Chapter XXVIII
Support Vector Machine:
Itself an Intelligent Systems

A. B. M. Shawkat Ali
Central Queensland University, Australia

ABSTRACT

From the beginning, machine learning methodology, which is the origin of artificial intelligence, has been rapidly spreading in the different research communities with successful outcomes. This chapter aims to introduce for system analysers and designers a comparatively new statistical supervised machine learning algorithm called support vector machine (SVM). We explain two useful areas of SVM, that is, classification and regression, with basic mathematical formulation and simple demonstration to make easy the understanding of SVM. Prospects and challenges of future research in this emerging area are also described. Future research of SVM will provide improved and quality access to the users. Therefore, developing an automated SVM system with state-of-the-art technologies is of paramount importance, and hence, this chapter will link up an important step in the system analysis and design perspective to this evolving research arena.

INTRODUCTION

Since the end of the last century, support vector machines (SVMs) have been introduced for classification and regression in the machine learning community. SVMs have a solid theoretical foundation rooted in statistical learning theory. SVMs work step by step. First, it maps the data into a high dimensional space via a nonlinear map, and in this high dimensional space it constructs an optimal separating hyperplane or linear regression function. This hyperplane or linear regression function obtained in the feature space couple with significant data points for prediction called support vectors (SVs). Therefore SVM do prediction based on SVs' information only. This process will involve a quadratic programming problem, and this will get a global optimal solution. This chapter formulates the statistical method of SVM based on classification and regression architecture. We explained both SVM classification methods: binary and muticlass classification. In each section we included a demonstration to easily understand SVM classification and regression methodology. Finally we give attention on how system analysers and designers can contribute to the construction of a full automated support vector learning algorithm.

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

BACKGROUND

The popular statistical learning algorithm, SVM, is an advanced version of the generalised portrait algorithm, which was developed in Russia in the late 60s (Smola & Schölkopf, 1998). After a large gap, Vapnik (1995) and his group simplified this theory and introduced SVM as an effective learning algorithm. Although SVM does not have a long history, it has already been successfully applied with significant outcomes in business, engineering, science, medicine, and many more.

SVM was introduced first to solve binary class pattern recognition problems, and then multiclass classification, regression estimation, and so many others. We can divide SVM literature into two parts: SVM classification and SVM regression. The following sections will cover both parts following the basic explanation with example.

SVM CLASSIFICATION

Let us consider a dataset D of l independently identically distributed (i.i.d) samples: $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)$. Each sample is a set of feature vectors of length m, $\mathbf{x}_i = \langle x_1, \cdots, x_m \rangle$ and the target value $y_i \in \{+1, -1\}$ that represents the binary class membership. Now, the pattern recognition problem or machine learning task is to learn the classes for each pattern by finding a classifier with decision functions $f(\mathrm{x}_i, \alpha_i)$, where, $f(\mathbf{x}_i, \alpha_i) = y_i, \alpha_i \in \Lambda, \forall \langle \mathbf{x}_i, y_i \rangle \in D$ and Λ is a set of abstract parameters.

Linear Hard Margin SVM

Now-a-days the general nonlinear SVM is quite popular to solve pattern recognition problems, but the root of this method is linear SVM. The original linear SVM was introduced to separate the binary class problem only.

Let us consider the above pattern recognition problem. Our aim is to find out the optimal hyperplane (OH) in the training phase with proper estimation of a weight vector w and the scalar bias factor b. All the training patterns are said to be linearly separable if there exists w and b such that the inequalities

$$(\omega \cdot \mathbf{x}_i) + b \geq 1 \qquad \text{if } y_i = 1 \text{ or } \bullet \qquad (1)$$

$$(\omega \cdot \mathbf{x}_i) + b \leq -1 \qquad \text{if } y_i = -1 \text{ or } \mathrm{C} \qquad (2)$$

These two sets of inequalities we can present into a single set such as

$$y_i = sign(\omega \cdot \mathbf{x}_i + b), \qquad i = 1, \ldots, \ell \qquad (3)$$

After extracting the OH, the whole set of vectors $\{\mathsf{w}\}$ will satisfy the Equation (3) as described in Figure 1.

The margin could be found by measuring the distance d between the binary classes' data points as shown in Figure 1 as follows:

$$\begin{aligned} d = d_{+1} + d_{-1} &= \min_{i:(y_i=+1)} d(\omega, b; \mathbf{x}_i) - \min_{i:(y_i=-1)} d(\omega, b; \mathbf{x}_i) \\ &= \min_{i:(y_i=+1)} \frac{|\langle \omega, \mathbf{x}_i \rangle + b|}{\|\omega\|} - \min_{i:(y_i=-1)} \frac{|\langle \omega, \mathbf{x}_i \rangle + b|}{\|\omega\|} \\ &= \frac{1}{\|\omega\|} \left(\min_{i:(y_i=+1)} |\langle \omega, \mathbf{x}_i \rangle + b| - \min_{i:(y_i=-1)} |\langle \omega, \mathbf{x}_i \rangle + b| \right) \\ &= \frac{2}{\|\omega\|} \end{aligned} \qquad (4)$$

where d_{+1} and d_{-1} are the distance of the closest positive and negative class data points from the OH.

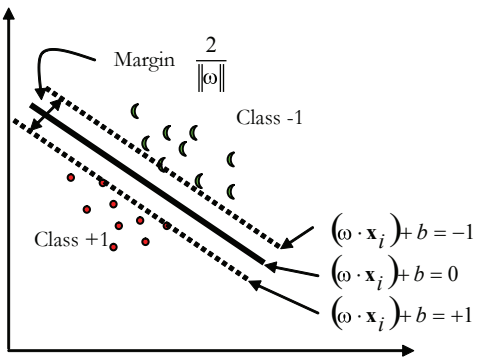
Now the OH can be obtained through the minimisation:

$$\Phi(\omega) = \frac{1}{2}\|\omega\|^2 \qquad (5)$$

subject to: $y_i[(\omega \cdot \mathbf{x}_i) + b] \geq 1$.

It is important to mention that the OH solution is independent with the scalar quantity bias b. The OH will shift up or down due to any changing of b, but the maximum margin will remain same.

Figure 1. The hard margin linear classification. The margin is the distance between the two dashed lines.



The above optimisation problem in Equation (5) can be solved using standard quadratic optimisation techniques. We construct the Lagrangian function as follows:

$$L(\omega,b,\alpha)=\frac{1}{2}\|\omega\|^2-\sum_{i=1}^{\ell}\alpha_i\left\{y_i\left(\omega\cdot\mathbf{x}_i+b\right)-1\right\} \tag{6}$$

where $\alpha_i = (\alpha_1,...,\alpha_l)$ are the Lagrange multipliers.

The solution of the Lagrangian optimisation problem can be determined by a saddle point of this Lagrangian, which is to be minimised with respect to ω and b and maximised with respect to non-negative α. By getting a differentiation of Equation (6) and setting the equality with zero we can obtain:

$$\frac{\partial L(\omega,b,\alpha)}{\partial\omega}=\omega-\sum_{i=1}^{l}\alpha_i y_i\mathbf{x}_i=0 \tag{7}$$

$$\frac{\partial L(\omega,b,\alpha)}{\partial b}=\sum_{i=1}^{l}\alpha_i y_i=0 \tag{8}$$

Now, using the upper score symbol (ω^o) to identify the optimal values of the cost function, we can obtain from Equation (7)

$$\omega^o=\sum_{i=1}^{l}\alpha_i^o\mathbf{x}_i y_i,\qquad \alpha_i^o\geq 0 \tag{9}$$

which shows the OH solution is a linear combination of the vectors of the training data points. It is noted that the training vectors $\mathbf{x}_i$ with $\alpha_O \geq 0$ only contribute to construct the OH. This fact follows the classical Karush-Kuhn-Tucker (KKT) conditions. Now substituting Equations (8) and (9) in Equation (6) and considering the KKT condition we can write

$$\begin{aligned}\alpha_i^o&=\frac{1}{2}\omega.\omega-\omega.\omega-0+\sum_{i=1}^{l}\alpha_i\\&=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\,\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)\end{aligned} \tag{10}$$

From this above expansion those training vectors that have nonzero coefficient α_i^o in the expansion of ω^o (Equation 9) are called *support vectors*, which play a significant role in constructing the OH.

