Feature Selection Methods Involving SVMs for Prediction of Insolvency in Non-life Insurance Companies

Sancho Salcedo-Sanz* [a], Mario DePrado-Cumplido [a],
María Jesús Segovia-Vargas [b], Fernando Pérez-Cruz [a] and
Carlos Bousoño-Calzón [a]

[a] *Department of Signal Theory and Communications, Universidad Carlos III de Madrid*

[b] *Department of Financial Economy and Accounting I, Universidad Complutense de Madrid*

Abstract

In this paper we propose two novel approaches for feature selection and ranking tasks based on Simulated Annealing (SA) and Walsh analysis, which use a Support Vector Machine (SVM) as underlying classifier. These approaches are inspired by one of the key problems in the insurance sector: predicting the bankruptcy of a non-life insurance company. This prediction is based on accounting ratios, which measure the health of the companies. The approaches proposed in this paper provide a set of ratios, the SA approach, and a ranking of the ratios, the Walsh analysis ranking, which would allow deciding about the financial state of each studied company. The proposed feature selection methods are applied for predicting the insolvency of several Spanish non-life insurance companies, yielding state-of-the-art results in the performed tests.

Keywords: Business failure, Insolvency, Non-life insurance companies, Support Vector Machines, Simulated Annealing, Walsh Analysis.

[1] This paper has been partially supported by the Project number 07T/0016/2003 of the Comunidad de Madrid.

1 Introduction

Financial risk assessment is one of the key issues when controlling the insurance market sector, due to the high amount of money to pay by the state insurance guaranty funds when an insurance company has gone bankrupt, and, at the same time, it is not desirable excessively perturb the market with unneeded interventions. Also protecting the society and the whole insurance sector against insolvent insurance companies is of great concern to auditors, governments and managers in the sector, because a bankruptcy reduces the public confidence in all the insurance companies. These facts explain the increasing interest in accurately predicting insurance companies failures. The European Union, through the *Solvency II Project*, has taken an active role in redefining a set of rules to provide the society information about how healthy their insurance companies are.

Controlling the solvency of non-life insurance companies, which is the aim of this publication, is a pattern recognition problem in which we have to decide whether a company is solvent or insolvent, or to predict if it would be insolvent in the years to come, from a given set of inputs. These inputs are accounting ratios that measure the financial health of the insurance companies.

There have been several previous proposals to applied operational research and pattern recognition methods to predict business failure, see works by [Ambrose (1994)], [Barniv (1990)], [Tam (1991)], but just a few have been applied to insurance sector. Among these approaches to the prediction of insolvency in insurance companies, the emergent classifiers algorithms such as Neural Networks [O'leary (1998)], [Serrano (1996)], Rough Set [Dimitras (1998)], Support Vector Machines (SVM) or Genetic Programming (GP) [Li (2000)], [Salcedo (2003)], have gained importance in the last few years as powerful methods which provide very good results in classification performance and generalization. This paper is focused on the application of SVM-based method for improving the prediction of the insolvency of non-life insurance companies.

Support Vector Machines (SVMs) are state-of-the-art tools for linear and nonlinear knowledge discovery problems [Schölkopf (2002)], [Vapnik (1998)]. SVMs are designed to work in high dimensional spaces even when very few input patterns are available, as in the situation at hand. The SVM is based on the maximum margin idea which states that, without any prior knowledge, the best classification boundary must correctly classify every given sample and be situated as far as possible from all the samples, reducing the risk of misplacing a new unseen pattern. But in many applications there are irrelevant features that, if the data are scarce, may bias the solution of the employed pattern classifier and will make it perform poorly with new unseen examples.

Feature Selection (FS) is an open problem in machine learning, which basically consists of finding a subset of input features that describes the underlying system structure as well as, or better, than all available features. In the ranking problem, the point is to weigh the features according to its relevance. The significance of FS appears when the given features are used to explain the achieved results. Note that in some applications such as financial ones, being able to explain the obtained solution (in terms of the selected input features) becomes as relevant as obtaining the best possible answer (accuracy of the subsequent classifier or regressor).

In this paper we propose two new features selection procedures that work jointly with a Support Vector Machine. The first proposal is based on a Simulated Annealing algorithm [Kirpatrick (1983)], [González (2002)], which uses the SVM validation error to select the most relevant features for the problem at hand. Our second proposal works with the Walsh expansion [Vose (1998)] of a particular binary function, which relates binary strings with the SVM soft output. This expansion will not be used to select the best features, but to rank them, giving richer information about the problem, which will allow to discuss the relevance of each of the feature according to this ranking.

The rest of the paper is organized as follows. We provide background material on SVM, Feature Selection, Simulated Annealing and Walsh analysis in Section 2. Section 3 is devoted to the proposed feature selection schemes. We show the actual validity of the proposed approaches with non-life insurance companies in Section 4 and we end the paper in Section 5 with some concluding remarks and suggested further work.

2 Background

2.1 Support Vector Machines

The Support Vector Machine has been reported as a powerful method for classification problems, with very good properties of generalization [Burgues (1998)]. This section provides a brief summary of the standard SVM for classification [2] applied to business failure, starting from the simple linear SVM and moving on to the nonlinear SVM.

Consider a set of l firms represented by the value of their n ratios $\{\mathbf{x}_i\}$, $i = 1, \ldots, l$, with $\mathbf{x}_i \in \mathbb{R}^n$, and a set of associated labels $y_i \in \{-1, 1\}$ which

[2] A more complete analysis as well as further results about SVMs can be found in [Burgues (1998)], [Schölkopf (2002)].

describe the firm as failed or healthy, respectively. First, imagine that this training set can be separated by a linear hyperplane (a line in 2D, a plane in 3D and so on). The SVM solves the following problem:

Find $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$, to minimize $\frac{1}{2}\|\mathbf{w}\|^2$, subject to:

$$y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 \qquad \forall i = 1, \ldots, l \tag{1}$$

Once such $\mathbf{w}$ and b have been found [3], our classification rule for new firms is given by $sign(\mathbf{w}^T \cdot \mathbf{x} + b)$. Thus, firms located in one side of the hyperplane will be healthy and on the other side will be failed, where the associated error to this classification, $R(\mathbf{w}, b)$, is defined as the percentage of misclassified firms.

Consider now the case when the points in the training set $\{\mathbf{x}_i\}$ are not linearly separable; then constraint (1) cannot be satisfied. We can introduce then some nonnegative slack variables ξ_i in order to overcome this difficulty. The SVM formulation results in this case:

Find $\mathbf{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$ and $\xi_i,\ i = 1, \ldots, l$, to minimize $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \xi_i$, under the constraints:

$$y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, l \tag{2}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, l \tag{3}$$

where C is a parameter of the classifier to be estimated.

Figure 1 shows an example of a classification problem formed by two classes (crosses and dots). The solution in Subplot (a) is obtained with a linear SVM, which defines a linear decision boundary ($\mathbf{w}$) unable to completely separate both classes. The dashed lines represent the margins, i.e the set of points that satisfies Equation (2), and therefore are the limits of each class. The samples over the margins, which are surrounded by circles, are the Support Vectors, the only information needed to plot the boundary. The samples located out of their regions are misclassifications (consequently with ξ_i greater than zero).

