SFB 649 Discussion Paper 2012-030

Support Vector Machines with Evolutionary Feature Selection for Default Prediction

Wolfgang Karl Härdle*
Dedy Dwi Prastyo*
Christian Hafner**

* Humboldt-Universität zu Berlin, Germany
** Université catholique de Louvain, Belgium

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk".

http://sfb649.wiwi.hu-berlin.de
ISSN 1860-5664

SFB 649, Humboldt-Universität zu Berlin
Spandauer Straße 1, D-10178 Berlin

Support Vector Machines with Evolutionary Feature Selection for Default Prediction

Wolfgang Karl Härdle
Humboldt-Universität zu Berlin, Ladislaus von Bortkiewicz Chair of Statistics
Spandauerstr. 1, 10178 Berlin, Germany
Email: haerdle@wiwi.hu-berlin.de

Dedy Dwi Prastyo
Humboldt-Universität zu Berlin, Ladislaus von Bortkiewicz Chair of Statistics,
Email: prastyod@hu-berlin.de
Department of Statistics, Institut Teknologi Sepuluh Nopember, Indonesia
Email: dedy-dp@statistika.its.ac.id

Christian Hafner
Institut de statistique, Université catholique de Louvain, Belgium
Email: christian.hafner@uclouvain.be

Abstract

Predicting default probabilities is at the core of credit risk management and is becoming more and more important for banks in order to measure their client's degree of risk, and for firms to operate successfully. The SVM with evolutionary feature selection is applied to the CreditReform database. We use classical methods such as discriminan analysis (DA), logit and probit models as benchmark On overall, GA-SVM is outperforms compared to the benchmark models in both training and testing dataset.

Keywords: SVM, Genetic algorithm, global optmimum, default prediction

JEL Classification: C14, C45, C61, C63, G33

1 Prediction methods for the probability of default

Default probability is defined as the probability that a borrower will fail to serve its obligation. Bonds and other tradable debt instruments are the main source of default for most individual and institutional investors. In contrast, loans are the largest and most obvious source of default for banks (Sobehart and Stein, 2000).

Predicting default probabilities is at the core of credit risk management and is becoming more and more important for banks in order to measure their client's degree of risk, and for firms to operate successfully. The Basel Committee on Banking Supervision established the borrower's rating as a crucial criterion for minimum capital requirements of banks to minimize their cost of capital and mitigate their own bankruptcy risk (Härdle *et al.*, 2009). Alterative methods to generate ratings have been developed essentially over the last 15 years (Krahnen and Weber, 2001).

There are basically two approaches that deal with default risk analysis: The market-based model, frequently denoted as structural model, and the statistical approach determined through an empirical analysis of historical data, e.g. accounting data. The market-based approach uses time series of the company data to predict the probability of default derived from an adapted Black-Scholes model (Black and Scholes (1973) and Vassalou and Xing (2004)). However, the most challenging requirement is the knowledge of market values of debt and equity. This precondition is a severe obstacle to using the Merton model (Merton, 1974) adequately as it is only satisfied in a minority of cases (Härdle *et al.*, 2009). The idea of Merton's model is that equity and debt could be considered as options on the value of the firm's assets. Unfortunately, long time series of market prices are not available for most companies. For companies that are not listed, their market price is unknown. In that case, it is necessary to choose a model which relies on cross-sectional data, financial statements or accounting data. Sobehart and Stein (2000) developed a hybrid model where the output is based on the statistical relationship to default of financial statement information, market information, ratings (when they exist) and a variant of Merton's contingent claims model expressed as distance to default.

The early studies about bankruptcy prediction attempted to identify the difference between financial ratios of solvent and insolvent firms (Ramser and Foster (1931), Winakor and Smith (1935) and Merwin (1942)). Then, parametric statistical models were introduced by using discriminant analysis (DA) for the univariate (Beaver, 1966) and multivariate case (Altman, 1968), also known as Z-score. DA was the dominant method in bankruptcy prediction up to the 1980s. The model separates defaulting from non-defaulting firms based on the discriminatory power of linear combinations of financial ratios. The logit and probit approach replaced the usage of DA during 1980s, see (Martin, 1977), (Ohlson, 1980), (Lo, 1986) and (Plat *et al.*, 1994). These approaches rely on a priori assumed dependence between predictors and risk default. The assumption in DA and logit (or probit) models often fail to meet the reality of observed data. Semiparametric logit models as in (Hwang *et al.*, 2007) incorporate the conventional linear model and a non-parametric approach.

If there is evidence that the separation mechanism is of a nonlinear kind, then the linear separating hyperplane approach is not suitable. In that case, Artificial Neural Network (ANN) is a non-parametric non-linear classification approach to solve the linear non-separability problem. ANN was introduced to analyze bankruptcy in the 1990s, see (Tam and Kiang (1992),Wilson and Sharda (1994) and Altman *et al.* (1994)) for details. ANN has often been criticized to be vulnerable to the multiple minima problem. Common to the Ordinary Least Square (OLS) and Maximum Likelihood Estimation (MLE) for linear models, ANN also uses the principle of minimizing empirical risk, which usually leads to poor classification performance for out-of-sample data (Haykin (1999), Gunn (1998) and Burges (1998)).

In contrast to the case of neural networks, where many local minima usually exist, Support Vector Machines (SVM) training always finds a global solution (Burges, 1998). SVMs is a state-of-the-art classification method and one of the most promising among recently developed non-linear statistical techniques. The idea of SVMs can be said to have started in the late 1970s by Vapnik (1979), but it was receiving increasing attention after the work in statistical learning theory (Boser *et al.* (1992), Vapnik (1995) and Vapnik (1998)). The SVM formulation embodies the Structural Risk Minimisation (SRM) principle (Shawe-Taylor *et al.*, 1996). At the first stages, SVM has been successfully applied to classify (multivariate) observations, see Blanz *et al.* (1996), Cortes and Vapnik (1995), Schölkopf *et al.* (1995), Schölkopf *et al.* (1996), Burges and Schölkopf (1997) and Osuna *et al.* (1997a). Later, SVM has been used in regression prediction and time series forecasting (Müller *et al.*, 1997).

The SVM has been applied to bankruptcy prediction and typically outperformed the competing models (Härdle and Simar (2012), Härdle *et al.* (2009), Zhang and Härdle (2010), Härdle *et al.* (2011) and Chen *et al.* (2011)). One of the important issues in SVM is the parameter optimization (feature selection). This chapter emphasizes the feature selection of SVM for bankruptcy prediction applied to a credit database. The SVM parameters are optimized by using an evolutionary algorithm, the so-called Genetic Algorithm (GA) introduced by Holland (1975). Some recent papers that deal with GA are Michalewicz (1996), Gen and Cheng (2000), Melanie (1999), Haupt and Haupt (2004), Sivanandam and Deepa (2008) and Baragona *et al.* (2011).

In the case of a small portion of samples belonging to a certain class (label) compared to the portion of other classes, this kind of data may tend to classify every sample as the class of the majority. This is the case in default and non-default datasets, and such models would be useless in practice. The fundamental issue with imbalanced learning problems is the property of imbal-

		sample (Y)	
		default (1)	non-default (-1)
predicted ($\widehat{Y}$)	(1)	True Positive (TP)	False Positive (FP)
	(-1)	False Negative (FN)	True Negative (TN)
total		P	N

Table 1: Contingency table for performance evaluation of two-class classification

anced data to significantly compromise the performance of most standard learning algorithms. He and Garcia (2009) provide a comprehensive and critical review of the development research in learning from imbalanced data.

Two of the methods to overcome this problem are the *down-sampling* and *over-sampling* strategies (Härdle *et al.*, 2009). Under-sampling works with bootstrap to select a set of majority class examples such that both the majority and minority classes are balanced. Due to the random sampling of bootstrap, the majority sample might cause the model to have the highest variance. An over-sampling scheme could be applied to avoid this unstable model building (Maalouf, 2011). The over-sampling method selects a set of samples from the minority class and replicates the procedure such that both majority and minority classes are balanced.

At first glance, the down-sampling and over-sampling appear to be functionally equivalent since they both alter the size of the original data set and can actually yield balanced classes. In the case of under-sampling, removing examples from the majority class may cause the classifier to miss important concepts pertaining to the majority class. With regards to over-sampling, multiple instances of certain examples become 'tied' which leads to overfitting (He and Garcia, 2009). Although sampling methods and cost-sensitive learning methods dominate the current research in imbalanced learning, kernel-based learning, i.e. SVM, have also been pursued. The representative SVMs can provide relatively robust classification results when applied to imbalanced data set (Japkowicz and Stephen, 2002).

2 Quality of default prediction

In classification, one of the most important issues is the discriminative power of classification methods. In credit scoring, for example, the classification methods are used for evaluating the credit worthiness of a client. Any classification errors can create damages to the resources of a credit institute. Therefore, assessing the discriminative power of rating systems is an important topic for banks and regulators.