The scalar bias (threshold) factor b can be obtained

$$b^o=\frac{1}{2}\left[\left(\omega^o\cdot\mathbf{x}_{+1}+\omega^o\cdot\mathbf{x}_{-1}\right)\right] \tag{11}$$

where $\mathbf{x}_{+1}$ and $\mathbf{x}_{-1}$ and $\mathbf{x}_{-1}$ are indicating the SVs belonging to the +1 and -1 classes.

Finally, the linearity of the dot product is the outcome of Equations (9) and (11), and then we can write the decision function for the hard margin classifier as follows:

$$\hat{f}(\mathbf{x})=sign(\omega^o\cdot\mathbf{x}+b)=sign\left(\sum_{SVs}\alpha_i^o y_i\left(\mathbf{x}_i\cdot\mathbf{x}\right)+b^o\right) \tag{12}$$

The hard margin classifier can not handle linearly nonseparable patterns though. Therefore we consider the soft margin classifier to improve the performance on linearly nonseparable patterns.

Linear Hard to Soft Margin SVM

The hard margin classifier fails to classify the linearly nonseparable patterns. Due to this, a soft margin classifier is introduced to classify linearly nonseparable patterns (Cortes & Vapnik, 1995). The soft margin hyperplane will not be able to classify with zero errors but the generalisation performance is expected to be better than the hard margin classifier. The soft margin hyperplane can be constructed by the vector $\mathbf{w}$ that minimises the function

$$\Phi(\omega,\xi)=\frac{1}{2}\|\omega\|^2+C\left(\sum_{i=1}^{l}\xi_i\right) \quad (13)$$

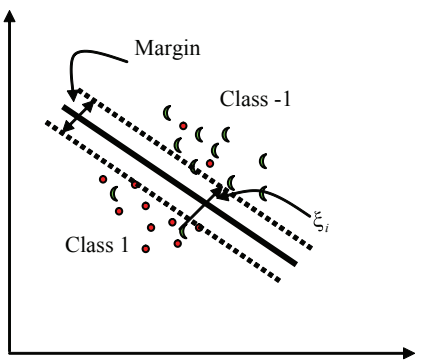
subject to: $y_i[(\omega\cdot\mathbf{x}_i)+b]\geq 1-\xi_i$

$\xi_i\geq 0$ where $i=1,\cdots,l$

where C is called hyperparameter and ξ is called slack variable. The maximum value of C could be up to infinity. The value of C is the boundary for finding α_i in the quadratic programming (QP) solution. Chapelle, Vapnik, Bousquet, and Mukherjee (2002) suggest a method for choosing the value of optimal C. The slack variable is determined by optimisation to minimise the violation of constraints.

In order to solve the problem we can construct the Lagrangian optimisation function as follows:

Figure 2. Problem of non separable patterns representation due to overlapping distributions



$$L(\omega,b,\xi,\alpha)=\frac{1}{2}\|\omega\|^2-\sum_{i=1}^{\ell}\alpha_i[y_i(\omega\cdot\mathbf{x}_i+b)-1+\xi_i]-\sum_{i=1}^{l}\alpha_i\xi_i+C\left(\sum_{i=1}^{l}\xi_i\right) \quad (14)$$

One can find the OH by following the hard margin Lagrangian quadratic solving procedure with the slightly different constraints such that:

$$0\leq\alpha_i\leq C$$
$$\sum_{i=1}^{l}\alpha_i y_i=0 \quad (15)$$

By following these procedures we can extract the SVs and then the decision function as follows:

$$\hat{f}(\mathbf{x})=sign(\omega^o\cdot\mathbf{x}+b)=sign\left(\sum_{SVs}\alpha_i^o y_i(\mathbf{x}_i\cdot\mathbf{x})+b^o\right) \quad (16)$$

Unfortunately, real world pattern recognition problems are almost always nonlinear. It is a difficult task to accurately classify nonlinear problems even by soft margin SVM because the margin construction procedure is linear. Therefore we need to construct the nonlinear margin for SVM. So we consider nonlinear SVM to handle nonlinear patterns.

Nonlinear SVM

In the previous sections we have explained the hard and soft margin-based linear hyperplane methodology, which is not suitable for most real world classification problems. Now we shall introduce the nonlinear SVM. We shall see that SVM can construct a nonlinear decision boundary in the nonlinear separable data space. The decision boundary for nonlinear patterns is conceptually quite simple and is done by mapping the input vectors $\mathbf{x}$ into a higher dimensional space, namely *feature space*, where a linear classification boundary can be constructed

easily. One can transform the input vectors $\mathbf{x}$ into a high dimensional feature space as follows:

$$\mathbf{x} \to \Phi(\mathbf{x}) = \left(a_1\Phi(\mathbf{x}), a_2\Phi_2(\mathbf{x}), \cdots, a_n\Phi_n(\mathbf{x}), \cdots\right) \tag{17}$$

where $\{a_n\}_{n=1}^{\infty}$ are the real numbers, which are the coefficients of the real functions $\{\Phi_n\}_{n=1}^{\infty}$. Now we can apply the soft margin construction procedure in the feature space. The solution of Equation (16) could be extended by considering the mapping as follows:

$$\hat{f}(\mathbf{x}) = sign\left(\omega^o \cdot \Phi(\mathbf{x}) + b\right) = sign\left(\sum_{i=1}^{l} \alpha_i^o y_i \Phi\left(\mathbf{x}_i\right)\Phi(\mathbf{x}) + b^o\right) \tag{18}$$

The product of $\Phi\left(\mathbf{x}_i\right)\Phi\left(\mathbf{x}_j\right)$ is a scalar quantity. Now it is wise to introduce the so called *kernel function K:*

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{19}$$

Finally we can construct the decision function by following the soft margin Lagrangian function for nonlinear SVM with the additional kernel property as follows:

$$\hat{f}(x) = sign\left(\sum_{i=1}^{l} \alpha_i^o y_i K(\mathbf{x}_i, \mathbf{x}_j) + b^o\right) \tag{20}$$

Therefore, the key property of nonlinear SVM is the kernel function that will be demonstrated in the next section.

KERNEL THEORY

To find an OH in a higher dimensional feature space is a complicated and computationally expensive task. Kernel theory has made it easier for the machine to extract the OH in higher dimensional feature space. The concept of the kernel method is older, but Vapnik reintroduced this method as a significant part of the statistical learning machine by combining it with SVM (Boser, Guyon, & Vapnik, 1992). Apart form SVM, kernel methods have been successfully implemented in kernel Fisher discriminant (KFD) (Baudat & Anouar, 2000), kernel principal component analysis (KPCA) (Schölkopf, Mika, Burges, Knirsch, Müller, Rätsch, et al., 1999), multiple additive regression kernel (MARK) (Bennett, Momma, & Embrechts, 2002), and many others. Like supervised learning methods, kernel functions have also been given similar attention in the unsupervised learning area (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001). Due to wide application and quite good performance, kernel-based algorithms have been renamed *kernel machines* (Schölkopf & Smola, 2000).

A simple kernel function is demonstrated on a binary class problem in Figure 3. The +1 and -1 classes are denoted by cross and rectangular symbols. The middle line in Figure 3 is indicated as an OH. Figure 3 (b) and (c) illustrate the linear and nonlinear functions of the kernel.