The classification obtained with the introduction of the slack variables ξ_i is still given by a linear frontier. The nonlinear SVM maps the input variable into a high dimensional (often infinite dimensional) feature space, and applies the linear SVM in this feature space. All the appearances of the mapping $\boldsymbol{\phi}$

[3] Note that $\mathbf{w}$ and b are the parameters which characterize the hyperplane.

are within dot products, which can be substituted by a kernel function. The nonlinear SVM with kernel K is equivalent to a regularization problem in the reproducing kernel Hilbert space H_K:

Find the mapping $\boldsymbol{\phi}(\mathbf{x}) = h(\mathbf{x}) + b$ with $h \in H_K$, $b \in \mathbb{R}$ and $\xi_i,\ i = 1, \ldots, l$, to minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \tag{4}$$

subject to

$$y_i(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, l \tag{5}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, l \tag{6}$$

Figure 1 (b) illustrates a nonlinear SVM classification with a Gaussian kernel. The sample set is identical to subplot (a), but in this case all samples are correctly classified. The resolution of the problem is obtained by a linear boundary in the Hilbert space generated by the kernel, which in the input space (the one represented in the figure) is a simple curve.

The kernel used in this article is the well-known Gaussian Kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\gamma}\right)$, where γ controls the width of the Gaussian [4]. The nonlinear SVM is able to classify any set of firms as healthy or failed, with a probability of error given by $R(\mathbf{w}, b)$.

Given a training set, the selection of the input variables (financial ratios) is an important issue to be considered, due to irrelevant or redundant ratios can affect in a negative way to the result given by the SVM. This is the so called *Feature Selection Problem* (FSP) [Weston (2000)], in which the features are the financial ratios. In the next subsection we give a brief review of the Feature Selection Problem focused in non-life insurance insolvency prediction.

2.2 Feature Selection

The Feature Selection Problem (FSP) in a learning from samples scheme can be addressed as follows: Given a set of labelled data points $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$, choose a subset of m features ($m < n$), that achieves the lowest classification error [Weston (2000)], [Salcedo (2002)]. Following [Weston (2000)], we will define the FSP as finding the optimum n-

[4] This value, jointly with C are the two tunable parameters of the SVM.

column vector $\boldsymbol{\sigma}$, where $\sigma_i \in \{0,1\}$, that defines the subset of selected features, which is found as

$$\boldsymbol{\sigma}^o = arg \min_{\boldsymbol{\sigma},\boldsymbol{\alpha}} \left(\int V(y, f(\mathbf{x} * \boldsymbol{\sigma}, \boldsymbol{\alpha}))dP(\mathbf{x}, y) \right), \tag{7}$$

where $V(\cdot,\cdot)$ is a loss functional, $P(\mathbf{x}, y)$ is the unknown probability function the data was sampled from and we have defined $\mathbf{x} * \boldsymbol{\sigma} \triangleq (x_1\sigma_1, \ldots, x_n\sigma_n)$. The function $y = f(\mathbf{x}, \boldsymbol{\alpha})$ is the classification engine that is evaluated for each subset selection, $\boldsymbol{\sigma}$, and for each set of its hyper-parameters, $\boldsymbol{\alpha}$.

In this problem, the objective is to process the data in order to extract valid, novel, potentially useful, and ultimately understandable structure in data by identifying relevant and meaningless features [Bradley (1999)]. This is the first step in a *knowledge discovery* learning scheme. In this context, two main approaches can be followed:

- The *wrapper approach* to the FSP was introduced in [John (1994)]. In this approach, the feature selection algorithm conducts a search for a good subset of features using the classifier itself as part of the evaluating function. Figure 2 (a) shows the idea behind the wrapper approach: the classifier is run on the training dataset with different subsets of features. The feature subset which produces the lowest estimated error in an independent but representative test set is chosen as the final feature set. For further considerations about wrappers methods, the following bibliography can be consulted [Kohavi (1997)], [Salcedo (2002)].
- In the *filter approach* to the FSP, the feature selection is performed based on the data, ignoring the classifier algorithm. An external measure calculated from the data must be defined in order to select a subset of features. After the search, the best feature subset found is evaluated on the data by means of the classifier algorithm. Note that filter algorithms performance completely depends on the measure selected for comparing feature subsets. Figure 2 (b) shows an example of how a filter algorithm works. Filter methods are usually faster than wrapper methods. However, their main drawback is that they totally ignore the effect of the selected feature subset on the performance of the classification algorithm during the search. Further analysis and application of filter methods can be found in [Chen (1999)], [Salcedo (2002)].

For both wrapper and filter methods, a binary representation can be used for the FSP, where a 1 in the i_{th} position of the binary vector $\boldsymbol{\sigma}$ means that the feature i is considered within the subset of features, and a 0 in the j_{th} position of the binary vector means that feature j is not considered within the subset of features. Note that using this notation is equivalent to encode the problem as the vector $\boldsymbol{\sigma}$ included in expression (7). Note also that there are 2^n different subsets of features (n total number of features), and the problem is to select the best one in terms of a certain measure, which can be either internal (wrapper methods) or previous (filter methods) to the classifier.

2.3 Simulated Annealing

Simulated Annealing (SA) is a powerful solving technique which has been successfully applied to a wide variety of optimization problems [Kirpatrick (1983)], [Wang (1997)], [González (2002)]. It is inspired by the physical process of heating a substance and then cooling it slowly, until a strong crystalline structure is obtained. This process is simulated by lowering an initial temperature by slow stages until the system reaches to an equilibrium point, and no more changes occur. Each stage of the process consists of changing the configuration several times, until a thermal equilibrium is reached, and a new stage starts, with a lower temperature. The solution of the problem is the configuration obtained in the last stage. In the standard SA, the changes in the configuration are performed in the following way: A new configuration is built by a random displacement of the current one. If the new configuration is better, then it replaces the current one, and if not, it may replace the current one probabilistically. This probability of replacement is high in the beginning of the algorithm, and decreases in every stage. This procedure allows the system to move toward the best configuration. However, SA is not guaranteed to find the global optima, it is better than others algorithms escaping from local optima. The solution found by SA can be considered a "good enough" solution, but it is not guaranteed to be the best.

The most important part in a SA algorithm are: the chosen representation for solutions, the objective function to be minimized during the process and the mutation or configuration change operator.

2.4 Walsh Analysis and Spectrum

Walsh analysis of a function is a commonly used method to study the internal structure of binary functions [Vose (1998)]. Walsh analysis is equivalent to a Fourier expansion of a function in the binary search space $\{0, 1\}^n$. The Walsh expansion of a function associates a Walsh coefficient w_j to a binary vector $\mathbf{j}$ (*partition*). The function can be completely reconstructed from partitions [5] $\mathbf{j}$s and Walsh coefficients w_j. Continuing with the notation, we give the main steps to define Walsh expansion of a function.

[5] Hereafter, we will denote in boldface a partition indexed as a binary vector, $\mathbf{j}$, and in normal type, j, its corresponding integer value.

