PREPRINT

Kernel Methods in Finance

by

Stephan K. Chalup and Andreas Mitschele

Citation:

Stephan K. Chalup and Andreas Mitschele. **Kernel Methods in Finance**. Chapter 27, pages 655-687 of: Detlef Seese, Christof Weinhardt and Frank Schlottmann. Handbook on Information Technology in Finance. Springer Berlin Heidelberg, 2008.

BibTeX record:

```
@incollection{ChalupMitschele2008,
title={Kernel Methods in Finance},
author={Chalup, Stephan~K. and Mitschele, Andreas},
pages={655--687},
chapter={27},
year={2008},
isbn={978-3-540-49486-7},
booktitle={Handbook on Information Technology in Finance},
series={International Handbooks Information System},
editor={Seese, Detlef and Weinhardt, Christof
and Schlottmann, Frank},
doi={10.1007/978-3-540-49487-4_27},
url={http://dx.doi.org/10.1007/978-3-540-49487-4_27},
publisher={Springer Berlin Heidelberg},
language={English}
}
```

Kernel Methods in Finance

Stephan K. Chalup[1] and Andreas Mitschele[2,3]

[1] School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia
chalup@cs.newcastle.edu.au
[2] GILLARDON AG financial software, Research Department, Alte Wilhelmstr. 15, D-75015 Bretten, Germany
[3] Institute AIFB, University of Karlsruhe (TH), D-76128 Karlsruhe, Germany
mitschele@aifb.uni-karlsruhe.de

Summary. Kernel methods are a class of powerful machine learning algorithms which are able to solve non-linear tasks. This chapter presents a concise overview of a selection of relevant machine learning methods and a survey of applications to show how kernel methods have been applied in finance. The overview of learning concepts addresses methods for dimensionality reduction, regression, and classification. The concept of kernelisation which can be used in order to transform classical linear machine learning methods into non-linear kernel methods is emphasised. The survey of applications of kernel methods in finance covers the areas of credit risk management, market risk management, and discusses possible future application fields. It concludes with a brief overview of relevant software toolboxes.

Keywords: Support Vector Machines, Dimensionality Reduction, Time Series Analysis, Credit Risk, Market Risk

1 Introduction

Kernel methods (Cristianini and Shawe-Taylor, 2000; Herbrich, 2002; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) can be regarded as machine learning techniques which are "kernelised" versions of other fundamental machine learning methods. The latter include traditional methods for linear dimensionality reduction such as principal component analysis (PCA) (Jolliffe, 1986), methods for linear regression and methods for linear classification such as linear support vector machines (Cristianini and Shawe-Taylor, 2000; Boser et al., 1992; Vapnik, 2006b). For all these methods corresponding "kernel versions" have been developed which can turn them into non-linear methods. Kernel methods are very powerful, precise tools that open the door to a large variety of complex non-linear tasks which previously were beyond the horizon of feasibility, or could not appropriately be analysed with traditional machine learning techniques. However, with kernelisation come a number of new tasks

and challenges that need to be addressed and considered. For example, for each application of a kernel method a suitable kernel and associated kernel parameters have to be selected. Also, high-dimensional non-linear data can be extremely complex and can feature counter-intuitive pitfalls (Verleysen and Francois, 2005).

Some kernel methods, for example non-linear Support Vector Machines (SVMs) (Vapnik, 2000, 1998; Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002; Burges, 1998; Evgeniou et al., 2000), have become very popular and useful tools in applications. They have, in recent years, substantially improved former benchmarks in application areas like bioinformatics, computational linguistics, and computer vision (Cristianini and Shawe-Taylor, 2000). Specific applications include genome sequence classification (Sonnenburg et al., 2005), other applications in computational biology (Schölkopf et al., 2004), time series prediction (Cao, 2003), text categorization (Fu et al., 2004), handwritten digit recognition (Vapnik, 2000), pedestrian detection (Kang et al., 2002; Chen et al., 2006), face recognition (Osuna et al., 1997; Li et al., 2004b; Carminati and Benois-Pineau, 2005), and other applications in bioengineering, signal- and image processing (Camps-Valls et al., 2007).

In machine learning applications the characteristics of the available data plays a crucial role. Of course this applies to applications in finance as well, including risk management, asset allocation issues, time series prediction, and financial instrument pricing. In risk management one may try to estimate the future price uncertainty of a financial instrument exposure using large historical data sets. The reliability of such an analysis is highly dependent on the adequacy of the chosen risk management model and the quality of the input data. However, financial data are generally known to be inherently noisy, non-stationary and deterministically chaotic (Cao and Tay, 2001). The potential of kernel machines to deal with such complex and possibly non-linear input data is therefore of particular value. Additionally, financial data sets exhibit highly heterogenous behavior. While market risk[4] data usually have extensive time series—especially when considering tick-wise data—credit risk[5] data show sparsity due to relatively rare credit events and short lengths of times series. Many well-approved techniques have been developed in the past to handle the deficiencies that these data sets exhibit. While some authors have employed artificial intelligence techniques to solve these problems, it can be stated that traditional statistical methods, like GARCH (generalized autoregressive conditional heteroskedasticity) or PCA are more common. Alexander (2001) gives an introductory overview to the more traditional techniques.

Some developments in kernel machines are very recent and have not yet had a chance to be applied in the area of finance. Our aim is to provide a

[4] *Market risk* arises due to adverse changes in the market prices of financial instruments, e.g. stocks, bonds or currencies.

[5] *Credit risk* describes potential losses for the creditor through a debtor who is not able or willing to repay his debt in full.

concise overview of fundamental concepts of kernel methods that are already common in finance applications, as well as aspects of some newer developments which may be of importance for financial applications in the future. There are several excellent texts on introductory or more theoretical aspects of kernel methods available which are recommended to complement and extend the present chapter. Among them are the introductory tutorial on SVMs and statistical learning theory by (Burges, 1998), the books (Vapnik, 1998, 2000, 2006b; Cristianini and Shawe-Taylor, 2000; Shawe-Taylor and Cristianini, 2004; Schölkopf and Smola, 2002; Herbrich, 2002; Suykens et al., 2002), as well as relevant book chapters in (Haykin, 1999; Gentle et al., 2004; Bishop, 2006), and a number of recent collections and overviews on associated topics such as (Müller et al., 2001; Burges, 2005; Chapelle et al., 2006).

The first part of this chapter provides some theoretical background and explains in Section 2 the general idea of kernel machines and kernelisation. The three fundamental machine learning paradigms of dimensionality reduction, regression, and classification, and some associated kernel methods are covered in Sections 3, 4, and 5, respectively. Section 6 addresses questions of kernel and parameter selection. The second part of the present chapter starts in Section 7 with a survey of typical questions and tasks arising in finance applications and how kernel methods have been applied to solve them. A brief overview of relevant software toolboxes follows in Section 8.

2 Kernelisation

Kernel methods employ a non-linear feature mapping

$$\phi : \mathcal{X} \longrightarrow \mathcal{H} \tag{1}$$

from an input space $\mathcal{X}$, for example $\mathcal{X} = \mathbf{R}^d$, to a high-dimensional possibly ∞–dimensional feature space $\mathcal{H}$. ϕ lifts a potentially non-linear task from $\mathcal{X}$ to $\mathcal{H}$ where "remotely" a satisfactory solution is sought via traditional typically linear tools (cf. Fig. 1). The idea is that the feature mapping ϕ only appears implicitly and does not need to be determined explicitly in any of the computations. Central to this approach is a continuous and symmetric function

$$K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbf{R} \tag{2}$$

which will be used to measure the similarity between inputs. Mercer's condition (Mercer, 1909; Courant and Hilbert, 1953; Cristianini and Shawe-Taylor, 2000; Herbrich, 2002) says if K is positive semi-definite then there exists a mapping ϕ as in (1) from $\mathcal{X}$ into some Hilbert space $\mathcal{H}$ (that is, a complete dot product space) such that K is a (Mercer) *kernel*, that is, it can be written as dot product as follows

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad \text{for} \quad i, j \in \{1, ..., k\}. \tag{3}$$

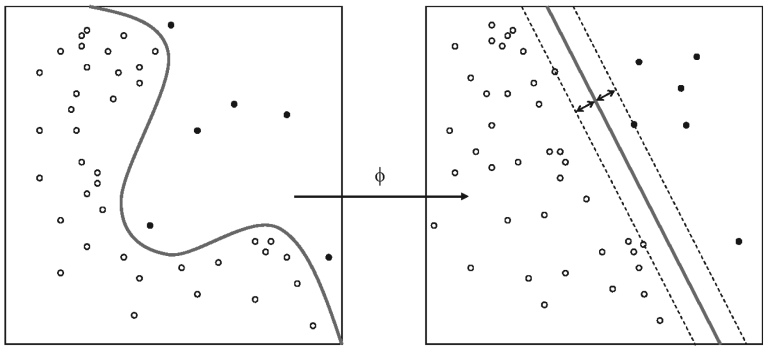

Fig. 1. The feature map of non-linear SVMs: A non-linear separator (thick line on the left) in input space is obtained by applying a linear maximal margin classifier in high-dimensional feature space (on the right).

Note that given a vector space $\mathcal{X} = \mathbf{R}^d$ the function K is positive semi-definite if $\sum_{i,j=1}^{k} c_i c_j K(x_i, x_j) \geq 0$ for all $k \in \mathbf{N}$, any selection of examples $x_1, ..., x_k \in \mathcal{X}$, and any coefficients $c_1, ..., c_k \in \mathbf{R}$. Kernelisation of any algorithm, for example for dimensionality reduction, regression, or classification, can be achieved by seeking a formulation of the algorithm where variables only occur in form of dot products $x_i^T x_j$. Then, after formally replacing all the $x_i \in \mathcal{X}$ by $\phi(x_i)$ the feature vectors only occur in the form of dot products $\phi(x_i)^T \phi(x_j)$ and finally, provided the selected kernel function K fulfills Mercer's condition, the substitution of equation (3) can be applied, which leads to a kernel method.

The computations of kernel methods thus only involve kernel values and do not require explicit knowledge of ϕ. This is often called *the Kernel Trick*. The basic idea of kernel methods was already developed in the 1960s (Aizerman et al., 1964) and incorporated into the SVM framework from 1992 (Boser et al., 1992; Vapnik, 2000, 2006a). The kernel trick was also employed to kernelise other methods such as PCA (Schölkopf et al., 1996, 1997; Schölkopf, 1997).

