REVIEW

# Support Vector Regression

Debasish Basak[1], Srimanta Pal[2] and Dipak Chandra Patranabis[3]

[1] Electrical Laboratory,
Central Institute of Mining and Fuel Research,
Barwa Road, Dhanbad-826001, INDIA
E-mail: deba65@yahoo.com

[2] Electronics & Communication Sciences Unit,
Indian Statistical Institute, 203 B.T. Road,
Kolkata- 700108, INDIA
E-mail: srimanta@isical.ac.in

[3] Department of Instrumentation and Electronics Engineering,
Jadavpur University, Salt Lake Campus, Kolkata – 700098, INDIA
Heritage Institute of Technology, Kolkata – 700107, INDIA
E-mail: dcp@iee.jusl.ac.in

(Submitted on July 12, 2007)

***Abstract* –** Instead of minimizing the observed training error, Support Vector Regression (SVR) attempts to minimize the generalization error bound so as to achieve generalized performance. The idea of SVR is based on the computation of a linear regression function in a high dimensional feature space where the input data are mapped via a nonlinear function. SVR has been applied in various fields – time series and financial (noisy and risky) prediction, approximation of complex engineering analyses, convex quadratic programming and choices of loss functions, etc. In this paper, an attempt has been made to review the existing theory, methods, recent developments and scopes of SVR.

**Keywords –** SVR, Ridge regression, Kernel methods, QP

## 1. Introduction

Support Vector Machines (SVM) are learning machines implementing the structural risk minimization inductive principle to obtain good generalization on a limited number of learning patterns. Structural risk minimization (SRM) involves simultaneous attempt to minimize the empirical risk and the VC (Vapnik–Chervonenkis) dimension. The theory has originally been developed by Vapnik and his co-workers on a basis of a separable bipartition problem at the AT & T Bell Laboratories. SVM implements a learning algorithm, useful for recognizing subtle patterns in complex data sets. The algorithm performs discriminative classification learning by example to predict the classifications of previously unseen data.

The VC dimension of a set of functions is the size of the largest data set due to that the set of functions can scatter. Let us consider a set of function $F= \{f(X, W)\}$ that map points from $R^n$ into the set {0, 1} or {-1, 1}. These are called *indicator* functions that map data points into one of two classes. If one considers Q points in $R^n$, each of these can be assigned (called labelling) a class of 0 or 1 randomly. Now, Q points can be labeled in $2^Q$ different ways. For example, for three points in the plane $R^2$, the eight possible labellings are shown in Fig. 1.

For the eight possible labellings the threshold logic neuron (TLN) can correctly separate or classify all eight configurations as shown in Fig 1. This is achieved by carefully placing the hyperplane to have the correct orientation such that all points to be classified as a +1 lie on the positive side of the hyperplane indicated by a small arrow. Now, it can be said that the VC dimension of the set of oriented straight lines in $R^2$ is three.

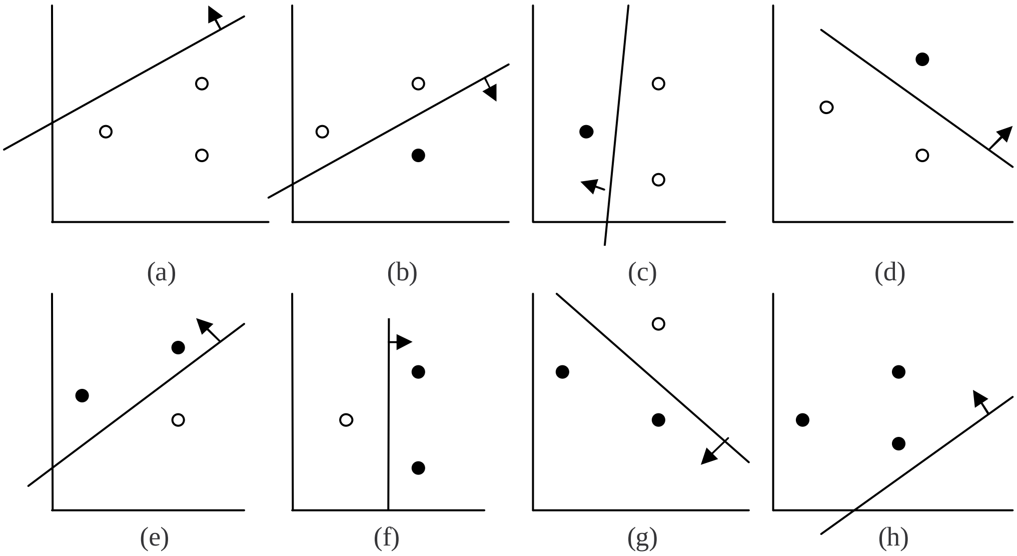

Fig.1: Three points in a plane.

The SV(Support Vector) algorithm is a nonlinear generalization of the generalized Portrait algorithm developed in Russia in the sixties [1, 2]. VC theory has been developed over the last three decades by Vapnik, Chervonenkis and others [3, 4, 5]. This theory characterizes properties of learning machines which enable them to effectively generalize the unseen data. In its present form, the SV machine has been developed at AT & T Bell Laboratories by Vapnik and co-workers [6]. Initial work has focused on OCR (optical character recognition). Within short period, SV classifiers have become competitive with the best available systems for both OCR and object recognition tasks [7]. Burges [8] published a comprehensive tutorial on SV classifiers. Excellent performances have been obtained in regression and time series prediction applications [9].

Statistical Learning Theory has provided a very effective framework for classification and regression tasks involving features. Support Vector Machines (SVM) are directly derived from this framework and they work by solving a constrained quadratic problem where the convex objective function for minimization is given by the combination of a loss function with a regularization term (the norm of the weights). While the regularization term is directly linked, through a theorem, to the VC-dimension of the hypothesis space, and thus fully justified, the loss function is usually (heuristically) chosen on the basis of the task at hand.

Traditional/statistical regression procedures are often stated as the processes deriving a function $f(x)$ that has the least deviation between predicted and experimentally observed responses for all training examples. One of the main characteristics of Support Vector Regression (SVR) is that instead of minimizing the observed training error, SVR attempts to minimize the generalized error bound so as to achieve generalized performance. This generalization error bound is the combination of the training error and a regularization term that controls the complexity of the hypothesis space.

Support vector machine (SVM) has been first introduced by Vapnik. There are two main categories for support vector machines: support vector classification (SVC) and support vector regression (SVR). SVM is a learning system using a high dimensional feature space. It yields prediction functions that are expanded on a subset of support vectors. SVM can generalize complicated gray level structures with only a very few support vectors and thus provides a new mechanism for image compression. A version of a SVM for regression has been proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola [6]. This method is called *support vector regression (SVR).* The model produced by support vector classification only depends on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is close (within a threshold ε) to the model prediction.

Support Vector Regression (SVR) is the most common application form of SVMs. An overview of the basic ideas underlying support vector (SV) machines for regression and function estimation has been given in [10]. Furthermore, it has included a summary of currently used algorithms for training SVMs, covering both the quadratic (or convex) programming part and advanced methods for dealing with large datasets. Finally, some modifications and extensions have been applied to the standard SV algorithm. It has discussed the aspect of regularization and capacity control from a SV point of view. The training data have been taken as $\{(x_1, y_1),...,(x_l, y_l)\} \subset \aleph \times \Re$, where $\aleph$ denotes the space of the input patterns – for instance $\Re^d$. In $\varepsilon - SV$ regression, the goal has been to find a function *f(x)* that has at most $\varepsilon$ deviation from the actually obtained targets $y_i$ for all the training data and at the same time as flat as possible. The case of linear function *f* has been described in the form as

$$f(x) = \langle \omega, x \rangle + b \qquad \text{with} \quad \omega \in \aleph, \quad b \in \Re \tag{1}$$

where $\langle .,. \rangle$ denotes the dot product in $\aleph$. Flatness in (1) means small $\omega$. For this, it is required to minimize the Euclidean norm i.e. $\|\omega\|^2$. Formally this can be written as a convex optimization problem by requiring

$$\text{minimize } \tfrac{1}{2}\|\omega\|^2$$

$$\text{subject to } \begin{cases} y_i - \langle \omega, x_i \rangle - b \le \varepsilon \\ \langle \omega, x_i \rangle + b - y_i \le \varepsilon \end{cases} \tag{2}$$

The above convex optimization problem is feasible in cases where *f* actually exists and approximates all pairs $(x_i, y_i)$ with $\varepsilon$ precision. Sometimes, some errors are allowed. Introducing slack variables $\xi_i, \xi_i^*$ to cope with otherwise infeasible constraints of the optimization problem (2), the formulation becomes

$$\text{Minimize } \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{subject to } \begin{cases} y_i - \langle \omega, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{3}$$

The constant C > 0 determines the trade off between the flatness of *f* and the amount up to which deviations larger than $\varepsilon$ are tolerated. $\varepsilon$ -intensive loss function $|\xi|_\varepsilon$ has been described by

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| < \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases} \tag{4}$$

Fig. 2. depicts the situation graphically.

The dual formulation provides the key for extending SV machine to nonlinear functions. The standard dualization method utilizing Lagrange multipliers has been described as follows:

$$\begin{aligned} L = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i - y_i + \langle \omega, x_i \rangle + b) - \\ \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle \omega, x_i \rangle - b) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \end{aligned} \tag{5}$$

The dual variables in (5) have to satisfy positivity constraints i.e. $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \ge 0$. It follows from saddle point condition that the partial derivatives of L with respect to the primal variables $(\omega, b, \xi_i, \xi_i^*)$ have to vanish for optimality.

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{6}$$

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)x_i = 0 \tag{7}$$

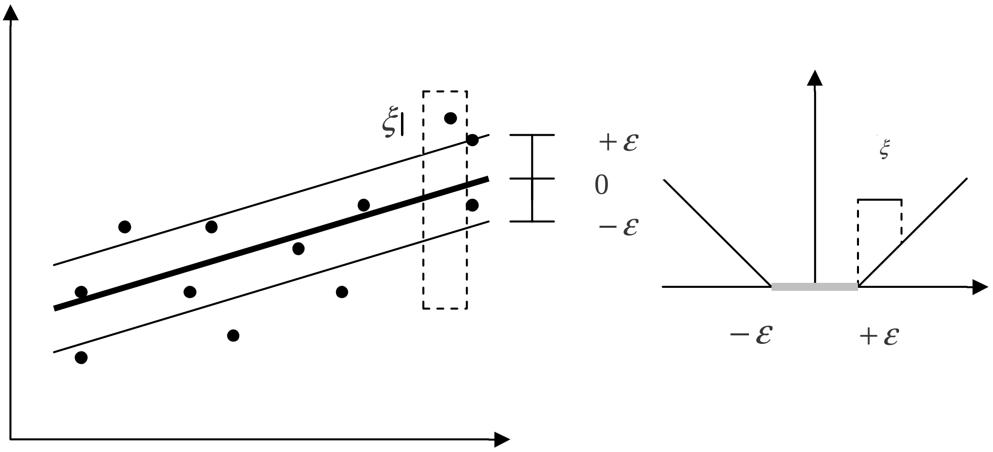

Fig. 2. The soft margin loss setting corresponds to a linear SV machine [11].

$$\frac{\partial L}{\partial \xi_i^{(*)}} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \tag{8}$$

