A Structural Pattern Mining Approach for Credit Risk Assessment

Bernardete Ribeiro[1(✉)], Ning Chen[2], and Alexander Kovačec[3]

[1] CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal
bribeiro@dei.uc.pt
[2] College of Computer Science and Technology (Software College), Henan Polytechnic University, 2001 Century Avenue, Jiaozuo 454003, Henan, People's Republic of China
nchenyx@outlook.com
[3] Department of Mathematics, University of Coimbra, Coimbra, Portugal
kovacec@mat.uc.pt

Abstract. In recent years graph mining took a valuable step towards harnessing the problem of efficient discovery of substructures in complex input data that do not fit into the usual data mining models. 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. When it comes to financial settings, data is very complex and in particular when risk factors relationships are not taken into account it seriously affects the goodness of predictions. In this paper, we posit that risk analysis can be leveraged if structure can be taken into account by discovering financial motifs in the input graphs. We use gBoost which learns from graph data using a mathematical linear programming procedure combined with a substructure mining algorithm. An algorithm is proposed which has shown to be efficient to extract graph structure from feature vector data. Furthermore, we empirically show that the graph-mining model is competitive with state-of-the-art machine learning approaches in terms of classification accuracy without increase in the computational cost.

Keywords: Graph mining · Classification · Financial applications

1 Introduction

Nowadays, data is naturally structured in form of trees or graphs, which are structures that may convey important information. 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 [1]. 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.

E. Onieva et al. (Eds.): HAIS 2015, LNAI 9121, pp. 73–84, 2015.
DOI: 10.1007/978-3-319-19644-2_7

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 and phylogenetic trees [2], molecular graphs [3], computer networks [4], hypertextual and XML documents, social networks, mobile call networks, among other important data [5].

Pattern mining takes essentially two approaches: statistical learning and structural. In the statistical learning, patterns are represented by feature vectors $\mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n$ 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 a financial risk analysis problem we take a structural pattern mining approach. We propose a graph construction algorithm on the basis of a qualitative data set of financial statements to gain further insights on the data structure, and then a graph-based model for pattern mining is generated via gBoost [7], a frequent subgraph discovery technique on the grounds of mathematical programming and gSpan algorithm [8]. This pattern-growth method uses Dept-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 Naive Bayes, Decision Trees and Support Vector Machines when unstructured dimensional feature vectors are used, without a substantial increase in the computational cost. Our case study encompasses a graph-based methodology that enables to unravel structural subtleties otherwise hidden in the data.

In the next section we will review the literature and background for the work. The graph classification model is explained in detail in Sect. 3. In Sect. 4 an algorithm is proposed for extracting graphs from feature vectors aiming at gathering the graph data. In the context of financial credit risk, we present in Sect. 5 the experiments including the research design and discussion of results. The paper will end with the conclusions and future work in Sect. 6.

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 intends to solve the problem stated as follows: given a number of companies labeled as bad / good credit or 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.

In the literature, a wide range of methods that can be divided into parametric methods, semi-parametric methods, and non-parametric methods have approached this problem from the viewpoint of model structure specification. Parametric methods (mainly referred to statistical methods) specify definitely the model structure and modeled process. Non-parametric methods that includes Artificial Neural Networks (ANNs), Fuzzy Set Theory (FST), Decision Trees (DTs), Case-Based Reasoning (CBR), Support Vector Machines (SVMs), Rough Set Theory (RST) among other intelligent methods, determine the model structure from the data. Semi-parametric methods define the modeled process with flexible structure [9,10].

When dealing with real world financial credit risk problems, there 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.) [11]. These performance key indicators are well fit to establish the relationships between nodes of financial companies.

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.

2.2 Graph-Based Pattern Mining

Graph classification can be investigated from two perspectives: graph classification (between graph) and vertex classification (within graph). In the former we aim at classifying individual graphs and in the latter we are interested in the classification of individual vertices within a graph. Either way, our focus is to find non-trivial characteristics (e.g. pattern subgraphs) that can determine class membership, that is, to assign a class label.

Classification of graphs has many applications. An obvious example are molecular structures that can be represented by graphs and classification is used to predict properties of molecules such as toxicity [3]. Another example comprises computer network traffic where the problem representation is embedded in graphs. Network traffic traces can represent the network behavior exposing better visual and structural differences among nodes and capturing many interesting patterns of node interactions [4].

Graph-based pattern mining took a new breed of approaches since the introduction of frequent pattern mining in [12]. In particular, many subgraph mining algorithms have been developed such as Apriori based methods like AGM [13], FSG [14], or pattern-growth methods like gSpan [8] and Gaston [15]. A major challenge in subgraph mining is the subgraph isomorphism, which is an NP-complete problem [8]. In gSpan, Dept-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 [7] 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, inductive programming logic and inductive database approaches.

