03-17

Suykens J.A.K., Van Gestel T., De Brabanter J., De Moor B., Vandewalle J., ``Support Vector Machines : Least Squares Approaches and Extensions", in Chapter 8 of *Advances in Learning Theory : Methods, Models and Applications*, (Suykens J.A.K., Horvath G., Basu S., Micchelli C. and Vandewalle J., eds.), IOS Press (NATO-ASI Series in Computer and Systems Sciences) (Amsterdam, The Netherlands), 2003, pp. 155-178., Lirias number: 181452.

Advances in Learning Theory: Methods, Models and Applications

Edited by

J.A.K. Suykens
G. Horvath
S. Basu
C. Micchelli
J. Vandewalle

To be published by

IOS Press: NATO-ASI Series in Computer and Systems Sciences, Amsterdam, The Netherlands, 2003

Chapter 8

Support Vector Machines: Least Squares Approaches and Extensions

Johan A.K. Suykens, Tony Van Gestel, Jos De Brabanter,
Bart De Moor and Joos Vandewalle[1]

Abstract. Support Vector Machines (SVMs) is a powerful methodology for nonlinear classification, function estimation and density estimation. However, due to its non-parametric nature, the original SVM methodology is more difficult to extend to other problem settings than e.g. classical multilayer perceptrons. In this Chapter we give a short overview on Least Squares SVMs (LS-SVMs) and demonstrate how several extensions are possible. LS-SVMs for classification and function estimation are closely related to kernel Fisher discriminant analysis, regularization networks and Gaussian processes, and aim at exploiting primal-dual formulations from the viewpoint of optimization theory. A Bayesian framework with three levels of inference is developed together with ways for achieving sparseness and robustness. The framework with primal-dual support vector machine formulations is extended to kernel-PCA, -CCA, -PLS, recurrent networks and optimal control. For very large scale problems, on-line learning and transductive inference, a method of fixed-size LS-SVM is proposed which allows modelling in the primal and the dual space in relation to a Nyström approximation with active selection of support vectors and leads to sparse representations.

[1]Research supported by *Research Council KUL*: GOA-Mefisto 666, IDO (IOTA oncology, genetic networks), several PhD/postdoc & fellow grants; *Flemish Government*: FWO: PhD/postdoc grants, G.0407.02 (support vector machines), projects G.0115.01 (microarrays/oncology), G.0240.99 (multilinear algebra), G.0080.01 (collective intelligence), G.0413.03 (inference in bioi), G.0388.03 (microarrays for clinical use), G.0229.03 (ontologies in bioi), G.0197.02 (power islands), G.0141.03 (identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), research communities (ICCoS, ANMMM); AWI: Bil. Int. Collaboration Hungary, Poland, South Africa; IWT: PhD Grants, STWW-Genprom (gene promotor prediction), GBOU-McKnow (knowledge management algorithms), GBOU-SQUAD (quorum sensing), GBOU-ANA (biosensors); Soft4s (softsensors) *Belgian Federal Government*: DWTC (IUAP IV-02 (1996-2001) and IUAP V-22 (2002-2006); PODO-II (CP/40: TMS and sustainibility); *EU*: CAGE; ERNSI; Eureka 2063-IMPACT; Eureka 2419-FliTE; *Contract Research/agreements*: Data4s, Electrabel, Elia, LMS, IPCOS, VIB; JS is a professor at K.U.Leuven Belgium and a postdoctoral researcher with FWO Flanders. TVG is postdoctoral researcher with FWO Flanders. BDM and JWDW are full professors at K.U.Leuven Belgium.

8.1 Introduction

In recent years considerable progress has been made in kernel based learning methods, after the introduction of Support Vector Machines (SVMs) [5, 30, 42, 43]. Up till now standard SVMs as proposed by Vapnik have been mainly developed for classification, nonlinear function estimation and density estimation. Interesting features of SVMs are sparse approximation by solving convex optimization problems and the primal-dual optimization problem formulations with use of Mercer's theorem. Furthermore, the use of the kernel trick (i.e. application of the Mercer theorem), has also been demonstrated for other problems such as principal component analysis and clustering [28, 29, 30]. In this way, the investigated models and methods have become important for the areas of neural networks, machine learning, pattern recognition, signal processing, datamining, systems and control with aspects relevant to mathematics, linear algebra, optimization and statistics. An interesting new aspect of SVM methods in relation to many classical methods is for example that the models are able to learn and generalize in huge dimensional input spaces which is important e.g. for novel applications in bioinformatics and textmining.

On the other hand, classical neural network methodologies such as multilayer perceptrons (MLP) [3] are currently still more flexible as a general plug-in solution thanks to the parametric nature of the model. For MLPs it is straightforward to parameterize classifiers, static nonlinear models, recurrent networks, unsupervised learning methods, controllers etc. SVM methods have often outperformed many other techniques but on the other hand it is mainly restricted to classification and static function estimation problems. For this purpose, least squares approaches are explored with the following motivations:

- Formulating SVM methods in terms of least squares cost functions and with equality constraints instead of inequality constraints usually leads to solving linear systems. Such problems are better understood from a theoretical, algorithmical and numerical point of view than solving quadratic programming problems. In view of interior point algorithms the obtained KKT systems can be considered as a core problem. Towards adaptive signal processing applications recursive least squares algorithms are well understood, e.g. in relation to Kalman filtering techniques.

- In terms of the simpler formulations it may become easier to understand the links between the new generation of SVM techniques and the related methods in different areas within the wide interdisciplinary context.

- It is shown that several extensions to the standard SVM formulations are possible in terms of least squares such as for PCA, PCR, PLS, CCA analysis problems and recurrent networks and control. In these formulations the primal-dual SVM optimization problem interpretations remain preserved. In general this leads to a new avenue of primal-dual modelling thinking where, for a given problem at hand, one may either solve the primal or the dual problem. The dual space is interesting for large dimensional inputs (e.g. classification of microarray data in bioinformatics), while the primal space can be interesting for a large number of training data (e.g. large scale datamining problems).

