Bankruptcy Prediction with Least Squares Support Vector Machine Classifiers

Tony Van Gestel*, Bart Baesens*, Johan Suykens*, Marcelo Espinoza*,
Dirk-Emma Baestaens¶, Jan Vanthienen*, Bart De Moor*
* Department of Electrical Engineering ESAT-SCD-SISTA, Katholieke Universiteit Leuven
* Department of Applied Economic Sciences, Katholieke Universiteit Leuven
¶ Financial Markets, Fortis Bank Brussels
Emails: tony.vangestel@esat.kuleuven.ac.be, bart.baesens@econ.kuleuven.ac.be

Abstract: *Classification algorithms like linear discriminant analysis and logistic regression are popular linear techniques for modelling and predicting corporate distress. These techniques aim at finding an optimal linear combination of explanatory input variables, such as, e.g., solvency and liquidity ratios, in order to analyse, model and predict corporate default risk. Recently, performant kernel based nonlinear classification techniques, like Support Vector Machines, Least Squares Support Vector Machines and kernel Fisher Discriminant Analysis, have been developed. Basically, these methods map the inputs first in a nonlinear way to a high dimensional kernel-induced feature space, in which a linear classifier is constructed in the second step. Practical expressions are obtained in the so-called dual space by application of Mercer's theorem. In this paper, we explain the relations between linear and nonlinear kernel based classification and illustrate their performance on predicting bankruptcy of mid-cap firms in Belgium and the Netherlands.*

Keywords: *Bankruptcy prediction, Least Squares Support Vector Machine Classifiers, Kernel Fisher Discriminant Analysis*

1 INTRODUCTION

The subject of bankruptcy prediction has been recognized as an important research area in the field of financial accounting. Financial failure of a firm occurs when the firm has chronic and serious losses or when the firm becomes insolvent with liabilities disproportionate to the assets. Causes and symptoms that are recognized to cause financial distress include poor management, autocratic leaders and difficulties to operate successfully in the market. The financial failure of a firm causes substantial losses to both the business community and society as a whole. Therefore, bankruptcy prediction is of critical importance to creditors, shareholders and employees, since it could provide timely warnings to management, investors, employees, shareholders and other interested parties who wish to reduce their losses. From a management perspective, financial failure forecasting tools allow to take timely strategic actions to avoid financial distress. E.g., for banks, efficient and automated credit rating tools provide facilities to early detect clients that will default their obligation. In this way, these tools may improve the efficiency of one of their core activities: commercial credit assignment.

The common assumption underlying bankruptcy prediction is that key macro-economic indicators (e.g. inflation, interest rates) together with firm characteristics (e.g. competition, management quality, market share) are appropriately reflected in the firm's financial statements. The future financial status of the firm can then be predicted by using data originating from these statements and advanced prediction models.

Many techniques have been used to predict bankruptcy. Early univariate approaches used ratio analysis to predict potential financial distress, comparing historical ratios of failed against non-failed firms. Since there is no single cause for corporate failure, multivariate approaches have been used to combine various (potentially conflicting) measures of profitability or financial soundness of a firm [1, 2, 7]. The most popular multivariate bankruptcy prediction models use linear multiple discriminant analysis or logistic regression [1, 23]. Linear multiple discriminant approaches, like Altman's Z-Scores, attempt to identify the most efficient hyperplane to linearly separate between successful and non-successful firms. However, these techniques typically rely on a linear separability assumption, as well as normality assumptions. Both empirical and theoretical evidence has shown that non-linear models may provide additional performance gains. From a theoretical viewpoint, it is reasonable to assume that an increase of the earnings/total assets ratio from -0.1 to 0.1 may have a larger impact on the probability of failure than an increase from 1.0 to 1.2 [3]. Furthermore, strong interaction effects may also exist since e.g. the impact of a negative cash flow on the probability of bankruptcy is typically higher when the firm also has large liabilities [3].

Neural networks have been suggested as interesting bankruptcy prediction tools because of their universal approximation property and nonlinear modelling capacities [8]. Many empirical studies report significant performance increases of neural networks when compared to discriminant analysis or logistic regression, see, e.g., [2, 3, 24, 29, 36]. Despite their success, a number of problems still remain when using neural networks like the non-convex training problem with multiple local minima and the choice of the number of hidden neurons. Recently, Support Vector Machines (SVMs) have been suggested in the literature as

promising nonlinear classification techniques [9, 26, 28, 34] that allows to construct a nonlinear classifier as the solution to a quadratic programming problem, which is convex and guarantees a global minimum of the error function.

In this paper, we will use Least Squares Support Vector Machine (LS-SVM) classifiers for bankruptcy prediction [28]. LS-SVMs are a modified version of SVMs resulting into a set of linear equations instead of a QP problem [27, 28]. LS-SVMs first map the data to a higher dimensional feature space in which the discriminant function is constructed. The feature space formulation is related to ridge regression, Fisher Discriminant Analysis and regularized discriminant analysis [6, 9, 18, 22, 26, 33] and yields comparable classification performances on 20 UCI benchmark datasets as the SVM formulation [31], the latter being computationally more complex for training. Comparisons will be made in this paper with linear discriminant analysis (LDA) and logistic regression (LOGIT) [8, 12, 25]. The performance will be quantified by using the classification accuracy, the confusion matrix and the area under the receiver operating characteristic curve (AUROC). A leave-one-out cross-validation experimental setup is used to compute these performance measures [13]. In a second step, a backward stepwise input selection experiment is carried out in order to reduce the number of explanatory variables of the trained bankruptcy prediction models and identify the most relevant inputs for predicting financial failure.

This paper is organised as follows. The basic steps from linear Fisher Discriminant Analysis to nonlinear kernel-based learning with Least Squares Support Vector Machines are discussed in Sections 2 and 3. The empirical results are reported and commented in Section 4.