3 Dimensionality Reduction

There are numerous motivations to perform dimensionality reduction including reduction of computational complexity, a better understanding of the data, and the avoidance of unwanted effects which can occur in high-dimensional spaces. In finance, particularly in risk management, dimensionality reduction plays a crucial role to enable the modeling of systems which often have several hundreds of risk factors. Hitherto, usually linear methods, such as PCA, were used in finance applications, like interest rate modeling (Alexander, 2001).

To describe the fundamental paradigm let $x_t \in \mathbf{R}^d$, $t = 1, ..., n$ be data points in the high dimensional space. The aim is to find a mapping

$$\begin{aligned} \mathbf{R}^d &\longrightarrow \mathbf{R}^r \\ x_t &\mapsto y_t \end{aligned} \tag{4}$$

for all $t = 1, ..., n$ such that $r < d$ and the new lower dimensional representation of the data in $\mathbf{R}^r$ has some equivalent topological structure and still contains all its essential information.

3.1 Classical Methods for Dimensionality Reduction

This section briefly describes principal component analysis (PCA) and multidimensional scaling (MDS). These two classical spectral methods for linear dimensionality reduction are the basis of kernel principal component analysis (KPCA) and isomap. The latter two methods can be regarded as kernel methods for non-linear dimensionality reduction (Williams, 2001; Ham et al., 2004) and will be addressed in section 3.2 below. PCA and MDS are in general not appropriate for processing data samples from non-linear manifolds. This is illustrated in Fig. 2 where the task is to reduce the embedding of the two-dimensional helical strip shown on the left from an embedding into $\mathbf{R}^3$ to an embedding into $\mathbf{R}^2$. As shown on the right hand side of Fig. 2 PCA and classical metric MDS, which both produce the same output (Cox and Cox, 2001; Xiao et al., 2006), both fail to unfold the helix correctly and instead project it onto the two-dimensional plane. The non-linear method isomap (Tenenbaum et al., 2000), however, is able to unfold the helix and thus leads to a topologically correct embedding of the helix into $\mathbf{R}^2$.

Principal Component Analysis

Principal component analysis (PCA) is a well-established method for linear dimensionality reduction (Hotelling, 1933; Jolliffe, 1986). Given a set of sample vectors $x_t \in \mathbf{R}^d$, $t = 1, ..., n$ the first step of PCA is to calculate the sample mean $\bar{x} = \frac{1}{n} \sum_{t=1}^{n} x_t$ and the sample covariance matrix

$$C = \frac{1}{n} \sum_{t=1}^{n} (x_t - \bar{x})(x_t - \bar{x})^T. \tag{5}$$

As C is a symmetric $(d \times d)$-matrix it is possible to compute its eigenvalues $\lambda_i \in \mathbf{R}$ and associated eigenvectors $v_i \in \mathbf{R}^d$, $i = 1, ..., d$, such that

$$V^T C V = \mathrm{diag}[\lambda_1, ..., \lambda_d] \tag{6}$$

where the $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$ are the eigenvalues of C in descending order, $\mathrm{diag}[\lambda_1, ..., \lambda_d]$ is the diagonal matrix with the eigenvalues as diagonal elements, and V is the matrix with the associated eigenvectors v_i, the principal

components, as columns. The obtained eigenvectors are mutually orthonormal and define a new basis aligned with the directions of maximum variance in the data. The eigenvalues represent the projected variance of the inputs along the new axes.

The number of significant eigenvalues, r, indicates the intrinsic dimensionality of the data set. A projection of the data into the r-dimensional subspace is given by $y_t = V_r^T(x_t - \bar{x}),\ t = 1, ..., n$ where V_r is the $(d \times r)$ submatrix of V containing the first r eigenvectors associated with the r largest eigenvalues as columns.

One possible geometric interpretation is that PCA seeks the orthogonal projection of the data onto a lower dimensional linear space such that the variance of the projected data becomes maximal (Jolliffe, 1986; Bishop, 2006).

Multidimensional Scaling

Classical Multidimensional Scaling (MDS) (Cox and Cox, 2001) can be motivated as a method which has the aim to find a faithful lower dimensional embedding of the given data $x_t \in \mathbf{R}^d,\ t = 1, ..., n$ under the constraint that pairwise Euclidean distances between data points are preserved as much as possible, that is, $\|x_i - x_j\| \approx \|y_i - y_j\|,\ i, j = 1, ..., n$. Under the assumption that the centroid of the data is at the origin, that is $\sum_{t=1}^{n} x_t = 0$, the following relation between the matrix of squared Euclidean distances $D = (\|x_i - x_j\|^2)_{i,j=1,...,n}$ and the Gram matrix $G = (x_i^T x_j)_{i,j=1,...,n}$ holds (Cox and Cox, 2001):

$$G = -\frac{1}{2} HDH \tag{7}$$

where $H = (H_{ij})_{i,j=1,...,n}$ is the centering matrix with

$$H_{ij} = \delta_{ij} - \frac{1}{n} \text{ and } \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \tag{8}$$

Let $X = [x_1, ..., x_n] \in \mathbf{R}^{d \times n}$ be the matrix with the data points as columns. Since $G = X^T X$ is symmetric there is an orthonormal basis of eigenvectors $w_1, ..., w_n$ such that

$$G = W \cdot \text{diag}[\mu_1, ..., \mu_n] \cdot W^T \tag{9}$$

where the $\mu_1 \geq \mu_2 \geq ... \geq \mu_n$ are the eigenvalues of G in descending order and W is the matrix with the associated eigenvectors w_i as columns. After selecting the r most significant eigenvalues a low dimensional representation of the data is obtained by

$$x_i \mapsto y_i = (\sqrt{\mu_1} w_{1i}, ..., \sqrt{\mu_r} w_{ri})^T,\ i = 1, ..., n. \tag{10}$$

An alternative motivation used in MDS is to preserve the dot products as far as possible, that is $x_i x_j^T \approx y_i y_j^T,\ i, j = 1, ..., n$, and start with an eigenvalue

decomposition of the Gram matrix as in (9). This approach directly works on the input data and not on the pairwise distances.

Metric MDS can be categorised as a kernel method because it can be interpreted as a form of kernel MDS or kernel PCA, respectively (Williams, 2001).

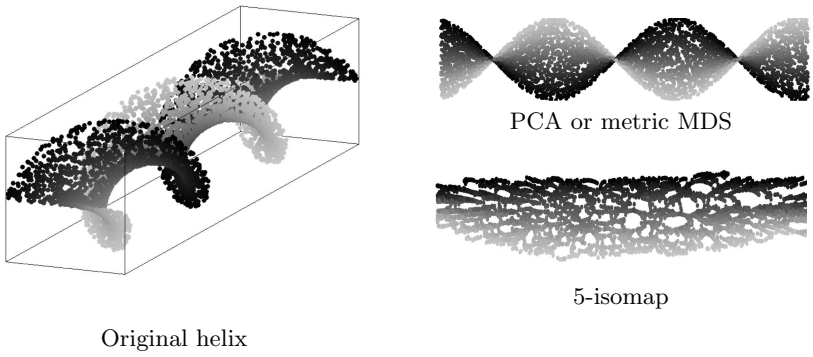

Fig. 2. Left: A two-dimensional helical strip is embedded in three-dimensional Euclidean space. Right top: Linear methods such as PCA or metric MDS project the data into $\mathbf{R}^2$. Right bottom: The non-linear method isomap with neighbourhood parameter 5 is able to unfold the helix and can approximately maintain topologically correct neighbourhood relationships between the data points.

3.2 Non-linear Dimensionality Reduction

Non-linear dimensionality reduction is often used synonymously with the term *manifold learning*. Manifolds are central objects in geometry and topology (Spivac, 1979). Examples of manifolds are locally Euclidean objects such as lines, circles, spheres, tori, subsets of those objects, and their generalisations to higher dimensions. The aim of manifold learning methods is to extract low-dimensional manifolds from high-dimensional data and faithfully embed them into a lower dimensional space.

The use of non-linear dimensionality reduction techniques in practical applications can come with technical and conceptual challenges. When using linear dimensionality reduction techniques the outcome has always been restricted because there exists essentially only one linear space in each dimension, while in the non-linear case there is a large variety of manifolds.

In recent years several new methods for non-linear dimensionality reduction have been developed and many of them are, similar to PCA and MDS, spectral methods (Xiao et al., 2006). These include, for example, Kernel Principal component Analysis (KPCA) (Schölkopf et al., 1998), Isometric Fea-

ture Mapping (Isomap) (Tenenbaum et al., 2000), Locally Linear Embedding (LLE) (Roweis and Saul, 2000), Maximum Variance Unfolding (MVU) (Weinberger and Saul, 2004; Weinberger et al., 2004, 2005), and Laplacian Eigenmaps (Belkin and Niyogi, 2003). The different methods have distinct individual advantages and it is a topic of current research to gain better understanding of their robustness and ability to recognise and preserve the underlying manifolds' topology and geometry.

Kernel Principal Component Analysis

The basic idea of kernel principal component analysis (KPCA) is to kernelise PCA, that is, to map the data $x_1, ..., x_n$ to a higher dimensional space using an implicit non-linear feature map $\phi : \mathbf{R}^d \longrightarrow \mathcal{H}$ and then to apply PCA (Schölkopf et al., 1997, 1998).

In practice, however, it is not easy to center the mapped data in feature space for calculation of the estimated covariance matrix

$$C = \frac{1}{n}\sum_{i=1}^{n}\left(\phi(x_i) - \frac{1}{n}\sum_{t=1}^{n}\phi(x_t)\right)\left(\phi(x_i) - \frac{1}{n}\sum_{t=1}^{n}\phi(x_t)\right)^T.$$

which is required for PCA.

According to (Schölkopf et al., 1997, 1998; Williams, 2001; Saul et al., 2006) there is an alternative solution to KPCA which utilises the Gram matrix in feature space

$$K = (\phi(x_i)^T\phi(x_j))_{i,j=1,...,n}.$$

After suitable kernel substitution with a positive semi-definite kernel, K is called the kernel matrix. A centered kernel matrix K^C can be calculated by

$$K^C = HKH \tag{11}$$

where $H = (H_{ij})_{i,j=1,...,n}$ is the centering matrix of MDS as in (8).

The remainder of the method follows the approach explained for MDS in (9): Since K^C is symmetric it can be diagonalised $K^C = W \cdot \mathrm{diag}[\mu_1, ..., \mu_n] \cdot W^T$ such that the eigenvalues μ_i are in descending order on the diagonal and W is the matrix with the associated eigenvectors w_i as columns. Selection of the r most significant eigenvalues leads to a low-dimensional representation of the data as in (10).