By implementing the kernel function, one can map the input data into a high dimensional feature space. The feature space could be of infinite dimension. After kernel transformation, it is straight forward to construct the OH in the feature space by minimising the error. One important advantage of the kernel machine is that it does not need any explicit parameter, for instance, *b*. For any continuous kernel function that satisfies the Mercer's condition of symmetry (Mercer, 1909), which is $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i)$, there exists a Hilbert space *H*, a map $\Phi: \Re^n \to H$, and positive numbers of α_i such as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1}^{\infty} \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{21}$$

where $\mathbf{x}_i, \mathbf{x}_j \in \Re^n$. Mercer's condition require that

$$\int K(\mathbf{x}_i,\mathbf{x}_j)g(\mathbf{x}_i)g(\mathbf{x}_j)d\mathbf{x}_i d\mathbf{x}_j \geq 0 \qquad (22)$$

is satisfied for all g such that

$$\int g^2(\mathbf{x})d\mathbf{x} < \infty . \qquad (23)$$

The integral is considered here over a compact subset of $\Re^n$. Finally the kernel function K can be represented as an inner product as follows:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) . \qquad (24)$$

The linear, polynomial, radial basis function (RBF), and sigmoidal are the most commonly used kernels for SVM. We formulate the SVM classical kernel as follows (Vapnik, 1999, 2000):

The linear kernel function is:

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right) = \left\langle \mathbf{x}_i^T \mathbf{x}_j \right\rangle . \qquad (25)$$

The dth order polynomial kernel function is:

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right) = \left\langle \mathbf{x}_i^T \mathbf{x}_j \right\rangle^d \quad \text{or} \qquad (26)$$

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right) = \left(\left\langle \mathbf{x}_i^T \mathbf{x}_j \right\rangle + 1 \right)^d \qquad (27)$$

Vapnik suggests choosing the second polynomial kernel function, which avoids the problems of the hessian matrix becoming zero (Boser et al., 1992).

Radial basis function has received significant attention in SVM implementation. The RBF kernel function is:

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right) = \exp\{-\left|\mathbf{x}_i - \mathbf{x}_j\right|^2\} \qquad (28)$$

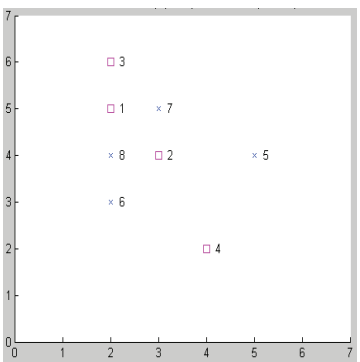
Boser et al. (Navarrete & Ruiz-del-Solar, 2003; Vapnik, 1998; McLachlan, 1992) modified the classical function by introducing a smoothing parameter σ as follows:

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right), \quad \text{where } \sigma > 0 . \qquad (29)$$

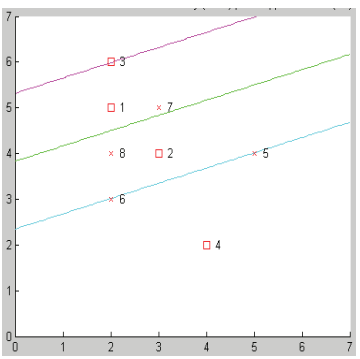
From the beginning of the nonlinear SVM, researchers have used these linear, polynomial, and RBF kernel for classification as well as regression problems. Therefore, these kernels are called SVM classical kernels.

The sigmoidal kernel (Vapnik, 1998) function is:

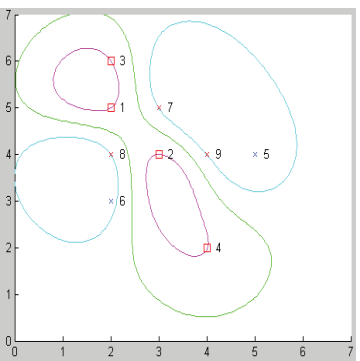
Figure 3. The kernel function: (a) The input space, (b) The linear OH construction with errors, and (c) The nonlinear OH construction without error by using kernel mapping to a 256 dimensional space



(a) Input space



(b) Linear separation



(c) Nonlinear separation

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\tanh(\eta\left(\mathbf{x}_i^T\mathbf{x}_j\right)+\theta) \qquad (30)$$

This kernel requires selection of two parameters, that is, η and θ.

Another two kernels, spline and multiquadratic, are also quite popular for some specific problems. The finite spline kernel (Gunn, 1998) can be described as follows:

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=1+\left(\mathbf{x}_i^T\mathbf{x}_j\right)+\frac{1}{2}\left(\mathbf{x}_i^T\mathbf{x}_j\right)\min\left(\mathbf{x}_i^T\mathbf{x}_j\right)^2-\frac{1}{6}\min\left(\mathbf{x}_i^T\mathbf{x}_j\right)^3 \qquad (31)$$

The multiquadratic positive semi definite kernel is (Evgeniou, Pontil, & Poggio, 2000)

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\left(\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2+\tau^2\right)^{\frac{1}{2}},\ \text{where } \tau>0 \qquad (32)$$

A graphical view of these kernels is shown in Figure 4 for an artificial dataset. The example shows the differences between the kernels and their functional behaviors. The linear kernel constructs the linear boundary for SVM, which is not suitable to classify nonlinearly separable patterns. The polynomial and RBF kernels construct the nonlinear boundary for SVM to classify the patterns. Both kernels showed better performance than the others. The RBF kernel produced zero classification errors. On the other hand spline, sigmoidal, and multiquadratic kernels are constructed with a near linear boundary for SVM to classify the patterns. So the classification error is higher than the RBF and polynomial kernels.

Some other suitable kernels include engineering kernel (Zien, Rätsch, Mika, Schölkopf, Lengauer, & Müller, 2000), ANOVA kernel (Stitson, Gammerman, Vapnik, Vovk, Watkins, & Weston, 1997), wavelet kernel (Strauss, Delb, Plinkert, & Jens, 2003), kernal with moderate decreasing (KMOD) (Ayat, Cheriet, Remaki, & Suen, 2001), semantic kernel (Siolas & d'Alche-Buc, 2000), scaling kernels (Zhang, Zhou, & Jiao, 2002), adaptive kernel (Zhao & Kuh, 2002), and tangent distance kernels (Haasdonk & Keysers, 2002), and have showed better performance based on specific problems than classical SVM kernels.

Like the evolution of SVM from hard margin to soft margin, linear to nonlinear, the SVM method has been also extended effectively from binary class to multiclass classification solver.

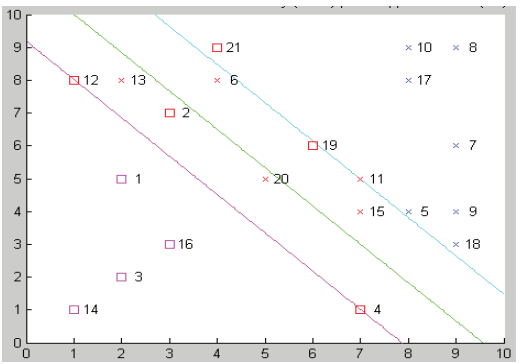
MULTICLASS SVM

SVM was first introduced as a binary classification solver. After that researchers extended it from binary to multiclass SVM in different ways, but still it is a research issue for an effective multiclass SVM. There are two types of multiclass SVM available in the literature. The first directly constructs a multiclass solver by considering all data points in one optimisation formulation (Crammer & Singer, 2000; Vapnik, 1998; Weston & Watkins, 1999) and the second constructs and combines several binary classifiers into a multiclass solver (Chih-Wei & Chih-Jen, 2000). There are three methods that have been developed for the second category of multiclass SVM: *one-against-one* (Kreßel, 1999; Friedman, 1996), o*ne-against-all* (Schölkopf, Burges, & Vapnik, 1995; Vapnik, 2000) and *directed acyclic graph* (DAG) SVM (Platt, Cristianini, & Shawe-Taylor, 2000). Weston and Watkins (1999) argue that direct multiclass SVM performs better than the second category multiclass SVM. Therefore, in this chapter we have summarised the direct method as follows (Weston, 1999):