The most frequent assessment metrics are *accuracy* and *misclassification rate*. A representation of two-class classification performances can be formulated by a contingency table (confusion matrix) as illustrated in Table 1. The accuracy (*Acc*) and misclassification rate (*MR*) are respectively defined as:

$$Acc = P(\widehat{Y} = Y) = \frac{TP + TN}{P + N}. \tag{1}$$

$$MR = P(\widehat{Y} \neq Y) = 1 - Acc. \tag{2}$$

Acc and MR can be deceiving in certain situations and are highly sensitive to changes in data, e.g., unbalanced two-class sample problems. *Acc* uses both columns of information in Table 1. Therefore, as class performance varies, measures of the performance will change even though the underlying fundamental performance of the classifier does not. In the presence of unbalanced data, it becomes difficult to do a relative analysis when the *Acc* measure is sensitive to the data distribution (He and Garcia, 2009).

Other evaluation metrics are frequently used to provide comprehensive assessments, especially for unbalanced data, namely, *specificity*, *sensitivity* and *precision*, which are defined as:

$$Spec = P(\widehat{Y} = -1|Y = -1) = \frac{TN}{N}. \tag{3}$$

$$Sens = P(\widehat{Y} = 1|Y = 1) = \frac{TP}{P}. \tag{4}$$

$$Prec = \frac{P(\widehat{Y} = 1|Y = 1)}{P(\widehat{Y} = 1|Y = 1) + P(\widehat{Y} = 1|Y = -1)} = \frac{TP}{TP + FP}. \tag{5}$$

Precision measures an exactness, but it can not assert how many default samples are predicted incorrectly.

2.1 AR and ROC

Many rating methodologies and credit risk modelling approaches have been developped. The question arises which of these methods are preferable to others. The most popular validation techniques currently used in practice are Cumulative Accuracy Profile (CAP) and Receiver Operating Characteristic (ROC) curve. Accuracy Ratio (AR) is the summary statistic of the CAP curve (Sobehart *et al.*, 2000). ROC has similar concept to CAP and has summary statistics, the area below the ROC curve (called AUC) (Sobehart and Keenan, 2001). Engelmann (2003) analyse the CAP and ROC from a statistical point of view.

Consider a method assign to each observed unit a score S as a function of the explanatory variables. Scores from total samples, S, have cdf F and pdf f, scores from default samples, $S|Y = 1$, have cdf F_1 as well as scores from non-default samples, $S|Y = -1$, have cdf F_{-1}.

The CAP curve is particularly useful as it simulataneously measures Type I and Type II errors. In statistical terms, the CAP curve represents the cumulative probability of default events for different percentiles of the risk score scale. The actual CAP curve is basically defined as the graph of all points $\{F, F_1\}$ where the points are connected by linear interpolation. A perfect CAP curve would assign the lowest scores to the defaulters, then increasing linearly and then staying at one. For a random CAP curve without any discriminative power, the fraction x of all events with the lowest rating scores will contain $x\%$ of all defaulters, $F_i = F_{1,i}$.

Therefore, AR is defined as the ratio of the area between actual and random CAP curves to the area between the perfect and random CAP curves (Figure 1). The classification method is the better the higher is AR, or the closer it is to one. Formally, if $y = \{0, 1\}$, the AR value is defined as:

$$AR = \frac{\int_0^1 y_{actual}\ F\ dF - \frac{1}{2}}{\int_0^1 y_{perfect}\ F\ dF - \frac{1}{2}} \tag{6}$$

If the number of defaulters and non-defaulters is equal, the AR becomes:

$$AR = 4\int_0^1 y_{actual}\ F\ dF - 2 \tag{7}$$

In classification, for example credit reating, assume future defaulters and non-defaulters will be predicted by using rating scores. A decision maker would like to introduce a cut-off value τ, and an observed unit with rating score less than τ will be classified into potential defaulters. A classified non-defaulter in an observed unit would have rating score greater than τ. Table 2 summarizes the possible decisions.

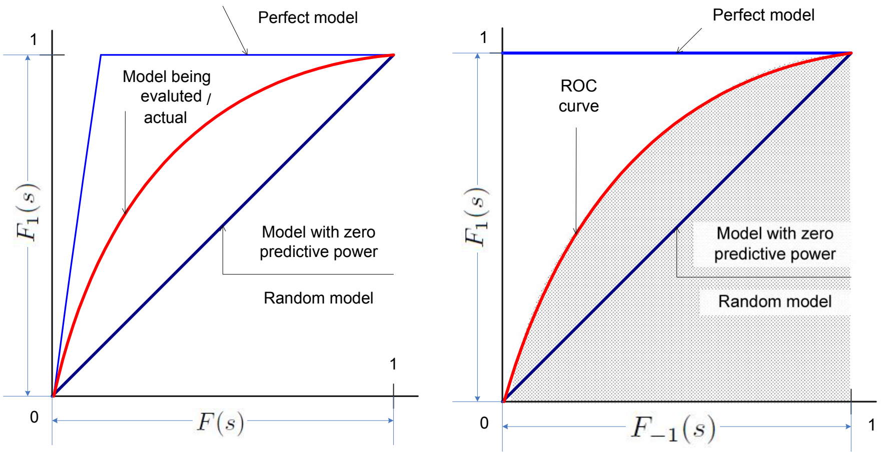


Figure 1: Cumulative Accuracy Profile (CAP) curve (left) and Receiver Operating Characteristic (ROC) curve (right).

		sample (Y)	
		default (1)	no default (-1)
predicted rating score	$\leq \tau$ (default)	correct prediction (hit)	wrong prediction (false alarm)
	$> \tau$ (no default)	wrong prediction (mass)	correct prediction (correct rejection)

Table 2: Classification decision given cut-off value τ

If the rating score is less than the cut-off τ conditionally on a future default, the decision was correct and it is called a *hit*. Otherwise, the decision wrongly classified non-defaulters as defaulters (Type I error), called *false alarm*. The hit rate, $HR(\tau)$, and false alarm rate, $FAR(\tau)$, are defined as ((Engelmann, 2003) and (Sobehart and Keenan, 2001)):

$$HR(\tau) = P(S|Y = 1 \leq \tau) \tag{8}$$

$$FAR(\tau) = P(S|Y = -1 \leq \tau) \tag{9}$$

Given a non-defaulter which has rating score greater than τ, the cassification is correct. Otherwise, a defaulter is wrongly classified as a non-defaulter (Type II error).

The ROC curve is constructed by plotting $FAR(\tau)$ versus $HR(\tau)$ for all given values τ. In other words, the ROC curve consists of all points $\{F_{-1}, F_1\}$ connected by linear interpolation (Figure 1). The area under the ROC curve (AUC) can be interpreted as the average power of the test on default or non-default corresponding to all possible cut-off values τ. A larger AUC characterized a better classification result. A perfect model has an AUC value of 1, and a random model without discriminative power has an AUC value of 0.5. The AUC is between 0.5 and 1.0 for any reasonable rating model in practice. The ralationship between AUC and AR is defined as (Engelmann, 2003):

$$AR = 2AUC - 1 \tag{10}$$

Sing *et al.* (2005) developed package `ROCR` in `R` to calculate performance measures under the ROC curve for classification analysis.

Similarly, the ROC curve is formed by plotting FP_{rate} over TP_{rate}, where

$$FP_{rate} = \frac{FP}{N}, \quad TP_{rate} = \frac{TP}{P}$$

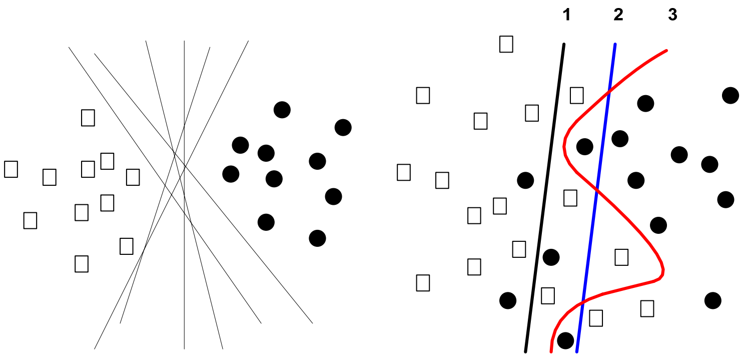


Figure 2: A Set of classification function in the case of linearly separable data (left) and linearly non-separable case (right).

and any point in the ROC curve corresponds to the performance of a single classifier on a given distribution. The ROC curve is useful because it provides a visual representation of the relative trade-offs between the benefits (reflected by TP) and cost (reflected by FP) of classification (He and Garcia, 2009).

3 SVM formulation

This section reviews the support vector machine methodology in classification. We first discuss classicial linear classification, both for linearly separable and non-separable scenarios, and then focus on non-linear classification.

SVM in the linearly separable case

Each observation consists of a pair of p predictors $x_i = (x_{i1}, ..., x_{ip}) \in \mathbb{R}^p$, $i = 1, \dots, n$ and the associated $y_i \in \mathcal{Y} = \{-1, 1\}$. We have a sequence

$$\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\} \in \mathcal{X} \times \{-1, 1\}, \tag{11}$$

of i.i.d pairs drawn from a probability distribution $F(x, y)$ over $X \times Y$. The domain $\mathcal{X}$ is some non-empty set from which x_i are drawn, and y_i are *targets* or *labels*. The indices $i, j = 1, \dots, n$ are always understood to run over the training set.