Isometric Feature Mapping

Isometric Feature Mapping (Isomap) (Tenenbaum et al., 2000) is a manifold learning method which can be seen as an extension of MDS where approximate geodesic distances are used as inputs instead of pairwise Euclidean distances. In contrast to Euclidean distances which are measured straight through the

surrounding space $\mathbf{R}^d$ geodesic distances can be longer because they are measured along (shortest) arcs within the manifold using its intrinsic metric.

As in (3) let $x_1, ..., x_n \in \mathbf{R}^d$, be given data points which are assumed to be sampled from a low-dimensional manifold $\mathcal{M}$ which is embedded in the high-dimensional input space $\mathbf{R}^d$. A basic version of Isomap using k-nearest neighbour graphs can then be explained in three steps (Tenenbaum et al., 2000):

1. Select a *neighbourhood parameter* $k \in \mathbf{N}$ and construct a neighbourhood graph $\mathcal{G}$ such that each point of the manifold is connected only to its k nearest neighbours. Weight existing connections between two neighbouring vertices $x_p, x_q \in \mathcal{G}$ by their Euclidean distance in $\mathbf{R}^d$.
2. Generate a distance matrix $D = (d_{ij})_{i,j=1,...,n}$ where each coefficient d_{ij} is the shortest path distance in $\mathcal{G}$ between each pair of the initially given sample points $x_i, x_j \in \mathbf{R}^d$, $i, j = 1, ..., n$. The shortest paths in $\mathcal{G}$ can be calculated, for example, by Dijkstra's algorithm (Cormen et al., 2001). The idea is that the path-length d_{ij} is an approximation of the geodesic distance between each pair of points $x_i, x_j \in \mathcal{M},\ i, j = 1, ..., n$.
3. Apply metric MDS using the $d_{ij},\ i, j = 1, ..., n$ as inputs.

Ham et al. (2004) showed that Isomap can be regarded as a form of KPCA, that is a kernel method, if the following conditionally positive definite kernel matrix is used

$$K_{\text{isomap}} = -\frac{1}{2} HDH, \tag{12}$$

where D is Isomap's distance matrix of step 2 above and H is the centering matrix of MDS which was defined in equation (8).

Example: Rotating Coin

The rotating coin data set consists of a series of 200 digital images taken of a gold coin while it was rotating. Each image had 168^2 pixel which means that it can be regarded as a point in a 28224-dimensional vector space. The essential underlying dynamics contained in the image sequence can be represented by a circle, that is, a 1-dimensional manifold embedded in the 2-dimensional plane. The task for the dimensionality reduction method was to recognise the underlying dynamics, extract the circle and reduce the dimensionality from 168^2 down to 2 dimensions. Fig. 3 shows the results obtained with Isomap (Tenenbaum et al., 2000) using $k = 3$ as the neighbourhood parameter. Similar results but with stronger geometric distortions and irregularities were obtained with KPCA and with Isomap when using $k > 3$. The traditional linear methods PCA and MDS were not able to solve this task.

4 Regression

The task of linear regression is to estimate a function

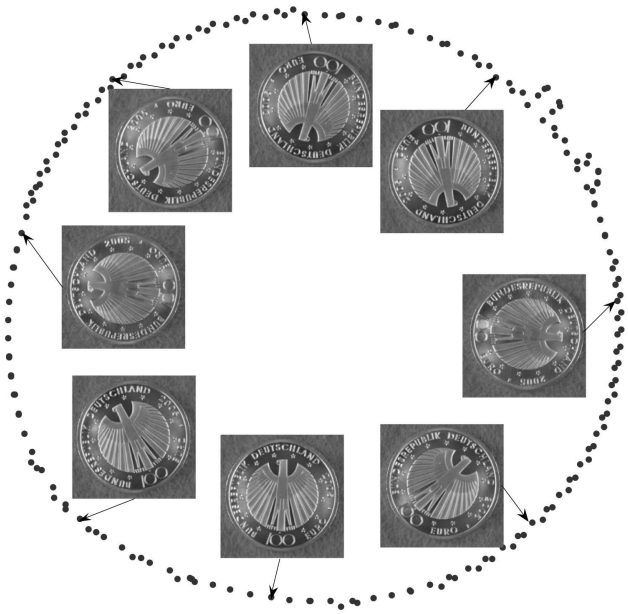

Fig. 3. Application of isomap (k =3) to the rotating coin data. Each point represents an image. Similar images are mapped to points in close vicinity. The rotation encoded in the sequence of high-dimensional pixel arrays is represented by the resulting one-dimensional circle graph.

$$f(x) = w^T x + b \tag{13}$$

by finding suitable parameters $w \in \mathbf{R}^m$ and $b \in \mathbf{R}$ such that $f(x_i) = y_i,\ i = 1, ..., n$ for a given set of iid data points

$$(x_1, y_1), ..., (x_n, y_n) \in \mathbf{R}^m \times \mathbf{R}. \tag{14}$$

In Finance, (mostly linear) regression represents a standard tool to perform time series analysis, especially for forecasting tasks (Alexander, 2001). A selection of applications of support vector machine regression in finance will be reviewed in section 7.2.

Geometrically linear regression corresponds to finding an offset and a direction of the line that best fits a set of given data points. A classical solution method for the task of line fitting is least squares optimisation which minimises the square loss $\sum_{i=1}^{m}(y_i - (wx_i + b))^2$ over all $(w, b) \in \mathbf{R}^m \times \mathbf{R}$.

Support Vector Machines (SVMs) for linear regression are a more recent method which can be used to solve this task by minimising the empirical risk

$$\frac{1}{n}\sum_{i=1}^{n} \|y_i - f(x_i)\|_\epsilon \tag{15}$$

$$\text{where } \|z\|_\epsilon = \begin{cases} 0 & \text{, if } \|z\| \leq \epsilon \\ \|z\| - \epsilon & \text{, otherwise,} \end{cases} \tag{16}$$

is Vapnik's ϵ-insensitive loss function (Vapnik, 1998).

This can be formulated as a constraint optimisation task as follows:

$$\begin{aligned} \text{Minimise } & \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \\ & \text{over all } w, b \in \mathbf{R} \text{ and } \xi_i, \xi_i^* \geq 0, \ i = 1, ..., n. \\ \text{subject to } & y_i - (w^T x_i + b) \leq \varepsilon + \xi_i, \ i = 1, ..., n \\ & (w^T x_i + b) - y_i \leq \epsilon + \xi_i^*, \ i = 1, ..., n. \end{aligned} \tag{17}$$

The parameters $\varepsilon > 0$ and $C \geq 0$ are to be selected separately by the user. $C \geq 0$ regulates the tolerance level provided by the slack variables $\xi_i, \xi_i^* \in \mathbf{R}_{\geq 0}$. If all data points lie within the ε-tube of equation (16) no slack variables are required.

Through application of the method of Lagrange multipliers the dual formulation of (17) can be obtained

$$\begin{aligned} \text{Maximise } & -\frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) x_i^T x_j \\ & -\varepsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i (\alpha_i - \alpha_i^*) \\ & \text{over all Lagrange multipliers } \alpha_i, \alpha_i^* \in [0, C] \\ \text{subject to } & \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0. \end{aligned} \tag{18}$$

The SVM in dual space is

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) x_i^T \cdot x + b \tag{19}$$

where α_i, α_i^* are obtained by solving (18) and b follows from the associated Karush-Kuhn-Tucker (KKT) conditions of optimality.

Through kernelisation the above procedure can be extended for non-linear function estimation. After substitution of x by $\phi(x)$ the function to be estimated in (13) becomes

$$f(x) = w^T \phi(x) + b. \tag{20}$$

The primal optimisation task is the same as described in (17) except that the x_i are formally replaced by $\phi(x_i)$. Application of Lagrangian optimisation leads to the dual formulation:

$$\begin{aligned} \text{Maximise } & -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j) \\ & -\varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*) \\ & \text{over all Lagrange multipliers } \alpha_i,\alpha_i^* \\ \text{subject to } & \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0 \\ & 0\le\alpha_i\le C,\ i=1,...,n \\ & 0\le\alpha_i^*\le C,\ i=1,...,n. \end{aligned} \tag{21}$$

Finally the SVM in dual space is

$$f(x)=\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)K(x_i,x)+b \tag{22}$$

where α_i,α_i^* are obtained by solving (21) and b follows from the associated conditions of optimality. This is the same as (18) and (19) except that the dot products $x_i^T x_j$ in (18) and (19) are formally replaced by the associated kernel values $K(x_i,x_j)$. The two free parameters ϵ and C in (21) control the VC-dimension (Vapnik, 2000) of $f(x)$ and have to be selected separately, for example, through cross-validation. It turns out that the set of data points x_i for which $(\alpha_i-\alpha_i^*)\neq 0$ is sparse. Its elements are called the *support vectors* of the machine (22). More details about this method are available in (Vapnik, 1998, 2000, 2006b; Cristianini and Shawe-Taylor, 2000; Mika et al., 2005; Shawe-Taylor and Cristianini, 2004; Suykens et al., 2002; Schölkopf and Smola, 2002).

5 Classification

Classification can be seen as a special case of the regression task (13)-(14) where the function values are integers, for example, $\{-1,1\}$ in the case of binary classification. Classification tasks are the second dominant application within the finance area (cf. Sec. 7.1). Kernel-based classification has been thoroughly used in credit risk management, for instance to differentiate between good and bad clients.

The perceptron (Rosenblatt, 1958; Mitchell, 1997) is one of the most basic methods for binary classification. It is often interpreted as an abstract neuron model

$$r=\varphi(w\cdot x+b) \tag{23}$$

where r is the associated return, $b\in\mathbf{R}$ is the bias, $w\cdot x=\sum_j w_j x_j$ is the dot product between a weight vector $w\in\mathbf{R}^d$ and an input vector $x\in\mathbf{R}^d$, and $\varphi:\mathbf{R}\longrightarrow\{-1,1\}$ is an activation function, for example, the signum function

$$\varphi : \mathbf{R} \longrightarrow \{0,1\},\ y \mapsto \varphi(y) = \begin{cases} -1 & \text{, if } 0 < y, \\ +1 & \text{, if } y \leq 0. \end{cases}$$

Geometrically the perceptron defines a hyperplane as the set of points which is perpendicular to the weight vector w and has distance $a = -\frac{b}{\|w\|}$ from the origin:

$$\{x; w \cdot x = -b\} = \{x; \frac{w}{\|w\|} \cdot x - a = 0\} \tag{24}$$

By dividing the input space into two halves a hyperplane can be employed for linear binary classification of input patterns. The perceptron training rule (Mitchell, 1997) is an iterative algorithm which adjusts the weight vector and bias to place the associated hyperplane in some sensible position between two classes of given training points.