2 LINEAR DISCRIMINANT ANALYSIS

Given a number n_x of explanatory variables or inputs $\boldsymbol{x} \in \mathbb{R}^{n_x}$ of a firm, the problem we are concerned with is to predict whether this firm will default its obligations ($y = -1$) or not ($y = +1$). This problem corresponds to a binary classification problem with class $\mathcal{C}_-(y = -1)$ denoting the class of (future) bankrupt firms and class $\mathcal{C}_+(y = +1)$ the class of solvent firms. Let $\mathrm{p}(\boldsymbol{x}|y)$ denote the class probability density of observing the inputs $\boldsymbol{x}$ given the class label y and let $\pi_+ = \mathrm{P}(y = +1)$, $\pi_- = \mathrm{P}(y = -1)$ denote the prior class probabilities, then the Bayesian decision rule to predict $\hat{y}$ is as follows

$$\begin{aligned} \hat{y} &= \mathrm{sign}[\mathrm{P}(y = +1|\boldsymbol{x}) - \mathrm{P}(y = -1|\boldsymbol{x})] \\ \hat{y} &= \mathrm{sign}[\mathrm{p}(\boldsymbol{x}|y = +1)\pi_+ - \mathrm{p}(\boldsymbol{x}|y = -1)\pi_-], \end{aligned} \tag{1}$$

where the second expression is obtained by applying Bayes' formula $\mathrm{p}(y|\boldsymbol{x}) = \mathrm{p}(\boldsymbol{x}|y)\mathrm{p}(y)/\mathrm{p}(\boldsymbol{x})$ and omitting the normalizing constant $\mathrm{p}(\boldsymbol{x})$. This Bayesian decision rule is known to yield optimal performance as it minimizes the risk of misclassification on each instance $\boldsymbol{x}$. In the case of Gaussian class densities with means $\boldsymbol{m}_-$, $\boldsymbol{m}_+$ and equal covariance matrix $\boldsymbol{\Sigma}_x$, the Bayesian decision rule becomes [8, 12, 25]

$$\hat{y} = \mathrm{sign}[\boldsymbol{w}^T\boldsymbol{x} + b] = \mathrm{sign}[z], \tag{2}$$

with latent variable $z = \boldsymbol{w}^T\boldsymbol{x} + b$ and where $\boldsymbol{w} = \boldsymbol{\Sigma}_x^{-1}(\boldsymbol{m}_+ - \boldsymbol{m}_-)$ and $b = \boldsymbol{w}^T(\boldsymbol{m}_+ + \boldsymbol{m}_-)/2 + \log(\pi_+/\pi_-)$. This result motivates the use of linear discriminant analysis.

As the class densities and the parameters $\boldsymbol{m}_+$, $\boldsymbol{m}_-$, $\boldsymbol{\Sigma}_x$ are typically unknown in practice, one has to estimate the decision rules (1)-(2) from given training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_\mathcal{D}}$. A common way to estimate the linear discriminant (2) is by solving

$$\min_{\boldsymbol{w},b} \frac{1}{2}\sum_{i=1}^{n_\mathcal{D}}(y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b))^2. \tag{3}$$

The solution $(\hat{\boldsymbol{w}}, \hat{b})$ follows from a linear set of equations of dimension $(n_x + 1) \times (n_x + 1)$. It is straightforward to show that the solution $\hat{\boldsymbol{w}}$ corresponds to the same discriminant function as found by Fisher Discriminant Analysis [16], which has been used in the pioneering paper of Altman [1]. More precisely, Fisher related the maximization of the Rayleigh quotient to a regression approach with targets $(-n_\mathcal{D}/n_\mathcal{D}^-, n_\mathcal{D}/n_\mathcal{D}^+)$, with $n_\mathcal{D}^+$ and $n_\mathcal{D}^-$ the number of positive and negative training instances. The solution only differs in the choice of the bias term. In empirical linear discriminant analysis, one plugs in the pooled sample covariance matrix and the sample means into the decision rule (2). It can be shown that this empirical LDA formulation yields again the same discriminant, but a different bias term than the other classifiers.

In this paper, we will use the least squares formulation with binary targets $(-1, +1)$. This formulation is often used in training neural network classifiers [8, 28]. It also corresponds to an asymptotically optimal least squares approximation to the discriminant function (1) [12]. In the reverse way, one may compute the posterior probabilities as $\mathrm{P}(y = +1|\boldsymbol{x}) \simeq \frac{1}{2}(1 + (\boldsymbol{w}^T\boldsymbol{x} + b))$ and $\mathrm{P}(y = -1|\boldsymbol{x}) \simeq \frac{1}{2}(1 - (\boldsymbol{w}^T\boldsymbol{x} + b))$. In practice, a softmax interpretation is often used to guarantee that the estimated probabilities are true probabilities between 0 and 1 and that sum up to 1:

$$\begin{aligned} \mathrm{P}(y = +1|\boldsymbol{x}) &= 1/(1 + \exp(-\boldsymbol{w}^T\boldsymbol{x} - b)) \\ \mathrm{P}(y = -1|\boldsymbol{x}) &= 1/(1 + \exp(\boldsymbol{w}^T\boldsymbol{x} + b)). \end{aligned}$$

Instead of calculating $\boldsymbol{w}$ and b from least squares and approximating the posterior probability, one may also optimize directly the probability of observing the given training data, as is done in logistic regression. Logistic regression is a popular technique for bankruptcy prediction [23]. It is known to be more robust against violations of the multivariate normality assumption. On the other hand, it is less optimal when the Gaussian assumptions do hold [25] and is computationally more expensive as there exists no closed form solution. The solution is often computed by means of an iteratively reweighted least squares algorithm.