3 gBoost Classifier

gBoost [7] is an extension of boosting for graphs and comprises a mathematical programming tool [16] 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 [16]. 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 [8]. gBoost uses first gSpan method [8] which finds frequent subgraphs and constructs a canonical search space in the form of a Depth-First Search (DFS) which is an algorithm for traversing or searching tree or graph data structures. With the proviso for achieving an optimal search the tree structure and the DFS code are necessary.

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 $\mid \mathcal{T} \mid$ 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:

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

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:

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

where $D = \frac{1}{cN}, c \in (0,1)$ is the cost classification parameter controlling the misclassification errors [7,16]. After solving the dual optimization problem, the primal solution α 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 [17]. 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

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

As defined aboveAs 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:

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

The set $\mathcal{T}^{(k)}$ is updated by adding the new index number $t^* : \mathcal{T}^{(k+1)} = \mathcal{T}^{(k)} \cup \{t^*\}$.

The iterative procedure proceeds 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 $h(\mathbf{x}, g_t)$ and takes the form:

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

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

4 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 in 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 is presented in Algorithm 1 and will be used in the next section with the benchmark qualitative data. We have coded the algorithm in Matlab for easiness of use with gBoost package. 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. The assessment of the gBoost classifier has been performed with the classification accuracy and test AUC performance metrics.

Algorithm 1. Graph Construction from Feature Vector Data

Input: For each collection of data $\mathcal{D}$, $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ with labels $\{y_n \in \{+1, -1\} \quad n = 1, 2, \cdots N\}$
/*Cycle over N rows*/
for all $n \leftarrow 1, \cdots, N$ **do**
  Initializations
  /* Cycle over the first row */
  for all $j \leftarrow 1, \cdots, NumNodes - 1$ **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 forOutput: 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

In the next section we will present the experimental setting, describe the research design including the dataset, the evaluation metrics and the empirical analysis.

5 Experiments

5.1 Dataset

The Qualitative Bankruptcy (QB) dataset can be download from https://archive.ics.uci.edu/ml/datasets/Qualitative_Bankruptcy. The attributes and samples of the dataset are from [18] where a rule-based approach was presented.

Table 1. QB Dataset: the attributes are qualitative (Positive, Average, Negative) and the assigned class is (B, NB).

	Financial indicators	Qualitative attributes
1	Industrial risk	{P,A,N}
2	Management risk	{P,A,N}
3	Financial flexibility	{P,A,N}
4	Credibility	{P,A,N}
5	Competitiveness	{P,A,N}
6	Operating risk	{P,A,N}
7	Class	{B,NB}

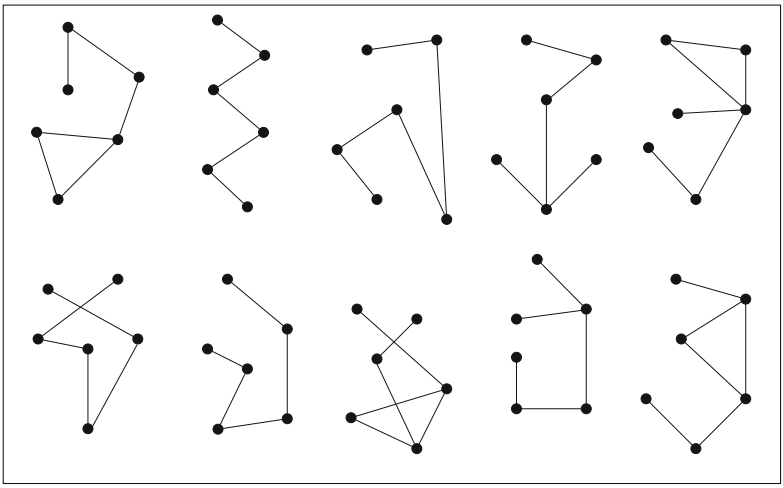

Fig. 1. Motifs for 6 node graph financial samples.

The sample size is 250 and the number of attributes is 6 each corresponding to Qualitative Parameters in Bankruptcy, namely, Industrial Risk, Management Risk, Financial Flexibility, Credibility, Competitiveness, and Operating Risk. The attribute information is nominal (P-Positive, A-Average, N-Negative) and there are two classes (B-Bankruptcy, NB-Non-Bankruptcy). The dataset is unbalanced being composed by 143 samples pertaining to the class NB and 107 samples pertaining to the class B. We assigned B to the positive class and NB to the negative class. The dataset is briefly described in Table 1. For example, sample #1 = (P, P, A, A, A, P) is assigned to class B while sample #250 = (P, N, N, N, A, A) pertains to the other class NB.