In contrast basic support vector machines (Cristianini and Shawe-Taylor, 2000) treat linear classification as convex optimisation task which in primal weight space represents a standard constraint optimisation problem: The margin of separation between the two point classes should be maximised under the constraint that the pattern should be classified correctly, that is in the case of separable patterns

$$y_i(w \cdot x_i + b) \geq 1,\ i = 1, ..., n \tag{25}$$

and in the case of non-separable patterns

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \tag{26}$$

with slack variables $\xi_i > 0$ which can take exceptions and outliers into account (Cortes and Vapnik, 1995).

After substituting the x_i by $\phi(x_i)$ the constraints for the non-linear case become

$$y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i. \tag{27}$$

As for regression this constraint optimisation task can be formulated in primal and dual form using the method of Lagrangian optimisation. In the dual form feature vectors will only occur as parts of dot products $\phi(x_i)^T\phi(x_j)$. Using a positive definite kernel function the latter can be replaced by kernel values according to equation (3) and a non-linear SVM is obtained by kernelising the concept of the linear maximum margin classifier.

It can be shown (Mangasarian, 1999; Pedroso and Murata, 2001; Ikeda and Murata, 2005) that if the distances between the feature vectors and the separating hyperplane are evaluated using a L_p–norm, that is,

$$\|w\|_p := \begin{cases} (\sum_{i=1}^n |w_i|^p)^{\frac{1}{p}} & ,\quad 1 \leq p < \infty \\ \max_{1 \leq i \leq n} |w_i| & ,\quad p = \infty \end{cases}$$

the problem of margin optimisation becomes a p-th order programming problem, for example

$p = 2$: quadratic programming is used by classical SVMs.
$p = 1$: linear programming is used by least squares support vector machines (LSSVMs) (Suykens and Vandewalle, 1999a,b; Suykens et al., 2002).

Support vector machines (SVM) (Vapnik, 2000; Cristianini and Shawe-Taylor, 2000; Suykens et al., 2002; Schölkopf and Smola, 2002) are among the most commonly used kernel machines. They can be regarded as a newer and, in practical applications, often better alternative to artificial neural networks for function approximation to solve classification or regression tasks. In contrast to traditional artificial neural networks no lengthy iterative training procedure is required. In most cases SVM "training" is faster and does not get stuck in local minima because it is achieved by a direct calculation for linearly constraint optimisation.

We have described the basic form of SVMs for regression and for binary classification. Other important variations and extensions of SVMs are ν-SVMs (Schölkopf and Smola, 2002), LS-SVMs (Suykens et al., 2002), SVMs for clustering (Tax and Duin, 1999; Ben-Hur et al., 2001; Yang et al., 2002b), and SVMs for multi-class classification (Rifkin and Klautau, 2004).

The SVM algorithms used before 1999 were typically slower than artificial neural networks with similar generalisation performance (Haykin, 1999, p.345). Significant speed improvements were achieved, for example, through the SMO algorithm by (Platt, 1999). Since then a variety of techniques have been investigated and evaluated to support application of SVMs to large data sets (Huang et al., 2006), data sets with small training samples (Hertz et al., 2006), and unbalanced data sets (Raskutti and Kowalczyk, 2004). The latter topics are still object of current research.

6 Kernels and Parameter Selection

The selection of suitable kernels and associated kernel parameters is an important task when using kernel methods for regression, classification, and dimensionality reduction. There are several possibilities to choose the kernel function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbf{R}$ (Cristianini and Shawe-Taylor, 2000).

If input space and feature space are identified, that is $\phi(x) = x$, this results in the linear kernel, which is the dot product of the two input vectors

$$K(x, y) = x \cdot y.$$

Examples of other common kernels are the polynomial kernel

$$K(x, y) = (1 + x \cdot y)^p$$

and the Gaussian or radial basis function (RBF) kernel

$$K(x, y) = e^{\frac{-d(x,y)^2}{\sigma^2}}.$$

The p-Gaussian kernel (Francois et al., 2005)

$$K(x,y) = e^{\frac{-d(x,y)^p}{\sigma^p}}$$

is a generalisation of the Gaussian kernel. The parameter p controls the smallest distance corresponding to the decreasing part of the kernel and can help to leviate some of the issues that standard Gaussian kernels can encounter in high dimensional spaces. For example, it can be shown that if data is uniformly distributed within the unit ball almost all will have norm equal to 1 in high dimensions (Verleysen and Francois, 2005). Consequently, in a high-dimensional Gaussian distribution most data will be contained in the tails of the distribution. This is in contrast to low dimensions where the Gaussian distribution is local and most data is close to zero.

In general, a kernel is only required to be positive semi-definite and should represent a plausible similarity measure between pairs of input examples (cf. Sec. 2). These rather loose prerequisites of the Mercer theorem allow for task specific kernel design.

Specialised string kernels have been developed and applied in biosequence analysis and text categorisation (Cristianini and Shawe-Taylor, 2000). Some recent developments in SVMs allow to learn task specific kernels or to use many kernels in parallel (Lanckriet et al., 2004; Sonnenburg et al., 2006).

For the dimensionality reduction paradigm a variation of KPCA, called Maximum Variance Unfolding (MVU), was proposed where the kernel matrices are optimised by semi-definite programming to preserve the geometry of the input data up to isometry (Weinberger et al., 2004). Experimental evaluation revealed that the positive semi-definite kernel matrices constructed by MVU have clear advantages over Gaussian, polynomial, or linear kernels (Weinberger et al., 2004). This shows that kernels which are successful for large margin classification are not necessarily suitable for dimensionality reduction and vice versa (Saul et al., 2006). The reason for this is that high-dimensional spaces have a few, sometimes counter-intuitive, properties which can have essential impact on kernel choice or design (Verleysen and Francois, 2005).

In addition to kernel selection, many kernel methods require that specific parameters have to be selected or adapted. These include the parameter C of C-SVMs for classification, the C and ϵ of SVMs for regression, the neighbourhood parameter k of isomap, and specific kernel parameters such as the width σ in Gaussian kernels.

Several of the kernel methods have their roots in statistical learning theory. For example, learning with SVMs and neural networks is based on Valiant's principle of PAC learning (Valiant, 1984). According to (Vapnik, 2006a) the bigger theoretical framework could be described as *empirical inference science*. SVMs approximately implement Vapnik's method of *structural risk minimisation* which defines a "trade-off between the quality of the approximation of the given data and the complexity of the approximating function" (Vapnik, 2000): The generalisation error is bounded by the sum of training error and a

function of the Vapnik-Chervonenkis (VC) dimension, which is a measure of the capacity, or flexibility of a function class. More specifically (Vapnik, 2000, 1998) proved that hyperplanes with $\|w\| < l$ have a Vapnik-Chervonenkis (VC) dimension which is bounded by $\min(int[R^2 l^2], n) + 1$ where R is the radius of the smallest ball containing all feature vectors.

SVM parameters can be selected to minimise this bound in order to achieve good generalisation ability of the model. In practice, however, these basic bounds, although predictive (Burges, 1998), are typically very conservative and refinements or alternative methods, such as cross-validation and its variants such as n-fold cross-validation (Mitchell, 1997; Stone, 1974), are often preferred. For binary classifiers an alternative method to evaluate the performance is the receiver operating curve (ROC) analysis (Fawcett, 2004).

7 Survey of Applications in Finance

The aim of this survey is to provide a structured overview of successful or promising kernel machine applications in the area of finance. While the survey does not claim completeness, it tries to highlight some of the main results.

Most of the hitherto performed analyses have their focus on risk management. In further areas, such as financial instrument pricing (e.g. option pricing) or asset allocation, there have hardly been applications yet. The survey has been divided into applications within the market risk context, which primarily concern time series forecasting, and within credit risk, which mainly involve rating/scoring schemes and bankruptcy prediction. Not least through the new capital regulations Basel II (Basel Committee on Banking Supervision 2006), that have been imposed on banks since the beginning of 2007 with a one year transition period, risk management has taken an even more prominent role in finance.

7.1 Credit Risk Management

The area of credit risk management has rapidly gained importance in recent years, particularly boosted through the new Basel II framework (Basel Committee on Banking Supervision 2006). The first Capital Accord, called Basel I, which has been published in 1988, already required banks to build buffers for possible losses through defaults of their clients. While these former regulations were rather undifferentiated, the new guidelines demand a considerably higher distinction. In this challenging context advanced statistical methods are frequently used by bank practitioners for the analysis of the newly built credit databases. Different machine learning approaches, including neural networks and kernel machines, have only recently been introduced to the area (Thomas et al., 2005).

Credit risk management represents a promising application area for kernel methods. SVMs have been successfully employed to derive bond ratings,

help banks to classify their loan base into "good" and "bad" clients, and to detect risks of business failure. Even though banks have put great efforts into extending their available credit data, actual databases are not very large yet. This is due to relatively rare nature of some events, for example defaults.

In a detailed research overview concerning bond rating estimation (Huang et al., 2004) showed that artificial intelligence (AI) methods, including rule-based expert systems, case-based reasoning and machine learning, proved to deliver very good results in comparison to conventional statistical methods like multiple linear regression or multiple discriminant analysis. The test set prediction accuracy of earlier AI investigations had ranged from around 50% to 88% for the classification problem. In their empirical study they compared back-propagation neural networks to SVMs for bond rating analysis using financial ratios and ratings from Taiwan (1998-2002) and the United States (1991-2000). In general they found that AI methods performed better than classical approaches with SVMs slightly ahead concerning the results.

(Chen and Shih, 2006) used a number of new input variables, including stock market information, financial support by the government and financial support by major shareholders, for their multi-class rating system, and also compared different SVM approaches with a back-propagation neural network. Using Taiwanese bank rating data with their SVM models they were able to further improve results with an accuracy rate of 89% to 100% in the training set and 73% to 85% in the test set, substantially outrivalling the benchmark NN approach.

Another study on bond rating estimation was performed by (Cao et al., 2006) using multi-class classification with "one-against-all", "one-against-one" and directed acyclic graph SVM (DAGSVM) approaches. All the SVM approaches, especially the latter one, significantly outperformed the commonly used traditional benchmarks (back-propagation neural networks, logistic regression and ordered probit regression) concerning classification accuracy. Through an additional sensitivity analysis of feature importance (Cao et al., 2006) were able to further enhance the SVM generalisation ability.