Now we have a machine whose task is to learn the information in a *training set*, $\mathcal{D}_n$, to predict the label y for any new observation. In the following we will call this machine learning a classifier. The label y_i from training data is then called *trainer* or *supervisor*. A nonlinear classifier function f may be described by a function class $\mathcal{F}$ which is fixed *a priori*, e.g. it can be the class of linear classifiers (hyperplanes).

First we will describe the SVM in the linearly separable case. A key concept to define a linear classifier is the dot product, also referred to as an *inner product* or *scalar product*, between two vectors defined as $x^\top w = \sum_i x_i w_i$. The family $\mathcal{F}$ of classification functions in the data space is given by:

$$\mathcal{F} = \left\{x^\top w + b, w \in \mathbb{R}^p, b \in \mathbb{R}\right\}, \tag{12}$$

where w is known as the *weight vector* and b is called *bias*.

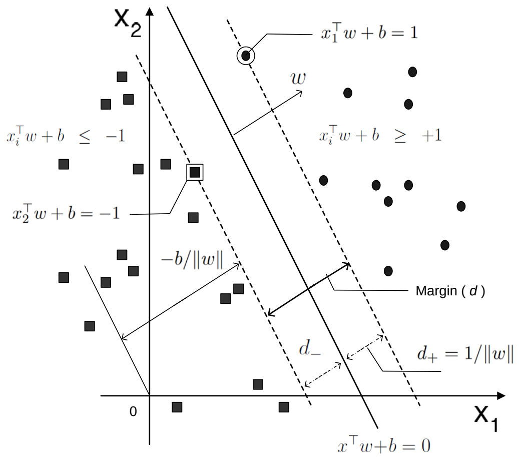


Figure 3: The separating hyperplane $x^\top w + b = 0$ and the margin in the linearly separable case.

The set of points x such that $f(x) = x^\top w = 0$ are all points that are perpendicular to w and go through the origin. The form of $f(x)$ is a line in two dimension, a plane in three dimension, and more generally, a *hyperplane* in the higher dimension. The bias b translates the hyperplane away from the origin (Figure 3).

The following decision boundary (separating hyperplane)

$$f(x) = x^\top w + b = 0, \tag{13}$$

divides the space into two regions as in Figure 3. The sign of $f(x)$ determines in which regions the points lie. The decision boundary defined by a hyperplane is said to be linear because it is linear in the inputs x_i. A so-called *linear classifier* is a classifier with a linear decision boundary. Furthermore, a classifier is said to be a *non-linear classifier* when the decision boundary depends on the data in a non-linear way.

In order to determine the support vectors we choose $f \in \mathcal{F}$ (or equivalently (w, b)) such that the so called *margin* – the corridor between the separating hyperplanes – is maximal. The margin is equal to $d_- + d_+$, where the signs $(-)$ and $(+)$ denote the two regions.

The classification function is a hyperplane plus the margin zone, where, in the separable case, no observations can lie. It separates the points from both classes with the highest 'safest' distance (margin) between them. It can be shown that margin maximization corresponds to the reduction of complexity as given by the VC-dimension (Vapnik, 1998) of the SVM classifier.

The length of vector w is denoted by *norm* $\|w\| = \sqrt{w^\top w}$. A unit vector $\hat{w}$, where $\|\hat{w}\| = 1$, in the direction of w is given by $\frac{w}{\|w\|}$. Furthermore, the margin of a hyperplane $f(x)$ with respect to a dataset $\mathcal{D}_n$ can be seen as follows,

$$d_{\mathcal{D}}(f) = \frac{1}{2}\hat{w}^\top (x_+ - x_-), \tag{14}$$

where the unit vector $\hat{w}$ is in the direction of w. It is assumed that x_+ and x_- are equidistant from the separating hyperplane

$$\begin{aligned} f(x_+) &= w^\top x_+ + b = a, \\ f(x_-) &= w^\top x_- + b = -a, \end{aligned} \tag{15}$$

with constant $a > 0$. Suppose to fix $a = 1$ in order to make the geometric margin meaningful and divide (14) by $\|w\|$ to obtain

$$\frac{d_{\mathcal{D}}(f)}{\|w\|} = \frac{1}{2}\hat{w}^{\top}(x_{+} - x_{-}) = \frac{1}{\|w\|}. \tag{16}$$

A bit of linear algebra shows that $\frac{1}{\|w\|}(x_i^{\top} w + b)$ is the signed distance of x_i from the decision boundary. Let $x^{\top} w + b = 0$ be a separating hyperplane and $y_i \in \{-1, +1\}$ codes a binary response for the i-th observation. Then d_{+} and (d_{-}) will be the shortest distance to the closest objects from the classes $+1$ and (-1). Since the separation can be done without errors, all observations $i = 1, 2, ..., n$ must satisfy:

$$\begin{array}{rcll} x_i^{\top} w + b & \geq & +1 & \text{for} \quad y_i = +1, \\ x_i^{\top} w + b & \leq & -1 & \text{for} \quad y_i = -1. \end{array}$$

We can combine both constraints into one as follows:

$$y_i(x_i^{\top} w + b) - 1 \geq 0 \qquad i = 1, \ldots, n. \tag{17}$$

Therefore the objective function of the linearly separable case would be maximizing (16) or equivalently,

$$\min_{w} \frac{1}{2} \|w\|^2, \tag{18}$$

under the constraint (17). The Lagrangian for the primal problem in this case is:

$$\min_{w,b} L_P\,(w, b) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \{y_i(x_i^{\top} w + b) - 1\}. \tag{19}$$

The Karush-Kuhn-Tucker (KKT) (Gale *et al.*, 1951) first order optimality conditions are:

$$\begin{array}{rcl} \frac{\partial L_P}{\partial w_k} = 0 & : & w_k - \sum_{i=1}^{n} \alpha_i y_i x_{ik} = 0, \quad k = 1, ..., d, \\ \frac{\partial L_P}{\partial b} = 0 & : & \sum_{i=1}^{n} \alpha_i y_i = 0, \end{array}$$

$$\begin{array}{rcl} y_i(x_i^{\top} w + b) - 1 & \geq & 0, \quad i = 1, \ldots, n, \\ \alpha_i & \geq & 0, \\ \alpha_i\{y_i(x_i^{\top} w + b) - 1\} & = & 0. \end{array}$$

From these first order conditions, we can derive $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ and therefore the summands in (19) would be

$$\begin{array}{rcl} \frac{1}{2}\|w\|^2 & = & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^{\top} x_j \\ \sum_{i=1}^{n} \alpha_i\{y_i(x_i^{\top} w + b) - 1\} & = & \sum_{i=1}^{n} \alpha_i y_i x_i^{\top} \sum_{j=1}^{n} \alpha_j y_j x_j - \sum_{i=1}^{n} \alpha_i \\ & = & \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^{\top} x_j - \sum_{i=1}^{n} \alpha_i \end{array}$$

Substituting this into (19), we obtain the Lagrangian for the dual problem:

$$L_D\,(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^{\top} x_j. \tag{20}$$

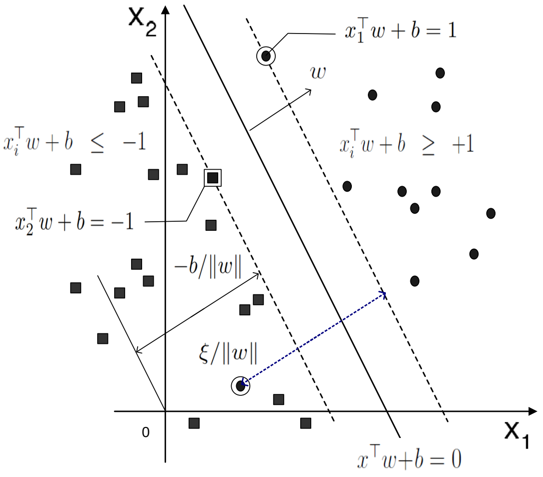


Figure 4: The separating hyperplane $x^\top w + b = 0$ and the margin in the linearly non-separable case.

Solving the primal and dual problems

$$\min_{w,b} L_P\,(w,b)$$

$$\max_{\alpha} L_D\,(\alpha) \quad \text{s.t.} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0.$$

give the same solution since the optimization problem is convex.

Those points i for which the equation $y_i(x_i^\top w + b) = 1$ holds are called *support vectors*. In Figure 3 there are two support vectors that are marked in bold: one solid rectangle and one solid circle. Apparently, the separating hyperplane is defined only by the support vectors that hold the hyperplanes parallel to the separating one.

After "training the support vector machine", i.e. solving the dual problem above and deriving Lagrange multipliers (which are equal to 0 for non-support vectors) one can classify an object, for example a company. One uses the classification rule

$$g(x) = \operatorname{sign}\left(x^\top w + b\right), \tag{21}$$

where $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ and $b = -\frac{1}{2}\left(x_{+1} + x_{-1}\right) w$, with x_{+1} and x_{-1} are two support vectors belonging to different classes for which $y(x^\top w + b) = 1$ hold. The value of the classification function (the score of a company) can be computed as

$$f(x) = x^\top w + b. \tag{22}$$

Each score $f(x)$ uniquely corresponds to a probability of default (PD). The higher $f(x)$, the higher also the PD.