The Walsh basis function for a partition $\mathbf{j}$, $\psi_{\mathbf{j}} : \{0,1\}^n \rightarrow \mathbb{R}$ is defined as

$$\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{i=1}^{n} (-1)^{x_i j_i}. \tag{8}$$

where x_i and j_i are the components of binary vectors $\mathbf{x}$ and $\mathbf{j}$.

Walsh functions form a complete orthogonal set of basis functions [Goldberg (1989)]. Every function $f : \{0,1\}^n \rightarrow \mathbb{R}$ can be expanded as

$$f(\mathbf{x}) = \sum_{j=0}^{2^n-1} w_j \psi_{\mathbf{j}}(\mathbf{x}). \tag{9}$$

where

$$w_j = \frac{1}{2^n} \sum_{x=0}^{2^n-1} f(\mathbf{x}) \psi_{\mathbf{j}}(\mathbf{x}). \tag{10}$$

The Walsh expansion captures the internal structure of a function: if this function has dependencies among variables, then its Walsh coefficients for partitions involving non-dependent variables are zero. For example, let $f(x_1, x_2, x_3, x_4) = f_1(x_1, x_2) + f_2(x_3, x_4)$, then $w_{0111} = w_{1110} = w_{0110} = 0$ [Kargupta (1999)]. For further analysis and details on Walsh analysis see [Vose (1998)].

The *Spectrum* of a function [Hordijk (1998)] is a graphic representation of the most important partitions of the function which is obtained from its Walsh expansion. The *order* of a partition $\mathbf{j}$ is defined as the number of 1s on it. Note that in the FSP with binary representation, the order of a partition is equivalent to the number of selected features.

Using the definitions above, the Spectrum of a function is defined starting from its Walsh coefficients as follows: Let $\wp$ be the set of all partitions belonging to the search space $S = \{0,1\}^n$. Let $\wp_p$ be the set of partitions belonging to S with order p. A total energy for the function is defined as:

$$\sigma^2 = \sum_{j \in \wp} w_j^2. \tag{11}$$

The energy for the partitions with order p is

$$\beta_p^2 = \sum_{j \in \wp_p} w_j^2, \tag{12}$$

and their normalized energy

$$B_p = \frac{\beta_p^2}{\sigma^2}. \tag{13}$$

Vector $\mathbf{B} = \{B_1 \ldots B_n\}$ is the *Spectrum* associated to the Walsh expansion of $f(\cdot)$. It can be readily shown that $B_p \geq 0$ and $\sum_p B_p = 1$. In this paper we will use a modification of the Spectrum in order to perform a ranking of the best features for a given FSP problem.

3 Proposed Feature Selection methods

3.1 Simulated Annealing with SVMs

The first method we present in this paper for FS is based on a Simulated Annealing algorithm, which performs a search over the space $\boldsymbol{\sigma}$ of binary strings (See definition of FSP in Section 2.2). As was mentioned above, the main parts of a SA algorithm are: the chosen representation for solutions, the objective function to be minimized during the process and the mutation or configuration change operator. We describe them following:

Problem representation:
We encode every solution to the FS as a binary string $\boldsymbol{\sigma}$, where $\sigma_i \in \{0, 1\}$ defines the subset of selected features. The length of binary vector $\boldsymbol{\sigma}$ will be equal to the total number of features in the problem.

Mutation Operator:
In this paper we consider a classical *Random Flip Mutation* operator (RFM), where N_f bits are randomly selected and flipped to obtain a configuration in the neighborhood of the current one.

Objective Function:
We use the probability of error in test given by a SVM as the objective function to be minimized by the SA. The SA algorithm will look for configurations (feature subsets) which provide the least error probability in the test set.

The complete algorithm:
The SA we use has the following pseudo-code:

Pseudo-code of the SA algorithm.

```
$k = 0$;
$T = T_0$;
Initialize the current configuration $\boldsymbol{\sigma}$ at random;
Run the SVM $\rightarrow f(\boldsymbol{\sigma}) = P_e(test)$;
repeat
    for $j = 0$ to $M$
      $\boldsymbol{\sigma}^{mut} = \mathbf{mutate}(\boldsymbol{\sigma})$;
      Run the SVM $\rightarrow f(\boldsymbol{\sigma}^{mut}) = P_e(test)$;
         if$((f(\boldsymbol{\sigma}^{mut}) < f(\boldsymbol{\sigma}))$ OR $(random(0,1) < e^{(\frac{-a}{T})}))$ then
         $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{mut}$;
         endif
    endfor
$T = f_T(T_0, k)$;
$k = k + 1$;
until$(T < T_{min})$;
```

where k counts the number of iterations performed; T keeps the current temperature; T_0 is the initial temperature; T_{min} is the minimum temperature to be reached; $\boldsymbol{\sigma}$ stands for the current configuration, $\boldsymbol{\sigma}^{mut}$ stands for the new configuration after mutation operator is applied. $f(\boldsymbol{\sigma})$ represents the objective function (probability of error in test provided by the SVM in this case); M is the number of changes performed with a given temperature T; f_T is the freezer function; and a is a previously fixed constant. Parameter a and the initial temperature T_0 are calculated in order to the initial acceptance probability to be 0.8, which is the value usually used. The freezer function is defined as

$$f_T = \frac{T_0}{1+k}. \tag{14}$$

The minimum temperature T_{min} is calculated on the basis of the desired number of iterations as:

$$T_{min} = f_T(T_0, numIt). \tag{15}$$

The current configuration of the SA algorithm in the last iteration is interpreted as the solution of the problem $\boldsymbol{\sigma}^o$.

3.2 Walsh analysis for ranking of features

The second approach to the FS we present consists of obtaining a ranking of features analyzing the intrinsic structure of a SVM classifier by means of a procedure of Walsh analysis, as follows:

The Spectrum of a function defined in [Hordijk (1998)] (also summarized in Section 2.4) represents the distribution of energies among different orders (number of 1s) of the partitions which form the search space. If the analyzed function is the objective function of a FSP, the Spectrum might give a measure of what are the most important features.

The definition of Spectrum from the Walsh analysis of a function given in [Hordijk (1998)] calculates the energy associated to partitions of the same order. For example, in the search space $\{0,1\}^4$, partitions $\{1011\}, \{1110\}, \{0111\}$ contribute to the same component of the Spectrum (B_3), however they represent three different sets of features. Thus, the "classical" definition of Spectrum cannot be used for ranking the features according to their relevance. In order to solve this problem, we propose a slight modification of the definition of the Spectrum, in the following way.

Let ζ_i be the set of partitions $\mathbf{j}$ with a 1 in the position i_{th}. In the example above partitions $\{1011\}$ and $\{1110\}$ belong to ζ_1 (they also belong to ζ_3) whereas partition $\{0111\}$ does not belong to ζ_1 (but it belongs to ζ_3). Therefore, a modified Spectrum, named the *prime Spectrum*, can be defined as follows:

$$B_i' = \frac{\sum\limits_{j \in \zeta_i} w_j^2}{\sum\limits_{\forall j} o(\mathbf{j}) \cdot w_j^2}, \tag{16}$$

where $o(\mathbf{j})$ is the order of partition $\mathbf{j}$.