While rating analysis is mainly relevant for larger companies who can afford to undergo the cost-intensive rating process, *credit scoring* plays a corresponding role within credit assessment for smaller companies and retail banking customers. (Baesens et al., 2003) provide an extensive study on state-of-the-art classification algorithms and their performance within credit scoring. Apart from common algorithms (for example logistic regression, discriminant analysis, k-nearest neighbour, neural networks and decision trees) they also considered kernel-based methods, namely SVMs and least-squares SVMs (LS-SVMs). They used eight real-life credit scoring data sets which they obtained from two banks and from publicly available sources. According to their chosen performance measures (percentage correctly classified cases, area under the receiver operating characteristic curve) LS-SVMs and neural network classifiers achieved the best results. However, in their study linear classifiers also yielded good outcomes, indicating that the analysed data sets

exhibited only weak non-linear behavior. In turn, the authors noted that even small improvements in scoring accuracy can lead to significant savings regarding the huge volume of banks' credit business.

(Schebesch and Stecking, 2005a) successfully employed SVMs to divide a set of labelled credit applicants into subsets of typical and critical patterns. While class labels for typical patterns were relatively easy to predict even using standard linear classification methods they stated that the more interesting critical patterns include less trivial training examples. Additionally they suggested using the SVM results as input for linear discriminant analysis in a hybrid approach that would improve generalisation ability. In another contribution (Stecking and Schebesch, 2006) outlined how appropriate kernels for the problem class could be chosen and compared. Their experiments with the relatively unknown *Coulomb kernel* $(1 + \|x_i - x_j\|^2/E)^{-\delta}$ (Hochreiter et al., 2003), which represents a localised kernel function similar to RBF kernels, indicated better classification performance based on a lower expected out-of-sample error than linear, polynomial, sigmoid, and RBF kernels.

As in credit scoring it is usually not possible to distinguish between absolutely "good" and "bad" debtors. (Wang et al., 2005) proposed fuzzy SVMs to evaluate credit risk. In their approach each customer was considered to belong both to the positive and to the negative class via a membership concept. Instances that were identified as outliers were assigned with low membership for the one class while the opposite class was assigned a higher membership. As each instance contributed two errors to the total error term they called their new hybrid approach *bilateral-weighted fuzzy SVM*. In their empirical study they analysed three different data sets: 60 UK corporations (30 failed and 30 non-failed with 12 characteristic variables each), 653 Japanese credit card application approval data (357 granted and 296 refused with 15 attributes each) and 1225 applicants (323 bad and 902 good with 12 variables each). Overall they found that their newly suggested fuzzy SVM almost consistently outperformed numerous comparable methods (linear/logit regression, neural network, standard and fuzzy SVM with varying kernels). Additionally it provided better generalisation ability than former fuzzy SVM approaches as the negative impact of outliers could be successfully reduced. They noted, however, that computational complexity increased considerably in their approach and that membership generation had to be thoroughly contemplated to avoid distortions.

Among other applications of SVMs in the area of credit scoring were (van Gestel et al., 2003a), (Sánchez et al., 2004), (Li et al., 2004a), again (Schebesch and Stecking, 2005b; Stecking and Schebesch, 2005) and (Lai et al., 2006b). Besides the assessment of creditworthiness, that is rating and scoring respectively, different authors have used SVM approaches to predict bankruptcy of companies. It has to be noted that this is also a very important task within the new Basel II guidelines (Basel Committee on Banking Supervision 2006).

(Härdle et al., 2005) used a SVM based approach to predict bankruptcy for 42 US companies that had filed for Chapter 11 of the US Bankruptcy Code

in 2001-2002. Based on information from the annual reports they computed a number of financial ratios concerning profit measures, for instance EBIT / TA (Earnings before income tax / total assets), leverage ratios, liquidity ratios and turnover ratios, for a total of 84 companies, that was 42 companies that went bankrupt and 42 other companies with similar characteristics that survived. Discriminant analysis suggested that the most significant predictors belonged to profit and leverage ratios. Through their analysis they found that SVMs were able to classify successful companies into one certain cluster, meaning that they had to have certain characteristics in common, while the failing companies were located outside this cluster. They also showed that SVMs provided superior classification information concerning defaulted companies compared to the common discriminant analysis approach. As their data set was relatively small the linear classifier delivered the best classification results. They noted, however, that for larger data sets non-linear classifiers would probably be better suited.

In their study (Shin et al., 2005) showed that SVMs outperformed back-propagation neural networks in the problem of corporate bankruptcy prediction. Particularly they claimed that SVMs could handle smaller sample sizes—that often occur within the credit risk context—more adequately. While they did not thoroughly analyse the optimality of the kernel parameters (Min and Lee, 2005) performed a detailed analysis in their contribution. First of all they compared SVM performance to the common benchmarks: multiple discriminant analysis, logistic regression and three-layer fully connected back-propagation neural networks. Their data sample covered a—in terms of credit risk—quite extensive data set of 1888 firms, with both 50% bankrupt and non-bankrupt firms. Inter alia they performed a principal component analysis and further steps to discern important features and ran an extensive grid search to determine optimal parameters for the SVM kernel. They found that SVMs outperform all benchmark methods and consistently achieved similar or better performance than the neural network approach. (Min and Lee, 2005) also proposed that SVMs tend to handle smaller sample sizes quite well while generally keeping their generalisation ability through the use of the structural risk minimisation principle.

In another hybridised approach Min et al. (2006) integrated genetic algorithms (GAs) with SVMs to predict bankruptcy for a data set of 614 Korean companies, half of which filed for bankruptcy between 1999 and 2002. The GA was used to optimise the feature subset and the SVM parameters. For the feature subset selection they used the so-called wrapper approach that trained the classifier with a certain subset and subsequently evaluated the corresponding classification error using a validation set. For the SVM's RBF kernel two parameters, C and σ^2, had to be optimised. As both the feature subset selection and the parameter optimisation were mutually dependent, they were optimised simultaneously. (Min et al., 2006) observed that the hybridised model significantly outperformed the common benchmarks logistic regression and artificial neural networks, and was additionally able to im-

prove prediction quality in comparison with the pure SVM approach. More studies on bankruptcy prediction with SVMs showing superior performance were conducted by (Fan and Palaniswami, 2000), (Shin et al., 2004), (van Gestel et al., 2003b), (Yun et al., 2004), (Lai et al., 2006a) and (Hui and Sun, 2006).

7.2 Market Risk Management

In recent years kernel methods have also been widely used within market risk management due to the well-known properties of financial time series which usually are inherently noisy, non-stationary and deterministically chaotic (Cao and Tay, 2001). The noisiness characteristic, which can lead to over- or underfitting when using estimation methods, implies that there is no complete information available from the past behavior of financial markets. Non-stationarity means that financial time series switch their dynamics between different regions. This behavior causes an ever changing dependency between input and output variables (Tay and Cao, 2001b). Thus, financial forecasting represents a promising area for the application of kernel methods.

(Cao and Tay, 2001) compared SVMs to a multi-layer perceptron with back-propagation in forecasting the S&P 500 Daily Index. They reported better forecasting abilities for the SVM approach and emphasised the low number of parameters (after the kernel had been specified) that had to be calibrated for the SVM. Remarkably, the choice of parameters had a minor influence on the results of their study when using SVMs while the neural network approach was much more affected by parameters. Also SVMs showed superior speed characteristics compared to back-propagation.

To further improve their results (Tay and Cao, 2001b) and (Cao, 2003) built a two-stage neural network architecture where they combined SVMs with a self organizing map (SOM) in a hybrid model. According to the 'divide-and-conquer' principle they first clustered the input data in disjoint regions, using SOMs. Secondly, they employed multiple SVMs (also called *SVM experts*) to fit each region separately with individual and most appropriate kernel functions and corresponding parameters. With this two-stage procedure they were able to significantly improve prediction performance for six different financial data sets compared to the above-mentioned single SVM benchmark model. Additionally, their model delivered more efficient learning through the data set splitting and it provided a sparser solution representation as less support vectors were used. Another hybrid variation of their standard model (Cao and Tay, 2001) was provided in (Tay and Cao, 2001a) where they used saliency analysis (SA) and genetic algorithms (GAs) to perform feature selection for the SVMs. Both methods improved convergence, generalisation performance, and training time, however, SA was slightly ahead of the GA version due to lower computational cost. With yet another contribution (Tay and Cao, 2002) enhanced SVMs to model non-stationarity of financial time series. Specifically

they included the assumption that more recent data may provide more relevant information than older data. The model was tested using real futures contracts and delivered again superior performance to standard SVMs.

In a comprehensive study (Hansen et al., 2006) compared SVMs to a selection of modern time-series prediction methods, namely exponential smoothing, autoregressive integrated moving average (ARIMA) and partially adaptive estimated ARIMA. Nine time-series from the Federal Reserve Economic Database (FRED) with widely differing characteristics were used to evaluate the model performances. Impressively, SVMs achieved the best predictive results for eight out of the nine investigated data sets.

The direction of daily stock price index changes was predicted by (Kim, 2003) in another study of financial time series forecasting using SVMs. According to his empirical investigation, SVMs outperformed back-propagation neural networks (BPN) and case-based reasoning (CBR). (Huang et al., 2005) also successfully predicted market movement direction for the NIKKEI 225 index with SVMs in comparison to a variety of benchmark methods. Remarkably a newly proposed hybrid approach consisting of a combination of all considered methods delivered even better classification results.

(Pérez-Cruz et al., 2003) employed SVMs to estimate the parameters of a GARCH model for the prediction of the conditional volatility of stock market returns. They found that—given normally distributed data—standard GARCH estimators (for instance maximum likelihood) return better results than SVMs, as they implicitly assumed the Gaussian distribution. However, in estimating non-normally distributed probability distribution functions (pdfs) SVMs outperformed standard methods substantially, being capable to approach any given distribution. It has to be noted that they only used linear SVMs, leaving improvement through the usage of kernels and non-linear SVMs open for further research.

Another detailed study on volatility forecasting and the specific problems of time series data was performed by (Gavrishchaka and Ganguli, 2003). The authors found that SVMs could successfully handle both long memory and multiscale effects of inhomogeneous markets without imposing the restrictive model assumptions of other methods. They emphasised the capability of SVMs to process real-time multiscale and high-frequency market data and their ability to tolerate data incompleteness. (Gavrishchaka and Banerjee, 2006) extended this analysis from foreign exchange data to stock market data (S&P 500 index).