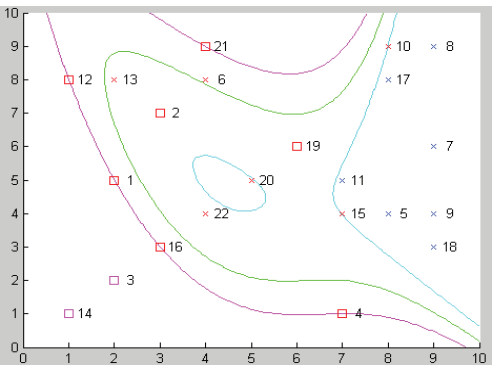
Let us consider the same dataset as described above with extended class values $y_i \in \{1,\cdots,k\}$ and the optimisation problem is:

$$\min_{\omega,\xi}\phi\left(\omega,\xi\right)=\frac{1}{2}\sum_{m=1}^{k}\left(\omega_m\cdot\omega_m\right)+C\sum_{i=1}^{l}\sum_{m\neq y_i}\xi_i^m \qquad (33)$$

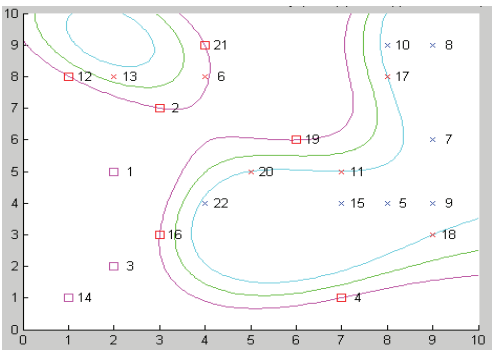
Figure 4. A pictorial view of the linear, polynomial, rbf, spline, sigmoidal and multiquadratic kernel on an artificial dataset. The cross and rectangular sign indicates the two classes of data. The middle lines (except sigmoidal) of the above graphs represent the OH for classification. Those data points placed on the hyperplane are called SVs [Ali, S. 2005]



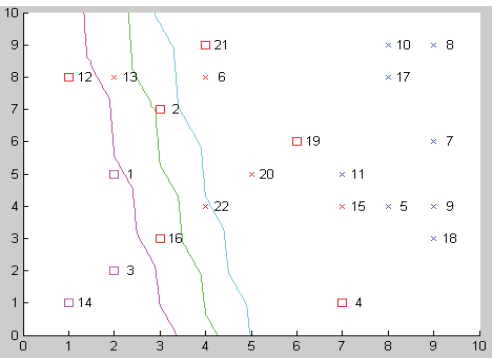
(a) linear kernel – 4 classification errors



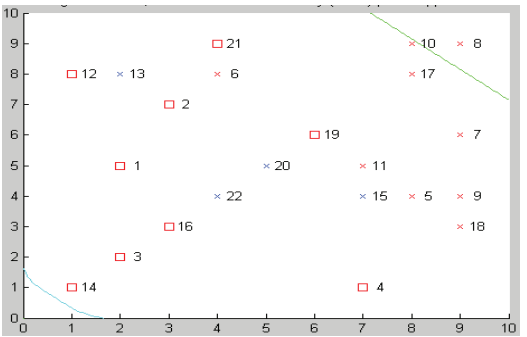
(b) polynomial kernel – 3 classification errors



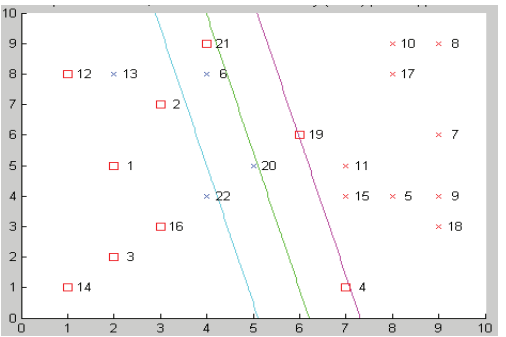
(c) rbf kernel – 0 classification errors



(d) spline kernel – 5 classification error



(e) sigmoidal kernel– 12 classification error



(f) m ultiquadratic kernel – 6 c lassification error

subject to: $\left(\omega_{y_i}\cdot\mathbf{x}_i\right)+b_{y_i}\geq\left(\omega_m\cdot\mathbf{x}_i\right)+b_m+2-\xi_i^m$

$$\xi_i^m\geq 0, i=1,\cdots,l \quad m\in\{1,\cdots,k\}\setminus y_i$$

Now we can solve this optimisation problem by constructing the similar quadratic approach to find the saddle point of the Lagrangian:

$$L(\omega,b,\xi,\alpha,\beta)=\frac{1}{2}\sum_{m=1}^{k}(\omega_m\cdot\omega_m)+C\sum_{i=1}^{\ell}\sum_{m=1}^{k}\xi_i^m-\sum_{i=1}^{\ell}\sum_{m=1}^{k}\alpha_i^m\left[\left(\left(\omega_{y_i}-\omega_m\right)\cdot\mathbf{x}_i\right)+b_{y_i}-b_m-2+\xi_i^m\right]-\sum_{i=1}^{\ell}\sum_{m=1}^{k}\beta_i^m\xi_i^m \tag{34}$$

with the dummy variables:

$$\alpha_i^{y_i}=0,\quad \beta_i^{y_i}=0,\quad \xi_i^{y_i}=0$$

subject to:

$$\alpha_i^m\geq 0,\ \beta_i^m\geq 0,\ \xi_i^m\geq 0, i=1,\cdots,\ell \text{ and } c\in\{1,\cdots,k\}\setminus y_i$$

which is maximised with respect to α and β minimised with respect to ω by ξ considering the notation:

$$c_i^n=\begin{cases}1 & if\ \ y_i=n\\ 0 & if\ \ y_i\neq n\end{cases}\quad\text{and}\quad A_i=\sum_{m=1}^{k}\alpha_i^m \tag{35}$$

After getting the differentiation, the optimal α is obtained as follows:

$$\alpha_i^o=2\sum_{i,m}\alpha_i^m+\sum_{i,j,m}\left[-\frac{1}{2}c_j^{y_i}A_iA_j+\alpha_i^m\alpha_j^{y_i}-\frac{1}{2}\alpha_i^m\alpha_j^{y_i}\right]\left(\mathbf{x}_i\cdot\mathbf{x}_j\right) \tag{36}$$

Finally the decision function for multiclass SVM is:

$$\hat{f}(\mathbf{x})=\arg\max_n\left[\sum_{i:y_i=n}A_i\left(\mathbf{x}_i\cdot\mathbf{x}\right)-\sum_{i:y_i\neq n}\alpha_i^n(\mathbf{x}_i\cdot\mathbf{x})+b_n\right] \tag{37}$$

The inner product ($\mathbf{x}_i\cdot\mathbf{x}$) can be replaced by the convolution inner product $K(\mathbf{x}_i,\mathbf{x}_j)$, also known as the kernel function.

A SIMPLE EXAMPLE FOR SVM CLASSIFICATION

In this section, the simple Boolean exclusive-OR (XOR) problem is solved by SVM second degree polynomial kernel to illustrate the approach. This is a nonlinear classification problem. The following XOR problem (Smith, 1999; Duda, Hart, & Stork, 2001) is described in Table 1.

We choose the polynomial kernel (as described in Equation [27]) with second degree. The solution does not consider any explicit bias. First, we transform the dataset by polynomial kernel as:

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i\cdot\mathbf{x}_j^T)^2$$

Here, $\mathbf{x}_i.\mathbf{x}_j^T=\begin{bmatrix}-1&-1\\-1&1\\1&-1\\1&1\end{bmatrix}\times\begin{bmatrix}-1&-1&1&1\\-1&1&-1&1\end{bmatrix}$,

Now we can construct the kernel matrix as follows:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\begin{bmatrix}9&1&1&1\\1&9&1&1\\1&1&9&1\\1&1&1&9\end{bmatrix}$$

We can write the maximisation term following Equation (10) as:

$$\alpha_i^o = \sum_{i=1}^{4} \alpha_i - \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$(= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}\, 9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\,\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2)$$

subject to

$$\sum_{i=1}^{4} y_i\alpha_i = \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0$$

$$0 \le \alpha_1$$
$$0 \le \alpha_2$$
$$0 \le \alpha_3$$
$$0 \le \alpha_4$$

Differentiation with respect to the Lagrangian parameters $\{\alpha_1,...,\alpha_4\}$, the following sets of simultaneous equations are

$$\begin{cases} 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1 \end{cases}$$

By solving these above equations we can write the solution to this optimisation problem as $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$.

