.
[14] V. Garcia, A.I. Marques, J.S. Sanchez, An insight into the experimental design for credit risk and corporate bankruptcy prediction systems, J. Intell. Inf. Syst. 44 (1) (2015) 159–189.
[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18, doi:10.1145/1656274.1656278.
[16] L. Holder, Greedy Search Approach of Graph Mining, Springer, US, 2011.
[17] R.C. Hwang, Ruey-Ching, H. Chung, C.K. Chu, Predicting issuer credit ratings using a semi-parametric method, J. Empir. Financ. 17 (1) (2010) 120–137.
[18] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, G. Varghese, Network monitoring using traffic dispersion graphs, in: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ACM, 2007, pp. 315–320.
[19] A. Inokuchi, T. Washio, H. Motoda, An apriori-based algorithm for mining frequent substructures from graph data, in: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD '00, Springer-Verlag, London, UK, 2000, pp. 13–23.
[20] M.-J. Kim, I. Han, The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms., Expert Syst. Appl. 25 (4) (2003) 637–646.
[21] E. Kirkos, Assessing methodologies for intelligent bankruptcy prediction, Artif. Intell. Rev. 43 (1) (2015) 83–123.
[22] S. Kramer, Inductive Database Approach to Graphmining, Springer, US, 2011.
[23] M. Kuramochi, G. Karypis, Frequent subgraph discovery, in: Proceedings of the IEEE International Conference on Data Mining, ICDM, 2001, pp. 313–320, doi:10.1109/ICDM.2001.989534.
[24] H. Li, H. Adeli, J. Sun, J.G. Han, Hybridizing principles of topsis with case-based reasoning for business failure prediction, Comput. Oper. Res. 38 (2) (2011) 409–419.
[25] Liang, CHEN, Muzi, YANG, Xiaoguang, Parametric and non-parametric combination model to enhance overall performance on default prediction, J. Syst. Sci. Complex. 27 (5) (2014) 950–969.
[26] D.G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, NY, 1969. Decision and control.
[27] G.D.S. Martino, A. Sperduti, Mining structured data, IEEE Comput. Intell. Mag. 5 (1) (2010) 42–49.
[28] S. Nijssen, J.N. Kok, The gaston tool for frequent subgraph mining, Electron. Notes Theor. Comput. Sci. 127 (1) (2005) 77–87. http://dx.doi.org/10.1016/j.entcs.2004.12.039.
[29] B. Ribeiro, N. Chen, Graph weighted subspace learning models in bankruptcy, in: Proceedings of the International Joint Conference on Neural Networks, 2011, pp. 2055–2061.
[30] B. Ribeiro, C. Silva, N. Chen, A. Vieira, J.C. das Neves, Enhanced default disk models with SVM+, Expert Syst. Appl. 39 (2012) 10140–10152.
[31] H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo, K. Tsuda, gBoost: a mathematical programming approach to graph classification and regression, Mach. Learn. 75 (1) (2009) 69–89.
[32] H. Saigo, T. Uno, K. Tsuda, Mining complex genotypic features for predicting HIV-1 drug resistance., Bioinformatics 23 (18) (2007) 2455–2462.
[33] G. Wang, J. Ma, A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine, Expert Syst. Appl. 39 (5) (2012) 5325–5331.
[34] J. Xuan, J. Lu, G. Zhang, X. Luo, Topic model for graph mining, IEEE Trans. Cybern. 45 (12) (2015) 2792–2803.
[35] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.
[36] X. Yan, J. Han, gSpan: graph based substructure pattern mining, in: Proceedings of the IEEE International Conference on Data Mining, Maebashi, Japan, 2002, pp. 721–724.
[37] J. Zaima, Bankruptcy prediction: the case of belgian SMEs, Rev. Account. Financ. 15 (1) (2016) 101–119.

Bernardete Ribeiro is Associate Professor with Habilitation at the Informatics Engineering Department, University of Coimbra, Portugal, from where she received a Ph.D. in Electrical Engineering, speciality of Informatics. Her research interests are in the areas of machine Learning, pattern recognition, risk analysis and signal processing and their applications to a broad range of fields. Bernardete Ribeiro is IEEE Senior Member, and member of International Association of Pattern Recognition (IAPR) and member of ACM.

Ning Chen in Computer Science as B.Sc. and M.Sc. by Shandong University in China in 1994 and 1997, respectively, then completed a Ph.D. degree in Chinese Academy of Sciences in 2001. She ever worked as researcher in City University of Hong Kong, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa (FCT-UNL), Chinese Academy of Sciences, and Instituto Superior de Engenharia do Porto, Instituto Superior Politecnico do Porto (ISEP-IPP). Since 2015 she is a professor in College of Computer Science and Technology (Software College), Henan Polytechnic University of China. Her research areas include data mining, machine learning, and risk analysis.

Alexander Kovačec is Professor at the Mathematics Department, University of Coimbra, Portugal. He received a Ph.D. in Mathematics from the University of Vienna, Austria. His main interests are in Linear Algebra and its Applications. He is a collaborator of Centre of Mathematics of University of Coimbra (CMUC).