3.1 SVM in the linearly non-separable case

In the linearly non-separable case the situation is illustrated in Figure 4. The slack variables ξ_i represent the violation of strict separation that allow a point to be in the margin error, $0 \leq \xi_i \leq 1$, or to be misclassified, $\xi > 1$. In this case the following inequalities can be induced (from Figure 4):

$$\begin{array}{rcl} w+b & \geq & 1-\xi_i \quad \text{for} \quad y_i = 1, \\ w+b & \leq & -(1-\xi_i) \quad \text{for} \quad y_i = -1, \\ \xi_i & \geq & 0, \end{array}$$

which could be combined into two constraints as follows:

$$y_i(x_i^\top w + b) \geq \ 1-\xi_i \tag{23a}$$

$$\xi_i \geq \ 0. \tag{23b}$$

SVM classification again maximizes the margin given a family of classification functions $\mathcal{F}$.

The penalty for misclassification is related to the distance of a misclassified point x_i from the canonical hyperplane bounding its class. If $\xi_i > 0$, an error in separating the two sets occurs. The objective function corresponding to penalized margin maximization is then formulated as:

$$\min_{w,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i, \tag{24}$$

with constraints as in equation (23). This formulation is called *soft-margin* SVM introduced by Cortes and Vapnik (1995).

The parameter C characterizes the weight given to the classification errors. The minimization of the objective function with constraints (23a) and (23b) provides the highest possible margin in the case when classification errors are inevitable due to the linearity of the separating hyperplane. Under such a formulation the problem is convex.

Non-negative slack variables ξ_i allow points to be on the wrong side of their *soft margin* ($x_i^\top w + b = \pm 1$), as well as the separating hyperplane. Parameter C is cost parameter that controls the amount of overlap. If the data are linearly separable, then for sufficiently large C the solution (18) and (24) coincide. If the data are linearly non-separable as C increases the solution approaches the minimum overlap solution with largest margin, which is attained for some finite value of C (Hastie *et al.*, 2004).

The Lagrange function for the primal problem is:

$$L_P(w,b,\xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \{ y_i \left(x_i^\top w + b \right) - 1 + \xi_i \} - \sum_{i=1}^{n} \mu_i \xi_i, \tag{25}$$

where $\alpha_i \geq 0$ and $\mu_i \geq 0$ are Lagrange multipliers. The primal problem is formulated as:

$$\min_{w,b,\xi} L_P(w,b,\xi). \tag{26}$$

The first order conditions of the primal problem are given by

$$\frac{\partial L_P}{\partial w_k} = 0 : \qquad w_k - \sum_{i=1}^{n} \alpha_i y_i x_{ik} = 0, \tag{27a}$$

$$\frac{\partial L_P}{\partial b} = 0 : \qquad \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{27b}$$

$$\frac{\partial L_P}{\partial \xi_i} = 0 : \qquad C - \alpha_i - \mu_i = 0. \tag{27c}$$

with the following conditions for the Lagrange multipliers:

$$\alpha_i \geq 0, \tag{28a}$$

$$\mu_i \geq 0, \tag{28b}$$

$$\alpha_i \{ y_i (x_i^\top w + b) - 1 + \xi_i \} = 0, \tag{28c}$$

$$\mu_i \xi_i = 0. \tag{28d}$$

Note that $\sum_{i=1}^{n} \alpha_i y_i b = 0$, similar to the linear separable case. The primal problem translates into the dual problem as follows:

$$\begin{aligned} L_D(\alpha) &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\top x_j - \sum_{i=1}^{n}\alpha_i y_i x_i^\top \sum_{j=1}^{n}\alpha_j y_j x_j \\ &\quad + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n}\mu_i\xi_i \\ &= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\top x_j + \sum_{i=1}^{n}\xi_i\left(C - \alpha_i - \mu_i\right). \end{aligned}$$

Since the last term is equal to zero, we derive the dual problem as:

$$L_D(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\top x_j, \tag{29}$$

and the dual problem is posed as:

$$\max_{\alpha} L_D(\alpha), \tag{30}$$

subject to:

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n}\alpha_i y_i = 0. \tag{31}$$

The sample x_i for which $\alpha > 0$ (support vectors) are those points that are on the margin, or within the margin when a soft-margin is used. The support vector is often sparse and the level of sparsity (fraction of data serving as support vector) is an upper bound for the misclassification rate (Schölkopf and Smola, 2002).

3.2 Non linear classification

We have not made any assumptions on the domain $\mathcal{X}$ other than being a set. We need additional structure in order to study machine learning to being able to generalize to unobserved data points. Given some new point $x \in \mathcal{X}$, we want to predict the corresponding $y \in \mathcal{Y} = \{-1, 1\}$. By this we mean that we choose y such that (x, y) is in some sense similar to the training examples. To this end, we need similarity measures in $\mathcal{X}$ and in $\{-1, 1\}$. The latter is easy, as two target values can only be identical or different (Chen *et al.*, 2005).

For the former, we require a similarity measure, i.e. a so called *kernel* function k, given two examples x_i and x_j, which returns a real number characterizing their similarity.

$$k \in K \quad : \quad \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \tag{32}$$

$$(x_i, x_j) \longmapsto k(x_i, x_j). \tag{33}$$

A type of similarity measure that is of particular mathematical appeal is the dot product. The dot product of two vectors $x_i, x_j \in \mathbb{R}^n$ is defined as

$$x_i \cdot x_j = x_i^\top x_j := \sum_{\ell=1}^{n} (x_i)_\ell \, (x_j)_\ell \,. \tag{34}$$

In order to be able to use a dot product as a similarity measure, we need to transform them into some dot product space, so called *feature space* $\mathcal{H} \in \mathbb{H}$, which need not be identical to $\mathbb{R}^n$.

$$\psi : \mathcal{X} \to \mathcal{H}. \tag{35}$$

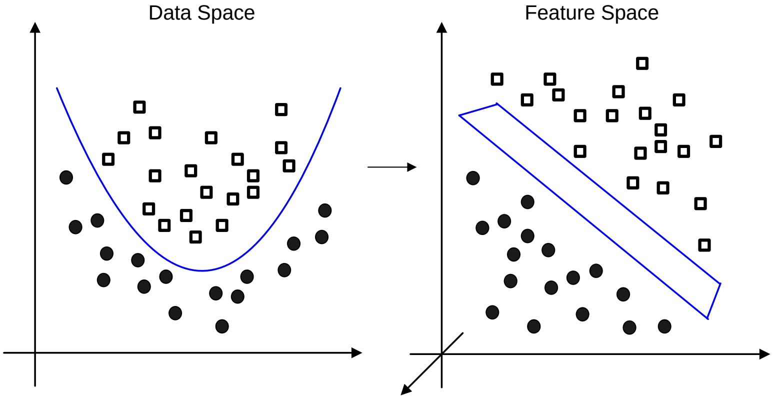


Figure 5: Mapping into a three dimensional feature space from a two dimensional data space $\mathbb{R}^2 \mapsto \mathbb{R}^3$. The transformation $\psi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top$ corresponds to the kernel function $K(x_i, x_j) = (x_i^\top x_j)^2$.

The SVMs can also be generalized to the nonlinear case. In order to obtain non-linear classifiers as in Figure 5 one maps the data with a non-linear structure via a function $\psi : \mathbb{R}^p \mapsto \mathbb{H}$ into a high dimensional space $\mathbb{H}$ where the classification rule is (almost) linear. Note that all the training vectors x_i appear in L_D (29) only as scalar products of the form $x_i^\top x_j$. In the nonlinear SVM situations this transforms to $\psi\left(x_i\right)^\top \psi\left(x_j\right)$.

The learning then takes place in the feature space, provided the learning algorithm can be expressed so that the data points only appear inside dot products with other points. This is often referred to as the *kernel trick* (Schölkopf and Smola, 2002). The *kernel trick* is to compute this scalar product via a kernel function. More precisely, the projection $\psi : \mathbb{R}^p \mapsto \mathbb{H}$ ensures that the inner product $\psi\left(x_i\right)^\top \psi\left(x_j\right)$ can be represented by kernel function

$$k(x_i, x_j) = \psi(x_i)^\top \psi(x_j). \tag{36}$$

If a kernel function k exists such that (36) holds, then it can be used without knowing the transformation ψ explicitly.

Given a kernel k and any data set $x_1, ..., x_n \in \mathcal{X}$ then the $n \times n$ matrix

$$K = k((x_i, x_j))_{ij}, \tag{37}$$

is called the kernel or *Gram* matrix of k with respect to $x_1, ..., x_n$. A necessary and sufficient condition for a symmetric matrix K, with $K_{ij} = K(x_i, x_j) = K(x_j, x_i) = K_{ji}$, to be a kernel is, by Mercer's theorem (Mercer, 1909), that K is positive definite:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j K(x_i, x_j) \geq 0. \tag{38}$$

The following is a simple example of a kernel trick. To obtain the discriminant function $f(x) = w^\top \psi(x) + b$, consider the case of a two-dimensional input space with mapping function given by a vector in terms of all degree-2 monomials,

$$\psi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top,$$

such that

$$w^\top \psi(x) = w_1x_1^2 + \sqrt{2}w_2x_1x_2 + w_3x_2^2.$$

The dimensionality of the feature space $\mathcal{F}$ is of quadratic order of the dimensionality of the original space. This quadratic complexity is feasible for low dimensional data. Kernel methods avoid the step of explicitly mapping the data into a high dimensional feature-space by the following steps

$$\begin{aligned} f(x) &= w^\top x + b \\ &= \sum_{i=1}^{n} \alpha_i x_i^\top x + b \\ &= \sum_{i=1}^{n} \alpha_i \psi(x_i)^\top \psi(x) + b \quad \text{in feature space } \mathcal{F} \\ &= \sum_{i=1}^{n} \alpha_i k(x_i, x) + b \end{aligned}$$

where the kernel associated with this mapping

$$\begin{aligned} \psi(x)^\top \psi(z) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(z_1^2, \sqrt{2}z_1z_2, z_2^2)^\top \\ &= x_1^2z_1^2 + 2x_1x_2z_1z_2 + x_2^2z_2^2 \\ &= (x^\top z)^2 \\ &= k(x, z) \end{aligned}$$

This example shows that the kernel can be computed without computing explicitly the mapping function ψ.