3 LEAST SQUARES SUPPORT VECTOR MACHINES

3.1 Primal Formulation

In Support Vector Machines (SVMs) and Least Squares Support Vector Machines (LS-SVMs) for nonlinear classification, the inputs $\boldsymbol{x}$ are first preprocessed in a nonlinear way by means of the nonlinear mapping $\boldsymbol{\varphi} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_\varphi} : \boldsymbol{x} \mapsto \boldsymbol{\varphi}(\boldsymbol{x})$ that maps the input vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$ to the typically high (and possibly) infinite dimensional feature vector $\boldsymbol{\varphi}(\boldsymbol{x}) \in \mathbb{R}^{n_\varphi}$. In this high dimensional feature space, the LS-SVM classifier takes the form

$$\hat{y} = \text{sign}[\boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}) + b]. \tag{4}$$

The LS-SVM classifier is then constructed as follows

$$\min_{\boldsymbol{w},b,\boldsymbol{e}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{n_\mathcal{D}} e_i^2 \tag{5}$$

$$\text{s.t. } e_i = y_i - (\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b), \quad i = 1, \ldots, n_\mathcal{D}. \tag{6}$$

The second term in (5) corresponds to the sum of squared errors[1] term in (3), while the first term can be interpreted as the regularization term in ridge regression [18], the weigth decay term in neural networks [8] or the large margin term in SVMs [34]. The regularization parameter $\gamma > 0$ determines the trade-off between regularization (γ small) and error minimization (γ large). The solution from the ridge regression problem (5)-(6) can be obtained in the primal space from a linear set of equations of dimension $(n_\varphi + 1) \times (n_\varphi + 1)$, but in SVMs and kernel methods in general, the feature vector $\boldsymbol{\varphi}(\boldsymbol{x})$ is only implicitly known from Mercer's theorem [21] by means of the positive definite kernel function

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j). \tag{7}$$

The feature vector may even become infinite dimensional. Some typical kernel functions are the linear kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T\boldsymbol{x}_j$ and the Radial Basis Function kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/\sigma^2)$ with bandwidth parameter σ. Figure 1 illustrates the kernel classification based on a linear classifier in the kernel induced feature space.

3.2 Dual Formulation

In LS-SVMs, one solves the constrained optimization problem (5)-(6) by constructing the Lagrangian

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{e}; \boldsymbol{\alpha}) = \boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{n_\mathcal{D}} e_i^2 - \sum_{i=1}^{n_\mathcal{D}} \alpha_i(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i - y_i),$$

[1] Note that the LS-SVM classifier formulation [27] started from $e_i = 1 - y_i(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b)$ in a similar way as the Vapnik SVM formulation [34]. Since $y_i^2 = 1$, this is equivalent to the regression formulation (5)-(6) [33].

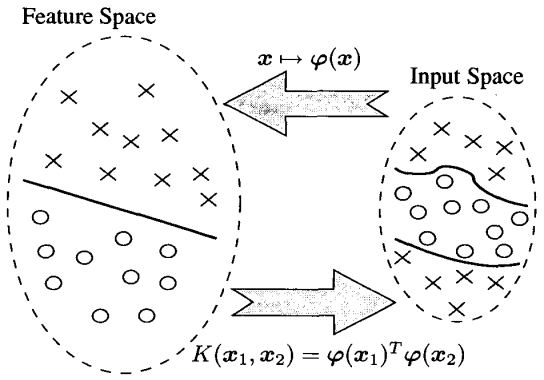


Figure 1: Illustration of kernel based classification. The inputs are first mapped in a nonlinear way to a high-dimensional feature space ($\boldsymbol{x} \mapsto \boldsymbol{\varphi}(\boldsymbol{x})$), in which a linear separating hyperplane is constructed. Applying the Mercer condition ($K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\varphi}(\boldsymbol{x}_1)^T\boldsymbol{\varphi}(\boldsymbol{x}_2)$), a nonlinear classifier in the input space is obtained.

where $\alpha_i \in \mathbb{R}$ are the Lagrange multipliers (support values). The conditions for optimality are given by:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_{i=1}^{n_\mathcal{D}} \alpha_i \boldsymbol{\varphi}(\boldsymbol{x}_i) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^{n_\mathcal{D}} \alpha_i = 0 \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, \quad i = 1, \ldots, n_\mathcal{D} \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow b = y_i - \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) - e_i, \quad i = 1, \ldots, n_\mathcal{D}. \end{cases}$$

As in standard SVMs, $\boldsymbol{w}$ and $\boldsymbol{\varphi}(\boldsymbol{x}_i)$ are never calculated and by elimination of $\boldsymbol{w}$ and $\boldsymbol{e} = [e_1; \ldots; e_{n_\mathcal{D}}]$ the following linear Karush-Kuhn-Tucker system of dimension $(n_\mathcal{D}+1) \times (n_\mathcal{D} + 1)$ is obtained in the dual space [27, 28, 33]:

$$\left[\begin{array}{c|c} 0 & \mathbf{1}^T \\ \hline \mathbf{1} & \boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I}_{n_\mathcal{D}} \end{array}\right] \left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \boldsymbol{y} \end{array}\right], \tag{8}$$

with $\boldsymbol{y} = [y_1; ...; y_{n_\mathcal{D}}]$, $\mathbf{1} = [1; ...; 1]$, and $\boldsymbol{\alpha} = [\alpha_1; ...; \alpha_{n_\mathcal{D}}]$ and where Mercer's theorem [9, 21, 26, 34] is applied within the $\boldsymbol{\Omega}$ matrix

$$\boldsymbol{\Omega}_{ij} = \boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{9}$$

In the optimum we have $\boldsymbol{w} = \sum_{i=1}^{n_\mathcal{D}} \alpha_i \boldsymbol{\varphi}(\boldsymbol{x}_i)$ and the LS-SVM classifier is obtained by applying the Mercer condition:

$$\begin{aligned} \hat{y} &= \text{sign}[\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}) + b] \\ &= \text{sign}[\sum_{i=1}^{n_\mathcal{D}} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b], \end{aligned} \tag{10}$$

with latent variable $z = \sum_{i=1}^{n_\mathcal{D}} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b$. The latent variable is obtained as a weighted sum of the kernel functions evaluated in the input vector and the training data points, adding also the bias term. The support values α_i $(i = 1, \ldots, n_\mathcal{D})$ in the dual classifier formulation determine the relative weight of each data point $\boldsymbol{x}_i$ in the classifier formulation.