Substituting (6), (7), and (8) into (5) yields the dual optimization problem.

$$\text{Maximize } \left\{ -\frac{1}{2}\sum_{i,j=1}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\left\langle x_i, x_j \right\rangle - \varepsilon \sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \right\} \tag{9}$$

$$\text{Subject to } \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]$$

Dual variables $\eta_i, \eta_i^*$ through condition (8) have been eliminated for deriving (9). Equation (7) can be rewritten as follows:

$$\omega = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i \text{ and therefore, } f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\left\langle x_i, x \right\rangle + b \tag{10}$$

This is the so-called support vector expansion, i.e. $\omega$ can be completely described as a linear combination of the training patterns $x_i$. Even for evaluating *f*(*x*), it is not needed to compute $\omega$ explicitly (although this may be computationally more efficient in the linear setting). Computation of *b* is done by exploiting Karush-Kuhn-Tucker (KKT) conditions [10] which states that at the optimal solution the product between dual variables and constraints has to vanish. In the SV case, this means

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \left\langle \omega, x_i \right\rangle + b) = 0 \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \left\langle \omega, x_i \right\rangle - b) = 0 \end{aligned} \tag{11}$$

and

$$\begin{aligned} (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0 \end{aligned} \tag{12}$$

Following conclusions can be made: (*i*) only samples $(x_i, y_i)$ with corresponding $\alpha_i^* = C$ lie outside the $\varepsilon$ -insensitive tube around *f*, (*ii*) $\alpha_i \alpha_i^* = 0$, i.e. there can never be a set of dual variables $\alpha_i, \alpha_i^*$ which are both simultaneously nonzero as this would require nonzero slacks in both directions. Finally for $\alpha_i^* \in (0, C)$, $\xi_i^* = 0$ and moreover the second factor in (11) has to vanish. Hence b can be computed as follows:

$$\begin{aligned} b = y_i - \left\langle \omega, x_i \right\rangle - \varepsilon \qquad \text{for } \alpha_i \in (0, C) \\ b = y_i - \left\langle \omega, x_i \right\rangle + \varepsilon \qquad \text{for } \alpha_i^* \in (0, C) \end{aligned} \tag{13}$$

From (11), it follows that only for $\left|f(x_i)-y_i\right|\geq\varepsilon$ the Lagrange multipliers may be nonzero, or in other words, for all samples inside the $\varepsilon$-tube, the $\alpha_i,\alpha_i^*$ vanish: for $\left|f(x_i)-y_i\right|<\varepsilon$ the second factor in (11) is nonzero, hence $\alpha_i,\alpha_i^*$ has to be zero such that the KKT conditions are satisfied. Therefore, a sparse expansion of $\omega$ exists in terms of $x_i$ (i.e., all $x_i$ are not needed to describe $\omega$). The examples that come with non-vanishing coefficients are called *Support Vectors*.

SV algorithm can be made nonlinear by simply preprocessing the training patterns $x_i$, by a map $\phi: X\rightarrow\Im$, into some feature space $\Im$ and then applying the standard SV regression algorithm. The expansion in (10) becomes

$$\omega=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\phi(x_i)$$

and therefore,

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(x_i,x)+b \tag{14}$$

The difference with the linear case is that $\omega$ is no longer explicitly given. In the nonlinear setting, the optimization problem corresponds to finding the *flattest* function in *feature* space, not in input space.

The standard SVR to solve the approximation problem is as follows:

$$f(x)=\sum_{i=1}^{N}(\alpha_i^*-\alpha_i)k(x_i,x)+b \tag{15}$$

where $\alpha_i^*$ and $\alpha_i$ are Lagrange multipliers. The kernel function $k(x_i,x)$ has been defined as a linear dot product of the nonlinear mapping, i.e.,

$$k(x_i,x)=\varphi(x_i)\varphi(x) \tag{16}$$

The coefficients $\alpha_i^*$ and $\alpha_i$ of (15) have been obtained by minimizing the following regularized risk functional

$$R_{reg}[f]=\tfrac{1}{2}\|\omega\|^2+C\sum_{i=1}^{l}L_\varepsilon(y) \tag{17}$$

The term $\|\omega\|^2$ has been characterized as model complexity, C as a constant determining the trade-off and the $\varepsilon$-insensitive loss function $L_\varepsilon(y)$ has been given by

$$L_\varepsilon(y)=\begin{cases}0, & \text{for } \left|f(x)-y\right|<\varepsilon\\ \left|f(x)-y\right|-\varepsilon, & \text{otherwise}\end{cases} \tag{18}$$

In classical support vector regression, the proper value for the parameter $\varepsilon$ is difficult to determine beforehand. Fortunately, this problem is partially resolved in a new algorithm, $\nu$ support vector regression ($\nu$-SVR), in which $\varepsilon$ itself is a variable in the optimization process and is controlled by another new parameter $\nu\in$ (0, 1). $\nu$ is the upper bound on the fraction of error points or the lower bound on the fraction of points inside the $\varepsilon$-insensitive tube. Thus a good $\varepsilon$ can be automatically found by choosing $\nu$, which adjusts the accuracy level to the data at hand. This makes $\nu$ a more convenient parameter than the one used in $\varepsilon$-SVR. Scholkopf, et. al. [12] have estimated the function in (1) from the empirical data $(x_1,y_1),\ldots,(x_i,y_i)\in\Re^N\times\Re$ by allowing an error of $\varepsilon$ at each point $x_i$. Everything above $\varepsilon$ is captured in slack variables $\xi_i^{(*)}$ ((*) being a shorthand implying both the variables with and without asterisks), which are penalized in the objective function via regularization constant C, chosen *a priori*. The tube size $\varepsilon$ is traded off against model complexity and slack variables via a constant $\nu\geq 0$ :

$$\text{Minimize}\quad \tau(\omega,\xi^{(*)},\varepsilon)=\frac{1}{2}\|\omega\|^2+C\left(\nu\xi+\frac{1}{l}\sum(\xi_i+\xi_i^*)\right) \tag{19}$$

$$\text{subject to}\quad ((\omega\cdot x_i)+b)-y_i\leq\varepsilon+\xi_i \tag{20}$$

$$y_i-((\omega\cdot x_i)+b)\leq\varepsilon+\xi_i^* \tag{21}$$

$$\xi_i^{(*)}\geq 0,\qquad \varepsilon\geq 0 \tag{22}$$

for $i$ = 1,2…,$l$. Introducing Lagrangian multipliers $\alpha_i^{(*)}, \eta_i^{(*)}, \beta \geq 0$, the Wolfe dual problem is obtained. Moreover, substituting a kernel $k$ for the dot product, corresponds to a dot product in some feature space related to input space via a nonlinear map $\phi$,

$$k(x, y) = (\phi(x) \cdot \phi(y)) \tag{23}$$

This leads to $\nu$ -SVR Optimization Problem**:** for $\nu \geq 0$, C > 0,

$$\text{Maximize} \quad W(\alpha^{(*)}) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) y_i - \frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(x_i, x_j) \tag{24}$$

$$\text{subject to} \quad \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 , \tag{25}$$

$$0 \leq \alpha_i^{(*)} \leq \frac{C}{l} , \tag{26}$$

$$\sum_{i=1}^{l} (\alpha_i + \alpha_i^*) \leq C.\nu , \tag{27}$$

The regression estimate can be shown to take the form

$$f(x) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) k(x_i, x) + b \tag{28}$$

where $b$ (and $\varepsilon$) can be computed by taking into account that (20) and (21) (substitution of $\sum_j (\alpha_i^* - \alpha_i) k(x_i, x)$ for (w.x) is understood) become equalities with $\xi_i^{(*)} = 0$ for points with $0 < \alpha_i^{(*)} < C / l$, following the *Karush-Kuhn-Tucker conditions*. The latter moreover imply that in the kernel expansion equation (28), only those $\alpha_i^{(*)}$ will be nonzero that correspond to a constraint in (20) and (21) which is precisely met. The respective patterns $x_i$ are referred to as *Support Vectors*. If $\nu$ > 1, then $\varepsilon$ = 0, since it does not pay to increase $\varepsilon$. If $\nu \leq$ 1, then $\varepsilon$ = 0, e.g. if the data are noise-free and can perfectly be interpolated with a low capacity model. The case $\varepsilon$ = 0, however, is not of interest; it corresponds to plain $L_1$ loss regression.

Scholkopf has given a brief description of the main ideas of statistical learning theory, support vector machines, and kernel feature spaces [13, 14]. Wahba [15] has suggested nonparametric regression and statistical model building as solutions to optimization problems in Reproducing Kernel Hilbert Spaces. Gaussian and non-Gaussian data, direct and indirect observations, splines and spline anova models and radial basis functions have been discussed as special cases.

The importance of data classification, interpolation, prediction, regression in information technology has been discussed [16]. In recent years, machine learning has become a focal point in Artificial Intelligence. Support vector machines are a relatively new, general formulation for learning machines. SVMs perform exceptionally well on pattern classification, function approximation, and regression problems.

Empirical comparisons between classical statistical methods (Akaike Information Criteria (AIC), Bayesian Information Criteria (BIC)) and the Structural Risk Minimization (SRM) method (based on VC-theory) for regression problems have been carried out [17, 18]. It has been intended to clarify the current state of affairs regarding practical usefulness of SRM model selection. In addition, they have addressed important methodological issues related to empirical comparisons in general and meaningful application of SRM model selection, in particular. In general, analytical estimates of (unknown) prediction risk $R_{est}$ as a function of (known) empirical risk $R_{emp}$ takes one of the following forms:

$$R_{est}(d) = R_{emp}(d).r(d, n) \tag{29}$$

or

$$R_{est}(d) = R_{emp}(d) + r(d / n, \sigma^2) \tag{30}$$

where $r$ ($d$, $n$) is often called the *penalization factor*, which is a monotonically increasing function of the ratio of model complexity (degrees of freedom) $d$ to the number of samples $n$. They have discussed three model selection methods. The first two are representative statistical methods:

*Akaike Information Criteria* (AIC) and *Bayesian Information Criteria* (BIC)

$$AIC(d) = R_{emp}(d) + \frac{2d}{n} \hat{\sigma}^2 \tag{31}$$

$$BIC(d) = R_{emp}(d) + (\ln n)\frac{d}{n}\hat{\sigma}^2 \tag{32}$$

In AIC and BIC, $d$ is the number of free parameters (of a linear estimator) and $\sigma$ denotes the standard deviation of additive noise in standard regression formulation under general setting for predictive learning i.e. y = g (**x**) + $\varepsilon$ where $\varepsilon$ is i.i.d. (independent and identically distributed) zero mean random error (noise), **x** is a multidimensional input and y is a scalar output. Both AIC and BIC are derived using asymptotic analysis (i.e. large sample size). In addition, AIC assumes that correct model belongs to the set of possible models. In practice, however, AIC and BIC are often used when these assumptions do not hold. When using AIC or BIC for practical model selection, one is faced with following two issues: (1) estimation and meaning of (unknown) noise variance, (2) estimation of model complexity. A third model selection method sometimes used in practice is based on Structural Risk Minimization (SRM) which provides a very general and powerful framework for model complexity control. Under SRM, a set of possible models forms a nested structure so that each element (of this structure) represents a set of models of fixed complexity. Hence, a structure provides natural ordering of possible models according to their complexity. Model selection amounts to choosing an optimal element of a structure using VC generalization bounds. For regression problems, the following VC – bound has been used:

$$R(h) \le R_{emp}(h)\left(1 - \sqrt{p - p\ln p + \frac{\ln n}{2n}}\right)^{-1} \tag{33}$$

where $p = h/n$ and $h$ is a measure of model complexity (called VC-dimension). The model complexity has been estimated accurately and the linear estimators first compared and used in crude heuristic estimates of model complexity for k-nearest neighbor regression where accurate estimates of model complexity are not known.