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. Each graph has 6 nodes. For easiness of handling the data we decided to assign

a corresponding weight to each qualitative value (for example, we assigned 2 to Positive, 1 to Average and 3 to Negative). From the labels of the qualitative data set (B, NB) we assigned the label (+1, −1) to the positive and negative class, respectively, for use in the gBoost algorithm. According to the partition in train and test we, respectively, built the label vectors $train_Y$ and $test_Y$ with the same size as the train and test graph samples.

5.2 Financial Motifs

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.

5.3 Evaluation Metrics

In order to evaluate a binary decision task we first define a contingency matrix representing the possible outcomes of the classification, as shown in Table 2.

Table 2. Contingency table for binary classification.

	Class positive	Class negative
Assigned positive	tp (True positives)	fp (False positives)
Assigned negative	fn (False negatives)	tn (True negatives)

Several measures have been defined based on the contingency Table 2, such as, error rate ($\frac{fp+fn}{tp+fp+tn+fn}$), 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.

AUC captures a single point on the Reception Operating Characteristic (ROC) curve. It is also known as ROC curve, and is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. Plotting the true positive rate against the false positive rate at various threshold settings creates the curve. AUC is sometimes referred as balanced accuracy.

5.4 Empirical Analysis

In this section we present several algorithms spanning over machine learning and data mining methods using the open source Weka Toolbox[1].

[1] http://www.cs.waikato.ac.nz/ml/weka/ [19].

Support Vector Machines (SVMs) 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. SVMs use the structural risk minimization principle to avoid overfitting. Since the introduction to the area of financial risk analysis [20], SVMs have gained wide popularity owing to the good generalization on a small amount of high-dimensional data [21]. Apart from SVM, we also used Neural Networks, Naive Bayes, Decision Trees, fuzzy grid, and random committee for comparison. J48 constructs a decision tree well adapted to the training data and then prunes the tree structure to avoid over-fitting. Naive Bayes (NB) estimates the probability of each class under the assumption of feature independence. Multi-Level Perceptron (MLP) and 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 classifiers to improve the classification accuracy. Since we are interested to compare the two categories of methods - unstructured versus structured - the model selection for each case was not so important. Therefore, we let the default parameters in all the algorithms tested.

Table 3. Classifier methods vs Accuracy ratio (%)

Classifier method	Accuracy ratio (%)	Classifier method	Accuracy ratio (%)
SVM	98.13	MLP	93.46
Naive Bayes	93.46	RBF	97.19
J48	95.32	Fuzzy grid	96.26
gBoost	98.13	Rand committee	96.26

In Table 3 the performance accuracy of various machine learning methods including SVM, J48, Naive Bayes and neural networks (Multi-Layer Perceptron (MLP) and radial Basis Functions (RBF)), fuzzy grid, and random committee are illustrated. The gBoost outperforms neural networks, decision trees and many other machine learning algorithms, while showing competitive performance as compared to SVM.

The test ROC AUC for gBoost found was 99.06 (see Fig. 3) while for the SVM the test ROC AUC was 98.1. Although these are preliminary results graph mining can boost classification in this financial setting via a small size dataset with low dimensionality. 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 $h(\mathbf{x}, g_t)$.

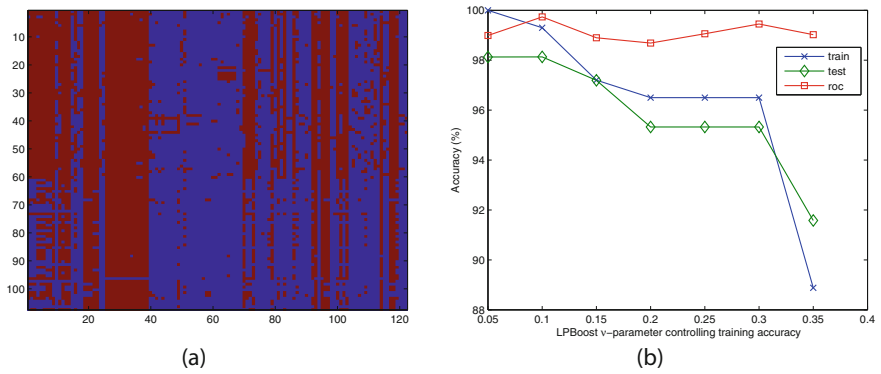

Fig. 2. (a) Graph test data $test_G$; (b) ν parameter controlling training accuracy.

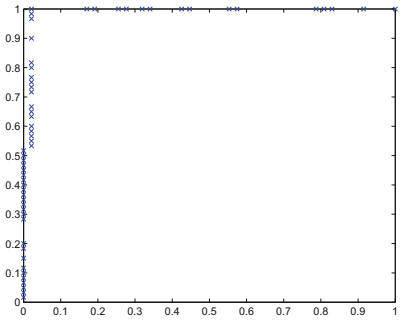

Fig. 3. ROC curve

For gBoost, the maximum pattern size, which in our case corresponds to the maximum number of nodes in a subgraph, was constrained up to 6, since we have 6 attributes defining the financial indicators. The regularization parameter ν is chosen from $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35\}$.