Using kernel methods (Ince and Trafalis, 2006) selected stocks for short-term portfolio management. In their study they examined SVMs and minimax probability machines (MPM), both of which provided similarly good results, depending on a sensible choice of free parameters. By assuming that the efficient market hypothesis does not hold around companies' earnings announcements (Ince and Trafalis, 2006) were able to earn excess returns through trading individual stocks after their earnings announcements. (Trafalis et al.,

Table 1. Publications in the area of market (MR) and credit risk (CR) with times series prediction (TS), rating/scoring (RS) and bankruptcy prediction (BP) as specific applications.

Area	Field	Support Vector Classification	Support Vector Regression	Kernel PCA	Hybrid Approaches	Total
MR	TS	(Zhang et al., 2006)	(Trafalis and Ince, 2000) (van Gestel et al., 2001) (Cao and Tay, 2001) (Yang et al., 2002a) (Pérez-Cruz et al., 2003) (Kamruzzaman et al., 2003) (Cao, 2003) (Kim, 2003) (Gavrishchaka and Ganguli, 2003) (Gavrishchaka and Banerjee, 2006) (Hansen et al., 2006) (Ni and Yin, 2006)	(Cao et al., 2003b) (Cao et al., 2003a) (Ince and Trafalis, 2004)	(Tay and Cao, 2001b) (Tay and Cao, 2001a) (Huang et al., 2005)	19
MR	Else	(Ince and Trafalis, 2006)	(Trafalis et al., 2003) (Takeuchi et al., 2006)	-	-	3
CR	RS	(van Gestel et al., 2003a) (Baesens et al., 2003) (Huang et al., 2004) (Li et al., 2004a) (Schebesch and Stecking, 2005b) (Stecking and Schebesch, 2005) (Stecking and Schebesch, 2006) (Cao et al., 2006)(Chen and Shih, 2006) (Lai et al., 2006b)	-	-	(Wang et al., 2005) (Schebesch and Stecking, 2005a) (Lai et al., 2006a)	13
CR	BP	(Fan and Palaniswami, 2000) (van Gestel et al., 2003b) (Sánchez et al., 2004) (Shin et al., 2004)(Yun et al., 2004) (Härdle et al., 2005) (Min and Lee, 2005) (Shin et al., 2005) (Hui and Sun, 2006) (Min et al., 2006)	-	-	(Min et al., 2006)	11
#		22	14	3	7	46

2003) also employed SVM Regression for option pricing and obtained minimum mean square error compared to RBF and MLP networks.

Further applications in financial forecasting were presented by (Trafalis and Ince, 2000; Ince and Trafalis, 2004), (van Gestel et al., 2001), (Yang et al., 2002a), (Zhang et al., 2006) and (Kamruzzaman et al., 2003). Another interesting hybrid approach that again combined self organizing maps with SVMs for exchange rate prediction was presented by (Ni and Yin, 2006).

7.3 Synopsis and Possible Future Application Fields

As set forth in the preceding sections, kernel based algorithms have successfully been introduced in many different financial application areas. In table 1 we present a structured overview of these applications showing their individual publication dates. It can be concluded from the table that support vector regression plays a very dominant role within market risk management while in credit risk management mostly kernel based classification methods are employed. There are also a smaller number of kernel PCA applications and some hybrid approaches throughout most of the considered areas. Interestingly, we found within the sample of publications of this review that the publication frequency reached another peak in 2006 after an early peak in 2003. These numbers underpin the timeliness of the proposed methods and the domain specific advantages that kernel methods may deliver within finance.

Possible Future Application Fields

Besides the presented overview of machine learning concepts and the review of ongoing research efforts in the subject matter it is an aim of this contribution to identify promising trends for future application of kernel methods. Due to their strengths in statistical data analysis, SVMs in particular can possibly improve performance in numerous further financial application fields.

While SVM algorithms have been extensively applied in specific credit risk domains, like rating/scoring and bankruptcy prediction, their usage in the analysis of other credit risk relevant parameters like Loss Given Default (LGD)[6] for instance has received very little attention until today. The adequate estimation of such parameters is very important for banks' internal credit risk models and also for the fulfilment of supervisory regulations. Even though it usually involves very heterogenous data sets with possibly non-linear relations, banks commonly still trust in linear methods, like linear regression, to derive their parameter estimates in practice.

Referring to the presented advanced dimensionality reduction methods in section 3 there are also promising new application fields. (Thomason, 1998)

[6] LGD is a highly relevant parameter from the Basel II context (Basel Committee on Banking Supervision 2006) and represents the percentage of an engagement that a financial institution looses if a specific obligor defaults. It has the following relation to the alternatively quoted recovery rate (RR): $LGD = 1 - RR$

reviewed PCA as a traditional dimensionality reduction method in the context of financial forecasting models. He stated that standard PCA using the normal distribution assumption may be not particularly suited for financial data. While he proposed a self-developed advanced PCA approach, a number of authors have already employed other recently developed dimensionality reduction algorithms such as KPCA.

(Cao et al., 2003a,b) compared the performance of PCA, KPCA and independent component analysis (ICA) for feature extraction of different time series data sets (including real futures contracts). They found that KPCA had the best characteristics. Within short term portfolio management (Ince and Trafalis, 2004) used KPCA and factor analysis, respectively, to identify the most influential inputs for a SVM based stock price forecasting model.

Furthermore, within market risk management there are different new areas where dimensionality reduction can also be applied, for instance term structure modeling. In these models a high number of possible input factors has to be reduced to eventually make the modeling possible (Alexander, 2001). However, to the knowledge of the authors, no kernel method applications have been reported in this area yet.

Apart from such modeling issues, market risk management often involves the approximation of high quantiles for a certain distribution. This is especially relevant in the context of portfolio risk management where value at risk (VaR) measures the risk of a loss within a specific time interval given a certain confidence level (quantile). In a novel application (Christmann, 2005) and subsequently (Takeuchi et al., 2006) used SVMs to estimate these high quantiles.

8 Overview of Software Tools

There are a large number of kernel machine implementations freely available through the internet[7] with SVM algorithms clearly dominating.

LIBSVM[8] (Chang and Lin, 2001) has probably become the most popular SVM software package offering a variety of algorithms, including support vector classification (C-SVC, ν-SVC, multi-class classification), regression (epsilon-SVR, ν-SVR) and distribution estimation (one-class SVM). Apart from the source code in C++ and Java the software comes with numerous interfaces to different other software packages. It has been integrated as package *e1071* into the R project which is a popular open source statistics software[9]. Another R package including parts of LIBSVM is *kernlab* which extends the algorithm spectrum by KPCA, spectral clustering and more.

[7] Links to a selection of software are available at the following web sites:
www.support-vector-machines.org/SVM_soft.html
www.kernel-machines.org/software.html

[8] www.csie.ntu.edu.tw/∼cjlin/libsvm/

[9] www.r-project.org

SVMlight[10] (Joachims, 1999) implements SVMs for pattern recognition, regression and learning of a ranking function in C code. One of the main strengths of this implementation is the fact that through scalable memory requirements it can handle problems with many thousands of support vectors and several hundred thousands of training vectors very efficiently. Additionally the package *klaR* for the R project and a Java version called *mySVM*[11] have been implemented. Users can define their own kernel functions in SVMlight and find some example problems on the author's web site.

LS-SVMlab[12] (Suykens et al., 2002) is a least-squares SVM toolbox for Matlab which is also available in C. This well-documented software package offers standard classification and regression using LS-SVM algorithms. Additionally KPCA, ultra large scale problems and a number of other advanced methods are supported.

WEKA[13] (Witten and Frank, 2005) is a sophisticated environment with graphical user interface for machine learning and data mining which is implemented in Java. It includes a large library of classification algorithms including SVMs and neural networks as well as evaluation tools such as ROC curves (Fawcett, 2004).

The Spider[14] is a Matlab based toolbox with interfaces to several Matlab and C/C++ libraries for kernel-based algorithms. It includes a large variety of tools for preprocessing, training and evaluation. Several kernel methods are implemented including SVMs for classification and regression, one-class SVMs, and KPCA. A WEKA interface has also been integrated.

SHOGUN[15] (Sonnenburg et al., 2006) is a new machine learning toolbox which focuses on SVMs and implements a variety of kernels including several string kernels which are important in computational biology. It offers the option to employ combined kernels which can be constructed by weighted linear combinations of sub-kernels. SHOGUN connects to LIBSVM (Chang and Lin, 2001) and SVMlight (Joachims, 1999). It is implemented in C++ and has interfaces to Matlab, Octave, Python and R.

In addition to the already mentioned implementations of KPCA in kernlab and LS-SVMlab, software toolboxes for dimensionality reduction methods are often available from the associated authors' webpages or general resource pages on manifold learning.[16]

[10] svmlight.joachims.org/
[11] www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html
[12] www.esat.kuleuven.ac.be/sista/lssvmlab/
[13] www.cs.waikato.ac.nz/∼ml/weka/
[14] www.kyb.tuebingen.mpg.de/bs/people/spider/main.html
[15] www2.fml.tuebingen.mpg.de/raetsch/projects/shogun
[16] www.cse.msu.edu/∼lawhiu/manifold/

9 Conclusion

The overview of kernel methods showed that the field has quickly advanced in recent years and provides an umbrella for some of the most successful algorithms for classification, regression, and dimensionality reduction. Non-linear methods such as KPCA, Isomap, and non-linear SVMs for regression and classification can be obtained through kernelisation of linear techniques.

The development on the machine learning side is rapid and new concepts and improvements, which have not been yet applied in finance, are continually emerging. Recent advances in the machine learning community to establish task specific kernel design offer new opportunities and challenges for financial applications.

Among the financial applications addressed in the review section notably the best results have been obtained in the area of credit risk whenever the underlying data exhibited non-linear characteristics, as for instance in (Baesens et al., 2003; van Gestel et al., 2003a, 2006).

Although SVMs and KPCA have been successfully applied in several studies on financial data the field of "Kernel Methods in Finance" is still in the early stages of development. The availability of kernel methods to accurately handle nonlinear dependencies has potential to further enhance current results. With respect to the high amounts that are dealt with on the financial markets even very small performance or accuracy improvements can result in considerable savings.

10 Acknowledgements

The authors are grateful to Stephen Young who worked as research assistant and supported generation of the figures. The second author would like to thank GILLARDON AG financial software for support of his work. Nevertheless, the views expressed in this chapter reflect the personal opinion of the authors and are neither official statements of GILLARDON AG financial software nor of its partners or its clients.

References

Aizerman, M., Braverman, E., Rozonoer, L. (1964) Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control 25, 821–837.

Alexander, C. (2001) Market Models: A Guide to Financial Data Analysis. John Wiley & Sons, Chichester.