In support vector machines, time complexity appears empirically to locally grow linearly with the number of examples, while generalization performance can be enhanced. Non-linear classification and function approximation is an important topic of interest with continuously growing research areas. Estimation techniques based on regularization and kernel methods play an important role. Support vector machines are a family of data analysis algorithms, based on *convex quadratic programming*. Their use has been demonstrated in classification, regression and clustering problems. Support vector regression (SVR) fits a continuous-valued function to data in a way that shares many of the advantages of support vector machines classification. Most algorithms for SVR [19, 11, 20, 21] require that the training samples be delivered in a single batch.

A toolbox LS-SVMlab for Matlab [22] has been presented first with implementations for a number of LS-SVM related algorithms. Most functions can handle datasets up to 20,000 data points or more. LS-SVMlabs interface for Matlab consists of a basic version for beginners as well as a more advanced version with programs for multi-class encoding techniques and a Bayesian framework. The Matlab toolbox is built around a fast LS-SVM training and simulation algorithm. The corresponding calls can be used for classification as well as function estimation. Additive models [23] have been described based on least square support vector machines (LS-SVM) which are capable of handling higher dimensional data for regression as well as classification tasks. Extensions of LS-SVMs towards robustness, sparseness and weighted versions, as well as different techniques for tuning of hyper-parameters have been included. Advantages of using componentwise LS-SVMs include the efficient estimation of additive models with respect to classical practice, interpretability of the estimated model, opportunities towards structure detection and the connection with existing statistical techniques.

Following an incremental support vector classification algorithm [24], an *accurate on-line support vector regression (AOSVR)* has been developed. AOSVR has efficiently updated a trained SVR function whenever a sample has been added to or removed from the training set. The updated SVR function has been identical to that produced by a batch algorithm.

SVR has been investigated as an alternate technique for approximating complex engineering analyses [25]. The computationally efficient theory behind SVR has been reviewed and SVR approximations have been compared against the different meta-modeling techniques using a testbed of 26 engineering analysis functions. SVR has achieved more accurate and more robust function approximations than the meta-modeling techniques. To overcome the huge time and computational costs of running complex engineering codes, an approximation of the complex analysis code known as "metamodels" has been described. Mathematically, if the inputs to the actual computer analysis are supplied in vector **x**, and the outputs from the analysis in vector **y**, the true computational code evaluates:

$y = f(x)$ where *f*(*x*) is a complex engineering analysis function.

The computationally efficient metamodel approximation is:

$\hat{y} = g(x)$ such that $y = \hat{y} + \varepsilon$ where $\varepsilon$ includes both approximation and random errors.

Lin and Weng [26] have proposed a simple approach for probabilistic prediction suitable for the standard SVR and they have started with generating out-of-sample residuals by *cross validation (CV)*, and then fitted the residuals by simple parametric models like Gaussian and Laplace. Support Vector Machines have emerged as a powerful multivariate modeling technique for classification as well as regression purposes.

Core Vector Machine (CVM) algorithm exploits the "*approximateness*" in the design of SVM implementations. The optimal solution is *approximated* by an iterative strategy. Typically, the stopping criterion utilizes either the precision of the Lagrange multipliers or the duality gap. The CVM algorithm has an asymptotic time complexity that is linear in m, m being the number of training patterns and a space complexity that is independent of m. Applicability of the CVM depends on the following conditions: (1) the kernel k satisfies k(x,x) = a constant; and (2) the QP (quadratic programming) of the kernel method is of a special form. There is no linear term in the QP's objective. The dual objective of SVR contains a linear term and is not of the required form. An enhancement of the CVM allowing a more general QP formulation lifts the condition on the kernel [27]. The resultant Core Vector Regression (CVR) algorithm can be used with any linear /nonlinear kernels and can obtain approximately optimal solutions. The resultant CVR procedure inherits the simplicity of CVM, and has small asymptotic time and space complexities. Experimentally, it is as accurate as existing SVR implementations, but is much faster and produces far fewer support vectors (and thus faster testing) on large data sets. This extension can also be used for scaling up other kernel methods, such as ranking SVM, SVMs in imbalanced learning problems, and SVMs with interdependent and structured outputs.

## 2. Gaussian Process Regression and Variance of Noise

A Gaussian process (GP) is specified by a mean and a covariance function. The mean is a function of **x** (which is often the zero function), and the covariance is a function C(**x**,**x**) which expresses the expected covariance between the value of the function *y* at the points **x** and **x**. The actual function *y*(**x**) in any data modeling problem is assumed to be a single sample from this Gaussian distribution. The equivalent kernel is a way of understanding how Gaussian process regression works for large sample sizes based on a continuum limit.

The use of Gaussian process (GP) has been investigated priors over functions, which permit the predictive Bayesian analysis for fixed values of hyper-parameters to be carried out exactly using matrix operations. Two methods using optimization and averaging (via Hybrid Monte Carlo) over hyper-parameters, have been tested on a number of challenging problems and have produced excellent results [28, 29]. This GP have been introduced for classification and regression. The background facts on the connection between Neural Networks and GPs and given a Bayesian probabilistic interpretation, explaining the use of hyper-parameters and implementation issues have been introduced [30].

Gaussian processes provide natural non-parametric prior distributions over regression functions. Regression problems where there is noise on the output have been considered [31] and the variance of the noise depends on the inputs. It has been assumed that the noise is a smooth function of the inputs and it is natural to model the noise variance using a second Gaussian process, in addition to the Gaussian process governing the noise-free output value and showed that prior uncertainty about the parameters controlling both processes can be handled and that the posterior distribution of the noise rate can be sampled from using Markov chain Monte Carlo methods. Their results on a synthetic data set have given a posterior noise variance that well-approximated the true variance.

Starting from Bayesian linear regression, Williams [32] showed how by a change of viewpoint one can see this method as a Gaussian process predictor based on priors over functions, rather than on priors over parameters. This has lead to a more general discussion of Gaussian processes and further issues, including hierarchical modelling and the setting of the parameters that control the Gaussian process, the covariance functions for neural network models and the use of Gaussian processes in classification problems. The Bayesian analysis of neural networks is difficult because the prior over functions has a complex form, leading to implementations that either make approximations or use Monte Carlo integration techniques.

In Gaussian process regression, the covariance between the outputs at input locations **x** and **x'** is usually assumed to depend on the distance $(\mathbf{x} - \mathbf{x}')^T \mathbf{W}(\mathbf{x} - \mathbf{x}')$ where **W** is a positive definite matrix. **W** is often taken to be diagonal [33], but if **W** is allowed to be a general positive definite matrix which can be tuned on the basis of training data, then an eigen-analysis of **W** shows that they are effectively creating hidden features, where the dimensionality of the hidden-feature space is determined by the data. They have demonstrated the superiority of predictions using the general matrix over those based on a diagonal matrix on two test problems.

*Support Vector Clustering (SVC)* methodology has been proposed in [34] with promising performance for high-dimensional and noisy data sets, and for clusters with arbitrary shape. Instead of searching for a single optimal configuration, they have involved generation, selection and combination of distinct clustering solutions that lead to a consensus clustering.

A method has been presented for the sparse greedy approximation of Bayesian Gaussian process regression, featuring a novel heuristic for very fast forward selection motivated by active learning [35]. It has been shown how a large number of hyper-parameters can be adjusted automatically by maximizing the marginal likelihood of the training data. The method is essentially as fast as an equivalent one which selects the "support" patterns at random, yet has the potential to outperform random selection on hard curve fitting tasks and at the very least leads to a more stable behaviour of first-level inference which makes the subsequent gradient-based optimization of hyper-parameters much easier. In line with the development of the method, they have presented a simple view on sparse approximations for GP models and their underlying assumptions and shown relations to other methods.

Support vector machine (SVM) regression problems have been solved as the maximum *a posteriori* prediction in the Bayesian framework and described an approximation technique that is useful in performing calculations for SVMs based on the mean field algorithm as proposed in Statistical Physics of disordered systems [36, 37]. Based on the mean field equation for a Gaussian process, an efficient iterative implementation algorithm has been derived. The mean field SVR method is moderately easy to implement and use.

The practical selection of hyper-parameters for support vector machines regression (that is, $\varepsilon$-insensitive zone and regularization parameter C) have been investigated [38]. The values of $\varepsilon$ and C parameters are obtained directly from the training data and (estimated) noise level. This approach is based on well-known theoretical understanding of SVM regression that provides the basic analytical form for parameter selection. Empirical tuning of these analytical dependencies has been performed using synthetic data sets. Extensive empirical comparisons have suggested that their selection yields good generalization performance of SVM estimates under different noise levels, types of noise, target functions and sample sizes. This approach can be applied in various application domains of SVM. The empirical results suggest that with the choice of $\varepsilon$, the value of regularization parameter C has only negligible effect on the generalization performance (as long as C is larger than a certain threshold determined analytically from the training data). The proposed value of C-parameter is derived for RBF kernels; however, the same approach can be applied to other kernels bounded in the input domain.

It has been also shown (*i*) how to approximate the equivalent kernel of the widely-used squared exponential (or Gaussian) kernel and related kernels, and (*ii*) how analysis using the equivalent kernel helps to understand the learning curves for Gaussian processes [39].

## 3. Ridge Regression and Kernel Methods

It is useful to first read the ridge regression. In kernel ridge regression (KRR), the final solution is not sparse in the variables of $\alpha$. A regression method which is sparse has been formulated with the concept of support vectors that determine the solution. The sparseness has come from complimentary slackness conditions which in turn have come from inequality constraints [40].

Platt [41] has proposed a new algorithm for training support vector machines: *Sequential Minimal Optimization (SMO)* which is a simple algorithm that can quickly solve the SVM QP (Support Vector Machine Quadratic Programming) problems without any extra matrix storage and without using numerical QP optimization steps at all. SMO decomposes the overall QP problem into QP sub-problems using Osuna's theorem to ensure convergence.

In the regression case, the loss function used penalizes errors which are greater than a threshold $\varepsilon$. Such a loss function typically has lead to a sparse representation of the decision rule giving significant algorithmic and representational advantages. If, however, $\varepsilon = 0$ in the case of optimizing the 2-norm of the margin slack vector, the regressor output has been recovered by a Gaussian process with corresponding covariance function, or equivalently the ridge regression function. These approaches have the disadvantage that since $\varepsilon = 0$, the sparseness of the representation has been lost [42].