Prime Spectrum fulfills $B_i' \geq 0$ and $\sum_i B_i' = 1$. In fact, it can be interpreted as the associated energy to every feature in the binary search space and thus, features with large values of B_i' are more relevant than features with small values of B_i'. Consequently, we propose using vector $\mathbf{B}'$ to perform ranking of features in the FSP.

Vector $\mathbf{B}'$ depends on the fitness function selected for the FSP through the values of its Walsh coefficients. Note that in large search spaces, the calculation of the Walsh expansion can be computationally infeasible, and estimation methods as the one proposed in [Hordijk (1998)] should be used.

In this paper we propose the calculation of the vector $\mathbf{B}'$ using as objective function the test error provided by a SVM in the test set. This way, vector

$\mathbf{B}'$ provides a measure of the intrinsic structure of the SVM when different features are removed from the data. The calculation of B' in pseudo-code is as follows:

Pseudo-code of the calculation of B' spectrum.

```
**for** $j = 0$ **to** $2^n - 1$
  $j \rightarrow \boldsymbol{\sigma}$ (Decimal to binary step)
      Run the SVM $\rightarrow f(\boldsymbol{\sigma}) = P_e(test)$;
       Calculate $w_j = \frac{1}{2^n} \sum_{x=0}^{2^n-1} f(\boldsymbol{\sigma})\psi_{\mathbf{j}}(\mathbf{x})$
**endfor**
**for** $i = 1$ **to** $n$
  $B'_i = \frac{\sum_{j \in \zeta_i} w_j^2}{\sum_{\forall j} o(\mathbf{j}) \cdot w_j^2}$
**endfor**
```

4 Experiments and results

4.1 Test data and input variables

In this Section we show the main characteristics of the data and variables that will be used to test the approaches presented in this paper. We have used the sample of firms also used by [Sanchis (2003)]. This data sample consists of Spanish non-life insurance firms data five years prior to failure. The firms were in operation or went bankrupt between 1983 and 1994. In each period, 72 firms (36 failed and 36 non-failed) are selected. As a control measure, a failed firm is matched with a non failed one in terms of premiums volume. In our study we have used data one year prior to the firms declare bankruptcy, due to this, it has to be noted that the prediction of the insolvency achieved by our method will be one year in advance.

In this research, each firm is described by 21 financial ratios (features) that have come from a detailed analysis of the variables and previous bankruptcy studies for non-life insurance. We have to pay particular attention to the fact that financial characteristics of insurance companies require general financial ratios as well as those that are specifically proposed for evaluating insolvency of insurance sector. Table 1 shows the 21 ratios which describe the firms. Ratios R1, R2, R3, R4 and R9 are general financial ratios and the rest are specific for insurance sector.

An important variable in this sector is the *Reinsurance.* It is possible to select ratios by means of distinguishing between ratios involving *earned premiums* and ratios involving *earned premiums net of reinsurance.* Thus we can separate the following couples of ratios: R5-R6, R7-R8, R11-R12, R13-R14, R15-R16, R17-R18 and R20-R21 (see Table 1). The majority of firms in our study have not reinsurance, so it is expected that the results for both sets of ratios to be similar, since general financial ratios are common for both sets. On the other hand, ratios 15 and 16 have been removed in our study, due to most of the firms have not “other income”; this reduce the number of used ratios to 19.

4.2 Experiments

Using the data defined above, we test the performance of the two approaches presented in this paper, by means of several experiments and comparisons: The performance of the SA with SVM algorithm described in Section 3 is tested in the selection of the best of features from the 19 initial ones. First, we choose the tunable SVM parameters C and γ (see 2.1) by means of a cross-validation scheme, following [Bishop (1995)].

The set of 72 firms are split in four sets, every set is formed by 18 firms (9 failed and 9 non-failed). The cross-validation procedure consists of training the SVM with three of the four sets, and validating the result with the remaining set. This process is repeated for each of the four possible combinations. The final result is the average of the four obtained results. Figure 3 shows an example of this process. Once C and γ are fixed, the SA described in Section 3 is run, using the test error of the SVM as the objective function value in the algorithm. Note that a different subset of features, indicated by $\boldsymbol{\sigma}$ is used in each SA step.

In this paper we compare the features obtained by this methods with the features achieved by a *Rough Set algorithm.* Rough Set theory was firstly developed by Pawlak [Pawlak (1991)] as a mathematical tool to deal with the uncertainty or vagueness inherent in a decision making process.

Briefly, the Rough Set approach works by discovering dependencies between attributes in an information table, and reducing the set of attributes by removing those that are not essential to characterize knowledge. A *reduct* is defined as the minimal subset of attributes which provides the same quality of classification as the set of all attributes. A reduced information table may provide decision rules of the form ”if conditions then decisions”. These rules specify what decisions (actions) should be undertaken when some conditions are satisfied, and can be used to assign new objects to a decision class by matching the condition part of one of the decision rule to the description of the object. We have performed the Rough Set analysis using the Rough Set system *ROSE*. For a more detailed description of the Rough Set theory and the *ROSE* software, see [Pawlak (1991)] and [Predki (1998)], [Predki (1999)].

The performance of the ranking provided by the Walsh analysis (spectrum $\mathbf{B}'$) is tested following. The function involved in the calculation of $\mathbf{B}'$ is given by the probability of error of a SVM for all possible binary strings $\boldsymbol{\sigma}$. In order to perform a better testing of the ranking provided by means of the prime spectrum $\mathbf{B}'$, we consider two groups of 12 features each: In a first group we include ratios G_1=[R1, R2, R3, R4, R5, R7, R9, R10, R11, R13, R17, R19, R20]. In the second group we include ratios G_2=[R1, R2, R3, R4, R6, R8, R9, R10, R12, R14, R18, R19, R21]. The difference between both groups is that the second one (G_2) contains the general ratios and the specific ones including the reinsurance variable (see Table 1). Recall that due to most of the firms considered have not reinsurance, both groups of ratios are similar, and therefore it is expected that the results obtained for both sets of ratios, G_1 and G_2, to be similar as well.

4.3 Results

4.3.1 Feature selection through Simulated Annealing and SVM

The best sets of features obtained with the SA algorithm are formed by ratios {R1, R9 , R13}, and {R3, R9, R19} both of them with a probability of error in test $P_e = 0.23$. The SA algorithm reached one of this sets in all simulations run. No other combination of features provide a better value of probability of error.

The reducts obtained using a Rough Set approach for all training sets contain from 4 to 6 attributes (ratios). The ratios that have the highest frequency of occurrence (more than 40%) in reducts are R1, R3, R4, R9, R17, R18 and R19. Note that there are several coincidences among variables provided by SA with SVM and Rough Set approaches. This would indicate that these variables are highly discriminatory between solvent and insolvent firms in our sample. Note also that the reducts obtained using the SA algorithm contain fewer ratios than the reducts provided by the Rough Set, ensuring the same quality than using the whole set of ratios. These results show that the SA algorithm provides a good feature selection, comparable with the obtained by existing methods, such as the Rough Set algorithm.

4.3.2 Ranking of features

It is important to note that feature selection methods in general provides very specific results, without information about the possible relationships among features. These relationships, which may be very important in some applications, can be obtained by performing a Ranking of features. A simple and commonly used method to tackle the ranking of variables is the so called *Fisher ranking*, where features are linearly classified following the value of a *Fisher score*. In this section we compare the proposed alternative ranking given by prime spectrum with the Fisher ranking.