Furthermore, to obtain non-linear classifying functions in the data space, a more general form is obtained by applying the kernel trick to (29) as follows:

$$\max_{\alpha} L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \tag{39}$$

subject to:

$$0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, n, \tag{40a}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{40b}$$

One of the most popular kernels used in SVM is the Radial Basis Function (RBF) kernel given by

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right). \tag{41}$$

The benefits of transforming the data into the feature space $\mathcal{H}$ (Chen *et al.*, 2005) summarize:

1. It lets us define a similarity measure from the dot product in $\mathcal{H}$,

$$k(x_i, x_j) := x_i^\top x_j = \psi(x_i)^\top \psi(x_j). \tag{42}$$

2. It allows us to deal with the patterns geometrically, and thus lets us study learning algorithms using linear algebra and analytical geometry.

3. The freedom to choose the mapping ψ will enable us to design a large set of learning algorithms. Consider a situation where the input already lives in a dot product space, in which case we could directly define a similarity measure as the dot product. However, we might still choose to first apply a non-linear mapping ψ to change the representation into one that is more suitable for a given problem and learning algorithm.

The resulting optimisation problems (39), which is a typical quadratic problem (QP), are dependent upon the number of training examples. The problem can easily be solved in a standard QP solver, i.e. package `quadprog` in `R` (Weingessel, 2004) or an optimizer of the interior point family ((Vanderbei, 1999) and (Schölkopf and Smola, 2002)) implemented to `ipop` in package `kernlab` in `R` (Karatzoglou *et al.*, 2005).

Osuna *et al.* (1997b) proposed exact methods by presenting a decomposition algorithm that is guaranteed to solve QP problem and that does not make assumptions on the expected number of support vectors. Platt (1998) proposed a new algorithm called Sequential Minimal Optimization (SMO) which decomposes the QP in SVM without using any numerical QP optimization steps. Some work on decomposition methods for QP in SVM was done by, for example, Joachims (1999), Keerthi *et al.* (2001), Hsu and Lin (2002). Subsequent developments were achieved by Fan *et al.* (2005) as well as Glasmachers and Igel (2006).

Due to the fast development and wide applicability, the existence of many SVM software routines is not surprising. The SVM software which is written in `C` or `C++` are `SVMTorch` (Collobert *et al.*, 2002), `SVMlight` (Joachims, 1999), `Royal Holloway Support Vector Machines` (Gammerman *et al.*, 2001), `libsvm` (Chang and Lin, 2001) which provides interfaces to MATLAB, `mySVM` (Rüping, 2004) and `M-SVM` (Guermeur, 2004). The SVM is also available in `MATLAB` (Gunn (1998), Canu *et al.* (2003) and Schwaighofer (2005)). Several packages in `R` dealing with SVM are `e1071` (Dimitriadou *et al.*, 1995), `kernlab` (Karatzoglou *et al.*, 2004), `svmpath` (Hastie *et al.*, 2004) and `klaR` (Roever *et al.*, 2005).

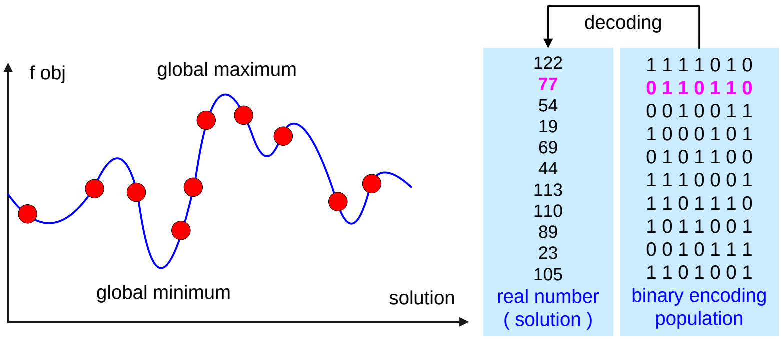
SVM recently has been developed by many researchers in various fields of application, i.e. Least Squares SVM (Suykens and Vandewalle, 1999), Smooth SVM or SSVM (Lee and Mangasarian, 2001), 1-norm SVM (Zhu *et al.*, 2004), Reduced SVM (Lee and Huang, 2007) and ν-SVM (Schölkopf *et al.* (2000) and Chen *et al.* (2005)). Hastie *et al.* (2004) viewed SVM as a regularised optimisation problem.

4 Evolutionary feature selection

During the learning process (training), an SVM finds the large margin hyperplane by estimating sets of parameters α_i and b. The SVM performance is also determined by another set of paramaters, the so-called *hypermarameters*: These are the soft margin constant C and the parameters of the kernel, σ, as in (41). The value of C determines the size of the margin errors. The kernel parameters control the flexibility of the classifier. If this complexity parameter is too large, then overfitting will occur.

Hastie *et al.* (2004) argue that the choice of the cost parameter (C) can be critical. They derive an algorithm, so-called `SvmPath`, that can fit the entire path of SVM solutions for every value of the cost parameter, with essentially the same computational cost as fitting one SVM model. The `SvmPath` has been implemented in the `R` computing environment via the library `svmpath`. Chen *et al.* (2011) use grid search methods to optimize SVM hyperparamaters to obtain the optimal classifier for a credit dataset. This chapter employs a Genetic Algorithm (GA) as an evolutionary algorithm to optimise the SVM parameters.

GA is an iterative procedure which follows the evolution of a population of individuals through successive generations. The idea of GA is based on the principle of *survival of the fittest.* Living beings are constituted by cells, with specialized tasks, which carry the genetic information of the whole individual. Each cell contains a fixed number of chromosomes composed by several genes. A gene is a piece of elementary information which may be conceptualized as a binary code. All information carried by genes of all chromosomes (the genotype) determines all characteristics of an individual (the phenotype). Each individual is evaluated to give measures of its fitness. Some individual undergo stochastic transformations by means of genetic operations to form a new individual. There are two types of transformation: *mutation* and *crossover* or *recombination.* Mutation creates a new individual by making changes in a single chromosome.



real number (solution)	binary encoding population
122	1 1 1 1 0 1 0
77	0 1 1 0 1 1 0
54	0 0 1 0 0 1 1
19	1 0 0 0 1 0 1
69	0 1 0 1 1 0 0
44	1 1 1 0 0 0 1
113	1 1 0 1 1 1 0
110	1 0 1 1 0 0 1
89	0 0 1 0 1 1 1
23	1 1 0 1 0 0 1
105	

Figure 6: Generating binary encoding chromosomes to obtain the global optimum solution through GA.

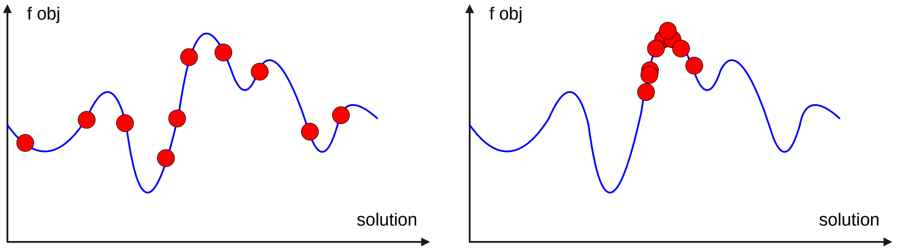


Figure 7: GA convergency: solutions at $1-st$ generation (left) and $g-th$ generation (right).

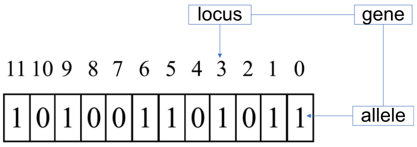


Figure 8: Chromosome.

Crossover creates new individuals by combining parts from two individuals represented by their chromosomes. Chromosomes are ordered in pairs, and when sexual reproduction takes place, children (new chromosome) or offspring receive, for each pair, one chromosome from each of their parents (old chromosomes). The children are then evaluated. A new population is formed by selecting fitter individuals from the parent population and the children population. After several generations (iteration), the algorithm converges to the best individual, which hopefully represents a (globally) optimal solution to the problem (Baragona *et al.* (2011) and Gen and Cheng (2000)).