Baesens, B., van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. A. K., Vanthienen, J. (2003) Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society 54 (6), 627–635.

Basel Committee (Juni 2006) Basel II: International convergence of capital measurement and capital standards: A revised framework - Comprehensive version. Basel Committee on Banking Supervision, Bank for International Settlements.

Belkin, M., Niyogi, P. (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15 (6), 1373–1396.

Ben-Hur, A., Horn, D., Siegelmann, H. T., Vapnik, V. N. (2001) Support vector clustering. Journal of Machine Learning Research 2, 125–137.

Bishop, C. M. (2006) Pattern Recognition and Machine Learning. Springer.

Boser, B. E., Guyon, I. M., Vapnik, V. N. (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory. ACM Press, pp. 144–152.

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

Burges, C. J. C. (2005) Geometric methods for feature extraction and dimensional reduction - a guided tour. In: Maimon and Rokach (2005), pp. 59–92.

Camps-Valls, G., Rojo-Álvarez, J. L., Martínez-Ramón, M. (Eds.) (2007) Kernel methods in bioengineering, signal and image processing. Idea Group Inc., Hershey, PA, USA.

Cao, L. (2003) Support vector machines experts for time series forecasting. Neurocomputing 51, 321–339.

Cao, L., Chua, K. S., Chong, W. K., Lee, H. P., Gu, Q. M. (2003a) A comparison of PCA, KPCA, ICA for dimensionality reduction in support vector machine. Neurocomputing 55, 321–336.

Cao, L., Chua, K. S., Guan, L. K. (2003b) Combining KPCA with support vector machine for time series forecasting. In: IEEE Computational Intelligence for Financial Engineering. pp. 325–329.

Cao, L., Guan, L. K., Jingqing, Z. (2006) Bond rating using support vector machine. Intelligent Data Analysis 10, 285–296.

Cao, L., Tay, F. E. H. (2001) Financial forecasting using support vector machines. Neural Computing & Applications 10, 184–192.

Carminati, L., Benois-Pineau, J. (2005) Support vector tracking of human faces with affine motion models. In: Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2005. Suisse (Montreux).

Chang, C.-C., Lin, C.-J. (2001) LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Chapelle, O., Schölkopf, B., Zien, A. (Eds.) (2006) Semi-Supervised Learning. The MIT Press, Cambridge, MA.

Chen, D., Cao, X. B., Xu, Y. W., Qiao, H. (2006) An evolutionary support vector machines classifier for pedestrian detection. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4223–4227.

Chen, W.-H., Shih, J.-Y. (2006) A study of Taiwan's issuer credit rating systems using support vector machines. Expert Systems with Applications 30 (3), 427–435.

Christmann, A. (2005) On a combination of convex risk minimization methods. In: Gaul, W., Weihs, C. (Eds.), Classification - the Ubiquitous Challenge. Springer, pp. 434–441.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2001) Introduction to Algorithms, 2nd Edition. MIT Press and McGraw-Hill.

Cortes, C., Vapnik, V. N. (1995) Support vector networks. Machine Learning 20, 273–297.

Courant, R., Hilbert, D. (1953) Methods of Mathematical Physics. Interscience Publishers, Inc, New York.

Cox, T. F., Cox, M. A. A. (2001) Multidimensional scaling, 2nd Edition. Chapman & Hall/CRC.

Cristianini, N., Shawe-Taylor, J. (2000) An introduction to support vector machines and other kernel based learning methods. Cambridge University Press.

Evgeniou, T., Pontil, M., Poggio, T. (2000) Regularization networks and support vector machines. Advances in Computational Mathematics 13 (1), 1–50.

Fan, A., Palaniswami, M. (2000) Selecting bankruptcy predictors using a support vector machine approach. In: IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (2000) Vol. 6. pp. 354–359.

Fawcett, T. (2004) ROC graphs: Notes and practical considerations for researchers. Tech. rep., HP Laboratories, Palo Alto, CA, USA, Tech Report HPL-2003-4.

Francois, D., Wertz, V., Verleysen, M. (2005) On the locality of kernels in highdimensional spaces. In: ASMDA2005, Applied Stochastic Models and Data Analysis. pp. 238–245.

Fu, X., Ma, Z., Feng, B. (2004) Kernel-based semantic text categorization for large scale web information organization. Vol. 3251 of Lecture Notes in Computer Science (LNCS). Springer, pp. 389–396.

Gavrishchaka, V. V., Banerjee, S. (2006) Support vector machine as an efficient framework for stock market volatility forecasting. In: Computational Management Science. Vol. 3. Springer, pp. 147–160.

Gavrishchaka, V. V., Ganguli, S. B. (2003) Volatility forecasting from multiscale and high-dimensional market data. Neurocomputing 55, 285–305.

Gentle, J. E., Härdle, W., Mori, Y. (Eds.) (2004) Handbook of Computational Statistics. Concepts and Methods. Springer.

Ham, J., Lee, D. D., Mika, S., Schölkopf, B. (2004) A kernel view of the dimensionality reduction of manifolds. In: Proceedings of the 21st International Conference on Machine Learning.

Hansen, J. V., McDonald, J. B., Nelsen, R. D. (2006) Some evidence on forecasting time-series with support vector machines. Journal of the Operational Research Society 57 (9), 1053–1063.

Härdle, W., Moro, R. A., Schäfer, D. (2005) Predicting bankruptcy with support vector machines. In: Cizek, P., Härdle, W., Weron, R. (Eds.), Statistical Tools for Finance and Insurance. Springer, pp. 225–248.

Haykin, S. (1999) Neural Networks. A Comprehensive Foundation, 2nd Edition. Prentice Hall.

Herbrich, R. (2002) Learning Kernel Classifiers. The MIT Press.

Hertz, T., Hillel, A.-B., Weinshall, D. (2006) Learning a kernel function for classification with small training samples. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh June 25-29, 2006.

Hochreiter, S., Mozer, M. C., Obermayer, K. (2003) Coulomb classifiers: Generalizing support vector machines via an analogy to electrostatic systems. In: Advances in Neural Information Processing Systems (NIPS). Vol. 15. The MIT Press, pp. 561–568.

Hotelling, H. (1933) Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24, 417–441, 498–520.

Huang, T.-M., Kecman, V., Kopriva, I. (2006) Kernel Based Algorithms for Mining Huge Data Sets. Supervised, Semi-supervised, and Unsupervised Learning. Vol. 17 of Studies in Computational Intelligence. Springer.

Huang, W., Nakamori, Y., Wang, S.-Y. (2005) Forecasting stock market movement direction with support vector machine. Computers & Operations Research 32, 2513–2522.

Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., Wu, S. (2004) Credit rating analysis with support vector machines and neural networks: a market comparative study. Decision Support Systems (37), 543–558.

Hui, X.-F., Sun, J. (2006) An application of support vector machine to companies' financial distress prediction. Vol. 3885 of Lecture Notes in Computer Science (LNCS). Springer, pp. 274–282.

Ikeda, K., Murata, N. (2005) Geometrical properties of nu support vector machines with different norms. Neural Computation 17, 2508–2529.

Ince, H., Trafalis, T. B. (2004) Kernel principal component analysis and support vector machines for stock price prediction. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks. Vol. 3. pp. 2053–2058.

Ince, H., Trafalis, T. B. (2006) Kernel methods for short-term portfolio management. Expert Systems with Applications 30, 535–542.

Joachims, T. (1999) Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods - Support Vector Learning. The MIT Press.

Jolliffe, I. T. (1986) Principal Component Analysis. Springer-Verlag, New York.

Kamruzzaman, J., Sarker, R. A., Ahmad, I. (2003) SVM based models for predicting foreign currency exchange rates. In: ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining. IEEE Computer Society, Washington, DC, USA, p. 557.

Kang, S., Byun, H., Lee, S.-W. (2002) Real-time pedestrian detection using support vector machines. Vol. 2388 of Lecture Notes in Computer Science (LNCS). Springer, pp. 268–277.

Kim, K.-j. (2003) Financial time series forecasting using support vector machines. Neurocomputing 55, 307–319.

Lai, K. K., Yu, L., Huang, W., Wang, S. (2006a) A novel support vector machine metamodel for business risk identification. Vol. 4099 of Lecture Notes in Computer Science (LNCS). Springer, pp. 980–984.

Lai, K. K., Yu, L., Zhou, L., Wang, S. (2006b) Credit risk evaluation with least square support vector machine. Vol. 4062 of Lecture Notes in Computer Science (LNCS). Springer, pp. 490–495.

Lanckriet, G. R. G., Bie, T. D., Cristianini, N., Jordan, M. I., Noble, W. S. (2004) A statistical framework for genomic data fusion. Bioinformatics 20, 2626–2635.

Li, J., Liu, J., Xu, W., Shi, Y. (2004a) Support vector machines approach to credit assessment. Vol. 3039 of Lecture Notes in Computer Science (LNCS). Springer, pp. 892–899.

Li, Y., Gong, S., Sherrah, J., Liddell, H. (2004b) Support vector machine based multi-view face detection and recognition. Image and Vision Computing 22, 413–427.

Maimon, O., Rokach, L. (Eds.) (2005) The Data Mining and Knowledge Discovery Handbook. Springer.

Mangasarian, O. L. (1999) Arbitrary-norm separating plane. Operations Research Letters 24, 15–23.

Mercer, J. (1909) Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Roy. Soc. London A 209, 415–446.

Mika, S., Schäfer, C., Laskov, P., Tax, D., Müller, K.-R. (2005) Support vector machines. In: Gentle, J. E., Härdle, W., Mori, Y. (Eds.), Handbook of Computational Statistics. Concepts and Methods. Springer, pp. 841–876.

Min, J. H., Lee, Y.-C. (2005) Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Systems with Applications 28, 603–614.

Min, S.-H., Lee, J., Han, I. (2006) Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Systems with Applications 31, 652–660.

Mitchell, T. (1997) Machine Learning. McGraw Hill.

Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B. (2001) An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12 (2), 181–201.

Ni, H., Yin, H. (2006) Recurrent self-organising maps and local support vector machine models for exchange rate prediction. Vol. 3973 of Lecture Notes in Computer Science (LNCS). Springer, pp. 504–511.

Osuna, E., Freund, R., Girosi, F. (1997) Training support vector machines: An application to face detection. In: CVPR '97: Proceedings of the 1997 Con-

ference on Computer Vision and Pattern Recognition (CVPR '97). IEEE Computer Society, Washington, DC, USA, pp. 130–136.