Ridge regression is a classical statistical technique that attempts to address the bias-variance trade-off in the design of linear regression models. A reformulation of ridge regression in dual variables permits a non-linear form of ridge regression via the well-known "*kernel trick*". Unfortunately, unlike support vector regression models, the resulting kernel expansion is typically fully dense. Ridge regression is a well-known technique from classical multiple linear regression that implements a regularized form of least-squares regression. Given, training data

$$D = \{x_i, y_i\}_{i=1}^{l}, \qquad x_i \in X \subset R^d, \qquad y_i \in Y \subset R,$$

the ridge regression algorithm determines the parameter vector, $w \in R^d$, and bias**,** $b \in R$, of a linear model, *f*(x) = **w.x** + *b*, via minimization of the following objective function:

$$L(w,b) = \frac{1}{2}\|w\|^2 + \frac{\gamma}{l}\sum_{i=1}^{l}(y_i - w.x_i - b)^2$$

This objective function used in ridge regression that implements a form of Tikhonov regularization of a SSE metric where $\gamma$ is a regularization parameter controlling the bias-variance trade-off. This corresponds to penalized maximum likelihood estimation of **w** and b, assuming the targets have been corrupted by an independent and identically distributed (i.i.d.) sample from a Gaussian noise process, with zero mean and fixed variance $\sigma^2$ i.e.

$$y_i = w.x_i + b + \varepsilon_i \qquad \varepsilon \in \mathrm{N}(0,\sigma^2).$$

A non-linear form of ridge regression, known as KRR can be obtained via the so-called “kernel trick”, whereby a linear regression model is constructed in a high dimensional feature space, $\mathrm{F}(\phi : \mathrm{X} \to \mathrm{F})$, induced by a non-linear kernel function defining the inner product $K(x,x') = \phi(x).\phi(x')$. The kernel function, $K : \aleph \text{ x } \aleph \to \Re$ may be any positive definite “Mercer” kernel. The objective function minimized in constructing a kernel ridge regression model is given by

$$L(w,b) = \frac{1}{2}\|w\|^2 + \frac{\gamma}{l}\sum_{i=1}^{l}(y_i - w.\phi(x_i) - b)^2$$

The solution of an optimization problem of this nature can be written in the form of a linear combination of the training patterns, i.e. $w = \sum_{i=1}^{l}\alpha_i K(x_i,x) + b$ . The output of the least squares support vector machine is then given by the kernel expansion

$$f(x) = \sum_{i=1}^{l}\alpha_i K(x_i,x) + b$$

A *reduced rank kernel ridge regression* (*RRKRR*) algorithm capable of generating an optimally sparse kernel expansion that is functionally identical to that resulting from conventional *kernel ridge regression (KRR)* has been introduced [43]. This method has been demonstrated to out-perform an alternative sparse kernel ridge regression algorithm on the Motorcycle and Boston Housing benchmarks [44].

A regularized kernel regression model has been introduced for data characterized by a heteroscedastic (input dependent variance) Gaussian noise process [45]. This model has provided more robust estimates of the conditional mean than standard models and also confidence intervals (error bars) on predictions.

It has been extended a form of *kernel ridge regression* for data characterised by a heteroscedastic (i.e. input dependent variance) Gaussian noise process [45, 46]. It has been shown that the proposed heteroscedastic kernel ridge regression model can give a more accurate estimate of the conditional mean of the target distribution than conventional KRR and also provide an indication of the spread of the target distribution (i.e. predictive error bars). The leave-one-out cross-validation estimate of the conditional mean has been used in fitting the model of the conditional variance in order to overcome the inherent bias in maximum likelihood estimates of the variance. The benefits of the proposed model have been demonstrated on synthetic and real-world benchmark data sets and for the task of predicting episodes of poor air quality in an urban environment.

A new and efficient algorithm for the sparse logistic regression problem has been developed [47]. This algorithm has been simple and based on the Gauss-Seidel method and asymptotically convergent and can be applied to a variety of real-world problems like identifying marker genes and building a classifier in the context of cancer diagnosis using microarray data.

Agarwal [48] has given a statistical interpretation of proximal support vector machines (PSVM) as linear approximates to (nonlinear) support vector machines and proved that PSVM using a linear kernel has been identical to ridge regression, a biased-regression method known in the statistical community for more than thirty years. Techniques from the statistical literature to estimate the tuning constant that appears in the SVM and PSVM framework have been discussed. Better shrinkage strategies that incorporate more than one tuning constant have been suggested. For nonlinear kernels, the minimization problem posed in the PSVM framework is equivalent to finding the posterior mode of a Bayesian model defined through a Gaussian process on the predictor space. Apart from providing new insights, these interpretations have helped to attach an estimate of uncertainty to their predictions and enable them to build richer classes of models. In particular, he has proposed a new algorithm called *PSVMMIX* which is a combination of ridge regression and a Gaussian process model.

A novel algorithm for sparse online greedy kernel-based nonlinear regression has implemented a form of gradient ascent and demonstrated its scaling and noise tolerance properties on three benchmark regression problems [49].

A new iterative algorithm for kernel logistic regression has been based on the solution of the dual problem using ideas similar to those of the SMO algorithm for Support Vector Machines. The algorithmic ideas can also be used to give a fast dual algorithm for solving the optimization problem arising in the inner loop of Gaussian Process classifiers. The well-known SMO algorithm of support vector machines has been then extended to Least Squares SVM formulations which include LS-SVM classification, kernel ridge regression and a particular form of regularized kernel fisher discriminant. Computational experiments have shown that the algorithm is fast and asymptotically convergent [50, 51].

A novel method for selecting descriptor subsets by means of Support Vector Machines in classification and regression - the *Incremental Regularized Risk Minimization (IRRM)* algorithm [52] has been presented. In contrast to many other wrapper methods, it is fully deterministic and computationally efficient. They have compared their method to existing algorithms and presented results on a *Human Intestinal Absorption (HIA)* classification data set [53] and Huuskonen regression data set for aqueous solubility [54, 55]

A mechanism to train SVMs with a hybrid kernel and minimal VC dimension has been presented [56]. After describing the VC dimension of sets of separating hyper-planes in a high-dimensional feature space produced by a mapping related to kernels from the input space, they have proposed an optimization criterion to design SVMs by minimizing the upper bound of the VC dimension. This method realizes a structural risk minimization and utilizes a flexible kernel function such that a superior generalization over test data can be obtained. In order to obtain a flexible kernel function, they have developed a hybrid kernel function and a sufficient condition to be an admissible Mercer kernel based on common Mercer kernels (polynomial, radial basis function, two-layer neural network, etc.). The nonnegative combination coefficients and parameters of the hybrid kernel are determined subject to the minimal upper bound of the VC dimension of the learning machine. The use of the hybrid kernel results in a better performance than those with a single common kernel.

In generalized Gaussian kernel regression models [57], each kernel regressor in the pool of candidate regressors has an individual diagonal covariance matrix. This matrix is determined by maximizing the absolute value of the correlation between the regressor and the training data using a repeated weighted search optimization. The standard orthogonal least squares algorithm is then applied to select a parsimonious model from the full regression matrix. Compared with the existing kernel regression modeling approaches which adopt a single common kernel variance for all regressors, their method has the advantages of improving modeling capability and producing sparser models. It has been proved that SVM regression result is effective and available for simulation. Cortical control of virtual cursor has been investigated by means of SVM where the training inputs of the regression estimation are firing rates of neuronal ensembles in motor and premotor cortex, and the outputs are trajectories of virtual cursors.

Statistical properties have been investigated for a broad class of modern *kernel based regression (KBR)* methods [58]. These kernel methods have been developed during the last decade and are inspired by convex risk minimization in infinite dimensional Hilbert spaces. One leading example is support vector regression. They have first described the relation between the loss function used in the KBR method and the tail of the response variable and then established the risk consistency for KBR which gives the mathematical justification for the statement that these methods are able to 'learn'. Then they have considered robustness properties of such kernel methods. In particular, the results have allowed choosing the loss function and the kernel to obtain computational tractable and consistent KBR methods having bounded influence functions. Furthermore, bounds for the sensitivity curve which is a finite sample version of the influence function have been developed and the relationship between KBR and classical M-estimators [59] has been discussed.

Kernel methods have gained a growing interest during the last few years for designing QSAR models [60] having a high predictive strength. One of the key concepts of SVMs is the usage of a so-called kernel function, which can be thought of as a special similarity measure. Fröhlich and Zell [61] have considered kernels for molecular structures, which are based on a graph representation of chemical compounds. The similarity score has been calculated by computing an optimal assignment of the atoms from one molecule to those of another one, including information on specific chemical properties, membership to a substructure (e.g. aromatic ring, carbonyl group, etc.) and neighbourhood for each atom. By using this kernel, they have achieved a generalization performance comparable to a classical model with a few descriptors, which are *a priori* known to be relevant for the problem and significantly better results than with and without performing an automatic descriptor selection. They have further explored the effect of combining kernel with a problem dependent descriptor set and also demonstrated the usefulness of an extension of their method to a reduced graph representation of molecules, in which certain structural features, like rings, donors or acceptors, have been represented as a single node in the molecular graph.

Support vector machines have become one of the most popular methods in Machine Learning during the last few years. A special strength is the use of a kernel function to introduce nonlinearity and to deal with arbitrarily structured data. Usually the kernel function depends on certain parameters, which, together with other parameters of the SVM, have to be tuned to achieve good results. However, finding good parameters can become a real computational burden as the number of parameters and the size of the dataset increases. Fröhlich and Zell [61] have proposed an algorithm to deal with the model selection problem, which is based on the idea of learning an *Online Gaussian Process* model of the error surface in parameter space and sampling systematically at points for which the so called *expected improvement* is highest. Their experiments have found good parameters very efficiently.

Recursive neural networks and kernel methods have been compared for structured data specifically support vector regression machine using a tree kernel, in the context of regression tasks for trees [62].

## 4. Convex Quadratic Programming and Loss Function

In regression problems, choice of different loss functions arises from various assumptions about the distribution of the noise in measurement. Several non-quadratic loss functions have been introduced to reduce the sensitivity of the outliers, such as the *Laplacian loss function* and the *Huber's loss function*.

The $\varepsilon$ -insensitive loss function (ILF) has been proposed by Vapnik [5] in Support Vector Machines (SVM) for regression (SVR). The performance of SVR crucially depends on the shape of the kernel function and other hyper-parameters that represent the characteristics of the noise distributions in the training data. Bayesian methods have been regarded as suitable tools to determine these hyper-parameters. However, these methods have been difficult to use SVR due to lack of smoothness of ILF.

A new smoothing strategy has been proposed for solving epsilon-SVR tolerating a small error in fitting a given data set linearly or nonlinearly [63]. Conventionally, this has been formulated as a constrained minimization problem, namely, a convex quadratic programming problem. The smoothing techniques that have been used for solving the support vector machine for classification has been applied to replace the epsilon-insensitive loss function by an accurate smooth approximation and allowed to solve epsilon-SVR as an unconstrained minimization problem directly and termed reformulated problem as epsilon- support vector regression. Newton-Armijo algorithm [64] has been shown to be convergent globally and quadratically to solve their problem has been prescribed. In order to handle the case of nonlinear regression with a massive data set, they also have introduced the reduced kernel technique to avoid the computational difficulties in dealing with a huge and fully dense kernel matrix.

The Bayesian evidence framework can be applied to both ε-SVR and ν-SVR algorithms [65]. Standard SVR training has been regarded as performing level one inference of the evidence framework, while levels two and three have allowed automatic adjustments of the regularization and kernel parameters respectively, without the need of a validation set.

Support Vector Machines for regression problems are trained by solving a quadratic optimization problem which needs memory and time resources of the order of $t^2$, where $t$ is the number of training examples. A decomposition algorithm, *SVMTorch*, has been proposed for classification problems, but adapted to regression problems [66]. This algorithm efficiently solves large-scale regression problems.