A binary string chromosome is composed of several genes. Each gene has a binary value (*allele*) and its position (*locus*) in a chromosome as shown in Figure 8. The binary string is decoded to the real number in a certain interval by the following equation

$$\theta = \theta_{lower} + (\theta_{upper} - \theta_{lower})\frac{\sum_{i=0}^{l-1} a_i 2^i}{2^l} \tag{43}$$

where θ is the solution (i.e. parameter C or σ), a is binary value (*allele)* and l is the chromosome length. In the encoding issue, according to what kind of symbol is used as the alleles of a gene, the encoding methods can be classified as follows: *binary* encoding, *real-number* encoding, *integer* or *literal permutation* encoding and *general data structure* encoding.

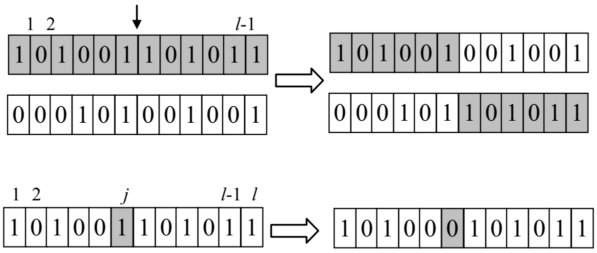


Figure 9: One-point crossover (top) and bit-flip mutation (bottom).

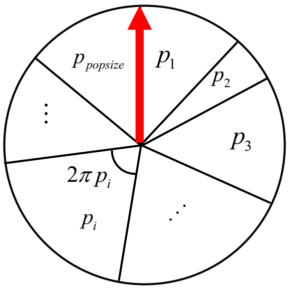


Figure 10: Probability of i-th chromosome to be selected in the next iteration (generation)

The current solution is evaluated to measure the fitness performance based on discriminatory power (AR or AUC), $f^*(C, \sigma)$. The next generation results from the reproduction process articulated in three stages of selection, crossover and mutation (Fig. 9). The selection step is choosing which chromosomes of the current population are going to reproduce. The most fitted chromosome should reproduce more frequently than the less fitted one.

If f_i^* is the fitness of i-th chromosome, then its probability of being selected (relative fitness) is

$$p_i = \frac{f_i^*}{\sum_{i=1}^{popsize} f_i^*}, \tag{44}$$

where *popsize* is the number of chromosomes in the population or population size. The *roulette wheel* method selects a chromosome with probability proportional to its fitness, see Fig. 10. To select the new chromosome, generate a random number $u \sim \mathrm{U}(0, 1)$, then select i-th chromosome if $\sum_{i=1}^{t} p_i < u < \sum_{i=1}^{t+1} p_i$, where $t = 1, \ldots, (popsize - 1)$. Repeat *popsize* times to get new population. The other popular selection operators are *stochastic universal sampling*, *tournament selection*, steady-state reproduction, sharing, ranking and scaling.

The selection stage produces candidates for reproduction (iteration). Randomly chosen pairs of chromosomes mate and produce a pair of offspring that may share genes of both parents. This process is called crossover (with fixed probability). One-point crossover can be extended to two-point or more crossover. Afterwards, the offspring is subject to the mutation operator (with small probability). Mutation introduces innovations into the population that cause the trapped local solutions to move out. The relationship of GA with evolution in nature is given in Table 3. Figure 11 shows how GA is applied to SVM optimization.

A too high crossover rate may lead to premature convergence of the GA as well as a too high mutation rate may lead to the loss of good solutions unless there is elitist selection. In elitism, the best solution in each iteration is maintained in another memory. When the new population will replace the old one, check whether best solution exists in the new population. If not, replace any chromosomes in the new population with the best solution we saved in another memory.

Nature	GA-SVM
Population	Set of parameters
Individual (phenotype)	Parameters
Fitness	Discriminatory power
Chromosome (genotype)	Encoding of parameter
Gene	Binary encoding
Reproduction	Crossover
Generation	Iteration

Table 3: Nature to GA-SVM mapping.

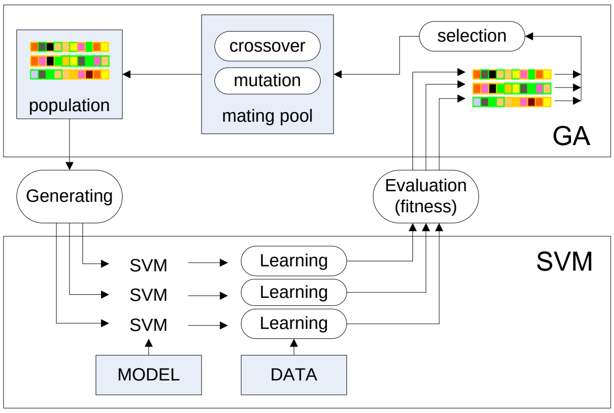


Figure 11: Iteration (generation) procedure in GA-SVM.

It is natural to expect that the adaptation of GA is not only for finding solutions, but also for tuning GA to the particular problem. The adaptation of GA is to obtain an effective implemetation of GA to real-world problems. In general, there are two types of adaptations: Adaptation to problems and adaptation to evolutionary processes (see Gen and Cheng (2000) for details).

5 Application

The SVM with evolutionary feature selection is applied to the CreditReform database consisting of $20,000$ solvent and $1,000$ insolvent German companies in the period from 1996 to 2002. Approximately 50% of the data are from the years 2001 and 2002. Table 4 describes the composition of the CreditReform database in terms of industry sectors. In our study, we only used the observations from the following industry sectors: manufacturing, wholesale and retail, construction, and real estate.

We excluded the observations of solvent companies in 1996 because of missing insolvencies in this year. The observations with zero values in those variables which were used as denominator to compute the financial ratios were also deleted. We also excluded the companies whose total assets were not in the range EUR $10^5 - 10^7$. We replace the extreme financial ratio values by the following rule: if $x_{ij} > q_{0.95}(x_j)$ then $x_{ij} = q_{0.95}(x_j)$ and if $x_{ij} < q_{0.05}(x_j)$ then $x_{ij} = q_{0.05}(x_j)$, where q is quartile. Table 5 describes the filtered data used in this study.

We predict the default based on 28 financial ratio variables as used in Chen *et al.* (2011) and Härdle *et al.* (2009). The GA was employed as an evolutionary feature selection of SVM. The population size is 20 chromosomes. We used a fixed number of iterations (generations)

type	solvent (%)	insolvent (%)	total (%)
Manufacturing	26.06	1.22	27.29
Construction	13.22	1.89	15.11
Wholesale and retail	23.60	0.96	24.56
Real estate	16.46	0.45	16.90
total	79.34	4.52	83.86
others	15.90	0.24	16.14

Table 4: Credit reform data based on industry sector.

year	solvent	insolvent	total
	number (%)	number (%)	number (%)
1997	872 (9.08)	86 (0.90)	958 (9.98)
1998	928 (9.66)	92 (0.96)	1020 (10.62)
1999	1005 (10.47)	112 (1.17)	1117 (11.63)
2000	1379 (14.36)	102 (1.06)	1481 (15.42)
2001	1989 (20.71)	111 (1.16)	2100 (21.87)
2002	2791 (29.07)	135 (1.41)	2926 (30.47)
total	8964 (93.36)	638 (6.64)	9602 (100)

Table 5: Filtered credit reform data.

Training	Training error (%)			Testing	Testing error (%)		
	DA	Logit	Probit		DA	Logit	Probit
1997	10.01	0	0	1998	9.13	9.00	8.88
1998	9.25	0	0	1999	11.08	10.82	10.82
1999	10.43	0	0	2000	9.20	9.31	9.31
2000	8.62	0	0	2001	6.86	7.78	7.78
2001	6.64	0	0	2002	7.95	7.16	7.16

Table 6: Percentage of training error and testing error from discriminant analysis, logit and probit model.

as a termination criterion. The number of generations is fixed at 100 with crossover rate 0.5, mutation rate 0.1 and elitism rate 0.2 of the population size. The obtained optimal parameters of GA-SVM are given by $\sigma = 1/178.75$ and $C = 63.44$.

We use classical methods such as discriminan analysis (DA), logit and probit models as benchmark (Table 6). Discriminant analysis shows a poor performance in both training and testing dataset. The financial ratios variables are collinear such that the assumptions in DA are violated. Logit and probit model show a perfect classification in training dataset with several variables are not significant. The best models of logit and probit, by excluding the nonsignificant variables, still show not significant different from as if we use the whole variables.

The GA-SVM yields also a perfect classification in the training dataset as in Table 7 which shows an overfitting. Overfitting means that the classification boundary is too curved, therefore has less ability to classify the unobserved data (i.e. testing data) correctly. The misclassification is zero for all training data such that the other discriminatory power measures, Acc, $Spec$, $Sens$, $Prec$, AR and AUC, attain one. A 5-fold cross-validation was used to measure the performance of GA-SVM in default prediction by omitting the overfitting effect. On overall, GA-SVM is outperforms compared to the benchmark models in both training and testing dataset.

Training	Training error (%)	$Acc, Spec, Sens$ $Prec, AR, AUC$	Cross validation	Testing	Testing error (%)
1997	0	1	9.29	1998	9.02
1998	0	1	9.22	1999	10.38
1999	0	1	10.03	2000	6.89
2000	0	1	8.57	2001	5.29
2001	0	1	4.55	2002	4.75

Table 7: Percentage of training error, discriminatory power, cross validation (5-fold) and testing error.