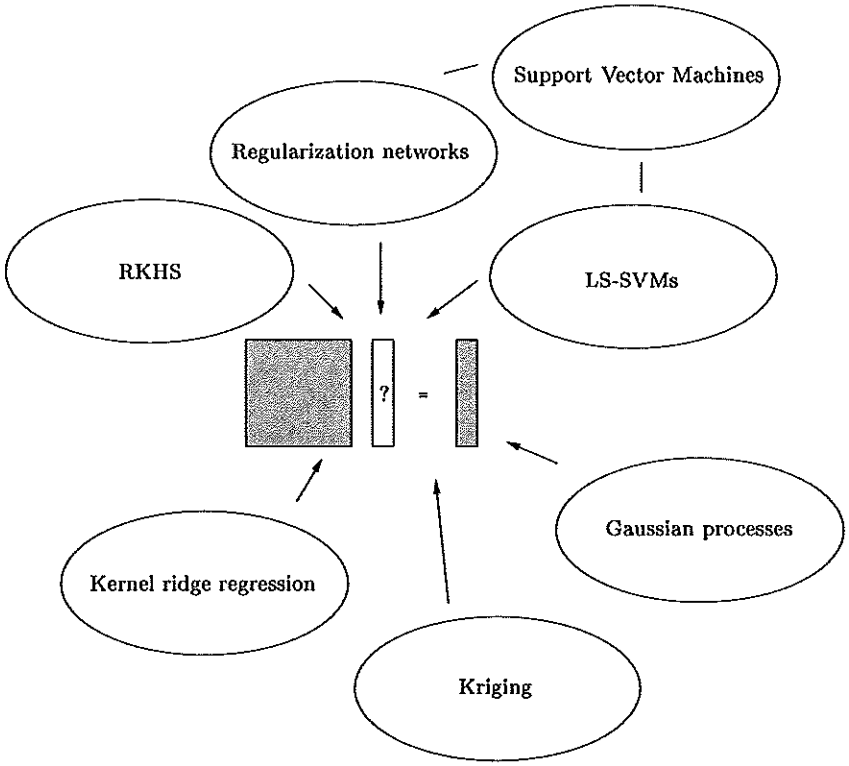


Figure 8.1: LS-SVMs are closely related to regularization networks, Gaussian processes, kriging and kernel ridge regression. The emphasis in the LS-SVM 'language' is on primal-dual optimization problem formulations as in standard SVMs.

In this chapter, we give an overview of developments in least squares support vector machines (LS-SVMs). In the function estimation case the solutions are mathematically equivalent to regularization networks and Gaussian processes (but it is more straightforward to handle bias terms in the model). The emphasis in LS-SVM models is on the primal-dual interpretations as for standard SVMs (Fig. 8.1). The classifier formulation is closely related to a kernel version of Fisher discriminant analysis. The use of least squares may have potential drawbacks such as the lack of sparseness and the robustness with respect to outliers or non-Gaussian noise distributions. Several ways are mentioned to overcome these drawbacks, such as the application of pruning techniques and the incorporation of methods from robust statistics. Probabilistic interpretations of the models are made within a Bayesian framework with three levels of inference. In the primal space, SVM models can be viewed as parametric (size of unknown vector is fixed) while in the dual space it becomes non-parametric (size of solution vector grows with the number of data). At the different levels of inference, one considers the unknown weights in the primal problem, next the hyperparameters and finally the tuning parameters of the kernel. Furthermore, support vector machine formulations are shown for PCA and CCA analysis and their kernel versions. Besides these extensions also recurrent versions of LS-SVMs have been formulated and use in optimal control. For very large scale problems and adaptive learning a method of fixed size LS-SVM has been proposed which makes use of the Nyström method with active selection of support vectors. Further details can be found in [33].

8.2 Least Squares SVMs for Classification and Function Estimation

8.2.1 LS-SVM classifiers and link with kernel FDA

Consider a given training set $\{x_k, y_k\}_{k=1}^N$ with input data $x_k \in \mathbb{R}^n$ and output data with class labels $y_k \in \{-1, +1\}$. The standard nonlinear SVM classifier [42] which takes the form

$$y(x) = \text{sign}\left[w^T\varphi(x) + b\right] \tag{8.1}$$

in the primal space with $\varphi(\cdot)$: $\mathbb{R}^n \to \mathbb{R}^{n_h}$, where n_h is the dimension of the (potentially infinite dimensional) feature space, and is determined by the optimization problem

$$\left[\begin{array}{lll} \boxed{\text{P}}: & \min\limits_{w,b,\xi} J_{\text{P}}(w,\xi) = & \frac{1}{2}w^Tw + c\sum\limits_{k=1}^{N}\xi_k \\ & \text{such that} & y_k\left[w^T\varphi(x_k) + b\right] \geq 1 - \xi_k, \quad k = 1, ..., N \\ & & \xi_k \geq 0, \quad k = 1, ..., N \end{array}\right] \tag{8.2}$$

where ξ_k are slack variables needed for tolerating misclassifications. In the objective function one penalizes the maximization of the soft-margin (by minimization of the regularization term) with respect to the sum of these slack variables, subject to the set of inequalities for correct classification of the training data, except for an amount of misclassified points determined by tuning parameter c. One solves then the dual problem in the unknown Lagrange multipliers associated to this problem.

For least squares support vector machine classifiers [34] this is reformulated into

$$\left[\begin{array}{lll} \boxed{\text{P}}: & \min\limits_{w,b,e} J_{\text{P}}(w,e) = & \frac{1}{2}w^Tw + \gamma\frac{1}{2}\sum\limits_{k=1}^{N}e_k^2 \\ & \text{such that} & y_k\left[w^T\varphi(x_k) + b\right] = 1 - e_k, \; k = 1, ..., N \end{array}\right] \tag{8.3}$$

where the set of inequalities are replaced by equalities. Instead of considering the value 1 as a threshold value, it is taken as a target value upon which an error variable is considered which is penalized in a least squares sense within the cost function with regularization constant γ.

The Lagrangian for the problem is

$$\mathcal{L}(w, b, e; \alpha) = J_{\text{P}}(w, e) - \sum_{k=1}^{N}\alpha_k\{y_k\left[w^T\varphi(x_k) + b\right] - 1 + e_k\} \tag{8.4}$$

where the α_k values are the Lagrange multipliers, which can be positive or negative due to the equality constraints.

The conditions for optimality yield

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \;\rightarrow\; w = \sum_{k=1}^{N} \alpha_k y_k \varphi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \;\rightarrow\; \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \alpha_k = \gamma e_k, \qquad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; y_k \left[w^T \varphi(x_k) + b \right] - 1 + e_k = 0, \quad k = 1, ..., N. \end{cases} \tag{8.5}$$

Defining $y = [y_1; ...; y_N]$, $1_v = [1; ...; 1]$, $e = [e_1; ...; e_N]$, $\alpha = [\alpha_1; ...; \alpha_N]$ and eliminating w, e, one obtains the following linear Karush-Kuhn-Tucker (KKT) system

$$\left[\begin{array}{l} \boxed{\text{D}}: \text{ solve in } \alpha, b: \\ \\ \left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega + I/\gamma \end{array} \right] \left[\begin{array}{c} b \\ \hline \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \hline 1_v \end{array} \right] \end{array} \right] \tag{8.6}$$

where $\Omega_{kl} = y_k y_l K(x_k, x_l)$ for $k, l = 1, ..., N$ and $K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l)$ is a positive definite kernel, according to the Mercer theorem. In the dual space the classifier takes the form

$$y(x) = \text{sign} \left[\sum_{k=1}^{N} \alpha_k y_k K(x, x_k) + b \right] \tag{8.7}$$

where α, b follows from solving the linear system. Any positive definite kernel can be taken such as e.g. the linear kernel $K(x_k, x_l) = x_k^T x_l$ and RBF kernel $K(x_k, x_l) = \exp(-\|x_k - x_l\|_2^2/\sigma^2)$.