Figure 4 shows the ranking obtained by means of the prime spectrum $\mathbf{B}'$ for sets of ratios G_1 (a) and G_2 (b). Recall that these two sets of ratios have similar properties due to most of the firms have not reinsurance, so the rankings in both sets should be also similar. It is easy to see that the ranking $\mathbf{B}'$ for the set of features G_1 highlights ratios R1, R9, R3 and R19, and also R2 and R4 with a slightly fewer value of $\mathbf{B}'$. On the other hand, ranking for set G_2 highlights ratios R1, R3 and R9. R19 and R4 are the following ratios in the ranking, but with a value of B' smaller.

Figure 5 shows a ranking of Fisher for the same sets G_1 (a) and G_2 (b), for comparison with our method. This ranking gives much importance to ratio R1 whereas other important ratios such as R3 or R19 for example are completely ignored. This indicates that our ranking algorithm performs better than Fisher ranking for this problem.

4.4 Analysis of the results and Discussion

The results obtained show the importance of the feature selection procedure used. In our sample, in spite of the initial large information system of 19 financial attributes, we could consider just three characteristics in order to check the solvency of a firm (solution provided by the SA algorithm). We have found two sets of three ratios each (five different ratios, since R1 is common to both sets), which seem essential to analyze the solvency of a non-life insurance company {R1, R3, R9, R13, R19}. They indicate that these variables are highly discriminatory between solvent and insolvent firms in our sample, and that we should have to take into account them if we want to evaluate the solvency of non life insurance companies.

The analysis of the results obtained by the ranking $\mathbf{B}'$ applied to sets G_1 and G_2 provides very interesting conclusions: First, since the majority of firms in our sample have not reinsurance the ranking of features provided by $\mathbf{B}'$ is similar in both sets G_1 and G_2, as expected. Second, the higher value of $\mathbf{B}'$ are for Ratios R1, R9, R3 and R19 (order of importance). Note the coincidence with the ratios provided by the SA algorithm. Note also that SA algorithm only choose these variables among the 19 possible in our sample, but B' in addition rank them in importance: R1 the more important, R9, R3 and finally R19.

A brief resume of the financial meaning of these ratios if given following:

- R1- One of the most important questions in order to assure the proper functioning of any firm is the need of having sufficient liquidity. But in the case of an insurance firm, the lack of liquidity should not arise due to ”productive activity inversion” which implies that premiums are paid in before claims occur. If an insurance firm can not pay the incurred claims, the clients and public in general could lose faith in that company. On the other hand, this ratio is a measure of financial equilibrium if it is positive as it implies that the working capital is also positive.
- R3- This ratio indicates that to obtain enough financial incomes is a critical issue because nowadays these incomes are the main source of benefit for an insurance company.
- R9- This ratio shows what proportion of the total liabilities represents the shareholders' funds (capital and reserves). This confirms the importance, from a solvency viewpoint, of the adequacy of these funds because these resources could be called on to meet the future claims obligations of the insurer due to some eventualities.
- R19- This ratio shows the importance of a proper reinsurance to evaluate the solvency in insurance firms.

5 Conclusions and further research

In this paper we have presented two feature selection methods based on Support Vector Machines (SVMs), and we have applied them to the prediction of insolvency in non-life insurance companies. We have chosen SVM-based methods due to SVM is considered a fast and robust classifier capable of obtaining accurate classifications in high dimensional problems with very few samples.

First we have presented a Simulated Annealing (SA) algorithm for feature selection and second we have considered an algorithm for ranking features using the prime Spectrum obtained from a Walsh analysis of a function which involves a SVM. The SA algorithm presented in this work can be useful in problems where the number of features is large and an exhaustive analysis of all combinations of features is unfeasible. In problems where the number of features allows an exact calculation of the SVM Walsh analysis, the prime spectrum gives a powerful tool for analyzing relations among features, ranking them in importance for the problem.

The presented feature selection algorithms have been used in order to estimate which financial ratios are the most adequate in the prediction of the insolvency of 72 Spanish non-life insurance companies. We have obtained good results, consisting of low probability of error given by the SVM discarding noisy and redundant ratios.

This work opens new lines of research, such as the application of the SVM to imputation problems (estimate of missed data or ratios) for having a more completed sample set for training and test a classifier. This would be specially useful in cases where is costly to obtain new data as in economics applications.

References

[Altman] Altman E. I., Marco, G. and Varetto, F. 1994. Corporate distress diagnosis: comparisons using discriminant analysis and neural networks (the italian experience), *Jounal of Banking and Finance*, 18:505-529.

[Ambrose (1994)] Ambrose, J. M., Carol, A. M. 1994. Using best ratings in life insurer insolvency prediction, *Journal of Risk and Insurance*, 61:317-327.

[Banister (1997)] Bannister, J. 1997. *Insurance solvency analysis*, LLP limited, second edition.

[Barniv (1990)] Barniv, R. 1990. Accounting procedures, Market data, cash-flow figures and insolvency classification: the case of the insurance industry, *The Accounting Review*, 65(3):578-604.

[Bishop (1995)] Bishop, C. M.: (1995), *Neural Networks for pattern recognition*, Oxford University press.

[Bradley (1999)] Bradley, P. S., Fayyad, U. M. and Mangasarian, O. L. 1999. Mathematical programming for data mining: formulations and challenges, *INFORMS Journal on Computing*, 11(3):217-238.

[Burgues (1998)] Burges, C. J. 1998. A tutorial on Support Vector Machines for pattern recognition, *Knowledge Discovery and Data Mining*, 2(2):121-167.

[Chen (1999)] Chen, H., Y., Chen, T. C., Min, D., Fischer, G. and Wu, Y. M. 1999. Prediction of tacrolimus blood levels by using the neural network with genetic algorithm in liver transplantation patients, *Therapeutic Drug Monitoring*, 21(1):50–56.

[Dimitras (1996)] Dimitras, A. I., Zanakis, S. H., and Zopounidis, C. 1996. A survey of businers failures with an emphasis on prediction methods and industrial applications, *European Journal of Operational Research*, 90(3):487-513.

[Dimitras (1998)] Dimitras, A. I., Slowinski, R., Susmaga, R. and Zopounidis, C. 1998. Business failure using rough set, *European Journal of Operational Research*, 114(2):263-280.

[Goldberg (1989)] Goldberg, D. E.: (1989), *Genetic algorithms in search, optimization and machine learning*, Reading, MA: Addison-Wesley.

[González (2002)] González, J., Rojas, I., Pomares, H., Salmerón, M. and Merelo, J. J. 2002. Web newspaper layout optimization using Simulated Annealing, *IEEE Trans. Systems, Man and Cybernetics*, 32(5):686-691.

[Hordijk (1998)] Hordijk, W. and Stadler, P. F. 1998. Amplitude spectra of fitness landscapes, *J. Complex Systems*, 1(1):39-66.