A randomized subset selection scheme has been suggested for SVR with respect to the so-called $\varepsilon$ -insensitive loss usually employed for SVR [67]. From the data points and from the value $\varepsilon$ received for the loss function, one constructs two new datasets, one by shifting the response up by $\varepsilon$ , and the other by shifting it down by $\varepsilon$ . For a large enough $\varepsilon$ for which hard tubes exist, these shifts amount to move the first dataset to stand fully above the regression hyperplane, and the second dataset to stand fully below it. Thus, the regression hyperplane has been effectively transformed into a classification hyperplane. Algorithms for training support vector machines in classification problems have been extended into algorithms for support vector regression problems under $\varepsilon$ -insensitive loss.

The dual formulation of support vector regression involves two closely related sets of variables. When the decomposition method is used, many existing approaches use pairs of indices from these two sets as the working set. Basically, they select a base set first and then expand it so that all indices are pairs. This makes the implementation different from that for support vector classification. In addition, a larger optimization sub-problem has to be solved in each iteration. Theoretical proofs and conducted experiments have been provided to show that using the base set as the working set leads to similar convergence (number of iterations). Therefore, by using a smaller working set while keeping a similar number of iterations, the program can be simpler and more efficient [68].

Practical selection of meta-parameters have been investigated for SVM regression (that is, $\varepsilon$-insensitive zone and regularization parameter *C*) advocating analytic parameter selection directly from the training data, rather than resampling approaches commonly used in SVM applications. Good generalization performance of the proposed parameter selection [17] has been demonstrated empirically using several low-dimensional and high-dimensional regression problems. Further, they have pointed out the importance of Vapnik's $\varepsilon$-insensitive loss function for regression problems with finite samples and compared generalization performance of SVM regression (with optimally chosen $\varepsilon$) with regression using 'least-modulus' loss ($\varepsilon$ = 0). These comparisons have indicated superior generalization performance of SVM regression for finite sample settings. They have also demonstrated the importance of $\varepsilon$- insensitive loss function on the generalization performance.

A methodology has been described for performing variable ranking and selection using support vector machines [69]. The method has constructed a series of sparse linear SVMs to generate linear models that can generalize well, and used a subset of nonzero weighted variables found by the linear models to produce a final nonlinear model. The method has exploited the fact that a linear SVM (no kernels) with *1*-norm regularization inherently performs variable selection as a side-effect of minimizing capacity of the SVM model. The distribution of the linear model weights has provided a mechanism for ranking and interpreting the effects of variables. They have illustrated the effectiveness of the methodology on synthetic data, benchmark problems, and challenging regression problems in drug design. This method can dramatically reduce the number of variables and outperforms SVMs using all attributes and using the attributes selected according to correlation coefficients. The visualization of the resulting models has been useful for understanding the role of underlying variables.

A unified framework for popular non-quadratic regression loss functions has been proposed in [70]. It has been done using a novel loss function known *soft insensitive loss function (SILF)* which possesses most of the virtues in popular non-quadratic loss functions such as insensitivity to the outliers, differentiability and sparseness in sample selection, etc. in addition, the introduction of SILF in regression makes it possible to apply Bayesian techniques on support vector methods, while preserving their individual advantages. Standard Gaussian processes for regression to set up the Bayesian framework, in which the unified loss function is used in the likelihood evaluation have been followed. The overall approach has the merits of support vector regression such as convex quadratic programming and sparsity in solution representation. Experimental results on simulated and real-world data sets have indicated that the approach has worked well even on large data sets. Moreover, confidence interval has been provided in prediction.

Training a SVR requires the solution of a very large QP (quadratic programming) optimization problem. Despite the fact that this type of problem is well understood, the existing training algorithms are very complex and slow. Linear and nonlinear regression models have been evaluated combining the *possibility and necessity estimation formulation* with the principle of SVM [71]. For data sets with crisp inputs and interval outputs, the possibility and necessity models have been recently utilized, which are based on quadratic programming approach giving more diverse spread coefficients than a linear programming one.

A heuristic method based on the similarity measurement, for reducing training data to accelerate support vector machines training, has been investigated [72]. The method (HSVM training) is fast enough without discounting the generalization performance. The extra computation cost for reducing training data is rather feasible since the total time spent on HSVM training is less than that done on SVM training with the complete training data. The algorithm transforms a QP problem of large size into one of small size without complex computations like other conventional methods. The presented method HSVM is more effective for dealing with dense training sets. The notion of similarity, as a measurement of the approaching degree between two samples, is introduced to select new training data. The similarity function S is defined as

$$S(x, y) = f\left(1/\|x-y\|_2\right) = f\left(1/\sqrt{\sum_{i=1}^{M}(x_i - y_i)^2}\right)$$

where *x*, *y* denote samples and M is the dimension of a sample. For the sake of simplification, let *f*(*t*) = *t*. It is clear that the larger the S, the closer are the two samples from the viewpoint of geometry. To obtain good generalization performance, the reduction process may not be carried out on the whole data instead on several groups. The reduction result depends on the selection of similarity threshold and the division of data groups. Too large similarity threshold will result in less data being reduced and the advantage, i.e., fast training in SVM, being concealed. On the other hand, too small similarity threshold will lead more data to be removed, and therefore the generalization error may increase greatly. Hence, the similarity threshold is actually a trade-off between training speed and predicting accuracy.

An approximate formulation of the SVM regression algorithm is desirable to avoid raising the quadratic programming problem of the classical formulation. Farag and Mohamed [73] have suggested an advanced approach utilizing some principles of the mean field theory to cope with the Gaussian classification problem. The basic idea of the mean field theory is to approximate the statistics of a random variable which is correlated to other random variables by assuming that the influence of the other variables can be compressed into a single effective mean “field” with a rather simple distribution. Mean field (MF) methods replace the intractable task of computing high dimensional sums and integrals by much easier problem of solving a system of linear equations. The regression problem is formulated so that the MF method can be used to approximate the learning procedure in a way that avoids the quadratic programming optimization. This approach is suitable for high dimensional regression problems.

## 5. Time Series Analysis and Financial Prediction

In the Bayesian framework, predictions for a regression problem are expressed in terms of a distribution of output values. The mode of this distribution corresponds to the most probable output, while the uncertainty associated with the predictions can conveniently be expressed in terms of error bars. The evaluation of error bars have been considered in the context of the class of generalized linear regression models and provided insights into the dependence of the error bars on the location of the data points. It derived an upper bound on the true error bars in terms of the contributions from individual data points which are themselves easily evaluated [74].

Support vector regression has been introduced to solve regression and prediction problems. It has become a hot topic of intensive study due to its successful application in classification tasks and regression tasks, especially on time series prediction and financial related applications. SVMs are state-of-the-art tools for non-linear input-output knowledge discovery. The support vector regressor (SVR) is for regression and function approximation. The combination of clustering and a state-of-the-art technique has been proposed for knowledge gain and accuracy improvement in a complex pharmacokinetic prediction problem [75]. The power and versatility of the SVR machines have allowed fast and reliable prediction schemes.

Use of an adaptive margin in SVR has also been suggested for financial prediction to minimize the downside risk, which is an essential part in financial prediction with volatile financial data [76]. More specifically, the two approaches are: one uses the *fixed and asymmetrical margins (FAAM)*, whereas the other uses *non-fixed and symmetrical margins (NASM)*. A key difference between FAAM and FASM is that there exist an up and a down margin that are asymmetrical. In the case of FAAM when the up margin is greater than the down margin, the predictive results tend to be smaller than the predictive results which are produced by using FASM. A general type of insensitive loss function in SVR and outline of the various margins, i.e., FASM, FAAM and NASM have been presented.

A novel algorithm based on SVM regression has been employed to obtain predictive *Quantitative Structure-Retention relationship (QSRR)* models using two-step computational strategy: (1) a sparse linear SVM is utilized as a variable selection method to identify relevant molecular descriptors, and (2) a set of nonlinear SVM models derived by kernel mapping using the selected features [77]. In addition, a statistical technique called *“bagging” (Bootstrap aggregation)* has been employed to improve model generalization performance.

The external administration of recombinant human erythropoietin is the chosen treatment for those patients with secondary anemia due to chronic renal failure in periodic hemodialysis. An individualized prediction of the EPO dosage has been carried out to be administered to those patients [78]. The high cost of this medication, its side-effects and the phenomenon of potential resistance which some individuals suffer --- all have justified the need for a model which is capable of optimizing dosage individualization. A group of 110 patients and several patient factors have been used to develop the models. The support vector regressor (SVR) has been benchmarked with the classical *multilayer perceptron* (*MLP*) and the *Autoregressive Conditional Heteroskedasticity* (*ARCH*) model. They have introduced a priori knowledge by relaxing or tightening the epsilon-insensitive region and the penalization parameter depending on the time period of the patients' follow-up. The so-called *profile-dependent SVR* (*PD-SVR*) has improved results of the standard SVR method and the MLP. They have performed sensitivity analysis on the MLP and inspected the distribution of the support vectors in the input and feature spaces in order to gain knowledge about the problem.

Experimental results have been conducted by studying the regression problem itself and the resultant predicted choice of $\varepsilon$ has been much closer to the experimentally observed optimal value, while again demonstrating a linear trend with the input noise [79].

Support vector regression (SVR) is a powerful technique for predictive data analysis. The goal of the regression problem is to approximate a function. The solution to a support vector regression problem is a function that accepts a data point and returns a continuous value. The support vector regression problem also allows for a '*zone of insensitivity*' defined typically by a parameter $\varepsilon$.

When using SVM in regression tasks, the support vector regressor must use a cost function to measure the empirical risk in order to minimize the regression error. For financial data, due to the embedded noise, one must set a suitable margin in order to obtain a good prediction. The financial data are usually highly noisy and contain outliers. A novel two-phase SVR training algorithm has been proposed to detect outliers and reduce their negative impact [80].

The use of Support Vector Machines (SVMs) has been presented for prediction and analysis of *antisense oligonucleotide* (*AO*) efficacy [81]. The collected database has comprised 315 AO molecules including 68 features, each inducing a problem well-suited to SVMs. The task of feature selection is crucial given the presence of noisy or redundant features, and the well-known problem of the *curse of dimensionality*. They have proposed a two-stage strategy to develop an optimal model: (1) feature selection using correlation analysis, mutual information, and *SVM-based recursive feature elimination (SVM-RFE),* and (2) AO prediction using standard and profiled SVM formulations. A profiled SVM has given different weights to different parts of the training data to focus the training on the most important regions. The SVM approach has been well suited to the AO prediction problem, and yielded a prediction accuracy superior to previous methods. The profiled SVM has been found to perform better than the standard SVM, suggesting that it could lead to improvements in other prediction problems as well.

Travel time is a fundamental measure in transportation. Accurate travel-time prediction also is crucial to the development of intelligent transportation systems and advanced traveler information systems. Support vector regression has been applied for travel-time prediction and compared its results to other baseline travel-time prediction methods using real highway traffic data [82]. Since support vector machines have greater generalization ability and guarantee global minima for given training data, it is believed that SVR would perform well for time series analysis. Compared to other baseline predictors, their results have shown that the SVR predictor could significantly reduce both relative mean errors and root-mean-squared errors of predicted travel times. They have also demonstrated the feasibility of applying SVR in travel-time prediction and proved that SVR is applicable to and performs well in traffic data analysis.