For a larger number of data points, iterative methods such as Conjugate Gradient (CG) algorithms [11] can be applied for solving the linear system. Due to the fact that a bias term is considered in the model, the linear system matrix is indefinite. However, it is straightforward to transform it into a positive definite system such that CG algorithms can be applied to it [33, 39]. Also SMO optimization has been developed for LS-SVM classifiers [47]. LS-SVM classifiers have been successfully tested on 20 UCI (binary and multiclass) benchmark data sets. The method consistently performs very well in comparison with many other methods on all these data sets and its performance is comparable to the standard SVM performance.

SVM models have neural network interpretations both in the primal and the dual space. In the primal space the problem is essentially parametric, while in the dual space it becomes non-parametric because the size of the solution vector α grows with the number of data points (Fig. 8.2). In this LS-SVM formulation, every data point is a support vector but some points contribute more than others as follows from the conditions for optimality with $\alpha_k = \gamma e_k$. For standard SVMs the support vectors are typically located close to the decision boundary, while in the LS-SVM case the points with large $|\alpha_k|$ values are located close and far from the decision boundary.

The LS-SVM classifier formulation can be related to a kernel version of Fisher Discriminant Analysis (FDA) [2, 23, 40]. In linear FDA the data are projected as

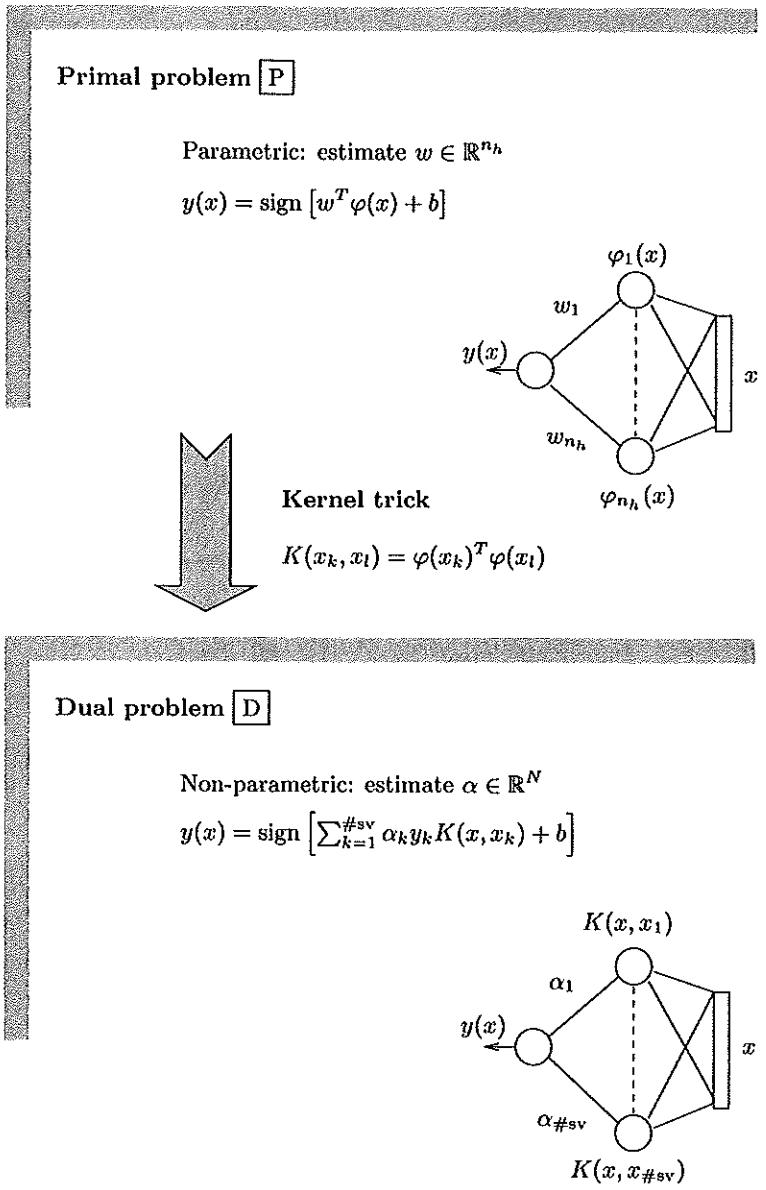


Figure 8.2: Primal-dual neural network interpretations of support vector machines.

follows

$$z = w^T x + b \tag{8.8}$$

where w is determined by optimizing the Rayleigh quotient

$$\max_w J_{\text{FD}}(w) = \frac{w^T \Sigma_B w}{w^T \Sigma_W w} \tag{8.9}$$

with Σ_B, Σ_W the between class and within class covariance matrix, respectively. In the dual space, the kernel FDA version with Fisher target ± 1 is closely related to the LS-SVM classifier solution.

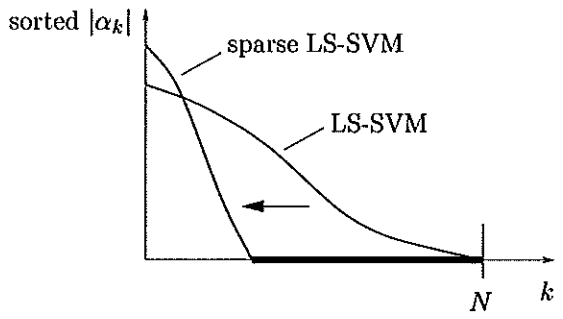


Figure 8.3: Pruning techniques as a simple method for imposing sparseness in LS-SVM models.