3.3 Links with Other Algorithms

Links exist between the LS-SVM classifier and other techniques. The relations explained above between the linear classifiers also hold in the kernel induced feature space and allow to relate the LS-SVM classifier with kernel Fisher Discriminant Analysis [6, 22, 33]. The regression interpretation to classification establishes links with Gaussian Process regression [35] and regularization networks [15]. The latter have already been successfully applied to option pricing [19]. Taking another cost function, one obtains the SVM classifier for which the solution follows from a computationally more complex quadratic programming problem instead of a linear KKT-system. On the other hand, the SVM formulation results into a sparse solution (with a number of zero support values α_i). In an extensive benchmark study [31] SVM and LS-SVM classifiers were evaluated on 10 binary and 10 multiclass UCI data sets and compared with statistical algorithms, decision trees and rules. From this comparison study it was concluded that the SVM and LS-SVM classifiers yielded the best average performances (with no significant difference between them) and obtained very good performances on each of the 20 individual data sets. It was also observed that the target choice $(-1, +1)$ yielded better performances than the Fisher targets $(-n_{\mathcal{D}}/n_{\mathcal{D}}^-, n_{\mathcal{D}}/n_{\mathcal{D}}^+)$, reflecting the better choice of the bias term as motivated above.

3.4 Hyperparameter Selection

The LS-SVM solution follows from a linear system (8) that is determined by the training data points $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_{\mathcal{D}}}$, the regularization parameter γ and the kernel function K with possible kernel parameter, like, e.g., the bandwidth parameter σ in the case of an RBF kernel. These so-called hyperparameters have to be selected carefully in order to obtain a good generalization behavior of the classifier. There exist different techniques for the hyperparameter selection, like Bayesian inference [4, 20, 32, 33], VC-bounds [10, 34] and cross-validation. The latter has proven to be a successful selection strategy in [31, 33] and will be used in this paper, as the number of training data points is rather limited. More precisely, the hyperparameters will be selected by means of a leave-one-out selection procedure [13]. Since the solution now follows from a linear system, this can be computed efficiently by means of the matrix inversion lemma, explained in the appendix. Compared with MLPs, this is another advantage, together with the lack of local minima and the reduced number of hyperparameters that have to be selected.

4 CASE STUDY

4.1 Data Set Description

The credit rating tool that is used here [5] is specifically designed for firms with middle-market capitalization (mid-cap firms) in the Benelux (Belgium, the Netherlands and Luxembourg). The market capitalization of mid-cap firms exceeds €10 mln, while generating a minimal turnover of €0.25 bln. Since these companies are not stocklisted, more advanced methods like option based valuation models are not applicable. Together with small and medium enterprises, mid-cap firms represent a large proportion of the economy in Belgium and the Netherlands. For Fortis Bank, the mid-cap market segment is especially important, reflecting its main business orientation.

The data set used in this experiment consists of $n_{\mathcal{D}} = 422$ observations, $n_{\mathcal{D}}^- = 74$ firms went bankrupt and $n_{\mathcal{D}}^+ = 348$ were solvent companies. The default data were collected from 1989-1997, while the other data was colleced in 1996-1997 only. Observe that a larger sample of solvent firms could have been selected, but involves training on an even more unbalanced training set. No distinction between Belgian or Dutch firms was made in the analysis. A total number of 40 candidate input variables was selected from balance sheet data, using a.o. liquidity, profitability and solvency measures. As can be seen from Table 1, both ratios as well as raw numbers were extracted from the data.

4.2 Performance Measures

The performance of all trained classifiers will be quantified using both the classification accuracy (PCC) and the area under the receiver operating characteristic curve (AUROC). The percentage of correctly classified (PCC) observations is used to report the classification accuracy. However, it tacitly assumes equal misclassification costs and balanced class distributions. The receiver operating characteristic curve (ROC) is a 2-dimensional graphical illustration of the sensitivity ('true alarms') on the Y-axis versus 1-specificity on the X-axis ('false alarms') for various values of the classification threshold. It basically illustrates the behaviour of a classifier without regard to class distribution or misclassification cost and provides information of the discrimination ability of the classifier.

We will use McNemar's test to compare the PCCs of different classifiers [14]. This chi-squared test is based upon contingency table analysis to detect statistically significant performance differences between classifiers. Statistically significant AUROC differences between classifiers are detected using the nonparametric chi-squared test of [11].

4.3 Empirical Results

We report the classification accuracy of Linear Discriminant Analysis (LDA), Logistic Regression (LOGIT) and LS-SVMs with RBF kernel by means of the percentage correctly classified instances (PCC). These performances were compared to the best algorithm (LS-SVM with RBF-kernel) using the McNemar test in order to detect significantly different classifier performances. The confusion matrices of the different algorithms are reported in Table 2, while the performances and p-values of the McNemar test are reported for the full candidate input set in Table 3. As the training data set is unbalanced with a relatively high number of sol-

Table 1: Fortis data set: description of the 40 candidate inputs. The inputs include various liquidity, solvency and profitability measures. Trends (Tr) are used to describe the evolution of the levels (L) and ratios (R). The results of backward input selection are presenting by reporting the number of remaining inputs in the LDA, LOGIT and LS-SVM model when an input is removed. These ranking numbers are underlined when the corresponding input is used in the model having optimal cross validation performance. Hence, inputs with low importance have a high number, while the most important input has rank 1.