Acknowledgement

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 'Economic Risk', Humboldt-Universität zu Berlin. Dedy Dwi Prastyo was also supported by a scholarship from Directorate General for Higher Education, Indonesian Ministry of Education and Cultur through Department of Statistics, Institut Teknologi Sepuluh Nopember (ITS), Indonesia.

References

Altman, E. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance* **23(4)**: 589-609.

Altman, E., Marco, G. and Varetto, F. (1994). Corporate distress diagnosis: comparison using linear discriminant analysis and neural network (the Italian experience). *Journal of Banking and Finance* **18**: 505-529.

Baragona, R., Battaglia, F. and Poli, I. (2011). *Evolutionary Statistical Procedures.* Springer: Heidelberg.

Beaver, W. (1966). Financial ratios as predictors of failures. Empirical research in accounting: selected studies, *Journal of Accounting Research* suplement to vol. 5: 71-111.

Black, F. and Scholes, M. (1973). The pricing of option and corporate liabilities. *The Journal of Political Economy* **81(3)**: 637-654.

Blanz, V., Schölkopf, B., Bülthoff, H., Burges, C., Vapnik, V., and Vetter, T. (1996). Comparison of view-based object recognition algorithm using realistic 3d models. *Artificial Neural Networks - ICANN'96*

Boser, B.E., Guyon, I.M. and Vapnik, V. (1992). A Training Algorithm for optimal margin classifier. In D. Haussler, editor, *5th annual ACM Workshop on COLT.* pages 144-152, ACM Press: Pittsburgh.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* **2**: 121-167.

Burges, C. and Schölkopf, B. (1996). Improving the accuracy and speed of support vector learning machines. In M. Mozer , M. Jordan, and T. Petsche, editors, *Advance in Neural Information Processing System 9.* pages 375-381, MIT Press: Cambridge.

Canu, S., Grandvalet, Y. and Rakotomamonjy, A. (2003). SVM and kernel methods MATLAB toolbox, Perception Systemes et Information, INA de Rouen, Rouen, France. URL http://asi.insa-rouen.fr/˜arakotom/toolbox/index.

Chen, S., Härdle, W. and Moro, R. (2011). Estimation of default probabilities with support vector machines *Quantitative Finance* **11**: 135-154.

Chang, C.C. and Lin, C.J., (2001). libsvm: a library for support vector machines. URL http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

Chen, P-H., Lin, C-J. and Schölkopf, B. (2005). A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry* **21**: 111-136.

Collobert, R., Bengio, S. and Mariethoz, J. (2002). Torch: A Modular Machine Learning Software Library. URL http://www.torch.ch/ and http://publications.idiap.ch/downloads/reports/2002/rr02-46.pdf

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning* **20**: 273-297.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. and Weingessel, A. (1995). e1071: misc functions of the department of statistics (e1071). TU Wien, Version 1.5-11., URL http://CRAN.R-project.org/.

Engelmann, B., Hayden, E. and Tasche, D. (2003). Measuring the discriminative power of rating system. *Discussion Paper* **2(1)**: Banking and Financial Supervision

Fan, D.R-E., Chen, P.-H. and Lin, C.-J. (2005). Working set selection using second order information for training SVM. *Journal of Machine Learning Research* **6**: 1889-1918.

Gammerman, A., Bozanic, N., Schölkopf, B., Vovk, V., Vapnik, V., Bottou, L., Smola, A., Watkins, C., LeCun, Y., Saunders, C., Stitson, M. and Weston, J. (2001). Royal holloway support vector machines. URL http://svm.dcs.rhbnc.ac.uk/dist/index.shtml.

Gale, D., Kuhn, H.W. and Tucker, A.W. (1951). Linear programming and the theory of games. *Proceeding: Activity Analysis of Production and Allocation*edited by Koopmans, T.C.: 317-329. John Wiley & Sons: New York.

Gen, M. and Cheng, R. (2000). *Genetic algorithms and engineering design.* John Willey & Sons, Inc.: New York.

Glasmachers, T. and Igel, C. (2006). Maximum-gain working set selection for support vector machines. *Journal of Machine Learning Research* **7**: 1437-1466.

Guermeur, Y. (2004). M-SVM. Lorraine laboratory of IT research and its applications. URL http://www.loria.fr/˜guermeur/.

Gunn, S. R., (1998). Support vector machines for classification and regression. *Technical Report.* UNSPECIFIED, Dept. of Electronics and Computer Science, University of Southampton

Hastie, T., Rosset, S., Tibshirani, R. and Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research* **5**: 1391-1415.

Haupt, R.L. and Haupt, S.E. (2004). *Practical Genetic Algorithms, 2nd ed.* John Wiley & Sons, Inc.: New Jersey.

Härdle, W., Hoffmann, L. and Moro, R. (2011). *Learning machines supporting bankruptcy prediction.* In Cizek, P., Härdle, W., Weron, R., editors, Statistical Tools for Finance and Insurance. 2nd ed., Springer Verlag: Heidelberg.

Härdle, W., Lee, Y.-J., Schäfer, D. and Yeh, Y.-R. (2009). Variable selection and oversampling in the use of smooth support vector machine for predicting the default risk of companies. *Journal of Forecasting* **28(6)**: 512-534.

Härdle, W. and Simar, L. (2012). *Applied multivariate statistical analysis. 3rd ed.* Springer Verlag: Heidelberg.

Haykin, S. (1999). *Neural network: a comprehensive foundation.* Prentice-Hall: Engelwood Cliffs.

He, H. and Garcia, E.A. (2009). Learning from imbalanced data. *IEEE transaction on Knowledge and Data Engineering* **21(9)**: Sep.

Holland, J.H. (1975). *Adaptation in natural and artificial systems.* University of Michigan Press.

Hsu, C.-W. and Lin, C.-J. (2002). A simple decomposition method for support vector machines. *Machine Learning* **46**: 291-314.

Hwang, RC., Cheng, K.F. and Jee, J.C. (2007). On the pricing of corporate debt: the risk structure of interest rates. *The Journal of Finance* **29**: 449-470.

Japkowicz, N. and Stephen, S. (2002). The class imbalanced problem: a systematic study. *Intelligent Data Analysis* **6(5)**: 429-449.

Joachims, T. (1998). Making large-scale SVM learning practical. In In Schölkopf, B. Burges, J.C., and Smola, A.J., editors, *Advances in Kernel Methods - Support Vector Learning.* pages 169-184, MIT Press: Cambridge.

Karatzoglou, A., Smola, A., Hornik, K. and Zeileis, A. (2005). kernlab kernel methods. R package, Version 0.6-2. URL http://CRAN.R-project.org/.

Karatzoglou, A., Smola, A., Hornik, K. and Zeileis, A. (2004). Kernlab an S4 package for kernel methods in R. *Journal of Statistical Software* **11(3)**.

Krahnen, J.P. and Weber, M. (2001). Generally accepted rating principles: a primer. *Journal of Banking and Finance* **209**: 415-446.

Keerthi, S.S., Shevade, S.K., Bhattacharya, C. and Murthy, K.R.K., (2000). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation* **13**: 637-649.

Lee, Y.-J. and Huang, S.-Y. (2007). Reduced support vector machines: a statistical theory. *IEEE Transactions on Neural Networks* **18(1)**: 1-13.

Lee, Y.-J. and Mangasarian, O.L. (2001). SSVM: A smooth support vector machine for classification. *Computational Optimization and Application* **20(1)**: 5-22.

Lo, A.W., (1986). Logit versus discriminant analysis: a specification test and application to corporate bankruptcies. *Journal Econometrics* **31(2)**: 151-178.

Maalouf, M. and Trafalis, T.B. (2011). Robust weighted kernel logistic regression in imbalanced and rare event data. *Computational Statistics and Data Analysis* **55**: 168-183.

Martin, D. (1977). Early warning of bank failure: a logit regression approach. *Journal of Banking and Finance* **1**: 249-276.

Mitchell, M. (1999). *An Introduction to Genetic Algorithms.* MIT Press: Massachusetts.

Merton, R. (1974). On the pricing of corporate debt: the risk structure of interest rates. *The Journal of Finance* **29**: 449-470.

Merwin, C. (1942). Financing small corporations in five manufacturing industries. *The Journal of Finance* 1926-36.

Michalewicz, Z. (1996). *Genetics Algorithm + Data Structures = Evolution Programs, 3rd ed.* Springer: New York.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London* **25**: 3-23.

Müller, K.-R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J. and Vapnik, V. (1997). Predicting time series with support vector machines. In Proceedings. *International Conference on Artificial Neural Networks* page 999. Springer Lecture Notes in Computer Science

Ohlson, J. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research* Spring: 109-131.

Osuna, E., Freund, R. and Girosi, F. (1997). Training support vector machines: an application to face detection. *Proc. IEEE Conference on Computer Vision and Pattern Recognition* pages 130-136.

Osuna, E., Freund, R. and Girosi, F. (1997). Improved training algorithm for support vector machines. In Principe, J., Gile, L., Morgan, N. and Wilson, E., In Neural Networks for Signal Processing VII. *Proceedings of the 1997 IEEE Workshop.* pp. 276 - 285, New York.