8.2.2 Function estimation case and equivalence to a regularization network solution

The LS-SVM function estimation formulation is given by

$$\left[\begin{array}{l} \boxed{\mathrm{P}}: \ \min_{w,b,e} J_{\mathrm{P}}(w,e) = \frac{1}{2}w^Tw + \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 \\ \qquad \text{such that} \quad y_k = w^T\varphi(x_k) + b + e_k, \ \ k = 1,...,N \end{array}\right] \tag{8.10}$$

which corresponds to a form of ridge regression [27]. Taking conditions for optimality with Lagrangian $\mathcal{L}(w,b,e;\alpha) = \mathcal{J}_{\mathrm{P}}(w,e) - \sum_{k=1}^{N}\alpha_k\{w^T\varphi(x_k) + b + e_k - y_k\}$ gives the dual problem after taking the conditions for optimality $\partial\mathcal{L}/\partial w = 0$, $\partial\mathcal{L}/\partial b = 0$, $\partial\mathcal{L}/\partial e_k = 0$, $(k = 1,..,N)$, $\partial\mathcal{L}/\partial\alpha_k = 0$, $(k = 1,..,N)$,

$$\left[\begin{array}{l} \boxed{\mathrm{D}}: \ \text{solve in } \alpha, b\text{:} \\ \qquad \left[\begin{array}{c|c} 0 & 1_v^T \\ \hline 1_v & \Omega + I/\gamma \end{array}\right]\left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right] \end{array}\right] \tag{8.11}$$

with $\Omega_{kl} = K(x_k, x_l)$ for $k,l = 1,...,N$. The solution is mathematically equivalent to regularization networks [8, 7, 24] although in the LS-SVM formulation it is more straightforward to take into account a bias term. Regularization networks are formulated in the context of functional analysis, while (LS)-SVMs are derived in view of primal-dual optimization problem formulations and invoke the Mercer theorem for linking the model between the primal and the dual space. LS-SVM and regularization network solutions also correspond to kernel ridge regression, Gaussian processes for function estimation and kriging [4, 19, 20, 44, 45].

8.2.3 Issues of sparseness and robustness

Potential drawbacks of the use of least squares cost functions are the lack of sparseness and the lack of robustness with respect to outliers and non-Gaussian noise distributions.

On the other hand there are several possible ways to overcome these drawbacks. A simple way to sparsify the obtained LS-SVM models is to apply neural network

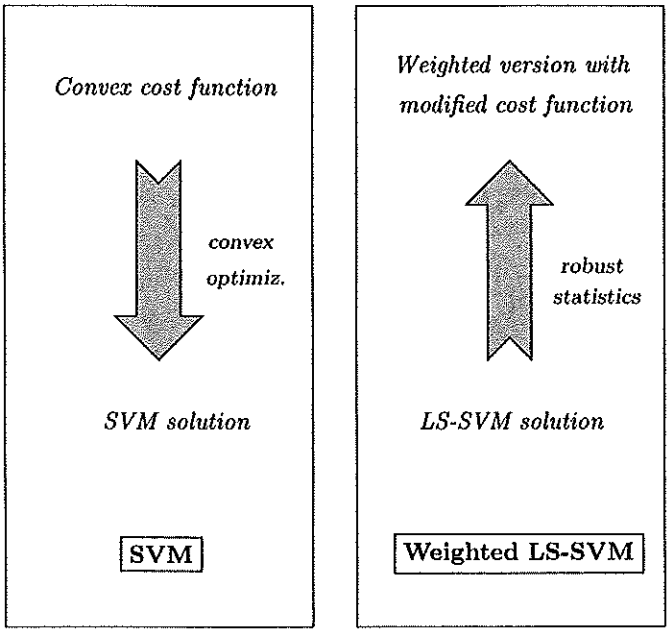


Figure 8.4: In standard SVMs one takes a top-down approach by choosing a convex cost function and finds the SVM solution by convex optimization. On the other hand one can also follow a bottom-up approach by starting from LS-SVM and define a weighting in terms of the error distribution based on robust statistics which implicitly corresponds to a modified cost function. In this way one aims at implicitly finding a most suitable cost function in view of robust statistics, with a good robustness-efficiency trade-off.

techniques of pruning (such as optimal brain damage and optimal brain surgeon) [13, 3]. Less relevant weights are gradually removed and the model is retrained (Fig. 8.3). A simple version with pruning of the smallest $|\alpha_k|$ values has been successfully applied to many classification and function estimation problem (although the obtained degree of sparsity is not extremely high due to the simplicity of this procedure). In this way one reduces the size of the model [35]. Another approach with a fixed size method will be explained in the sequel of this chapter. Furthermore, it is also known that there is a close connection between SVMs and sparse approximation in general [10].

In the standard SVM function estimation case the Vapnik epsilon insensitive loss function is usually employed. Essentially, this is an L_1 estimator (which ensures robustness) but with an epsilon insensitive zone around the origin for achieving sparseness. The SVM methodology has been extended to any convex cost function (Fig. 8.4). In order to robustify LS-SVM models it is straightforward to apply an additional weighted version. The weightings are determined as a function of the empirical distribution of the error variables e_k with a robust scale estimation based on robust statistics [16]. In standard SVMs a *convex* cost function is chosen which gives the solution in a top-down fashion by applying a convex optimization algorithm, e.g. by using interior point algorithms. The reduced KKT system to be solved within interior point algorithms takes the same form as one single LS-SVM KKT system. Hence, conceptually one may view it as solving a sequence of LS-SVM KKT systems. In the weighted LS-SVM case

one rather works in a bottom-up way, by first solving the unweighted LS-SVM and then solve a weighted version, where one additional weighting step is often sufficient [35]. Implicitly this weighted version corresponds to solving a modified cost function. At this point one should note that several popular loss functions in robust statistics (Andrews, Hampel, Tukey) are non-convex and cannot be plugged-in into standard SVM formulations, except for the Huber loss function [12].

8.2.4 Bayesian inference of LS-SVMs and Gaussian processes

A Bayesian framework with three levels of inference has been developed for LS-SVM classifiers and regression (Fig. 8.5) [40]. At this point there are a few differences with related work in the area of Gaussian Processes (GP) [45]. In the LS-SVM models it is straightforward to handle a bias term, while in GP this should be done by taking an additional constant within the kernel. The LS-SVM classifier case can be treated as a regression problem with targets ± 1 while the classifier case in GP needs the application of sampling techniques. Furthermore, tuning parameters of the kernel are treated at a different level of inference. The Bayesian framework for LS-SVM models is done in the same way as the framework developed by MacKay for MLPs [22].

At the first level, inference is considered in the primal weight parameters w, b. It leads to a probabilistic interpretation of the output and an additional correction for the choice of the bias term. At the second level, inference of the hyperparameters is done. In addition to the regularization constant γ, which is associated to the least squares cost function, an additional hyperparameter is needed which is associated which is related to the regularization term (hyperparameters μ, ζ). This leads to expressions for the effective number of parameters in terms the eigenvalues of the centered kernel matrix. At the third level of inference different models $\mathcal{H}_\sigma$ (assume e.g. models parameterized by a kernel tuning parameter σ) can be compared, where the kernel width σ is selected by maximizing the level 3 posterior. By taking as many σ_i values as components of the input vector (equivalent to taking a weighted norm [24]), these σ_i values can be inferred at level 3. In this way one can assess the relevance of the input variables of the model. This procedure is called Automatic Relevance Determination (ARD).