The decision function in the inner product representation is

$$D(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i^* y_i H(\mathbf{x}_i, \mathbf{x}) = (0.125)\sum_{i=1}^{4} y_i \left[(\mathbf{x}_i \cdot \mathbf{x}) + 1\right]^2$$

Table 1. Boolean XOR Problem

Input data **x**	Output class y
(-1,-1)	-1
(-1,+1)	+1
(+1,-1)	+1
(+1,+1)	-1

This decision function separates the data with a maximum margin that will be demonstrated later in Figure 5.

Since the kernel considers the inner product of the input vectors, we can write the second degree polynomial kernel function as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i, \mathbf{x}_j) + 1)^2$$
$$(= x_{i1}x_{j1} + x_{i2}x_{j2})^2 + 2(x_{i1}x_{j1} + x_{i2}x_{j2}) + 1$$
$$= 1 + (x_{i1}x_{j1})^2 + 2(x_{i1}x_{j1})(x_{i2}x_{j2}) + (x_{i2}x_{j2})^2 + 2(x_{i1}x_{j1}) + 2(x_{i2}x_{j2})$$
$$= \Phi(\mathbf{x}_i)^T(\mathbf{x}_j)$$

Now we can write the second degree polynomial transformation function as:

$$\Phi(\mathbf{x}_i) = [1, x_{i1}{}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}{}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]^T$$

The six-dimensional feature space, where the decision function is linear with maximum margin, can be represented in Table 2.

By substituting the value of α and $\Phi(x_i)$ in Equation (9), we can construct the optimum weight vector as:

$$\omega^o = \sum_{i=1}^{4} \alpha_i y_i \Phi(\mathbf{x}_i)$$

$$\frac{1}{8}[-\Phi(\mathbf{x}_1) + \Phi(\mathbf{x}_2) + \Phi(\mathbf{x}_3) - \Phi(\mathbf{x}_4)]$$

$$\frac{1}{8}\left[-\begin{bmatrix}1\\1\\\sqrt{2}\\1\\-\sqrt{2}\\-\sqrt{2}\end{bmatrix} + \begin{bmatrix}1\\1\\-\sqrt{2}\\1\\-\sqrt{2}\\\sqrt{2}\end{bmatrix} + \begin{bmatrix}1\\1\\-\sqrt{2}\\1\\\sqrt{2}\\-\sqrt{2}\end{bmatrix} - \begin{bmatrix}1\\1\\\sqrt{2}\\1\\\sqrt{2}\\\sqrt{2}\end{bmatrix}\right] = \begin{bmatrix}0\\0\\-1/\sqrt{2}\\0\\0\\0\end{bmatrix}$$

Finally the optimal hyperplane can be defined (without any external bias) as $(\omega^o)^T \Phi(\mathbf{x}) = 0$

So,

$$\left[0,0,\frac{-1}{\sqrt{2}},0,0,0\right]\begin{bmatrix}1\\ x_1^2\\ \sqrt{2}x_1x_2\\ x_2^2\\ \sqrt{2}x_1\\ \sqrt{2}x_2\end{bmatrix}=0$$

Therefore the optimal hyperplane function for this XOR problem is:

$$\hat{f}(\mathbf{x}) = -x_1x_2$$

This is the SVM 2nd degree polynomial solution for XOR problem shown in Figure 4. Due to the nonlinear nature of this data, the linear kernel is unable to separate the classes like a polynomial kernel.

In Figure 4, the polynomial, RBF, and sigmoidal kernels are capable of classifying all patterns. But the other kernels fail to classify all patterns correctly.

EXPERIMENTAL RESULTS: CLASSIFIER PERFORMANCE

We consider all the algorithms from Waikato environment for knowledge analysis (WEKA) release 3.1.8 with default parameter settings. We considered in our experiment the 100 classification problems. We choose eight popular classifiers namely IBK, C4.5, partial tree (PART), kernel density (KD), naive Bayes (NB), OneR, SVM, and finally neural network (NN). The machine configuration is Pentium IV, CPU 2.66 GHz and 1 GB RAM. The average accuracy is the combination of true positive rate (TPR), true negative rate (TNR), percentage of correct classification, and weighted F-measure. The computational complexity considers both the model train time as well as the test set evaluation time (Ali, 2005).

From this experiment we observed that the SVM performed better in terms of accuracy. On the other hand the computational performance is average for the SVM. But it is faster than popular classifier NN.

SVM Regression

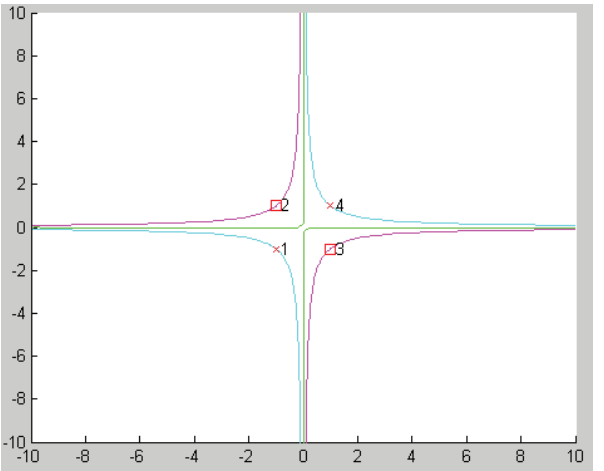
SVM has been introduced for the first time for binary class pattern recognition problems, but the application of SVM to regression problems has shown many breakthroughs and plausible performance. Moreover, applications of support vector regression (SVR) (Gunn, 1998), such as forecasting of financial market (Yang, Chan, & King, 2002), estimation of power consumption (Chen, Chang, & Lin, 2001), reconstruction of chaotic systems (Matterra & Haykin, 1999), and prediction of highway traffic flow (Ding, Zhao, & Jiao, 2002), are also under development. The time-varying properties of SVR applications resemble the time-dependency of traffic forecasting, combined with many successful results of SVR predictions encouraged also in many regression-based modeling.

Let us consider a set of training data $\{(x_1,y_1),....,(x_\ell,y_\ell)\}$, where each $x_i \subset R^n$ de-

Table 2. 2nd degree polynomial kernel feature space

Input space	Feature space						Target
(x_1, x_2)	1	x_1	x_2	x_1x_2	x_1^2	x_2^2	y
(1,1)	1	1	1	1	1	1	1
(1,-1)	1	1	-1	-1	1	1	-1
(-1,-1)	1	-1	-1	1	1	1	1
(-1,1)	1	-1	1	-1	1	1	-1

Figure 4. Pictorial view of the XOR problem classification with SVM. The polynomial kernel function is $\hat{f}(\mathbf{x}) = -(x_1 x_2)$



notes the input space of the sample and has a corresponding target value $y_i \subset R$ for $i = 1,..., l$ where l corresponds to the size of the training data (Vapnik, 2000). The goal for the regression problem is to determine a function that can approximate future values accurately.