Pedroso, J. P., Murata, N. (2001) Support vector machines with different norms: Motivation, formulations, and results. Pattern Recognition Letters 22, 1263–1272.

Pérez-Cruz, F., Afonso-Rodríguez, J. A., Giner, J. (2003) Estimating GARCH models using support vector machines. Quantitative Finance 3 (3), 163–172.

Platt, J. (1999) Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods - Support Vector Learning. The MIT Press.

Raskutti, B., Kowalczyk, A. (2004) Extreme re-balancing for svms: a case study. SIGKDD, Explorations 6 (1), 60–69.

Rifkin, R., Klautau, A. (2004) In defense of one-vs-all classification. Journal of Machine Learning Research 5, 101–141.

Rosenblatt, F. (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65 (6), 386–408.

Roweis, S. T., Saul, L. K. (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290 (5500), 2323–2326.

Sánchez, M., Prats, F., Agell, N., Rovira, X. (2004) Kernel functions over orders of magnitude spaces by means of usual kernels. Application to measure financial credit risk. Vol. 3040 of Lecture Notes in Computer Science (LNCS). Springer, pp. 415–424.

Saul, L. K., Weinberger, K. Q., Sha, F., Ham, J., Lee, D. D. (2006) Spectral Methods for Dimensionality Reduction, Ch. 16. In: Chapelle et al. (2006), pp. 293–308.

Schebesch, K. B., Stecking, R. (2005a) Support vector machines for classifying and describing credit applicants: detecting typical and critical regions. Journal of the Operational Research Society 56 (9), 1082–1088.

Schebesch, K. B., Stecking, R. (2005b) Support vector machines for credit scoring: Extension to non standard cases. In: Baier, D., Wernecke, K.-D. (Eds.), Innovations in Classification, Data Science, and Information Systems. Proceedings of the 27th Annual Conference of the Gesellschaft für Klassifikation e.V., Brandenburg University of Technology, Cottbus, March 12-14, 2003. Springer, pp. 498–505.

Schölkopf, B. (1997) Support vector learning. R. Oldenbourg Verlag, Munich.

Schölkopf, B., Smola, A., Müller, K.-R. (1996) Nonlinear component analysis as a kernel eigenvalue problem. Tech. Rep. 44, Max-Planck-Institut für biologische Kybernetik.

Schölkopf, B., Smola, A., Müller, K.-R. (1997) Kernel principal component analysis. Vol. 1327 of Lecture Notes in Computer Science (LNCS). Springer, Berlin, pp. 583–588.

Schölkopf, B., Smola, A., Müller, K.-R. (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319.

Schölkopf, B., Smola, A. J. (2002) Learning with Kernels. The MIT Press, Cambridge, Massachusetts.

Schölkopf, B., Tsuda, K., Vert, J.-P. (Eds.) (2004) Kernel Methods in Computational Biology. The MIT Press.

Shawe-Taylor, J., Cristianini, N. (2004) Kernel methods for pattern analysis. Cambridge University Press.

Shin, K.-s., Lee, K. J., Kim, H.-j. (2004) Support vector machines approach to pattern detection in bankruptcy prediction and its contingency. Vol. 3316 of Lecture Notes in Computer Science (LNCS). Springer, pp. 1254–1259.

Shin, K.-s., Lee, T. S., Kim, H.-j. (2005) An application of support vector machines in bankruptcy prediction model. Expert Systems with Applications 28, 127–135.

Sonnenburg, S., Raetsch, G., Schaefer, C., Schoelkopf, B. (2006) Large scale multiple kernel learning. Journal of Machine Learning Research 7, 1531–1565.

Sonnenburg, S., Rätsch, G., Schölkopf, B. (2005) Large scale genomic sequence svm classifiers. In: Proceedings of the International Conference on Machine Learning, ICML 2005.

Spivac, M. (1979) A Comprehensive Introduction to Differential Geometry, 2nd Edition. Publish or Perish, Inc.

Stecking, R., Schebesch, K. B. (2005) Informative patterns for credit scoring using linear svm. In: Weihs, C., Gaul, W. (Eds.), Classification - the Ubiquitous Challenge: Proceedings of the 28th Annual Conference of the Gesellschaft für Klassifikation e.V., University of Dortmund. Springer, pp. 450–457.

Stecking, R., Schebesch, K. B. (2006) Comparing and selecting svm-kernels for credit scoring. In: Spiliopoulou, M., Kruse, R., Borgelt, C., Nürnberger, A., Gaul, W. (Eds.), From Data and Information Analysis to Knowledge Engineering. Proceedings of the 29th Annual Conference of the Gesellschaft für Klassifikation e.V., University of Magdeburg, March 9-11, 2005. Springer, pp. 542–549.

Stone, M. (1974) Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society. Series B (Methodological) 36 (2), 111–147.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J. (2002) Least Squares Support Vector Machines. World Scientific Pub. Co., Singapore.

Suykens, J. A. K., Vandewalle, J. (1999a) Least squares support vector machine classifiers. Neural Processing Letters 9, 293–300.

Suykens, J. A. K., Vandewalle, J. (1999b) Training multilayer perceptron classifiers based on a modified support vector method. IEEE Transactions on Neural Networks 10 (4), 907–911.

Takeuchi, I., Le, Q. V., Sears, T. D., Smola, A. J. (2006) Nonparametric quantile estimation. Journal of Machine Learning Research 7, 1231–1264.

Tax, D. M. J., Duin, R. P. W. (1999) Support vector domain description. Pattern Recognition Letters 20 (11–13), 1191–1199.

Tay, F. E. H., Cao, L. (2001a) A comparative study of saliency analysis and genetic algorithm for feature selection in support vector machines. Intelligent Data Analysis 5 (3), 191–209.

Tay, F. E. H., Cao, L. (2001b) Improved financial time series forecasting by combining support vector machines with self-organizing feature map. Intelligent Data Analysis 5 (4), 339–354.

Tay, F. E. H., Cao, L. (2002) Modified support vector machines in financial time series forecasting. Neurocomputing 48, 847–861.

Tenenbaum, J. B., de Silva, V., Langford, J. C. (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323.

Thomas, L. C., Oliver, R. W., Hand, D. J. (2005) A survey of the issues in consumer credit modelling research. Journal of the Operational Research Society 56, 1006–1015.

Thomason, M. R. (1998) Non-traditional PCA for dimensionality reduction of financial forecasting models. Journal of Computational Intelligence in Finance 6 (4), 34–38.

Trafalis, T. B., Ince, H. (2000) Support vector machine for regression and applications to financial forecasting. In: IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000. Vol. 6. pp. 348–353.

Trafalis, T. B., Ince, H., Mishina, T. (2003) Support vector regression in option pricing. Conference on Computational Intelligence for Financial Engineering: Workshop Paper, Hong Kong Baptist University.

Valiant, L. G. (1984) A theory of the learnable. Communications of the ACM 27 (11), 1134–1142.

van Gestel, T., Baesens, B., Garcia, J., van Dijcke, P. (2003a) A support vector machine approach to credit scoring. Bank en Financiewezen 2, 73–82.

van Gestel, T., Baesens, B., Suykens, J., Baestaens, D., Vanthienen, J., De Moor, B. (2003b) Bankruptcy prediction with least squares support vector machine classifiers. In: Proceedings of the Conference on Computational Intelligence for Financial Engineering (CIFEr'03), March 21-23, Hong Kong. IEEE.

van Gestel, T., Baesens, B., Suykens, J. A. K., van den Poel, D., Baestaens, D.-E., Willekens, M. (2006) Bayesian kernel based classification for financial distress detection. European Journal of Operational Research 172, 979–1003.

van Gestel, T., Suykens, J. A. K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B., Vandewalle, J. (2001) Financial time series prediction using least squares support vector machines within the evidence framework. IEEE Transactions on Neural Networks 12 (4), 809–821.

Vapnik, V. N. (1998) Statistical Learning Theory. John Wiley & Sons, NY.

Vapnik, V. N. (2000) The nature of statistical learning theory. Springer, NY.

Vapnik, V. N. (2006a) Estimation of Dependencies Based on Empirical Data, Ch. Empirical Inference Science, Afterword 2006. In: Vapnik (2006b), pp. 400–505.

Vapnik, V. N. (2006b) Estimation of dependencies based on empirical data, reprint of 1982 edition, 2nd Edition. Springer.

Verleysen, M., Francois, D. (2005) The curse of dimensionality in data mining and time series prediction. Vol. 3512 of Lecture Notes in Computer Science (LNCS). Springer, pp. 758–770.

Wang, Y., Wang, S.-Y., Lai, K. K. (2005) A new fuzzy support vector machine to evaluate credit risk. IEEE Transactions on Fuzzy Systems 13 (6), 820–831.

Weinberger, K. Q., Packer, B. D., Saul, L. K. (2005) Unsupervised learning of image manifolds by semidefinite programming. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics. Barbados.

Weinberger, K. Q., Saul, L. K. (2004) Unsupervised learning of image manifolds by semidefinite programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04). Vol. 2. Washington D. C., pp. 988–995.

Weinberger, K. Q., Sha, F., Saul, L. K. (2004) Learning a kernel matrix for nonlinear dimensionality reduction. In: Proceedings of the Twenty First International Conference on Machine Learning (ICML-04). Banff, Canada, pp. 839–846.

Williams, C. K. I. (2001) On a connection between kernel pca and metric multidimensional scaling. In: Advances in Neural Information Processing Systems (NIPS). Vol. 13.

Witten, I. H., Frank, E. (2005) Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco.

Xiao, L., Sun, J., Boyd, S. (2006) A duality view of spectral methods for dimensionality reduction. In: Proceedings of the 23rd International Conference on Machine Learning (ICML 2006). pp. 1041–1048.

Yang, H., Chan, L., King, I. (2002a) Support vector machine regression for volatile stock market prediction. Vol. 2412 of Lecture Notes in Computer Science (LNCS). Springer, pp. 391–396.

Yang, J., Estivill-Castro, V., Chalup, S. (2002b) Support vector clustering through proximity graph modelling. In: Proceedings, International Conference on Neural Information Processing (ICONIP'2002). pp. 898–903.

Yun, Y., Yoon, M., Nakayama, H., Shiraki, W. (2004) Prediction of business failure by total margin support vector machines. Vol. 3213 of Lecture Notes in Computer Science (LNCS). Springer, pp. 441–448.

Zhang, Z.-y., Shi, C., Zhang, S.-l., Shi, Z.-z. (2006) Stock time series forecasting using support vector machines employing analyst recommendations. Vol. 3973 of Lecture Notes in Computer Science (LNCS). Springer, pp. 452–457.