8.3 Primal-dual Formulations to Kernel PCA and CCA

8.3.1 Kernel PCA as a one-class modelling problem and a primal-dual derivation

In view of the least squares formulations for classification and nonlinear function estimation, extensions of the framework can be made towards kernel Principal Component Analysis (PCA) [28], kernel canonical correlation analysis (CCA) [1] and kernel partial least squares (PLS) [26].

In order to obtain a support vector machine formulation of kernel PCA with primal-dual interpretation [38], the PCA analysis problem is viewed as a one-class modelling problem with target zero (instead of targets ± 1 as for the classification case) upon which

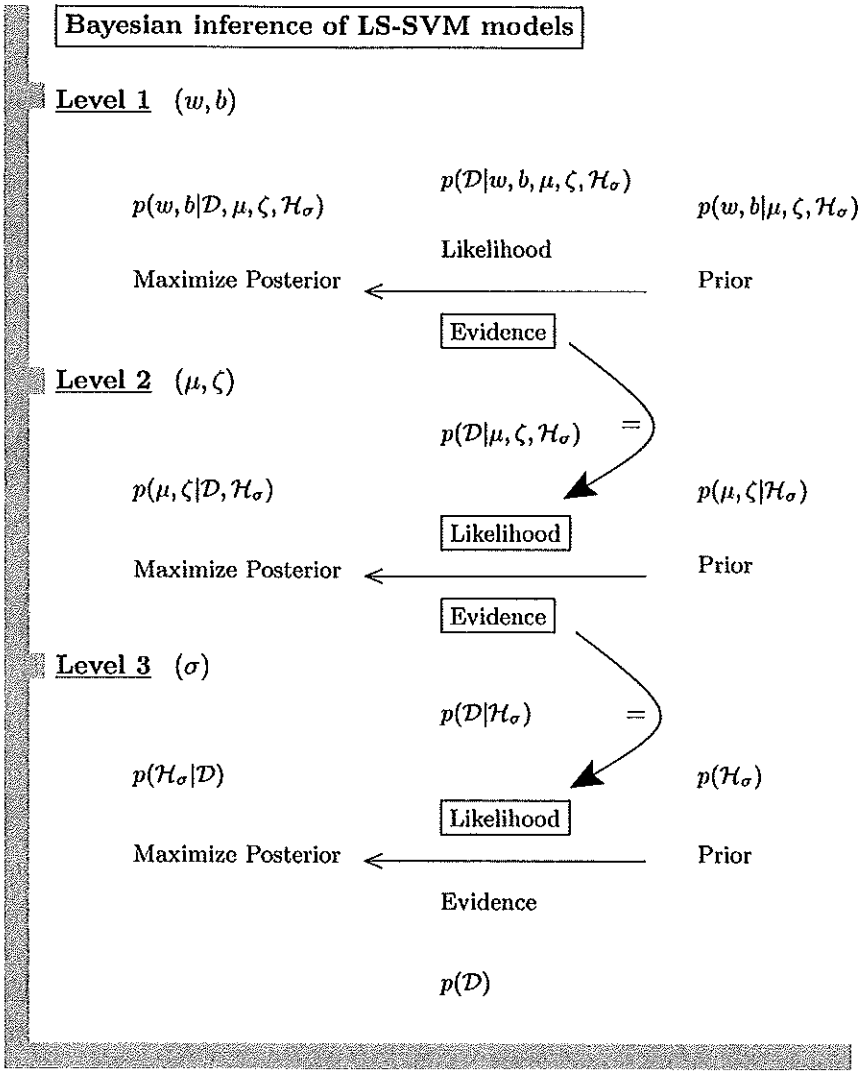


Figure 8.5: Illustration of Bayesian inference for LS-SVM models with several levels of inference.

an error variable is considered and the sum squared error is maximized (in comparison with minimization for the classification case). These error variables correspond to the score variables (Fig. 8.6).

The objective is then the following

$$\max_{w} \sum_{k=1}^{N} \left(0 - w^T(\varphi(x_k) - \widehat{\mu}_\varphi)\right)^2 \tag{8.12}$$

with $\widehat{\mu}_\varphi = (1/N)\sum_{k=1}^{N} \varphi(x_k)$ and $\varphi(\cdot)$: $\mathbb{R}^n \to \mathbb{R}^{n_h}$ the mapping to a high dimensional feature space which might be infinite dimensional. Instead of this centering approach, also a bias term can be taken. The following optimization problem is formulated in

the primal weight space

$$\left[\begin{array}{ll} \boxed{\mathrm{P}}: & \max_{w,e} J_{\mathrm{P}}(w,e) = \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2}w^T w \\ & \text{such that} \quad e_k = w^T\left(\varphi(x_k) - \widehat{\mu}_\varphi\right),\ k=1,...,N. \end{array}\right] \tag{8.13}$$

This gives the Lagrangian

$$\mathcal{L}(w,e;\alpha) = \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2}w^T w - \sum_{k=1}^{N}\alpha_k\left(e_k - w^T(\varphi(x_k) - \widehat{\mu}_\varphi)\right) \tag{8.14}$$

with conditions for optimality

$$\left\{\begin{array}{lll} \frac{\partial\mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum_{k=1}^{N}\alpha_k(\varphi(x_k) - \widehat{\mu}_\varphi) \\ \frac{\partial\mathcal{L}}{\partial e_k} = 0 & \rightarrow & \alpha_k = \gamma e_k, \qquad\qquad k=1,...,N \\ \frac{\partial\mathcal{L}}{\partial\alpha_k} = 0 & \rightarrow & e_k - w^T(\varphi(x_k) - \widehat{\mu}_\varphi) = 0, \quad k=1,...,N. \end{array}\right. \tag{8.15}$$

By elimination of the variables e, w one obtains

$$\frac{1}{\gamma}\alpha_k - \sum_{l=1}^{N}\alpha_l(\varphi(x_l) - \widehat{\mu}_\varphi)^T(\varphi(x_k) - \widehat{\mu}_\varphi) = 0\ ,\ k=1,...,N. \tag{8.16}$$