The generic SVR estimating function is as follows:

$$f(x) = (w \cdot \Phi(x)) + b \tag{38}$$

where $w \subset R^n$, $b \subset R$, and Φ denotes a nonlinear transformation from R^n to high dimensional space. Our aim is to find the value of w and b such that values of x can be determined by minimising the regression risk:

$$R_{reg}(f) = C \sum_{i=0}^{\ell} \Gamma(f(xi) - yi) + \frac{1}{2}\|w\|^2 \tag{39}$$

where $\Gamma(\cdot)$ is a cost function, C is a constant, and vector w can be written in terms of data points as:

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)\Phi(x_i) \tag{40}$$

Table 3. Formulated ranking averaged across test set classification problems based on a variety of measures (where a rank of 1 means best performing algorithm, and 0 means worst performing algorithm)

Classifier	IBK	C4.5	PART	KD	NB	OneR	SVM	NN
TPR	0.595	0.595	0.595	0.61	0.495	0.365	0.565	**0.645**
TNR	0.54	0.6	0.6	0.6	0.44	0.385	0.595	**0.605**
% of correct classification	0.505	0.615	0.55	0.565	0.385	0.325	**0.62**	0.615
F-measure	0.565	**0.625**	**0.625**	0.575	0.48	0.365	0.56	0.62
Average accuracy	0.551	0.609	0.593	0.588	0.45	0.36	0.585	**0.621**

Table 4. Average ranking of computational performance

Classifier	IBK	C4.5	PART	KD	NB	OneR	SVM	NN
Execution Time	0.535	0.535	0.52	0.51	0.705	**0.995**	0.5	0.015

By substituting Equation (40) into Equation (38), the generic equation can be rewritten as:

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)(\Phi(x_i) \cdot \Phi(x)) + b = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) k(x_i, x) + b \quad (41)$$

In Equation (41) the dot product can be replaced by the kernel function as described early in this chapter.

In SVM literature the quality of estimation is measured by the loss function $L(y, f(\mathbf{x},\omega)$. We use a new type of loss function called ε-insensitive loss function SVM regression proposed by Vapnik (1998, 1999):

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \varepsilon, & for \quad |f(x) - y| \geq \varepsilon \\ 0 & otherwise \end{cases} \quad (42)$$

The empirical risk is:

$$R_{emp}(\omega) = \frac{1}{n} \sum_{i=1}^{n} L_{\varepsilon}(y_i, f(\mathbf{x}_i, \omega)) \quad (43)$$

It is important that ε-insensitive loss coincides with least-modulus loss and with a special case of Huber's robust loss function (Vapnik, 1995, 1999) when ε =0. Now we can compare prediction performance of SVM (with proposed chosen ε) with regression estimates obtained using least-modulus loss(ε =0) for various noise densities.

By solving the quadratic optimisation problem in Equation (44), the regression risk in Equation (39) and the ε -insensitive loss function in Equation (42) can be minimised:

$$\frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(x_i, x_j) - \sum_{i=1}^{\ell} \alpha i^* (y_i - \varepsilon) - \alpha_i (y_i + \varepsilon) \quad (44)$$

subject to

$$\sum_{i=1}^{\ell} \alpha_i - \alpha_i^* = 0, \quad \alpha_i, \alpha_i^* \in [0, C]$$

The Lagrange multipliers, α_i and α_i^*, represent solutions to the above optimisation problem that act as forces pushing predictions towards target value y_i. Among these values only the nonzero values of the Lagrange multipliers in Equation (44) are useful in forecasting the regression line and are known as support vectors. For all points inside the ε-tube as presented in Figure 5 the Lagrange multipliers equal to zero do not contribute to the regression function.

The basic regression line fitting of the SVM method in the training data points are explained in Figure 5.

The constant *C* introduced in Equation (39) always determines penalties to estimation errors. A large *C* assigns higher penalties to errors so that the regression is trained to minimise error with lower generalisation while a small *C* assigns fewer penalties to errors; this allows the minimisation of margin with errors, thus higher generalisation ability. If *C* goes to infinitely large, SVR would not allow the occurrence of any error and result in a complex model. Whereas when *C* goes to zero,

the result would tolerate a large amount of errors and the model would be less complex.

A Simple Example for SVM Regression

This is a simple problem, solving a simple regression task using LS-SVMlab (Suykens, Gestel, Brabanter, Moor, & Vandewalle, 2002). A dataset is constructed in the correct formatting. The data are represented as matrices where each row contains one data point (Box 1).

The first two variables (X and Y) we use to construct a SVM regression model and the next two variables (Xt and Yt) we will use to evaluate the model.

In order to make an LS-SVM model with RBF kernel fitting we need to initialize two extra parameters: γ (`gama`) (described as C in Equation 39) is the regularisation parameter, which determines the trade-off between the fitting error minimisation, and the RBF kernel function smoothness of the estimated function σ^2 (`sigma`^{2}). We initialised gama = 10 and sigma = 0.3 for this above problem. After completing the model generation we found the two parameter values for Equation (41). The optimisation parameter α has been summerised in the above table and the bias factor b = -0.1394.

Now the LS-SVM performance can be displayed if the dimension of the input data is 1 or 2 as described in Figure 6.

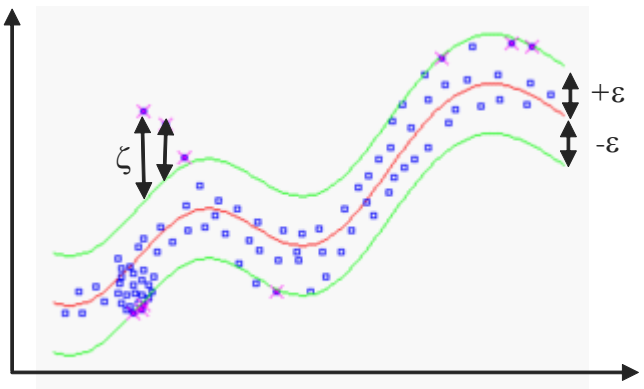
Prediction Performance

We present a comparison for the best prediction performance of SVM, back-propagation neural network (BNN), multiple discriminant analysis (MDA), and logistic regression analysis (logit) in training and holdout data, and show that SVM outperforms BPN, MDA, and Logit by 0.5, 4.8, and 3.9%, respectively, for the holdout data (Min & Lee, 2005) in Table 5.

FUTURE TRENDS

It is well known that SVM generalisation performance (estimation accuracy) depends on a good setting of metaparameters C, ε, and the automated kernel and its parameters selection. Ali and Smith (in press, 2005) already published some solutions about parameter and automatics kernel selection

Figure 5. Support vector regression to fit a tube with radius ε to the data and positive slack variables ζ measuring the points lying outside of the tube



Box 1.

X =	Y =	α	Xt=	Yt
-3.0000	0.0986	0.7225	-3.0000	0.1488
-2.8000	0.0150	-2.5448	-2.9000	-0.0207
-2.6000	0.1492	0.6324	-2.8000	-0.0179
-2.4000	0.1495	1.2348	-2.7000	0.0707
-2.2000	0.0872	1.4470	-2.6000	0.1827
-2.0000	-0.1004	-0.7677	-2.5000	0.0419
-1.8000	-0.1987	-0.4472	-2.4000	0.0060
-1.6000	-0.2266	0.7385	-2.3000	0.1000
-1.4000	-0.3348	-2.2723	-2.2000	0.0785
-1.2000	-0.2615	-2.7375	-2.1000	0.0954
-1.0000	0.1472	3.8079	-2.0000	-0.0595
-0.8000	0.2394	0.3383	-1.9000	-0.0667
-0.6000	0.3828	-2.1291	-1.8000	-0.1474
-0.4000	0.7527	2.1717	-1.7000	-0.1594
-0.2000	0.8227	-0.2337	-1.6000	-0.0357
0	0.8651	-1.1451	-1.5000	-0.2729
0.2000	0.9094	0.7907	-1.4000	-0.3510
0.4000	0.8522	2.8846	-1.3000	-0.1512
0.6000	0.5174	-0.1629	-1.2000	-0.2463
0.8000	0.2995	1.2449	-1.1000	-0.0858
1.0000	-0.1168	-3.0861	-1.0000	-0.0628
1.2000	-0.2020	-0.2473	-0.9000	0.1628
1.4000	-0.2425	0.9768	-0.8000	0.2892
1.6000	-0.3105	-1.4638	-0.7000	0.3475
1.8000	-0.2359	-2.1312	-0.6000	0.2991
2.0000	0.0931	3.3496	-0.5000	0.6499
2.2000	0.0862	0.2270	-0.4000	0.9161
2.4000	0.0616	-1.8873	-0.3000	0.9602
2.6000	0.1970	2.0950	-0.2000	0.7774
2.8000	0.0900	0.7937	-0.1000	0.9758
3.0000	-0.0990	-2.1995	0	0.9318
			0.1000	0.8812
			0.2000	0.8121
			0.3000	0.8873
			0.4000	0.7139
			0.5000	0.6422
			0.6000	0.4678
			0.7000	0.3214
			0.8000	0.2710
			0.9000	0.1821
			1.0000	0.2112
			1.1000	-0.2252
			1.2000	-0.2582
			1.3000	-0.0943
			1.4000	-0.2552
			1.5000	-0.3503
			1.6000	-0.1577
			1.7000	0.0038
			1.8000	-0.0332
			1.9000	0.1440
			2.0000	0.0505
			2.1000	0.2333
			2.2000	0.0511
			2.3000	-0.0020
			2.4000	0.1050
			2.5000	0.2463
			2.6000	0.0048
			2.7000	0.1589
			2.8000	0.0067
			2.9000	0.0890
			3.0000	-0.1100