Support vector machines have been developed by Vapnik as an effective algorithm for determining an optimal hyperplane to separate two classes of patterns. Comparative studies have shown that support vector classification (SVC) usually has given better predictions than other classification methods. In a short period of time, SVC has found significant applications in bioinformatics and computational biology, such as cancer diagnosis, prediction of protein fold, secondary structure, protein-protein interactions, and sub-cellular localization. Using various loss functions, the support vector method has been extended for regression (SVR). SVR can have significant applications in *QSAR* (*quantitative structure-activity relationships*) if it is able to predict better than other well-established QSAR models. QSAR models obtained with multiple linear regression (MLR) and SVR has been compared for the benzodiazepine receptor affinity using a set of 52 pyrazolo[4,3-*c*]quinolin-3-ones [83].

## 6. Usefulness of SVR

The problem of the determination of developmental age of an embryo from its segmentation gene expression patterns in Drosophila has been addressed [84]. By applying support vector regression, they have developed a fast method for automated staging of an embryo on the basis of its gene expression pattern. Support vector regression is a statistical method for creating regression functions of arbitrary type from a set of training data. The training set has been composed of embryos for which the precise developmental age has been determined by measuring the degree of membrane invagination. Testing the quality of regression on the training set has shown good prediction accuracy. The optimal regression function has been used for the prediction of the gene expression based age of embryos in which the precise age has not been measured by membrane morphology. Moreover, they have shown that the same accuracy of prediction can be achieved when the dimensionality of the feature vector is reduced by applying factor analysis. The data reduction has allowed them to avoid over-fitting and to increase the efficiency of the algorithm.

Support vector regression has been applied to the geolocation problem with the addition of Kalman-Bucy filtering to smooth location estimates in a mobile tracking scenario [85]. Mobile positioning using radiolocation techniques usually involves time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), signal strength (SS) measurements or some combination of these methods. They have described two-step location estimation algorithm to estimate the location of a mobile user. The first step employs an approach in

SVR to obtain an initial estimate of the mobile location. Measurements have been taken at known locations *a priori* and a training database has been created.

Functional magnetic resonance imaging (fMRI) is a noninvasive technique for mapping brain function by the *blood oxygenation level dependent* (*BOLD*) effect. An approach has been presented utilizing spatio-temporal support vector regression (SVR), within which the intrinsic spatio-temporal auto correlations in fMRI data have been reflected [86]. The following major advantages of the approach have been found: avoidance of interpolation after motion estimation, embedded removal of low-frequency noise components, and easy incorporation of multi-run, multi-subject, and multi-task studies into the framework.

Blind identification of *single-input multiple-output* (*SIMO*) channels is a common problem encountered in communications, sonar and seismic signal processing. SIMO channels appear either when the signal is oversampled at the receiver or from the use of an array of antennas. Blind identification has been formulated as a support vector regression problem and an iterative procedure has also been suggested to solve it avoiding trivial solution [87]. According to the SRM principle the proposed cost function establishes a trade-off between a loss function and the complexity of the solution: this regularized functional leads to a robust solution to either channel order overestimation or badly conditioned channels.

An application of support vector machine has been demonstrated to the oceanic disasters search and rescue operation [88]. The support vector regression for system identification of a nonlinear black-box model has been utilized in this research. The SVR-based ocean model has helped the search and rescue unit by predicting the disastrous target's position at any given time instant. The closer the predicted location to the actual location would shorten the searching time and minimize the loss. One of the most popular ocean models, namely the Princeton ocean model, has been applied to provide the ground truth of the target leeway. From the experiments [88], the results on the simulated data showed that the SVR-based ocean model has provided a good prediction compared to the Princeton ocean model. Moreover, the experimental results on the real data collected by the Royal Thai Navy also has shown that this model can be used as an auxiliary tool in the search and rescue operation.

A performance analysis has been reported between Multivariate Adaptive Regression Splines (MARS), neural networks and support vector machines. The MARS procedure has built flexible regression models by fitting separate splines in to distinct intervals of the predictor variables [89].

The ultrasonic motor (USM) has excellent performance and many useful features which electromagnetic type motors do not have. A characteristic of the USM that is affected by friction is strong nonlinearity, which makes it difficult to control. A position control method has been proposed for the USM using SVR, which is a regression method for SVM [80]. It has been a newly proposed method of machine learning that does not have the disadvantages of a Neural Network such as a large number of learning times, local minima, overfitting and so on. They have used an SVR controller combined with a PI controller. The SVR controller performs nonlinear input-output mapping of the USM. The learning of the SVR controller has used training data obtained from experiments. The effectiveness of this control method has been confirmed by experiments [90].

The identification of discriminative genes for categorical phenotypes in microarray gene expression data analysis has been extensively studied, especially for disease diagnosis. In recent biological experiments, continuous phenotypes have also been dealt with. For example, the extent of programmed cell death (apoptosis) can be measured by the level of caspase 3 enzyme. Thus, an effective gene selection method for continuous phenotypes is desirable. Two new support vector formulations have been proposed for ordinal regression, which optimize multiple thresholds to define parallel discriminant hyperplanes for the ordinal scales. Both approaches have guaranteed that the thresholds are properly ordered at the optimal solution [91].

## 7. Conclusion

It has been experimentally determined the values of $\varepsilon$ that lead to the lowest generalization error [92]. A good agreement with the values that has previously been predicted by a theoretical argument based on the asymptotic efficiency of a simplified model of SV regression has been found for various noise models and SV parameter settings. The existence of differential gloss would impair perceived image quality. As a result, it is desirable to measure the amount of differential gloss existing on a printed image. There are two approaches to quantify the differential gloss: direct instrument measurement and indirect estimation based on *a priori* model. A support vector regression technique based on 1-norm penalty function has been adopted to identify (density/gloss) patches that are important under various printing processes and paper types [93]. As a result, these patches can be printed and measured such that a mapping function f can be established, which, in turn, is used to quantify the differential gloss in a printed image.

The algorithm of SVR has been utilized to perform the compression of gray images [94]. Blocks of image have been represented by the values of the Lagrange multipliers of support vectors using the linear function, the polynomial function and the Gaussian radial basis function as the kernel functions. If the parameters of the learning machine are carefully adjusted, most of the multipliers will be zero. Since only very few of the parameters have been saved, the purpose of compression could be achieved.

Active Set Support Vector Regression (ASVR) algorithm consists of partitioning the dual variables ά and $\alpha$ into non-basic and basic variables. The non-basic variables are those which are set to zero. The values of the basic variables are determined by finding the gradient of the objective function with respect to these variables, setting this gradient equal to zero and solving the resulting linear equations for the basic variables. If the basic variable takes on a negative value after solving the linear equations, it is set to zero and becomes non-basic. This is the essence of the algorithm. A new active set strategy has been presented to solve a straightforward formulation of the standard support vector regression problem [95]. They have demonstrated that active set algorithm can be used effectively for the support vector regression problem. ASVR is fast, finite, simple and capable of performing support vector regression on datasets with millions of points. ASVR requires nothing more complex than a commonly available linear equation solver for solving small systems with few variables even for massive datasets.

The disappointing performance of value and small cap strategies shows that style consistency may not provide the long-term benefits often assumed in the literature. It has been examined whether the short-term variation in the U.S. size and value premium is predictable and documented with style-timing strategies based on technical and macro-economic predictors using a recently developed artificial intelligence tool called support vector regressions [96]. SVR are known for their ability to tackle the standard problem of over-fitting, especially in multivariate settings. Their findings have indicated that both premiums to be predictable under fair levels of transaction costs and various forecasting horizons.

A support vector regression filtering approach has been introduced for reduction of *False Positive signals (FPR)* in an automated mass detection system [97]. The proposed approach consists of two steps: the first one belongs to the family of neural filters but has been based on SVM, a class of learning algorithm which has been proved in recent years to be superior to the conventional Neural Network method for both classification and regression tasks, hence its application to neural-like image processing looks very appealing. The second step is a simple way to take into account information given by SVR filter, in order to decide whether the analysed signal is a False Positive (FP) or not. $\nu$ -support vector regression ($\nu$ -SVR) has been considered for regression estimation. The SVR algorithm has acted as a filter because it has been capable to associate to each input crop an output image, which has been subsequently used to determine if the crop has contained or not a mass.

Minimizing bounds of leave-one-out errors is an important and efficient approach for support vector machine model selection. Past research has focused on their use for classification but not regression. Chang and Lin [98] have derived various leave-one-out bounds for support vector regression and discussed the difference from those for classification. Experiments have demonstrated that the proposed bounds are competitive with Bayesian SVR for parameter selection. They have also discussed the differentiability of leave-one-out bounds.

Most speech processing systems perform a frame-by-frame analysis of the speech data. Support vector regression problem has been formulated to estimate the *Target-to-Interferer Ratio* (*TIR*) by observing the usual speech features [99]. A fusion method based on support vector regression has been proposed. The basic idea is to determine a mapping function which maps the points in the feature space constructed from the usable speech measures into the TIR space. The TIR is an unobservable metric and hence the regression model would be determined in controlled environment, when the TIR values can be measured.

The connection between higher order *total variation* (*TV*) regularization and SVR with spline kernels in a one-dimensional discrete setting has been studied and proved that the contact problem arising in the tube formulation of the TV minimization problem is equivalent to the SVR problem [100].

Many more aspects such as more data dependent generalization bounds, efficient training algorithms, automatic kernel selection procedures, vector valued regression and many techniques that already have made their way into the standard neural networks toolkit, will have to be considered in the future [21].

## References

[1] V. Vapnik and A. Lerner, (1963), "Pattern recognition using generalized portrait method", *Automation and Remote Control*, 24.

[2] V. Vapnik and A. Chervonenkis, (1964), "A note on one class of perceptrons", *Automation and Remote Control*, 25.

[3] V. Vapnik and A. Chervonenkis, (1974), “Theory of Pattern Recognition” (in Russian), Nauka, 1974.

[4] V. Vapnik, (1982), “Estimation of dependences based on empirical data”, Springer Verlag.

[5] V. Vapnik, (1995), “The Nature of Statistical Learning Theory”, Springer Verlag.

[6] V. Vapnik, S. Golowich and A. Smola, (1997), “Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing”, in M. Mozer, M. Jordan, and T. Petsche (eds.), *Neural Information Processing Systems*, Vol. 9. MIT Press, Cambridge, MA.

[7] B. Schölkopf, P.L. Bartlett, A. Smola, and R. Williamson, (1998), “Support vector regression with automatic accuracy control”, In L. Niklasson, M. Boden, and T. Ziemke, editors, *Perspectives in Neural Computing: Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN'98),* pages 111-116. Springer-Verlag.

[8] C.J.C. Burges, (1998), “A tutorial on support vector machines for pattern recognition”, *Knowledge Discovery and Data Mining*, 2(2).

[9] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik, (1997), “Support vector regression machines” In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 155-161, Cambridge, MA, MIT Press.

[10] A.J. Smola, and B. Schölkopf, (1998), “A Tutorial on Support Vector Regression”, NeuroCOLT, *Technical Report NC-TR-98-030*, Royal Holloway College, University of London, UK.