[John (1994)] John, G., Kohavi, R. and Pfleger, K. 1994. Irrelevant features and the subset selection problem, In *Proc. of the 11th International Conference on Machine Learning.* Aug. 1994, pp. 121–129, Morgan Kaufmann.

[Kargupta (1999)] Kargupta, H. 1999. A striking property of genetic code-like transformations, Technical Report EECS-99-04, EECS School, Washington State University, Washington.

[Kirpatrick (1983)] Kirpatrick, S., Gerlatt, C. D. and Vecchi, M. P. 1983. Optimization by simulated annealing, *Science*, 220:671-680.

[Kohavi (1997)] Kohavi, R. and John, G. H. 1997. Wrappers for features subset selection, *Int. J. Digit. Libr.*, 1:108–121.

[Li (2000)] Li, J. and Tsang, E. P. K. 2000. Reducing failures in investments recommendations using Genetic Programming, In *Proc. of the 6th International Conference on Computing in Economics.* Barcelona, Jul. 2000.

[O'leary (1998)] O'Leary, D. E. 1998. Using neural networks to predict corporate fealure, *International Jounal of Intelligent Systems in Accounting Finance and Management*, 7:187-197.

[Patuwo (1993)] Patuwo, E., Wu, M. Y., and Hung, M. S. 1993. Two group classification using neural networks, *Decision Sciences*, 23:899-916.

[Pawlak (1991)] Pawlak, Z. 1991. *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, London.

[Predki (1998)] Predki, B., Slowinski, R., Stefanowski, J., Susmaga, R. and Wilk, S. 1998. ROSE: Software Implementation of the Rough Set Theory, *Rough Sets and Current Trends in Computing*, Lecture Notes in Artificial Intelligence, vol. 1424. Springer-Verlag, p. 605-608.

[Predki (1999)] Predki, B. and Wilk, S. 1999. Rough Set based data exploration using ROSE system. *Foundations of Intelligent Systems*, Lecture Notes in Artificial Intelligence, vol. 1609, Springer-Verlag; p. 172-180.

[Salcedo (2002)] Salcedo-Sanz, S., DePrado-Cumplido, M., Pérez-Cruz, F. and Bousoño-Calzón, C. 2002. Feature selection via genetic optimization, In *Proc. of the International Conference on Artificial Neural Networks, ICANN2002*, Madrid, Spain., Aug 2002, pp. 547–552.

[Salcedo (2003)] Salcedo-Sanz, S., Fernández-Villacañas, J. L., Segovia-Vargas, M. J. and Bousoño-Calzón, C. 2002. Genetic programming for the prediction of insolvency in non-life insurance companies, *Computers & Operations Research*, in press.

[Sanchis (2003)] Sanchis, A., Gil, J. A., and Heras, A. 2003. El análisis discriminante en la previsión de la insolvencia en la empresa de seguros no-vida, *Revista Española de Financiación y Contabilidad*, 116.

[Segovia (2003)] Segovia-Vargas, M. J., Salcedo-Sanz, S. and Bousoño-Calzón, C. 2003. Prediction of insolvency in non-life insurance companies using Support Vector Machines and genetic algorithms. In proc. *X SIGEF Congress in Emergent Solutions for the Information and Knowledge Economy*, León, Spain, October 2003.

[Serrano (1996)] Serrano-Cinca, C. 1996. Shelf organizing neural networks for financial diagnosis, *Decision Support Systems*, 17:227-238.

[Schölkopf (2002)] B. Schölkopf, B. and A.J. Smola A. J. 2002. *Learning with Kernels*, MIT press, Cambridge, MA.

[Tam (1991)] Tam, K. Y. 1991. Neural network models and the prediction of bankruptcy, *Omega*, 19(5):429-445.

[Tam (1992)] Tam, K. Y., and Kiang, M. Y.: (1992), Managerial applications of neural networks: the case of bank failure predictions, *Management Science*, 38(7):926-947.

[Vafaie (1992)] Vafaie, H., and De Jong, K. A.: (1992), Genetic Algorithms as a Tool for Features Selection in Machine Learning, *Proc. of the 4th Intl. Conf. on Tools with Artificial Systems*, IEEE computer society press, Arlintong, VA, 200-204.

[Vapnik (1998)] Vapnik, V. N.: (1998), *Statistical Learning Theory*, John Wiley & Sons, New York.

[Vose (1998)] Vose M. D. and Wright, A. H.: (1998), The simple genetic algorithm and the Walsh transform: Part I, theory, *Evol. Comput.*, 6(3):253-273.

[Wang (1997)] Wang, G. and Ansari, N.: (1997), Optimal broadcast scheduling in packet radio networks using mean field annealing, *IEEE J. Select. Areas Commun.*, 15(2):250-259.

[Weston (2000)] Weston, H., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V.: (2000), Feature Selection for SVMs, Advances in NIPS 12, MIT Press, 526-532.

[Wilson (1994)] Wilson, R. L., and Sharda, R.: (1994), Bankruptcy prediction using neural networks, *Decision Support Systems*, 11:545-557.

[Zopounidis (1998)] Zopounidis. C., and Dimitras, A.: (1998), *Multicriteria decision aid methods for the prediction of business failure*, Kluwer.

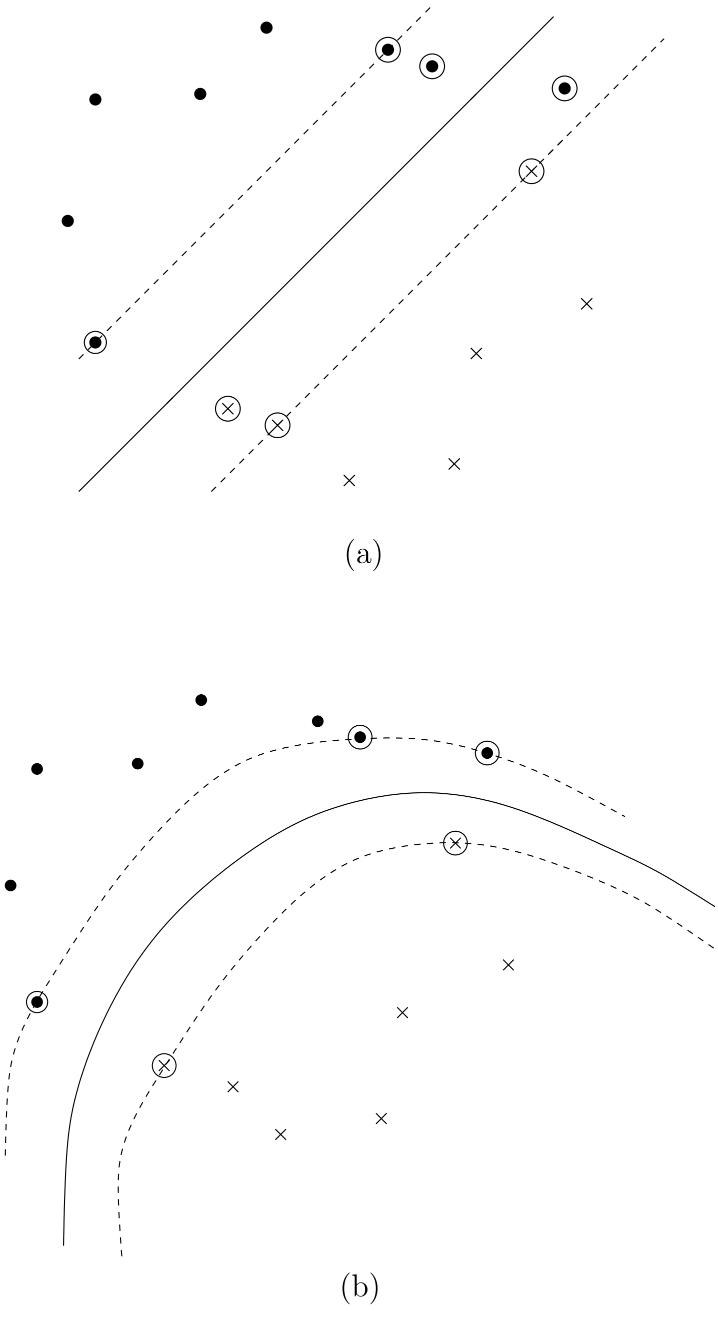
[Zopounidis (1999)] Zopounidis, C.: (1999), Multicriteria decision aid in financial management, *European Journal of Operational Research*, 119(2):404-415.