Input	LDA	LOGIT	LS-SVM
Current ratio (Tr)	27	27	29
Quick ratio (Tr)	24	26	39
Numbers of days to customer credit (Tr)	13	20	15
Numbers of days of supplier credit (Tr)	12	25	6
Current ratio (R)	6	12	31
Quick ratio (R)	35	8	17
Numbers of days to customer credit (R)	32	6	23
Numbers of days of supplier credit (R)	20	31	20
Capital and reserves (Tr)	18	38	40
Financial debt payable after one year (Tr)	31	36	37
Financial debt payable within one year (Tr)	37	35	30
Solvency Ratio (%)(Tr)	14	18	36
Capital and reserves (L)	38	10	2
Financial debt payable after one year (L)	23	17	26
Financial debt payable within one year (L)	17	28	7
Solvency Ratio (%)(R)	3	7	3
Turnover (Trend)	39	24	38
Added value (Tr)	19	22	34
Total assets (Tr)	29	40	33
Current profit/Current loss before taxes (Tr)	34	23	18
Current profit/Current loss (Tr)	33	34	28
Gross operation margin (%)(Tr)	40	29	9
Net operation margin (%)(Tr)	30	11	19
Added value/sales (%)(Tr)	36	30	32
Added value/pers. employed (Tr)	22	32	22
Cash-flow/equity (%)(Tr)	21	21	13
Return on equity (%)(Tr)	9	33	11
Net return on total assets before taxes and debt charges (%)(Tr)	26	37	35
Turnover (Level)	2	2	1
Added value (Ratio)	7	1	10
Current profit/Current loss before taxes (R)	8	16	16
Current profit/Current loss (R)	15	13	8
Gross operation margin (%)(R)	11	39	14
Net operation margin (%)(R)	25	19	4
Added value/sales (%)(R)	4	5	25
Added value/pers. employed (R)	16	15	24
Cash-flow/equity (%)(R)	28	4	21
Return on equity (%)(R)	10	3	5
Net return on total assets before taxes and debt charges (%)(L)	1	14	27
Total assets (L)	5	9	12

Table 2: Leave-one-out cross validation set confusion matrices of LDA, LOGIT and LS-SVM obtained with the full candidate input set (40 inputs, left) and the optimal reduced input set (right).

LDA	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	43.24%	56.76%
$y=+1$	06.32%	93.68%

LDA	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	33.78%	66.22%
$y=+1$	00.86%	99.14%

LOGIT	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	47.30%	52.70%
$y=+1$	08.62%	91.38%

LOGIT	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	51.35%	48.65%
$y=+1$	03.16%	96.84%

LS − SVM	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	45.95%	54.05%
$y=+1$	02.30%	97.70%

Full input set

LS − SVM	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	54.05%	45.95%
$y=+1$	00.86%	99.14%

Reduced input set

vent firms with respect to bankrupt firms, a bias term correction was carried out in order to improve the PCC. Changing $b \mapsto b + \Delta b$ corresponds to other choices for the prior class probabilities or misclassification costs [33]. The area under receiver operating characteristic curve AUROC was estimated using the trapezoidal rule [17] based on the leave-one-out latent variables, while the variance and covariances for pairwise comparisons are used for the significance test with respect to the best algorithm [11]. These results are reported in the same way as the classification accuracies in Table 3. The corresponding ROC-curves are depicted in Figure 3. As can be seen from Table 3, the LS-SVM classifier yields a statistically better PCC and AUROC at the 1% level than both the LDA and LOGIT classifier for the full models. The confusion matrices in Table 2 indicate that all classifiers perform especially well in classifying the healthy firms as healthy. Most of the errors concern misclassifying unhealthy firms as healthy.

Table 3: Leave-one-out cross validation set performances PCC and AUROC of LDA, LOGIT and LS-SVM obtained with the full (F) candidate input set (40 inputs) and with the reduced (R) optimized input set obtained by backward input selection. The corresponding p-values for the significance tests comparing LDA and LOGIT with the LS-SVM are reported between parentheses. A low p-value (e.g., below 5% or 1%) allows to reject the H0 hypothesis of no improvement of the LS-SVM classifier over the other classifiers.

	LDA	LOGIT	LS-SVM
PCC (F)	84.83 (0.0046)	84.12 (0.0031)	88.62
PCC (R)	87.68 (0.0071)	88.86 (0.0662)	91.23
AUROC (F)	75.67 (0.0020)	76.98 (0.0055)	84.08
AUROC (R)	80.68 (0.0081)	82.97 (0.0653)	86.82

Given the full candidate input set and the corresponding hyperparameters, a backward input selection step is performed by in turn removing each candidate input and comparing the corresponding classification performances. During this step, we kept the hyperparameters fixed, but refine them on the selected model with optimal reduced input set. After backward input selection, the LS-SVM still performs better in terms of PCC and AUROC than the LDA classifier at the 1% level. The PCC and AUROC of the pruned LOGIT classifier are different from the pruned LS-SVM classifier at the 10% level but not at the 5% level. Again, when inspecting the confusion matrices of the reduced models (see Table 3), the same patterns are present as for the full models. The ROC curves of Figures 2 and 3 clearly illustrate the better performance of the LS-SVM classifier when compared to the LDA and LOGIT classifier. It can be observed from Table 1 that the pruned LDA classifier has 8 inputs, the pruned LOGIT classifier 11 inputs and the pruned LS-SVM classifier 19 inputs. This clearly illustrates that the proposed input pruning procedure results in more concise and parsimonious classifiers with a higher performance in terms of both PCC and AUROC than the classifiers trained on all 40 inputs (see Table 3). For the sake of completeness, we mention that we also implemented a kernel LOGIT classifier and obtained a maximal classification accuracy of 89.34% after input selection [30]. As particularly the linear classifiers yield high misclassification error rates on the bankrupt firms (due to both overlap and somewhat unbalanced training data), the same input and hyperparameter selection procedure was performed with the AUROC as selection criterion, which is in-