Figure 6. A simple regression problem is solved by LS-SVM. The solid line indicates the estimated output. The dotted line represents the true underline function. The star indicates the support vectors for the above line fit.

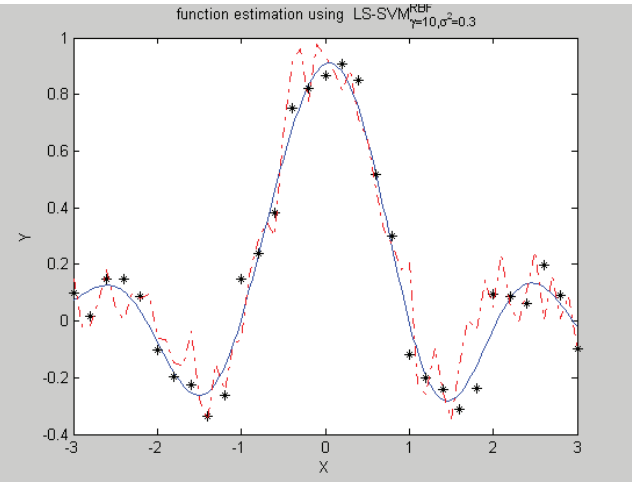


Table 5. The best prediction accuracy of SVM, BPN, MDA, and Logit (hit ratio: %)

	SVM	**BNN**	**MDA**	**Logit**
Training data	88.0132	85.2474	78.8079	79.8676
Holdout data	83.0688	82.5397	79.1391	78.3069

for SVM. They propose data dependent solutions for kernel and its parameter selection.

Each dataset can be described by simple, distance, and distribution-based statistical measures. They explained $X^{i}_{k,j}$ to be the value of the jth variable (column) in the kth example (row) of dataset i. These three types of measures characterise the dataset matrix in different ways. First, the simple classical statistical measures identify the data characteristics based on variable-to-variable comparisons (i.e., comparisons between columns of the dataset). Then, the distance-based measures identify the data characteristics based on sample-to-sample comparisons (i.e., between rows of the dataset). Finally, the density-based measures consider the relationships between single data points and the statistical properties of the entire data matrix to identify the datasets characteristics. The simple statistical measures are calculated within each column, and then averaged over all columns to obtain global measures of the dataset. Likewise, the distance measures are averaged over all pairwise comparisons, and the density-based measures are averaged across the entire matrix.

For each dataset *i,* a total of 29 measures are calculated (11 statistical, 3 distance-based, and 15 density-based). The dataset characteristics matrix is then assembled with the columns comprising the 29 measures, and the rows comprising the 112 datasets. Finally, by combining the dataset characteristics with the experimental results they generated rules with the help of decision tree algorithm C5.0 (Quinlan, 1993) for automatic kernel and its parameter selection.

This is not a state-way solution for automatic kernel and its parameter selection. More research is required for an optimal solution for SVM parameter selection. The new research can bring SVM into a fully more efficient automated system for predictions. SVM has a wide spectrum of applications including search engines, medical diagnosis, bioinformatics and cheminformatics, detecting credit card fraud, stock market analysis, market promotion identification, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing, and robot locomotion. For more information, please visit http://www.clopinet.com/isabelle/Projects/SVM/applist.html. There are a number of complex problems in medical sciences, which are really hard to carry out the acceptable solutions. For instance, the tumor node and metastasis (TNM) classification is an internationally agreed benchmark for assessing cancer severity, treatment options, and prognostic outcomes of patients with cancer. The TNM benchmark method has been used for over 50 years with various revisions made by American Joint Committee on Cancer (AJCC) (Burke, 2004). The TNM classification is derived mainly from two major sources: clinical, includes examination, imaging, endoscopy, biopsy, and surgical exploration, and pathological resection or biopsy of the primary tumor (Gospodarowicz, Miller, Groome, Greene, Logan, & Sobin, 2003). However, Burke (2004) raises concerns on the current ability of predicting survival rate using TNM classification. For instances, the five year disease-specific survival for newly diagnosed cancer patient is predicted to be the same as the mean survival of all those patients who are diagnosed with cancer before five years. Further, the TNM classification system is also unable to provide information regarding a natural history of the cancer progression, for instances, what happens after a certain period of time such spread, recurrence, or metastasis of a primary tumor.

CONCLUSION

The material covered in this chapter is an elaborate explanation of the SVM theory. SVMs have been formulated for supervised classification (both binary and multi-class) and regression. For simplicity, we demonstrated a binary classification scenario with a well known XOR problem. In the regression section we just used synthetic data to demonstrate how SVM regression methodology works. Both demonstrations made it easy to understand SVM. Moreover, the basic explanation of SVM classification included hyperplane construction procedure and the heart of SVM kernel activity. It is easily explained that, when it is possible to linearly separate two classes, an optimum separating hyperplane can be found by minimising the squared norm of the separating hyperplane. The minimisation has been done by QP problem, in which the training data are represented as a matrix of inner products between feature vectors. Once the optimum separating hyperplane is found, the machine opens the support vector points at the same time and the solution is an expansion on these points only. Other data points may be ignored for SVM prediction. There have been many benefits of the SVM method. The solution to the optimisation problem is a global minimum, whereas other machine learning methods, such as neural networks, can often terminate in local minima, therefore there is a chance of modelling the training data inaccurately. The SVM solution is an expansion on a subset of the original training data, resulting in a sparser model and comparatively less computation time required for subsequent classification. Finally, SVM always minimises the expected generalisation error, rather than just the empirical error, on the training data. The kernel method and the empirical risk analysis made SVM more attractive in the different research community. Thus, it can be proven that SVMs should generalise better than many of their counterparts. Some challenges of SVM have been explained with sufficient help text towards the end of this chapter. Therefore this research could be very useful for an efficient machine learning-based system analysis and design.

REFERENCES

Ali, S. (2005). *Automated support vector learning algorithms.* Unpublished doctoral thesis, Monash University, Australia.

Ali, S., & Smith, K. A. (in press). Automatic kernel selection for support vector machine. *Neurocomputing.* Elsevier Science.

Ali, S., & Smith, K. A. (in press). On optimal degree selection for polynomial kernel with support vector machines: Theoretical and empirical investigations. *International Journal of Knowledge-Based and Intelligent Engineering Systems*.

Ali, S., & Smith, K. A. (2005). Kernel width selection for SVM classification: A meta-learning approach. *International Journal of Data Warehousing and Mining*, 78-97. Hershey, PA: Idea Group.

Ayat, N. E., Cheriet, M., Remaki, L., & Suen, C. Y. (2001). KMOD: A new support vector machine kernel with moderate decreasing for pattern recognition. Application to digit image recognition. In *Proceedings of the 6th International Conference on Document Analysis and Recognition* (pp. 1215-1219).

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, *12*(10), 2385-2404.

Bennett, K., Momma, M., & Embrechts, J. (2002). MARK: A boosting algorithm for heterogeneous kernel models. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining,* Canada.