List of Tables

List of Figures

Table 1
Definition of the Ratios

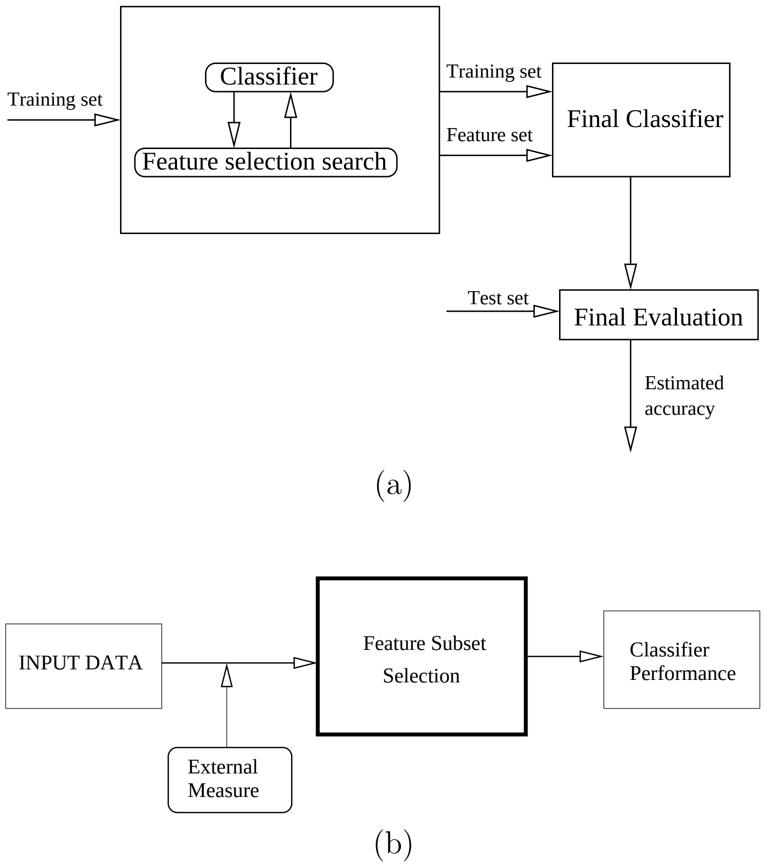
Ratio	Definition
R1	$\frac{Working\ Capital}{Total\ Assets}$
R2	$\frac{Earnings\ Before\ Taxes\ (EBT)}{(Capital+Reserves)}$
R3	$\frac{Investment\ Income}{Investments}$
R4	$\frac{EBT+Reserves\ for\ Depreciation+(Extraordinary\ Income-Extraordinary\ charges)}{Total\ Liabilities}$
R5	$\frac{Earned\ Premiums}{(Capital+Reserves)}$
R6	$\frac{Earned\ Premiums\ net\ of\ Reinsurance}{(Capital+Reseves)}$
R7	$\frac{Earned\ Premiums}{(Capital+Reserves+Technical\ Provisions)}$
R8	$\frac{Earned\ premiums\ Net\ of\ Resinsurance}{(Capital+Reserves+Technical\ Provisions)}$
R9	$\frac{(Capital+Reserves)}{TotalLiabilities}$
R10	$\frac{Technical\ Provisions}{(Capital+Reserves)}$
R11	$\frac{Claims\ Incurred}{(Capital+Reserves)}$
R12	$\frac{Claims\ Incurred\ Net\ of\ Reinsurance}{(Capital+Reserves)}$
R13	$\frac{Claims\ Incurred}{(Capital+Reserves+TechnicalProvisions)}$
R14	$\frac{Claims\ Incurred\ Net\ of\ Reinsurance}{(Capital+Reserves+Technical\ Provisions)}$
R15	$\frac{Claims\ Incurred}{Earned\ Premiums} + \frac{Other\ Charges\ and\ Commisions}{Other\ Income}$
R16	$\frac{Claims\ Incurred\ Net\ of\ Reinsurance}{Earned\ Premiums\ Net\ of\ Reinsurance} + \frac{Other\ Charges\ and\ Commissions}{Other\ income}$
R17	$\frac{Claims\ Incurred\ +Other\ Charges\ and\ Commisions}{Earned\ Premiums}$
R18	$\frac{Claims\ Incurred\ Net\ of\ Reinsurance+Other\ Charges\ and\ Commisions}{Earned\ Premiums\ Net\ of\ Reinsurance}$
R19	$\frac{Technical\ provisions\ of\ Assigned\ reinsurance}{Technical\ Provisions}$
R20	$\frac{Claims\ Incurred}{Earned\ Premiums}$
R21	$\frac{Claims\ Incurred\ Net\ of\ Reinsurance}{Earned\ Premiums\ net\ of\ Reinsurance}$



(a)

(b)

Figure 1. (a) Illustration of linear SVM for solving a two class classification problem. Note that some samples of dot class are misclassified; (b) Illustration of the same problem using a nonlinear SVM (Radial Basis Function Kernel). Now the boundary obtained is capable of completely separing both sample sets.



(a)

(b)

Figure 2. (a) Outline of a Wrapper method; (b) Outline of a Filter method.