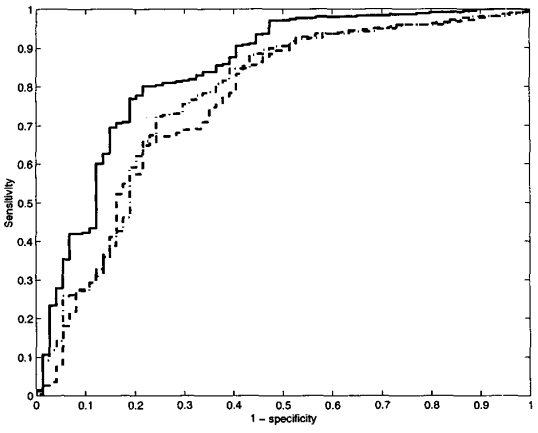


Figure 2: Receiver Operating Characteristic curves obtained with LDA (dashed line), LOGIT (dash-dotted line) and LS-SVM (full line) using the full candidate input set.

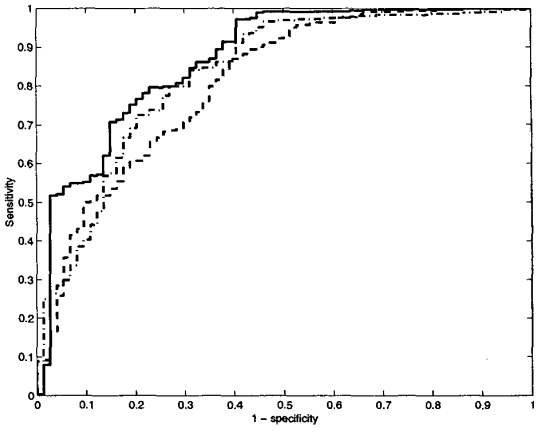


Figure 3: Receiver Operating Characteristic curves obtained with LDA (dashed line), LOGIT (dash-dotted line) and LS-SVM (full line) using the optimized input set.

dependent of the choice of the bias term. As can be seen from Tables 4 and 5, this yields slightly better results on recognizing bankrupt firms, while the overall performance of the classifiers LDA, LOGIT and LS-SVM are similar as in the case of the PCC input selection criterion.

Summarizing the results, it is observed in Table 1 that all classifiers agree on the importance of the Turnover as one of the most important inputs, which may correspond somewhat to the firm size variable of the Altman model [1]. The five most important inputs of the LS-SVM classifier are: Turnover (Level), Capital and Reserves (Level), Solvency Ratio, Net operation Margin, and Return on Equity. These 5 inputs already give a PCC of 89.91% and an AUROC of 85.64%.

Table 4: Leave-one-out cross validation set confusion matrices of LDA, LOGIT and LS-SVM obtained with the full candidate input set (40 inputs, left) and the optimal reduced input set (right) using the AUROC as selection criterion.

LDA	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	43.24%	56.76%
$y=+1$	06.32%	93.68%

LDA	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	33.78%	66.22%
$y=+1$	00.86%	99.14%

LOGIT	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	47.30%	52.70%
$y=+1$	04.60%	95.40%

LOGIT	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	48.65%	51.35%
$y=+1$	03.74%	96.26%

LS − SVM	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	47.30%	52.70%
$y=+1$	02.87%	97.13%

Full input set

LS − SVM	$\hat{y}=-1$	$\hat{y}=+1$
$y=-1$	62.16%	37.84%
$y=+1$	02.87%	97.13%

Reduced input set

Table 5: Leave-one-out cross validation set performances PCC and AUROC of LDA, LOGIT and LS-SVM using the AUROC as selection criterion.

	LDA	LOGIT	LS-SVM
PCC (F)	84.83 (0.0051)	84.12 (0.0027)	88.39
PCC (R)	86.97 (0.0147)	87.91 (0.0485)	91.00
AUROC (F)	75.67 (0.0006)	76.98 (0.0020)	84.59
AUROC (R)	86.68 (0.0106)	86.02 (0.0034)	91.63

5 CONCLUSIONS

Linear and nonlinear bankruptcy prediction techniques are an important research topic for modelling, predicting and understanding corporate failure. In this paper, we used Least Squares Support Vector Machine (LS-SVM) classifiers as a computationally simple, but powerful nonlinear classifier to predict financial distress of mid-cap companies in the Benelux. The LS-SVM classifier can be understood as applying linear Fisher Discriminant Analysis or ridge regression in the high dimensional kernel induced feature space, while practical expressions for model training and evaluation are obtained in terms of the kernel function. Comparing the results of the different classfiers, significantly better leave-one-out classification performances are obtained with the LS-SVM classifier using an RBF kernel.

ACKNOWLEDGMENTS

This work was carried out at the ESAT-SCD-SISTA laboratory, the institute LIRIS of the K.U.Leuven and the Interdisciplinary Center of Neural Networks ICNN of the K.U.Leuven and supported by grants and projects from the Flemish Government: (Research council K.U.Leuven: Grants, GOA-Mefisto 666; FWO-Flanders: Grants, research projects (G.0407.02, G.0080.01, G.0256.97, G.0115.01, G.0240.99, G.0197.02) and communities (ICCoS and ANMMM); AWI: Bil. Int. Collaboration South Africa, Hungary and Poland; IWT: Soft4s, STWWW, GBOU, Eureka); from the Belgian Federal Government (Interuniversity Attraction Poles: IUAP-P4/02, P4/24); from the European Commission: (TMR Network (Alapedes, Niconet); Science: ERNSI). TVG and JS are postdoctoral researchers with the Fund for Scientific Researchers FWO-Vlaanderen. BB is a research assistant at the K.U.Leuven, Dept. of Applied Economic Sciences.