Platt, H., Platt, M. and Pedersen, J. (1994). Bankruptcy discrimination with real variables. *Journal of Business Finance and Accounting* **21:4**: 491-510.

Platt, J.C. (1998). Fast training of support vector machines using sequential minimal optimization, In Schölkopf, B. Burges, J.C., and Smola, A.J., editors. *Advances in Kernel Methods - Support Vector Learning.* MIT Press: Cambridge.

Ramser, J. and Foster, L.A. (1931). A demonstration of ratio analysis. *Bureau of Business Research*, **40**. University of Illinois.

Roever, C., Raabe, N., Luebke, K. and Ligges, U. (2005). klaR - classification and visualization. R package, Version 0.4-1. URL http://CRAN.R-project.org/.

Rüping, S. (2004). mySVM A Support Vector Machine. University of Dortmund, Computer Science. URL http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html.

Schölkopf, B., Burges, C. and Vapnik, V. (1995). Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceeding, First International Conference on Konwledge Discovery and Data Mining.* AAAI Press, Menlo Park, CA.

Schölkopf, B., Burges, C. and Vapnik, V. (1996). Incorporating invariances in support vector learning machines. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Artificial Neural Networks - ICANN'96.* pages 47-52, Berlin. Springer Lecture Note in Computer Science, Vol. 1112.

Schölkopf, B., Smola, A.J., Williamson, R.C. and Bartlett, P.L. (1999). New support vector algorithm. *Neural Computation* **12**: 1207-1245.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press: Massachusetts.

Schwaighofer, A. (2005). SVM Toolbox for MATLAB, Intelligent Data Analysis group (IDA). Fraunhofer FIRST, URL http://ida.first.fraunhofer.de/˜anton/software.html.

Shawe-Taylor, J., Bartlett, P.L., Williamson, R.C. and Anthony, M. (1998). A framework for structural risk minimization. *In Proceedings 9th Annual Conference on Computational Learning Theory* pages 6876.

Sing, T. Sander, O. Beerenwinkel, N. and Lengauer, T. (2005). ROCR: visualizing classifier performance in R. *Bioinformatics* **21(20)**: 3940-3941.

Sivanandam, S.N. and Deepa, S.N. (2008). *Introduction to Genetic Algorithms.* Springer-Verlag: Heidelberg.

Sobehart, J. and Stein, R. (2000). *Moody's public firm risk model: a hybrid approach to modeling short term default risk.* Moody Investors Service, Rating Methodology.

Sobehart, J., Keenan, S. and Stein, R. (2000). *Benchmarking Quantitative Default Risk Models: A Validation Methodology.* Moody Investors Service.

Sobehart, J. and Keenan, S. (2001). Measuring default accurately, credit risk special report. *Risk* **14**: 31-33.

Suykens, J.A.K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters* **9(3)**: 193-300.

Tam, K. and Kiang, M. (1992). Managerial application of neural networks: the case of bank failure prediction. *Management Science* **38**: 926-947.

Vanderbei, R. (1999). LOQO: An interior point code for quadratic programming. *Optimization Methods and Software* **12**: 251-484.

Vapnik, V. (1979). *Estimation of Dependencies Based on Empirical Data.* Nauka: Moscow.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer Verlag: New York.

Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley: New York.

Vassalou, M. and Xing, Y. (2004). Default risk in equity returns. *The Journal of Finance* **19(2)**: 831 - 868.

Weingessel, A. (2004). quadprog functions to solve quadratic programming problems. R package, Version 1.4-7. URL http://CRAN.R-project.org/.

Wilson, R.L. and Sharda, R. (1994). Bankruptcy prediction using neural network. *Decision Support System* **11(5)**: 545 - 557.

Winakor, A. and Smith, R. (1935). Changes in the financial structure of unsuccessful industrial corporations. *Bureau of Business Research* **51**, University of Illinois.

Zhang, J. L. and Härdle, W. (2010). The bayesian additive classification tree applied to credit risk modelling. *Computational Statistics and Data Analysis* **54**: 1197-1205.

Zhu, J., Rosset, R., Hastie, T.and Tibshirani, R. (2004). 1-norm support vector machine, In *Proceeding of Advances in Neural Information Processing System 16.*

SFB 649 Discussion Paper Series 2012

For a complete list of Discussion Papers published by the SFB 649, please visit http://sfb649.wiwi.hu-berlin.de.

001 "HMM in dynamic HAC models" by Wolfgang Karl Härdle, Ostap Okhrin and Weining Wang, January 2012.

002 "Dynamic Activity Analysis Model Based Win-Win Development Forecasting Under the Environmental Regulation in China" by Shiyi Chen and Wolfgang Karl Härdle, January 2012.

003 "A Donsker Theorem for Lévy Measures" by Richard Nickl and Markus Reiß, January 2012.

004 "Computational Statistics (Journal)" by Wolfgang Karl Härdle, Yuichi Mori and Jürgen Symanzik, January 2012.

005 "Implementing quotas in university admissions: An experimental analysis" by Sebastian Braun, Nadja Dwenger, Dorothea Kübler and Alexander Westkamp, January 2012.

006 "Quantile Regression in Risk Calibration" by Shih-Kang Chao, Wolfgang Karl Härdle and Weining Wang, January 2012.

007 "Total Work and Gender: Facts and Possible Explanations" by Michael Burda, Daniel S. Hamermesh and Philippe Weil, February 2012.

008 "Does Basel II Pillar 3 Risk Exposure Data help to Identify Risky Banks?" by Ralf Sabiwalsky, February 2012.

009 "Comparability Effects of Mandatory IFRS Adoption" by Stefano Cascino and Joachim Gassen, February 2012.

010 "Fair Value Reclassifications of Financial Assets during the Financial Crisis" by Jannis Bischof, Ulf Brüggemann and Holger Daske, February 2012.

011 "Intended and unintended consequences of mandatory IFRS adoption: A review of extant evidence and suggestions for future research" by Ulf Brüggemann, Jörg-Markus Hitz and Thorsten Sellhorn, February 2012.

012 "Confidence sets in nonparametric calibration of exponential Lévy models" by Jakob Söhl, February 2012.

013 "The Polarization of Employment in German Local Labor Markets" by Charlotte Senftleben and Hanna Wielandt, February 2012.

014 "On the Dark Side of the Market: Identifying and Analyzing Hidden Order Placements" by Nikolaus Hautsch and Ruihong Huang, February 2012.

015 "Existence and Uniqueness of Perturbation Solutions to DSGE Models" by Hong Lan and Alexander Meyer-Gohde, February 2012.

016 "Nonparametric adaptive estimation of linear functionals for low frequency observed Lévy processes" by Johanna Kappus, February 2012.

017 "Option calibration of exponential Lévy models: Implementation and empirical results" by Jakob Söhl und Mathias Trabs, February 2012.

018 "Managerial Overconfidence and Corporate Risk Management" by Tim R. Adam, Chitru S. Fernando and Evgenia Golubeva, February 2012.

019 "Why Do Firms Engage in Selective Hedging?" by Tim R. Adam, Chitru S. Fernando and Jesus M. Salas, February 2012.

020 "A Slab in the Face: Building Quality and Neighborhood Effects" by Rainer Schulz and Martin Wersing, February 2012.

021 "A Strategy Perspective on the Performance Relevance of the CFO" by Andreas Venus and Andreas Engelen, February 2012.

022 "Assessing the Anchoring of Inflation Expectations" by Till Strohsal and Lars Winkelmann, February 2012.

SFB 649, Spandauer Straße 1, D-10178 Berlin
http://sfb649.wiwi.hu-berlin.de

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk".

SFB 649 Discussion Paper Series 2012

For a complete list of Discussion Papers published by the SFB 649, please visit http://sfb649.wiwi.hu-berlin.de.

023 "Hidden Liquidity: Determinants and Impact" by Gökhan Cebiroglu and Ulrich Horst, March 2012.

024 "Bye Bye, G.I. - The Impact of the U.S. Military Drawdown on Local German Labor Markets" by Jan Peter aus dem Moore and Alexandra Spitz-Oener, March 2012.

025 "Is socially responsible investing just screening? Evidence from mutual funds" by Markus Hirschberger, Ralph E. Steuer, Sebastian Utz and Maximilian Wimmer, March 2012.

026 "Explaining regional unemployment differences in Germany: a spatial panel data analysis" by Franziska Lottmann, March 2012.

027 "Forecast based Pricing of Weather Derivatives" by Wolfgang Karl Härdle, Brenda López-Cabrera and Matthias Ritter, March 2012.

028 “Does umbrella branding really work? Investigating cross-category brand loyalty” by Nadja Silberhorn and Lutz Hildebrandt, April 2012.

029 “Statistical Modelling of Temperature Risk” by Zografia Anastasiadou, and Brenda López-Cabrera, April 2012.

030 “Support Vector Machines with Evolutionary Feature Selection for Default Prediction” by Wolfgang Karl Härdle, Dedy Dwi Prastyo and Christian Hafner, April 2012.

SFB 649, Spandauer Straße 1, D-10178 Berlin
http://sfb649.wiwi.hu-berlin.de

This research was supported by the Deutsche
Forschungsgemeinschaft through the SFB 649 "Economic Risk".