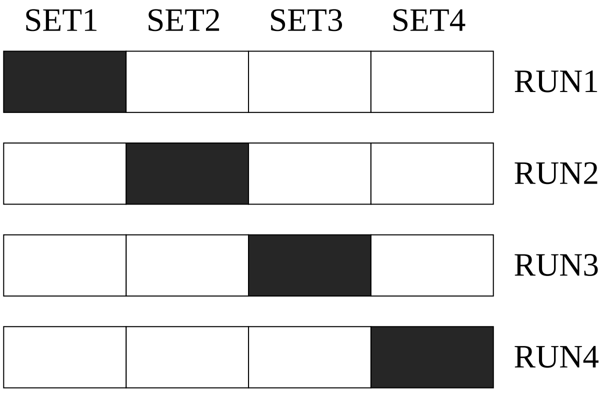
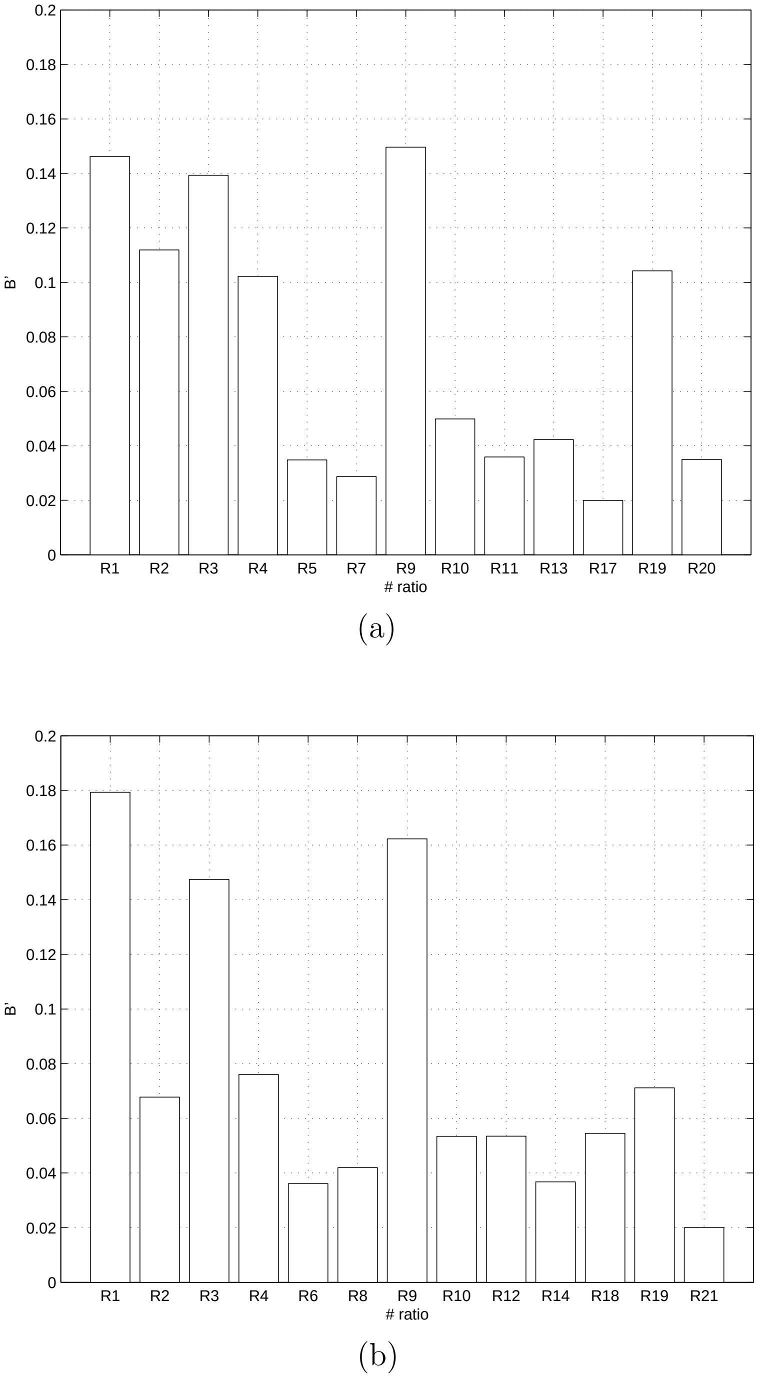


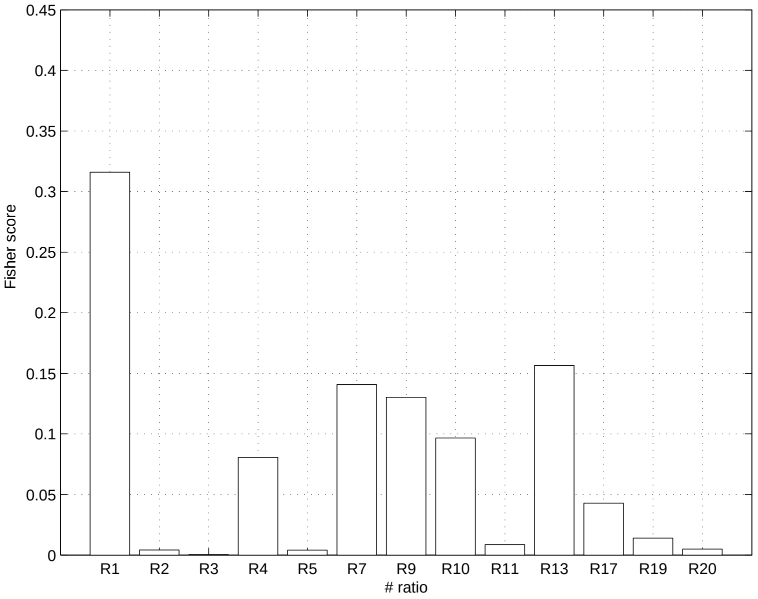
Figure 3. Schematic illustration of the crossvalidation procedure. The NN is trained four times, each time using a version of the data set in which one of the segments (shaded) is omitted. Each trained network is then tested on the data from the set wich was omitted during training. The final result is the average over the four sets.



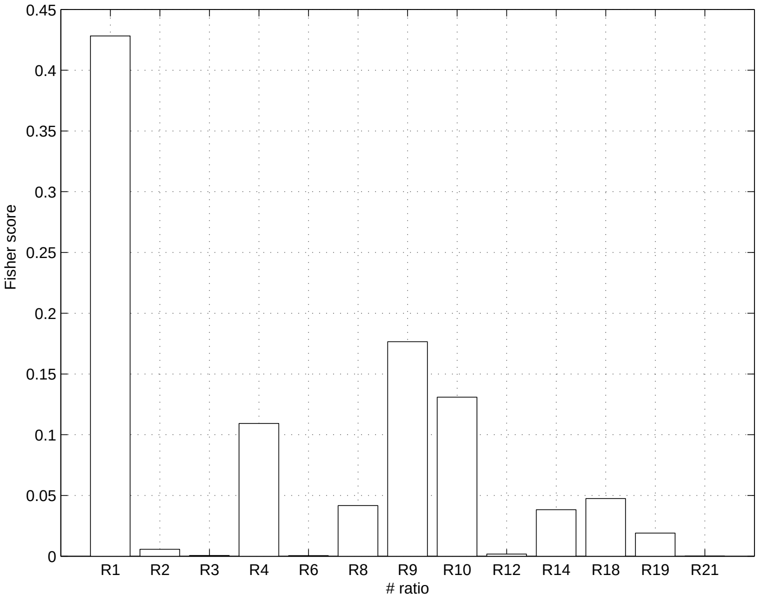
(a)

(b)

Figure 4. (a) Ranking provided by the prime Spectrum $\mathbf{B}'$ using ratios in set G_1; (b) Ranking provided by the prime Spectrum $\mathbf{B}'$ using ratios in set G_2.



(a)



(b)

Figure 5. (a) Ranking provided by the Fisher score using ratios in set G_1; (b) Ranking provided by the Fisher score using ratios in set G_2.