REFERENCES

[1] Altman, E.I. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Journal of Finance*, 23:589–609, 1968.

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

[3] Atiya, A.F. Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on Neural Networks*, 12(4):929–935, 2001.

[4] Baesens, B., Viaene, S., Van den Poel, D., Vanthienen, J., and Dedene, G. Using bayesian neural networks for repeat purchase modelling in direct marketing. *European Journal of Operational Research*, 138(1):191–211, 2002.

[5] Baestaens, D.-E. Credit Risk Modelling Strategies: The Road to Serfdom. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 8:225–235, 1999.

[6] Baudat, G. and Anouar, F. Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation*, 12:2385–2404, 2000.

[7] Beaver, W.H. Financial Ratios as Predictors of Failure. *Empirical Research in Accounting Selected Studies, supplement to the Journal of Accounting*, 5:71–111, 1966.

[8] Bishop, C.M. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[9] Cristianini, N., and Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[10] Cucker, F., and Smale, S. On the mathematical foundations of learning theory. *Bulleting of the AMS*, 39:1–49, 2002.

[11] De Long, E.R., De Long, D.M. and Clarke-Pearson, D.L. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44:837–845, 1988.

[12] Duda, R.O., and Hart, P.E. *Pattern Classification and Scene Analysis*. John Wiley, New York, 1973.

[13] Eisenbeis, R. Pitfalls in the application of discriminant analysis in business. *The Journal of Finance*, 32(3):875–900, 1977.

[14] Everitt, B.S. *The analysis of contingency tables*. Chapman and Hall, London, 1977.

[15] Evgeniou, T., Pontil, M., and Poggio, T. Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics*, 13:1–50, 2001.

[16] Fisher, R.A. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

[17] Hanley, J.A. and McNeil, B.J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148:839–843, 1983.

[18] Hoerl, A.E. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59, 1962.

[19] Hutchinson, J.M., Lo, A.W., and Poggio, T. A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks. *Journal of Finance*, 49:851–889, 1994.

[20] MacKay, D.J.C. Bayesian Interpolation. *Neural Computation*, 4:415–447, 1992.

[21] Mercer, J. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 209:415–446, 1909.

[22] Mika, S., Rätsch, G., Weston, J., Schölkopf, B., and Müller, K.-R. Fisher discriminant analysis with kernels. In *Proceedings Neural Networks for Signal Processing Workshop IX (NNSP'99)*, Hu, Y.-H., Larsen, J., Wilson, E., Douglas, S. (Eds.), pp. 41–48, 1999. IEEE.

[23] Ohlson, J.A. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, 18:109–131, 1980.

[24] Poddig, T. *Neural networks in the Capital Markets (Editor: Apostolos-Paul Refenes)*, chapter Bankruptcy prediction: a comparison with discriminant analysis, pages 311–323. John Wiley and Sons, 1995.

[25] Ripley, B.D. *Pattern Classification and Neural Networks*. Cambridge University Press, 1996.

[26] Schölkopf, B., and Smola, A. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[27] Suykens, J.A.K., and Vandewalle, J. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[28] Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle J. *Least Squares Support Vector Machines*. World Scientific, in press, 2002.

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

[30] Van Gestel, T., Baesens, B., Suykens, J., Willekens, M., Baestaens, D.E., Vanthienen, J., and De Moor, B. From Linear to Nonlinear Kernel Discriminant Analysis for Bankruptcy Prediction, Working Paper, Dept. Electrical Engineering, K.U.Leuven, Belgium, 2000.

[31] Van Gestel, T., Suykens, J.A.K., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B., and Vandewalle, J. Benchmarking Least Squares Support Vector Machine Classifiers. *Machine Learning*, in press.

[32] Van Gestel, T., Suykens, J.A.K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B., and Vandewalle, J. Predicting financial time series using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks (Special Issue on Financial Engineering)*, 12:809–821, 2001.

[33] Van Gestel, T., Suykens, J.A.K., Lanckriet, G., Lambrechts, A., De Moor, B., and Vandewalle, J. A Bayesian framework for Least Squares Support Vector Machine Classifiers, Gaussian Processes and kernel Fisher Discriminant Analysis. *Neural Computation*, 14:1115–1147, 2002.

[34] Vapnik, V. *Statistical Learning Theory*. Wiley, New-York, 1998.

[35] Williams, C.K.I. Prediction with Gaussian Processes: from Linear Regression to Linear Prediction and Beyond. In *Learning and Inference in Graphical Models*, Jordan, M.I. (Ed.), pp. 599–621, 1998. Kluwer Academic Press.

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

Appendix: Efficient Implemenation of Leave-One-Out Cross-Validation for LS-SVMs

We are concerned with solving the linear Karush-Kuhn-Tucker (8) in a leave-one-out cross-validation setup. This implies solving $n_{\mathcal{D}}$ times a linear set of $n_{\mathcal{D}}+1$ equations, which may become time consuming. A more efficient implementation is derived as the $n_{\mathcal{D}}$ KKT systems for the leave-one-out problems only differ from the full $n_{\mathcal{D}}+1$ KKT system by just one data point.