Defining $\lambda = 1/\gamma$ one gets the following dual problem

$$\left[\begin{array}{ll} \boxed{\mathrm{D}}: & \text{solve in } \alpha: \\ & \Omega_c\alpha = \lambda\alpha \end{array}\right] \tag{8.17}$$

with

$$\Omega_c = \left[\begin{array}{ccc} (\varphi(x_1) - \widehat{\mu}_\varphi)^T(\varphi(x_1) - \widehat{\mu}_\varphi) & \ldots & (\varphi(x_1) - \widehat{\mu}_\varphi)^T(\varphi(x_N) - \widehat{\mu}_\varphi) \\ \vdots & & \vdots \\ (\varphi(x_N) - \widehat{\mu}_\varphi)^T(\varphi(x_1) - \widehat{\mu}_\varphi) & \ldots & (\varphi(x_N) - \widehat{\mu}_\varphi)^T(\varphi(x_N) - \widehat{\mu}_\varphi) \end{array}\right] \tag{8.18}$$

with as elements for the centered kernel matrix

$$\Omega_{c,kl} = (\varphi(x_k) - \widehat{\mu}_\varphi)^T(\varphi(x_l) - \widehat{\mu}_\varphi)\,, \quad k,l=1,...,N. \tag{8.19}$$

For the centered kernel matrix one can apply the kernel trick for given points x_k, x_l:

$$\begin{aligned} &(\varphi(x_k) - \widehat{\mu}_\varphi)^T\ (\varphi(x_l) - \widehat{\mu}_\varphi) \\ &= K(x_k,x_l) - \frac{1}{N}\sum_{r=1}^{N}K(x_k,x_r) - \frac{1}{N}\sum_{r=1}^{N}K(x_l,x_r) + \frac{1}{N^2}\sum_{r=1}^{N}\sum_{s=1}^{N}K(x_r,x_s). \end{aligned} \tag{8.20}$$

For the linear PCA case this new formulation shows the link with principal co-ordinate analysis [17] as the dual problem. In this sense, kernel PCA as proposed by Schölkopf *et al.* corresponds to the kernel version of principal co-ordinate analysis [17].

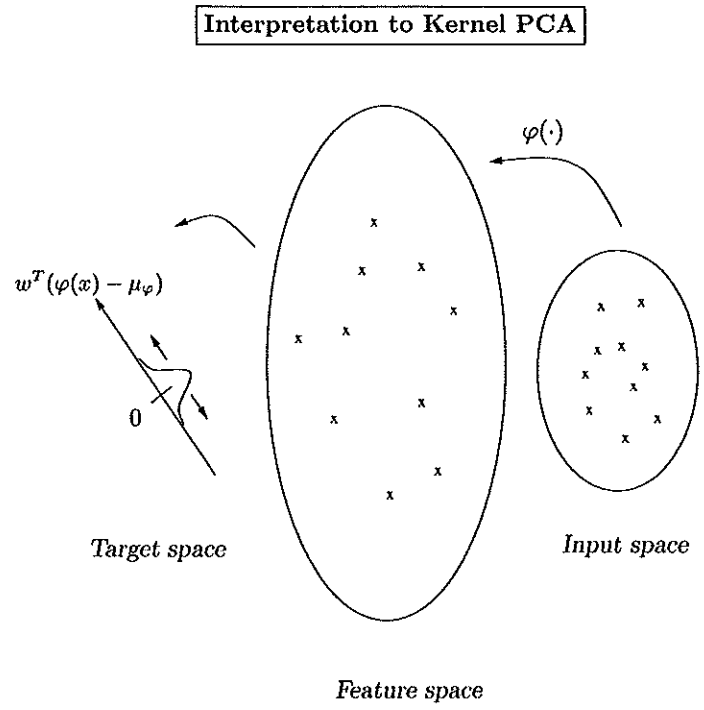


Figure 8.6: Interpretation of kernel PCA towards a primal-dual support vector machine formulation.

8.3.2 A support vector machine formulation to Kernel CCA

The problem of canonical correlation analysis (CCA) is closely related to the PCA analysis problem [17]. In CCA analysis (originally studied by Hotelling [15]) one is interested in finding maximal correlation between projected variables $z_x = w^T x$ and $z_y = v^T y$ where $x \in \mathbb{R}^{n_x}, y \in \mathbb{R}^{n_y}$ denote given random vectors with zero mean. Linear CCA analysis has been applied e.g. in subspace algorithms for system identification, with links to system theory, information theory and signal processing.

The objective function is to maximize the correlation coefficient

$$\begin{aligned} \max_{w,v} \rho &= \frac{\mathcal{E}\left[z_x z_y\right]}{\sqrt{\mathcal{E}\left[z_x z_x\right]}\sqrt{\mathcal{E}\left[z_y z_y\right]}} \\ &= \frac{w^T C_{\mathrm{xy}} v}{\sqrt{w^T C_{\mathrm{xx}} w}\sqrt{v^T C_{\mathrm{yy}} v}} \end{aligned} \tag{8.21}$$

with $C_{\mathrm{xx}} = \mathcal{E}\left[xx^T\right]$, $C_{\mathrm{yy}} = \mathcal{E}\left[yy^T\right]$, $C_{\mathrm{xy}} = \mathcal{E}\left[xy^T\right]$.

Towards a kernel CCA version [1] and its primal-dual support vector machine formulation, one considers the score variables [33]

$$\begin{aligned} z_x &= w^T\left(\varphi_1(x) - \widehat{\mu}_{\varphi_1}\right) \\ z_y &= v^T\left(\varphi_2(y) - \widehat{\mu}_{\varphi_2}\right) \end{aligned} \tag{8.22}$$

where $\varphi_1(\cdot)$: $\mathbb{R}^{n_x} \to \mathbb{R}^{n_{h_x}}$ and $\varphi_2(\cdot)$: $\mathbb{R}^{n_y} \to \mathbb{R}^{n_{h_y}}$ are mappings (which can be chosen to be different) to high dimensional feature spaces and $\widehat{\mu}_{\varphi_1} = (1/N)\sum_{k=1}^{N}\varphi_1(x_k)$, $\widehat{\mu}_{\varphi_2} = (1/N)\sum_{k=1}^{N}\varphi_2(y_k)$. It is important to take centering into account for the nonlinear CCA problem.