[11] A. Smola and B. Scholkopf, (1997), “On a Kernel-based Method for Pattern Recognition, Regression, Approximation and Operator Inversion”, *Algorithmica* 22(1064), pp. 211 – 231.

[12] B. Schölkopf, P.L. Bartlett, A. Smola and R. Williamson, (1999), “Shrinking the tube: a new support vector regression algorithm”, In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 330 -- 336, Cambridge, MA. MIT Press.

[13] B. Schölkopf, (2000), “A short tutorial on kernels”, Tutorial given at the *NIPS'00 Kernel Workshop*.

[14] B. Schölkopf, (2000), “Statistical learning and kernel methods”, MSR-TR 2000-23, *Microsoft Research*.

[15] G. Wahba, (2000), “An introduction to model building with reproducing kernel Hilbert spaces”, TR 1020, Statistical Department University of Wisconsin-Madison. *Computing Science and Statistics*, vol. 33. Proceedings of the Interface 2000, E. Wegmann, ed.

[16] M. Palaniswami, (2002), “Machine learning using support vector machines”, *IEEE Signal Processing Society* - North Melbourne.

[17] V. Cherkassky, and Y. Ma, (2002), “Comparison of Model Selection for Regression”, Under Review in *Neural Computation*, [cherkass, myq]@ece.umn.edu.

[18] V. Cherkassky, and Y. Ma, (2002), “Practical Selection of SVM Parameters and Noise Estimation for SVM Regression”, submitted to *Neurocomputing, special issue on SVM 2002.*

[19] C.C. Chang and C.J. Lin (2002), “Training $\nu$ -Support Vector Regression: Theory and Algorithms”, Neural *Computation*, Volume 14, Number 8, 1 August, pp. 1959-1977(19). The MIT Press.

[20] A. Smola, B. Schölkopf and K.R. Müller, (1998), “General cost functions for support vector regression”, In T. Downs, M. Frean, and M. Gallagher, editors, *Proc. of the Ninth Australian Conf. on Neural Networks*, pages 79 - 83, Brisbane, Australia, 1998, University of Queensland. A slightly shorter version was published in: L. Niklasson and M. Boden and T. Ziemke (eds.), *Proceedings of the 8th International Conference on Artificial Neural Networks,* Springer Verlag, Perspectives in Neural Computing.

[21] A.J. Smola and B. Schölkopf, (1998), “On a Kernel-Based Method for Pattern Recognition, Regression, Approximation, and Operator Inversion”, *Algorithmica 22*(1/2): 211-231 (1998).

[22] K. Pelckmans, J.A.K. Suykens, T.V. Gestel, J.D. Brabanter, B. Hamers, D. Moor, and J. Vandewalle, (2002), “LS-SVMlab: a MATLAB/C toolbox for Least Squares Support Vector Machines”, http://www.esat.kuleuven.ac.be/sista/lssvmlab, presented at *Neural Information Processing Systems (NIPS 2002).*

[23] K. Pelckmans, I. Goethals, J.D. Brabanter, J.A.K. Suykens, and B.D. Moor, (2005), “Componentwise Least Squares Support Vector Machines”, arXiv:cs.LG/0504086 v1 19 Apr 2005.

[24] J. Ma, J. Theiler, and S. Perkins, (2003), “Accurate On-line Support Vector Regression”, *Neural Comput.,* November 1; 15(11): 2683 - 2703.

[25] S.M. Clarke, J.H. Griebsch, and T.W. Simpson, (2003), “Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses”, ASME Design Engineering Technical Conferences – Design Automation Conference, Shimada, K. ed. (September 2-6, 2003, Chicago, IL, paper no. DETC

2003/DAC- 48759.

[26] C.J. Lin and R.C. Weng, "Simple Probabilistic Predictions for Support Vector Regression", (cjlin@csie.ntu.edu.tw), http://csie.ntu.tw/~cjlin/papers/svrprob.pdf.

[27] I.W. Tsang, J.T. Kwok and K.T. Lai, (2005), "Core Vector regression for Very Large Regression Problems", Appearing in *Proceedings of the 22nd International Conference on machine Learning*, Bonn, Germany.

[28] C. K. I. Williams, (1995), "Regression with Gaussian Processes**",** Paper presented at the Mathematics of Neural Networks and Applications conference, Oxford, UK, July 1995. In *Mathematics of Neural Networks: Models, Algorithms and Applications*. Eds. S.W. Ellacott, J.C. Mason and I.J. Anderson, Kluwer, 1997.

[29] C.K.I. Williams and C.E. Rasmussen, (1996), "Gaussian Processes for Regression", In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 514-520. MIT Press.

[30] D.J.C. MacKay, (1997), "Introduction to gaussian processes", Extended version of a tutorial at *ICANN'97*.

[31] P.W. Goldberg, C.K.I. Williams, and C.M. Bishop, (1998), "Regression with Input-dependent Noise: A Gaussian Process Treatment**",** *Advances in Neural Information Processing Systems 10*. Editor: M. I. Jordan and M. J. Kearns and S. A. Solla. MIT Press.

[32] C.K.I. Williams, (1998)," Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond**",** *"Learning and Inference in Graphical Models"*, ed. M. I. Jordan, Kluwer, 1998.

[33] F. Vivarelli and C.K.I. Williams, (1999), "Discovering hidden features with Gaussian processes regression**",** *Advances in Neural Information Processing Systems 11*, eds. M. J. Kearns, S. A. Solla and D. A. Cohn. MIT Press, 1999.

[34] C.H. Wu, C.C. Wei, D.C. Su, M.H. Chang and J.M. Ho, (2003), "Travel Time Prediction with Support Vector Regression," in the *Proceedings of IEEE Intelligent Transportation Systems Conference*, October 2003.

[35] M. Seeger, (2004), "Gaussian processes for machine learning". *International Journal of Neural Systems*, 14(2):1-38.

[36] J.B. Gao, S.R. Gunn, and C.J. Harris, (2003), "Mean Field Method for the Support Vector Machine Regression", *Neurocomputing*, Volume 50, pages 391-405.

[37] J.B. Gao, S.R. Gunn, and C.J. Harris, (2003), "SVM Regression through Variational Methods and its Sequential Implementation", *Neurocomputing*, (Accepted for Publication).

[38] V. Cherkassky, and Y. Ma, (2004), "Practical selection of SVM parameters and noise estimation for SVM regression", *Neural Networks*, 17, 113-126.

[39] P. Sollich and C.K.I. Williams, (2005), "Using the Equivalent Kernel to Understand Gaussian Process Regression**",** *Advances in Neural Information Processing Systems 17*, MIT Press.

[40] M. Welling, "Support Vector Regression", (www.ics.uci.edu/~welling/classnotes /papers_class/svregression.pdf).

[41] J.C. Platt, (1998), "Fast training of support vector machines using sequential minimal optimization", In Schölkopf, B., Burges, C.J.C. and Smola, A.J. (eds), *Advances in Kernel Methods* **-** *Support Vector Learning*, pp. 185-208, MIT Press, Cambridge, MA.

[42] N. Cristianini, and J. Shawe-Taylor, (2000), "An Introduction to Support Vector Machines and other kernel-based learning methods", *Cambridge University Press*.

[43] G.C. Cawley, and N.L.C. Talbot, (2002), "Reduced rank kernel ridge regression" *Neural Processing Letters*, 16(3):293-302, December.

[44] (http:// libstat.cmu.edu/datasets/boston).

[45] R.J. Foxall, G.C. Cawley, N.L.C. Talbot, S.R. Dorling, and D.P. Mandic, (2002), "Heteroscedastic regularised kernel regression for prediction of episodes of poor air quality." In *Proceedings of the European Symposium on Artifical Neural Networks (ESANN-2002)*, pages 19-24, Bruges, Belgium, April 24-26.

[46] G.C. Cawley, N.L.C. Talbot, R.J. Foxall, S.R. Dorling, and D.P. Mandic, (2004), "Heteroscedastic kernel ridge regression", *Neurocomputing*, 57:105-124, March.

[47] S.K. Shevade and S.S. Keerthi, (2002), "A Simple and Efficient Algorithm for Gene Selection using Sparse Logistic Regression", Technical Report No. CD-02-22, Control Division, Department of

Mechanical Engineering, National University of Singapore – 117576, submitted for publication in *Bioinformatics* journal.

[48] D. K. Agarwal, (2002), “Shrinkage estimator generalizations of proximal support vector machines”, In Proc. 8th Int. Conf. Knowledge Discovery and Data Mining, Edmonton, Canada, 2002. http://citeseer.ist.psu.edu/agarwal02shrinkage.html.

[49] Y. Engel, S. Mannor, and R. Meir, (2002), “Sparse Online Greedy Support Vector Regression”, yaki@alice.nc.huji.ac.i1, {shie@tx,rmeir@ee}.technion.ac.il, *ECML 2002.*

[50] S.S. Keerthi, K. Duan, S.K. Shevade, and A.N. Poo, (2002), “A Fast Dual Algorithm for Kernel Logistic Regression”, submitted for publication in *Machine Learning,* July 4.

[51] S.S. Keerthi, and S.K. Shevade, (2003), “SMO Algorithm for Least Squares SVM Formulations”, *Neural Computation,* Vol. 15, Feb., pp 487-507.

[52] H. Fröhlich, J. Wegner and A. Zell, (2004), “Towards Optimal Descriptor Subset Selection with Support Vector Machines in Classification and Regression”, *QSAR & Combinatorial Science*, 23:311 - 318, 2004.

[53] http://www.chemsilico.com /CS_prHIAexp.html

[54] J. Huuskonen, (2000), “Aqueous solubility for a diverse set of organic compounds based on molecular topology”*, J.Chem. Inf. Comput. Sci*., 40 (2000), 773-777.

[55] J. Huuskonen, M. Salo, and J. Taskinen, (1998), “Aqueous Solubility Prediction of Drugs Based on Molecular Topology and Neural Network Modeling”, *J.Chem. Inf. Comput. Sci.*, 1998, 38, 450-456.

[56] Y. Tan and J. Wang, (2004), “A Support Vector Machine with a Hybrid Kernel and Minimal Vapnik-Chervonenkis Dimension”, *IEEE Transactions on Knowledge and Data Engineering,* April (Vol. 16, No. 4) pp. 385-395.

[57] X.X. Wang, S. Chen and D.J. Brown, (2004), “An approach for constructing parsimonious generalized Gaussian kernel regression models”, *Neurocomputing,* 62, 441-457.

[58] A. Christmann, and I. Steinwart, (2005), “Consistency and Robustness of Kernel Based Regression”, Los Alamos National Laboratory *Technical Report LA-UR-04-8797*, submitted for publication, 2005.

[59] Newey-MC-Fadden, *Handbook on Econometrics*, chapter 36, pp 2115.

[60] http://netsci.org/science/compchem/feature19.html

[61] H. Fröhlich and A. Zell, (2005), “Efficient Parameter Selection for Support Vector Machines in Classification and Regression via Model-Based Global Optimization”, *Proc. Int. Joint Conf. on Neural Networks (IJCNN*), pp. 1431 - 1438, 2005.

[62] A. Micheli, F. Portera and A. Sperduti, (2005), “A preliminary empirical comparison of recursive neural networks and tree kernel methods on regression tasks for tree structured domains”, *Neurocomputing,* 64, 73-92.