In Fig. 2 results from running gBoost on the dataset are illustrated (a) Graph Test Data $test_G$ and (b) performance of gBoost by varying ν parameter controlling training accuracy. This parameter is used in the graph optimization process LPBoost for finding frequent subgraphs [16]. The convergence tolerance ϵ used in the runs was set to 0.05. For a short range of the parameter ν the results are encouraging and outperform well-known classifier methods as illustrated in Table 3.

The classification algorithm gBoost takes into account the structure embedding information that is advantageous as compared to the traditional two-dimensional feature vectors framework. As such, it incorporates relations among the entities that integrate extra knowledge capable of better models, which fosters the goodness of predictions.

6 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 a credit risk problem. Furthermore we used gBoost classifier to mine specific sampled graphs that are able to predict the samples category in either bankrupt or non-bankrupt. The structure pattern mining methodology can be simply extended to other kind of data.

Despite the myriad of statistical learning models able to tackle this problem we empirically showed that by using structural approaches the performance results can be enhanced in particular if graphs to cast data are carefully built.

Future work will trace the performance of this approach while spanning over the historic data in a real-world data set. Another future issue will be to check the scalability to large graph data.

References

1. Martino, G.D.S., Sperduti, A.: Mining structured data. IEEE Comput. Intell. Mag. **5**(1), 42–49 (2010)
2. Pimentel, C., Van Der Straeten, D., Pires, E., Faro, C., Rodrigues-Pousada, C.: Characterization and expression analysis of the aspartic protease gene family of cynara cardunculus l. FEBS J. **274**(10), 2523–2539 (2007)
3. Borgelt, C., Berthold, M.: Mining molecular fragments: finding relevant substructures of molecules. In: Proceedings of the IEEE International Conference on Data Mining, ICDM 2002, pp. 51–58, December 2002
4. Iliofotou, M., Pappu, P., Faloutsos, M., Mitzenmacher, M., Singh, S., Varghese, G.: Network monitoring using traffic dispersion graphs (tdgs). In: Dovrolis, C., Roughan, M. (eds.) Internet Measurement Conference, pp. 315–320. ACM (2007)
5. Saigo, H., Uno, T., Tsuda, K.: Mining complex genotypic features for predicting HIV-1 drug resistance. Bioinformatics **23**(18), 2455–2462 (2007)
6. Bunke, H., Riesen, K.: Recent advances in graph-based pattern recognition with applications in document analysis. Pattern Recogn. **44**(5), 1057–1067 (2011)
7. Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., Tsuda, K.: gBoost: a mathematical programming approach to graph classification and regression. Mach. Learn. **75**(1), 69–89 (2009)
8. Yan, X., Han, J.: gspan: graph based substructure pattern mining. In: IEEE International Conference on Data Mining, Maebashi City, Japan, pp. 721–724, December 2002
9. Cheng, K.F., Chu, C.K., Hwang, R.: Predicting bankruptcy using the discrete-time semi-parametric hazard model. Quant. Finance **10**(9), 1055–1066 (2010)
10. Hwang, R.C., Chung, H., Chu, C.K.: Predicting issuer credit ratings using a semiparametric method. J. Empirical Finance. **17**(1), 120–137 (2010)
11. Ribeiro, B., Silva, C., Chen, N., Vieira, A., das Neves, J.C.: Predicting issuer credit ratings using a semiparametric method. Expert Syst. Appl. **39**, 10140–10152 (2012)

12. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD 1993, pp. 207–216. ACM Press, New York, NY, USA(1993)
13. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 13–23. Springer, Heidelberg (2000)
14. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Proceedings of the IEEE International Conference on Data Mining, ICDM 2001, pp. 313–320 (2001)
15. Nijssen, S., Kok, J.N.: The Gaston tool for frequent subgraph mining. Electron. Notes Theor. Comput. Sci. **127**(1), 77–87 (2005). Proceedings of the International Workshop on Graph-Based Tools (GraBaTs 2004) Graph-Based Tools 2004
16. Demiriz, A., Bennett, K.P., Shawe-Taylor, J.: Linear programming boosting via column generation. Mach. Learning **46**(1–3), 225–254 (2002)
17. Luenberger, D.G.: Optimization by Vector Space Methods Decision and Control. Wiley, New York (1969)
18. Kim, M.J., Han, I.: The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. Expert Syst. Appl. **25**(4), 637–646 (2003)
19. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. newsl. **11**(1), 10–18 (2009)
20. Min, J.H., Lee, Y.C.: Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Syst. Appl. **28**(4), 603–614 (2005)
21. Yang, Z., You, W., Ji, G.: Using partial least squares and support vector machines for bankruptcy prediction. Expert Syst. Appl. **38**(7), 8336–8342 (2011)