One starts from the primal problem

$$\left[\begin{array}{l} \boxed{\mathrm{P}}: \quad \max_{w,v,e,r} \quad \gamma\sum_{k=1}^{N} e_k r_k - \nu_1\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \nu_2\frac{1}{2}\sum_{k=1}^{N} r_k^2 - \frac{1}{2}w^T w - \frac{1}{2}v^T v \\ \qquad \text{such that} \quad e_k = w^T\left(\varphi_1(x_k) - \widehat{\mu}_{\varphi_1}\right), \qquad k = 1,...,N \\ \qquad\qquad\quad\; r_k = v^T\left(\varphi_2(y_k) - \widehat{\mu}_{\varphi_2}\right), \qquad k = 1,...,N \end{array}\right] \tag{8.23}$$

with Lagrangian

$$\begin{aligned} \mathcal{L}(w,v,e,r;\alpha,\beta) = \gamma\sum_{k=1}^{N} e_k r_k - \nu_1\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \nu_2\frac{1}{2}\sum_{k=1}^{N} r_k^2 - \frac{1}{2}w^T w - \\ \frac{1}{2}v^T v - \sum_{k=1}^{N}\alpha_k\left[e_k - w^T\left(\varphi_1(x_k) - \widehat{\mu}_{\varphi_1}\right)\right] - \sum_{k=1}^{N}\beta_k\left[r_k - v^T\left(\varphi_2(y_k) - \widehat{\mu}_{\varphi_2}\right)\right] \end{aligned} \tag{8.24}$$

where α_k, β_k are Lagrange multipliers. Note that w and v might be infinite dimensional.

The conditions for optimality are

$$\left\{\begin{array}{ll} \frac{\partial\mathcal{L}}{\partial w} = 0 \;\to\; w = \sum_{k=1}^{N}\alpha_k\left(\varphi_1(x_k) - \widehat{\mu}_{\varphi_1}\right) & \\ \frac{\partial\mathcal{L}}{\partial v} = 0 \;\to\; v = \sum_{k=1}^{N}\beta_k\left(\varphi_2(y_k) - \widehat{\mu}_{\varphi_2}\right) & \\ \frac{\partial\mathcal{L}}{\partial e_k} = 0 \;\to\; \gamma v^T\left(\varphi_2(y_k) - \widehat{\mu}_{\varphi_2}\right) = \nu_1 w^T\left(\varphi_1(x_k) - \widehat{\mu}_{\varphi_1}\right) + \alpha_k & \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 1,...,N & \\ \frac{\partial\mathcal{L}}{\partial r_k} = 0 \;\to\; \gamma w^T\left(\varphi_1(x_k) - \widehat{\mu}_{\varphi_1}\right) = \nu_2 v^T\left(\varphi_2(y_k) - \widehat{\mu}_{\varphi_2}\right) + \beta_k & \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 1,...,N & \\ \frac{\partial\mathcal{L}}{\partial\alpha_k} = 0 \;\to\; e_k = w^T\left(\varphi_1(x_k) - \widehat{\mu}_{\varphi_1}\right) & \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 1,...,N & \\ \frac{\partial\mathcal{L}}{\partial\beta_k} = 0 \;\to\; r_k = v^T\left(\varphi_2(y_k) - \widehat{\mu}_{\varphi_2}\right) & \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 1,...,N & \end{array}\right. \tag{8.25}$$

which results in the following dual problem after defining $\lambda = 1/\gamma$

$$\left[\begin{array}{l} \boxed{\mathrm{D}}: \text{ solve in } \alpha,\beta: \\ \qquad \begin{bmatrix} 0 & \Omega_{c,2} \\ \Omega_{c,1} & 0 \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ \qquad = \lambda\begin{bmatrix} \nu_1\Omega_{c,1} + I & 0 \\ 0 & \nu_2\Omega_{c,2} + I \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \end{array}\right] \tag{8.26}$$

where

$$\begin{aligned} \Omega_{c,1_{kl}} &= (\varphi_1(x_k) - \widehat{\mu}_{\varphi_1})^T (\varphi_1(x_l) - \widehat{\mu}_{\varphi_1}) \\ \Omega_{c,2_{kl}} &= (\varphi_2(y_k) - \widehat{\mu}_{\varphi_2})^T (\varphi_2(y_l) - \widehat{\mu}_{\varphi_2}) \end{aligned} \tag{8.27}$$

are the elements of the centered Gram matrices for $k, l = 1, ..., N$. In practice these matrices can be computed by $\Omega_{c,1} = M_c \Omega_1 M_c$, $\Omega_{c,2} = M_c \Omega_2 M_c$ with centering matrix $M_c = I - 1_v 1_v^T / N$. The eigenvalues and eigenvectors that give an optimal correlation coefficient value are selected. The resulting score variables can be computed by applying the kernel trick with kernels $K_1(x_k, x_l) = \varphi_1(x_k)^T \varphi_1(x_l)$, $K_2(y_k, y_l) = \varphi_2(y_k)^T \varphi_2(y_l)$. This kernel CCA formulation can also be further related to kernel partial least squares (PLS), which has been studied in [14, 26].

8.4 Large Scale Methods and On-line Learning

8.4.1 Nyström method

For linear (LS)-SVMs one may in fact solve the primal problem instead of the dual. The primal is advantageous for solving problems with a large number of given training data N, while the dual is suitable for large dimensional input spaces. For the nonlinear case on the other hand the situation is more complicated. For many choices of the kernel, $\varphi(\cdot)$ may become infinite dimensional and hence also the w vector. However, one may still try to find meaningful estimates for $\varphi(x_k)$.

A technique to find such estimates is the Nyström method, which is well known in the area of integral equations and has been successfully applied in the context of Gaussian processes by Williams & Seeger in [46]. The method is related to finding a low rank approximation to the given kernel matrix by randomly choosing M rows/columns of the kernel matrix. Let us denote the *big* kernel matrix by $\Omega_{(N,N)} \in \mathbb{R}^{N \times N}$ and the *small* kernel matrix based on the random subsample $\Omega_{(M,M)} \in \mathbb{R}^{M \times M}$ with $M < N$ (in practice often $M \ll N$). Consider the eigenvalue decomposition of the *small* kernel matrix $\Omega_{(M,M)}$

$$\Omega_{(M,M)} \overline{U} = \overline{U}\, \overline{\Lambda} \tag{8.28}$$

where $\overline{\Lambda} = \mathrm{diag}([\overline{\lambda}_1; ...; \overline{\lambda}_M])$ contains the eigenvalues and $\overline{U} = [\overline{u}_1 ... \overline{u}_M] \in \mathbb{R}^{M \times M}$ the corresponding eigenvectors. This is related to eigenfunctions $\phi_i(x)$ and eigenvalues λ_i of the integral equation

$$\int K(x, x') \phi_i(x) p(x) dx = \lambda_i \phi_i(x') \tag{8.29}$$

as follows

$$\begin{aligned} \widehat{\lambda}_i &= \tfrac{1}{M} \overline{\lambda}_i \\ \widehat{\phi}_i(x_k) &= \sqrt{M}\, \overline{u}_{ki} \\ \widehat{\phi}_i(x') &= \tfrac{\sqrt{M}}{\overline{\lambda}_i} \sum_{k=1}^{M} \overline{u}_{ki} K(x_k, x') \end{aligned} \tag{8.30}$$

where $\widehat{\lambda}_i$ and $\widehat{\phi}_i$ are estimates to λ_i and ϕ_i, respectively, for the integral equation, and $\overline{u}_{ki}$ denotes the ki-th entry of the matrix $\overline{U}$. This can be understood from sampling the

integral by M points $x_1, x_2, ..., x_M$. For the *big* kernel matrix one has the eigenvalue decomposition

$$\Omega_{(N,N)}\, \widetilde{U} = \widetilde{U}\, \widetilde{\Lambda}. \tag{8.31}$$