[63] K.K. Lee, C.J. Harris, S.R. Gunn, and P.A.S. Reed, (2001), “Regression Models for classification to Enhance Interpretability”, *Proc. Intelligent Processing and Manufacturing of Materials,* (Vancouver, Canada), 2001.

[64] http://www.cs.wisc.edu/~yuh-jyc/prelim.ps, http://ftp.math.ncsu.edu/FTP/Kelley/PAPERS/WPI.PS.pdf

[65] M.H. Law and J.T. Kwok, (2001), “Bayesian support vector regression”, In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics* (AISTATS), pp. 239--244, Key West, Florida, USA, January.

[66] R. Collobert, and S. Bengio, (2000), “Support vector machines for large-scale regression problems”, *Technical report, IDIAP.*

[67] J. L. Balcazar, Y. Dai, and O. Watanabe, (2002), “Provably Fast Support Vector Regression Using Random Sampling”, *Proceedings of SIAM Workshop on Discrete Mathematics and Data Mining,* Arlington VA, April, 2002, pp 19-29.

[68] S.P. Liao, H.T. Lin, and C.J. Lin, (2002), “A Note on the Decomposition Methods for Support Vector Regression”, *Neural Computation*; 14:1267-1281.

[69] J. Bi, K.P. Bennett, M. Embrechts, C.M. Breneman, and M. Song, (2003), “Dimensionality reduction via sparse support vector machines”, *Journal of Machine Learning Research*, Vol. 3, pp. 1229-1244.

[70] W. Chu, S.S. Keerthi, and C.J. Ong, (2004), “Bayesian support vector regressions using a unified loss function”, *IEEE Transactions on Neural Networks* 15(1):29-44.

[71] D.H. Hong, and C. Hwang, (2004), “Support Vector Machine for Interval Regression”, *Proceedings of the Autumn Conference*, Korean Statistical Society.

[72] W. Wang and Z. Xu, (2004), “A heuristic training for support vector regression”, *Neurocomputing, 61*, 259-275.

[73] A. Farag, and R.M. Mohamed, (2004), “Regression Using Support Vector Machines: Basic Foundations”, *Technical report*, December.

[74] C.S. Qazaz, C.K.I. Williams and C..M. Bishop, (1977), “An Upper Bound on the Bayesian Error Bars for Generalized Linear Regression**”,** *Mathematics of Neural Networks: Models, Algorithms and Applications*. Eds. S. W. Ellacott, J. C. Mason, I. J. Anderson. Kluwer.

[75] G. Camps-Valls, E. Soria-Olivas, J.J. Perez-Ruixo, F. Perez-Cruz, A.R. Figueiras-Vidal, and A. Artes-Rodriguez, (2001), “Cyclosporine Concentration Prediction using Clustering and Support Vector Regression Methods”, *IEE2001CyA.pdf*.

[76] H. Yang, L. Chan, Laiwan and I. King, (2002), “Support Vector Machine Regression for Volatile Stock Market Prediction”, {hqyang,lwchan,king} @cse.cuhk.edu.hk, H.Yin et. al. (Eds): *IDEAL 2002*, LNCS 2412, pp 391-396, Springer-Verlag Berlin Heidelberg.

[77] M. Song, C.M. Breneman, J. Bi, N. Sukumar, K.P. Bennett, S. Cramer and N. Tugcu, (2002), “Prediction of Protein Retention Times in Anion-exchange Chromatography Systems using Support Vector Machine Regression”**,** *JCICS (Journal of Chemical Information and Computer Science),* 42(6), 1347-1357 NOV-DEC 2002**,** http://dx.doi.org/10.1021/ci025580t.

[78] J.D. Martin-Guerrero, G. Camps-Valls, E. Soria-Olivas, A.J. Serrano-Lopez, J.J. Perez-Ruixo, N.V. Jimenez-Torres, (2003), “Dosage individualization of erythropoietin using a profile-dependent support vector regression”, *IEEE Trans Biomed Eng*. Oct;50(10):1136-42.

[79] J.T. Kwok and I.W. Tsang, (2003), “Linear Dependency Between $\varepsilon$ and the Input Noise in $\varepsilon$ -Support Vector Regression”, *IEEE Transactions on Neural Networks*, 14(3): 544-553, May.

[80] H. Yang, K. Huang, L. Chan, I. King and M.R. Michael, (2004), “Outliers Treatment in Support Vector Regression for Financial Time series Prediction”, *ICONIP-2004:1260-1265.*

[81] G. Camps-Valls, A.M. Chalk, A.J. Serrano-López, J.D. Martín-Guerrero, José D and E.L.L. Sonnhammer, (2004), “Profiled support vector machines for antisense oligonucleotide efficacy prediction”, *BMC Bioinformatics*, 5:135 ; http://www.biomedcentral.com/1471-2105/5/135.

[82] C.H. Wu, C.C. Wei, D.C. Su, M. H. Chang and J. M. Ho, (2004), “Travel-time prediction with support vector regression”, *IEEE Transactions on Intelligent Transportation Systems*, Dec. 2004, Volume: 5, Issue: 4, page(s): 276- 281.

[83] O. Ivanciuc, (2005),” Support Vector Regression Quantitative Structure-Activity Relationships (QSAR) for Benzodiazepine Receptor Ligands”, *Internet Electron. J. Mol. Des. 4*, 181-193.

[84] E. Myasnikova, A. Samsonova and J. Reinitz, (2002), “Support vector regression applied to the determination of the developmental age of a *Drosophila* embryo from its segmentation gene expression patterns”, *Bioinformatics*, Vol. 18 no. 90001, Pages S87-S95.

[85] S. Gezici, H. Kobayashi, and H.V. Poor, (2003), “A New Approach to Mobile Position Tracking”, Proc. IEEE Sarnoff Symposium on Advances in Wired and Wireless Communication, pp 204-207, EWING, NJ, March 11-12.

[86] Y. Wang, R.T. Schultz, R.T. Constable and L.H. Staib, (2003), “Nonlinear Estimation and Modeling of fMRI Data using Spatio-Temporal Support Vector Regression”, *Information Processing in Medical Imaging Proceedings*, 647-659.

[87] I. Santamaria, J. Via and C.C. Gaudes, (2004), “Robust Blind Identification of SIMO Channels: A Support Vector Regression Approach”, *Proc. of the 2004 IEEE International Conference on Accoustics, Speech and Signal Processing, ICASSP 2004,* May 17-21, Montreal, Quebec, Canada, Mayo.

[88] N. Theera-Umpon and U. Boonprasert, (2004), “Leeway Prediction of Oceanic Disastrous Target via Support Vector Regression”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.8, No.6 pp. 606-612.

[89] C. Mukkamala, A.H. Sung, A. Abraham, and V. Ramos, (2004), “Intrusion Detection Systems Using Adaptive Regression Splines”, in *ICEIS-04, 6th International Conf. on Enterprise Information Systems,* I. Seruca, J. Filipe, S. Hammoudi and J. Cordeiro (Eds), ISBN 972-8865-00-7, Vol. 3, pp 26-33, Porto, Portugal, April 14-17.

[90] M. Kobayashi, Y. Konishi, S. Fujita, and H. Ishigaki, (2005), "Position Control of Ultrasonic Motor using Support Vector Regression", *Acta Press, Proceeding (487) Computational Intelligence.*

[91] W. Chu, and S.S. Keerthi, (2005), "New approaches to support vector ordinal regression", *International Conference on Machine Learning* (ICML-05):145-152.

[92] A. Chalimourda, B. Schölkopf, and A.J. Smola, (2000), "Choosing in Support Vector Regression with Different Noise Models: Theory and Experiments", *IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00) Volume 5, 2000:* 199-204.

[93] C. Kuo, Y. Ng, and C.J. Wang, (2002), "Gloss Patch Selection Based on Support Vector Regression", *PICS 2002: IS&T's PICS Conference, An International Technical Conference on Digital Image Capture and Associated System, Reproduction and Image Quality Technologies*, Portland, Oregon; April; p.121-125.

[94] J.H. Jeng, and M.H. Cheng, (2003), "Image Coding using Support Vector Machine", jjeng@isu.edu.tw, ssbn3067@seed.net.tw, Paper ID: 2003A203.

[95] D.R. Musicant and A. Feinberg, (2004), "Active Set Support Vector regression", *IEEE Transactions on Neural Networks*, Vol. 15, No. 2, 268-275.

[96] G. Nalbantov, R. Bauer and I. Sprinkhuizen-Kuyper, (2004**), "**Equity Style Timing Using Support Vector Regressions", Social Science Research Network, *Tomorrow's Research Today*.

[97] E. Angelini, R. Campanini, and A. Riccardi, (2005), "Support vector regression filtering for reduction of false positives in a mass detection CAD Scheme: Preliminary Results", submitted to the IEEE *ICIP05. angelini05support.pdf*.

[98] M.W. Chang, and C.J. Lin, (2005), **"**Leave-One-Out Bounds for Support Vector Regression Model Selection", Neural *Computation*, 2005; 17:1188-1222.

[99] N. Chandra, and A.N. Iyer, "Support Vector regression for Fusion of Usable Speech Measures", (http://www.personal.psu.edu/users/a/n/ani103/files/chandra_citsa04_ labs.pdf).

[100] G. Steidl, S. Didas and J. Neumann, (2005), "Relations Between Higher Order TV Regularization and Support Vector Regression", R. Kimmel, N. Sochen, J.Weickert (Eds.); *Scale-Space 2005*, LNCS 3459, pp. 515-527, Springer-Verlag Berlin Heidelberg.

**Debasish Basak** received B.E. (Electrical Engg.) in 1986 from National Institute of Technology (formerly Regional Engineering College), Durgapur, India and M.E. (Electrical Engg.) with specialization in System Engineering & Operations Research in 1988 from I.I.T. Roorkee (formerly University of Roorkee), India and **MBA** in Operations Management in 2002 from IGNOU, New Delhi, India. He has worked nearly one year at IIT Kharagpur, India as "Junior Research Engineer" in a project sponsored by ISRO, Bangalore, India. In 1989, he joined Central Institute of Mining and Fuel Research (erstwhile Central Mining Research Institute), Dhanbad, India where he is currently Scientist "EII" and head of Electrical Laboratory. He has published about 25 research papers in national/international conferences/journals. His research interest includes support vector machines, optimization, reliability, non-destructive study of steel wire ropes.

**Srimanta Pal** obtained B.Sc. (Mathematics) from University of Calcutta in 1978, B.Tech. (Instrumentation and Electronics Engineering) and MBA (Operations Research) from Jadavpur University in 1982 and 1989 Respectively, M.Tech. (Computer Science.) from Indian Statistical Institute in 1984 and Ph.D. (Computer Science and Engineering) from Indian Institute of Technology, Kharagpur in 1992. He has co-guest edited the special issues on *Computational Intelligence for Pattern Recognition* in IJPRAI, Vol 16(2), World Scientific, 2002 and *Advances in Pattern Recognition* in *Pattern Recognition Letters*, Vol 26(4), Elsevier, 2005. He is also co-edited the volumes *Advances in Pattern Recognition, ICAPR 2003* and *Neural Information Processing, ICONIP-2004*, LNCS, Springer. At present he is an Associate Professor in Indian Statistical Institute, Calcutta.

**Dipak Chandra Patranabis** is currently an Emeritus Professor in the Department of Instrumentation, Jadavpur University, Calcutta.