Starting from the full KKT system which is written as

$$\boldsymbol{F}\boldsymbol{u} = \boldsymbol{t}, \tag{11}$$

using the notation $\boldsymbol{F} = [0\ \mathbf{1}^T;\ \mathbf{1}\ \boldsymbol{\Omega}+\gamma^{-1}\boldsymbol{I}_{n_{\mathcal{D}}}]$, $\boldsymbol{u} = [\hat{b};\ \boldsymbol{\alpha}]$ and $\boldsymbol{t} = [0;\ \boldsymbol{y}]$, respectively, the solution $\boldsymbol{u}$ is obtained from

$$\boldsymbol{u} = \boldsymbol{F}^{-1}\boldsymbol{t} = \boldsymbol{H}\boldsymbol{t}, \tag{12}$$

with $\boldsymbol{H} \stackrel{\triangle}{=} \boldsymbol{F}^{-1}$. Now consider the smaller leave-one-out training problem on the first $n_{\mathcal{D}}-1$ data points (removing the last data point) and partition (11) and (12) into the block matrix equations

$$\begin{bmatrix} \boldsymbol{F}_{11} & \boldsymbol{f}_{12} \\ \boldsymbol{f}_{12}^T & f_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_1 \\ t_2 \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} \boldsymbol{u}_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{h}_{12} \\ \boldsymbol{h}_{12}^T & h_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{t}_1 \\ t_2 \end{bmatrix}. \tag{14}$$

From the matrix inversion lemma, several relations between the block matrices have been shown like $\boldsymbol{H}_{11} = \boldsymbol{F}_{11}^{-1} + \boldsymbol{F}_{11}^{-1}\boldsymbol{f}_{12}f_{22}^{-1}\boldsymbol{f}_{12}^T\boldsymbol{F}_{11}^{-1}$, $\boldsymbol{h}_{12} = -\boldsymbol{F}_{11}^{-1}\boldsymbol{f}_{12}\boldsymbol{f}_{22}^{-1}$ and $\boldsymbol{F}_{11}^{-1} = \boldsymbol{H}_{11} - \boldsymbol{h}_{12}h_{22}^{-1}\boldsymbol{h}_{12}^T$.

The training problem for the first $n_{\mathcal{D}}-1$ data points corresponds to the linear KKT system

$$\boldsymbol{F}_{11}\boldsymbol{u}' = \boldsymbol{t}_1, \tag{15}$$

where the accent is used to denote the solution to the smaller problem. Given the relations between the inverses of a block matrix, one obtains the following solution

$$\begin{aligned} \boldsymbol{u}' &= \boldsymbol{F}_{11}^{-1}\boldsymbol{t}_1 = \boldsymbol{H}_{11}\boldsymbol{t}_1 - \boldsymbol{h}_{12}h_{22}^{-1}\boldsymbol{h}_{12}^T\boldsymbol{t}_1 \\ &= \boldsymbol{u}_1 - \boldsymbol{h}_{12}(t_2 + h_{22}^{-1}\boldsymbol{h}_{12}^T\boldsymbol{t}_1) \\ &= \boldsymbol{u}_1 - \boldsymbol{h}_{12}(t_2 + h_{22}^{-1}(u_2 - h_{22}t_2)) \\ &= \boldsymbol{u}_1 - \frac{u_2}{h_{22}}\boldsymbol{h}_{12}. \end{aligned} \tag{16}$$

Given the solution $\boldsymbol{u}$ and matrix inverse $\boldsymbol{H}$ of the full $(n_{\mathcal{D}}+1)\times(n_{\mathcal{D}}+1)$ KKT system, the solution of an $n_{\mathcal{D}}$ subblock KKT system is efficiently computed by calculating a scalar inverse and a vector sum. This is far more efficient instead of solving the $n_{\mathcal{D}}\times n_{\mathcal{D}}$ linear set of equations.

The leave-one-out cross-validation for LS-SVM classifiers is obtained in the following steps:

1. Calculate the inverse $\boldsymbol{H} = [0\ \mathbf{1}^T;\mathbf{1}\ (\boldsymbol{\Omega}+\gamma^{-1}\boldsymbol{I}_{n_{\mathcal{D}}})]^{-1}$ and compute $\boldsymbol{\alpha}$ and b for the full set from (8).
2. For each of the $n_{\mathcal{D}}$ data points compute the leave-one-out estimate as follows:
   (a) Calculate $\boldsymbol{u}' = [b';\ \boldsymbol{\alpha}']$ from
   $$\begin{bmatrix} \hat{b}' \\ \boldsymbol{\alpha}' \end{bmatrix} = \begin{bmatrix} \hat{b} \\ \alpha_1 \\ \vdots \\ \alpha_{i-1} \\ \alpha_{i+1} \\ \vdots \\ \alpha_{n_{\mathcal{D}}} \end{bmatrix} - \frac{\alpha_i}{\boldsymbol{H}(i+1,i+1)} \begin{bmatrix} \boldsymbol{H}(1,i) \\ \boldsymbol{H}(2,i) \\ \vdots \\ \boldsymbol{H}(i-1,i) \\ \boldsymbol{H}(i,i+1) \\ \vdots \\ \boldsymbol{H}(i,n_{\mathcal{D}}+1) \end{bmatrix}.$$
   (b) Evaluate the latent variable z_i from
   $$z_i = \sum_{k\neq i}\alpha_k' K(\boldsymbol{x}_k,\boldsymbol{x}_i) + \hat{b}'$$
   and obtain the classification result $\hat{y}_i = \mathrm{sign}(z_i)$.
3. Store the latent variables $\boldsymbol{z} = [z_1;\ldots;z_i;\ldots;z_{n_{\mathcal{D}}}]$ and classification decisions $\hat{\boldsymbol{y}} = [\hat{y}_1;\ldots;\hat{y}_i;\ldots;\hat{y}_{n_{\mathcal{D}}}]$.

This procedure allows to efficiently evaluate the leave-one-out performance of LS-SVM classifiers.