Boser, B. E., Guyon, I., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop of Computational Learning Theory* (Vol. 5, pp. 144-152). Pittsburgh: ACM Press.

Burke, H. (2004). Outcome prediction and the future of the TNM staging system. *Journal of the National Cancer Institute, 96*(19), 1408-1409.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning, 46*(1), 31-159.

Chen, B. J., Chang, M. W., & Lin, C. J. (2001). *Load forecasting using support vector machines: A study on EUNITE Competition 2001. Report for EUNITE competition for Smart Adaptive System.* Retrieved April 3, 2008, from http://www.eunite.org

Chih-Wei, H., & Chih-Jen, L. (2002). A comparison of methods for multiclass support vector machines**,** *IEEE Transactions on Neural Networks*, *13*(2), 415-425.

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

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265-292.

Ding, A., Zhao, X., & Jiao, L. (2003). *Traffic flow time series prediction based on statistics learning theory.* Paper presented at the IEEE 5th International Conference on Intelligent Transportation Systems (pp. 727-730).

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. John Wiley & Sons, Inc.

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

Friedman, J. (1996). *Another approach to polychotomous classification* (Tech. Rep.). Stanford University Department of Statistics, Stanford, CA.

Gospodarowicz, M. K., Miller, D., Groome, P. A., Greene, F. G., Logan, P. A., & Sobin, L. H. (2003). The process of continuous improvement of the TNM classification. *American Cancer Society*, 1-5.

Gunn, S. R. (1998). *Support vector machine for classification and regression* (Tech. Rep.) University of Shouthampton, UK.

Haasdonk, B., & Keysers, D. (2002). Tangent distance kernels for support vector machines. In *IEEE Proceedings of the 16th International Conference on Pattern Recognition* (Vol. 2, pp. 864-868).

Kreßel, U. (1999). Pairwise classification and support vector machines. In B. Schölkopf et al., (Eds.), *Advances in kernel methods-support vector learning* (pp. 255-268). Cambridge: MIT Press.

Matterra, D., & Haykin, S. (1999). Support vector machines for dynamic reconstruction of a chaotic system. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods* (pp. 211-241). MIT Press. ISBN 0-262-19416-3.

McLachlan, G. (1992). *Discriminate analysis and statistical pattern recognition*. New York: John Wiley and Sons, Inc.

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.

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*(4), 603-614.

Navarrete, P., & Ruiz-del-Solar, J. (2003). Kernel-based face recognition by a reformulation of kernel machines. In J. Benitez & F. Hoffmann (Eds.), *Advances in soft computing: Engineering, design and manufacturing* (pp. 183-196). Springer-Verlag.

Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (2000). Large margin DAG's for multiclass classification. *Advances in Neural Information Processing Systems, 12*, 547-553. Cambridge: MIT Press.

Quinlan, R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufman Publishers.

Schölkopf, B., Burges, C., & Vapnik, V. (1995). Extracting support data for a given task. In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining* (pp. 252-257). Menlo Park: AAAI Press.

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K.-R., Rätsch, G., et al. (1999). Input space versus feature space in kernel based methods. *IEEE Transaction on Neural Networks*, *10*, 1000-1017.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*(7), 1443-1472.

Schölkopf, B., & Smola, A. (2000). *Kernel machines*. Retrieved April 3, 2008, from http://www.kernel-machines.org

Siolas, G., & d'Alche-Buc, F. (2000). Support vector machines based on a semantic kernel for text categorization. In *IEEE-INNS-ENNS Proceedings of International Joint Conference on Neural Networks* (Vol. 5, pp. 205-209).

Smith, K. A. (1999). *Introduction to neural networks and data mining for business applications*. Australia: Eruditions Publishing.

Smola, A., & Schölkopf, B. (1998). *A tutorial on support vector regression* (Tech. Rep.). NeuroCOLT2.

Stitson, M., Gammerman, A., Vapnik, V. N., Vovk, V., Watkins, C., & Weston, J. (1997). *Support vector regression with ANOVA decomposition kernels* (Tech. Rep. CSD-97-22). University of London, Royal Holloway.

Strauss, D. J., Delb, W., Plinkert, P. K., & Jens, J. (2003). Hybrid wavelet-kernel based classifiers and novelty detectors in biosignal processing. In *IEEE Proceedings of the 25th Annual International Conference of the Engineering in Medicine and Biology Society* (Vol. 3, pp. 2865- 2868).

Suykens, J. A. K., Gestel, T.V., Brabanter, J. D., Moor, B. D., & Vandewalle, J. (2002). *Least squares support vector machines*. Singapore: World Scientific. ISBN 981-238-151-1.

Vapnik, V. (1995). *The nature of statistical learning theory* (1st ed.). New York: Springer-Verlag.

Vapnik, V. (1998). *Statistical learning theory*. John Wiley and Sons.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transaction on Neural Networks*, *10*(5), 988-999.

Vapnik, V. (2000). *The nature of statistical learning theory* (2nd ed.). New York: Springer-Verlag.

Weston, J. A. E. (1999). *Extensions to the support vector machine*. Unpublished doctoral thesis, University of London, Royal Holloway, England.

Weston, J., & Watkins, C. (1999). Multi-class support vector machines. In M. Verleysen (Ed.), *Proceedings of the 7th European Symposium on Artificial Neural Networks*. Belgium: Bruges.

Yang, H., Chan, L., & King, I. (2002). Support vector machine regression for volatile stock market prediction. In *Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning* (Vol. 2412, pp. 391-396). Springer

Zhang, L., Zhou, W., & Jiao, L. (2004). Wavelet support vector machine. *IEEE Transactions on Systems, Man and Cybernetics*, *34*(1), 34-39.

Zhao, X., & Kuh, A. (2002). Adaptive kernel least square support vector machines applied to recover DS-CDMA signals. In *IEEE Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers* (Vol. 1, pp. 943-947).

Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., & Müller, K.-R. (2000). Engineering support vector machine kernels that recognize translation initiation sites in DNA. *Bioinformatics*, *16*, 799-807.

KEY TERMS

Classification: This is a task of a machine learning algorithm, which labels training data into a finite number of output classes. A model that classifies training examples is sometimes referred to as a classifier. Generally a classifier's performance is measured by its ability to correctly label unseen test cases with *accuracy*. The alternative name of accuracy is called *error rate*.

Data Mining: The applied branch of machine learning that can automatically detect the trends and associations of data, which are always hidden in the data base. This hidden information is useful in making an expert decision.

Kernel: It is the heart of SVM, which is used to transform the data. After kernel transformation, nonlinear data always become a linear shape. Then, it is easy to learn the data for the SVM method.

Neural Network: It is a well established machine learning method. Neural network is a complex nonlinear modeling technique based on a model of a human neuron. It is a useful learning method for classification as well as regression tasks.

Optimisation: Optimisation is a mathematical method that always offers a best, or optimal, solution for a model.

Polynomial Kernel: This is one of the classical kernels adopted in SVM methodology. This kernel follows the polynomial transformation rule during the kernel feature space construction.

RBF Kernel: One of the popular classical kernel in SVM. The RBF kernel nonlinearly maps the example into a higher dimensional space so it can handle the example better when the relation between class labels and attributes is nonlinear.

Regression: As like classification, regression is another popular applied branch of machine learning. The prediction of regression is always a continuous value. A model or algorithm that esti-

mates a continuous value is sometimes referred to as a regressor. Generally a regressor's performance is measured by its ability to predict a value that is near to the actual value, such as with a correlation coefficient.

Supervised Learning: Machine learning techniques used to learn the relationship between independent attributes and a dependent attribute. Most popular learning algorithms fall into the supervised learning category.

SVM: Support vector machine (SVM) is a statistical-based learning algorithm that has been widely used by researchers in various fields including business, text categorisation, pattern recognition, to protein function prediction. Recently researches added a new dimension for SVM in the cancer classification ability to deal with high dimensional data. Moreover, SVM can handle any classification, clustering, regression, and even novelty detection problems.