Furthermore, as explained in [46] one has

$$\begin{aligned} \widetilde{\lambda}_i &= \tfrac{N}{M}\overline{\lambda}_i \\ \widetilde{u}_i &= \sqrt{\tfrac{N}{M}}\tfrac{1}{\overline{\lambda}_i}\Omega_{(N,M)}\overline{u}_i. \end{aligned} \tag{8.32}$$

One can then show that

$$\Omega_{(N,N)} \simeq \Omega_{(N,M)}\Omega_{(M,M)}^{-1}\Omega_{(M,N)} \tag{8.33}$$

where $\Omega_{(N,M)}$ is the $N \times M$ block matrix taken from $\Omega_{(N,N)}$. These insights are used then for approximately solving the linear system in combination with applying the matrix inversion lemma.

8.4.2 Basis construction in the feature space using fixed size LS-SVM

In the Nyström method an approximate solution to the linear system is computed based upon a random subsample of the given training data set. In the method of fixed size LS-SVM [33] the number of support vectors is decided beforehand and the support vectors are actively selected instead of taken at random. The method makes use of the Nyström method but additionally links it to entropy criteria and density estimation. The support vectors are selected according to an entropy criterion and the estimation is done in the primal space by exploiting the primal-dual LS-SVM formulations (Fig. 8.7). In this way one selects a basis in the feature space. Other methods related to basis construction in the feature space are e.g. [6, 29, 32].

The estimation in the primal weight space is done in the unknowns w, b

$$\min_{w\in\mathbb{R}^{n_h}, b\in\mathbb{R}} \frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{k=1}^{N}\left(y_k - (w^T\varphi(x_k) + b)\right)^2 \tag{8.34}$$

with estimations for $\varphi(x_k)$ provided by the Nyström method. One chooses a fixed size M ($M \leq N$ and typically $M \ll N$) for a working set of support vectors where the value of M is related to the Nyström subsample. Using the expression (8.30) one obtains

$$\varphi_i(x') = \sqrt{\overline{\lambda}_i}\,\widehat{\phi}_i(x') = \frac{\sqrt{M}}{\sqrt{\overline{\lambda}_i}}\sum_{k=1}^{M}\overline{u}_{ki}K(x_k, x'). \tag{8.35}$$

Hence one constructs the $M \times M$ kernel matrix, takes its eigenvalue decomposition, and computes the eigenfunctions based upon the eigenvalue decomposition of the kernel matrix which gives the expression for $\varphi(x')$ evaluated at any point x'. This can be applied both to function estimation and classification problems.

The model takes the form

$$\begin{aligned} y(x) &= w^T\varphi(x) + b \\ &= \sum_{i=1}^{M} w_i \frac{\sqrt{M}}{\sqrt{\lambda_i}} \sum_{k=1}^{M} \overline{u}_{ki} K(x_k, x). \end{aligned} \tag{8.36}$$

The support values corresponding to the number of M support vectors are then

$$\alpha_k = \sum_{i=1}^{M} w_i \frac{\sqrt{M}}{\sqrt{\lambda_i}} \overline{u}_{ki} \tag{8.37}$$

if one represents the model as

$$y(x) = \sum_{k=1}^{M} \alpha_k K(x_k, x). \tag{8.38}$$

This approach gives explicit links between the primal and the dual representation.

In order to make a more suitable selection of the support vectors instead of a random selection, one can relate the Nyström method to kernel principal component analysis, density estimation and entropy criteria, as discussed in [9] (Fig. 8.8). An analysis is done of the quadratic Renyi entropy [25]

$$H_R = -\log \int p(x)^2 dx \tag{8.39}$$

in relation to kernel PCA and density estimation with

$$\int \widehat{p}(x)^2 dx = \frac{1}{N^2} 1_v^T \Omega 1_v \tag{8.40}$$

where a normalized kernel is assumed with respect to density estimation. One chooses a fixed size M then and actively selects points from the pool of training data as candidate support vectors (Fig. 8.7). In the working set of support vectors a point is randomly selected and replaced by a randomly selected point from the training data set if the new point improves the entropy criterion. Illustrative examples for a regression and classification problem are shown in Fig. 8.9-8.10.

The fixed size LS-SVM method is suitable for *adaptive* signal processing applications where on-line updating of w, b and recursive methods for the eigenvalue decomposition can be applied. Both for recursive least squares and singular value decomposition updating, various efficient algorithms have been developed in the literature, which can be used at this point. Also for transductive inference the use of fixed size LS-SVM is natural due to the fact that the search of support vectors is done in an unsupervised way. In *transductive inference* [41, 43] one is interested in finding a model which is performing well on a specific future data point (or set of data points) rather than a general model (as obtained by inductive inference). If for fixed size LS-SVM one knows future data (in addition to the given training data) on which the model should perform well, then one may take these points into account (without class labels or target values) for possible selection of support vectors.

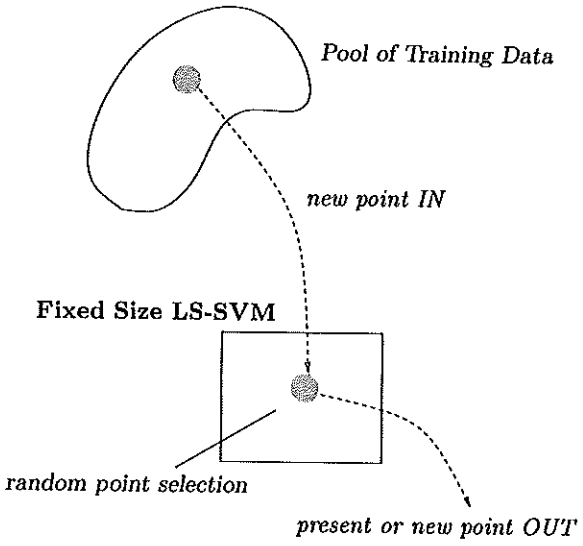


Figure 8.7: Fixed size LS-SVM: the number of support vectors is fixed beforehand and the support vectors are actively selected from the pool of training data. After estimating eigenfunctions the model is computed in the primal space with calculation of w, b. In the working set of support vectors a point is randomly selected and replaced by a randomly selected point from the training data if this new point improves an entropy criterion which is related to density estimation and kernel PCA.

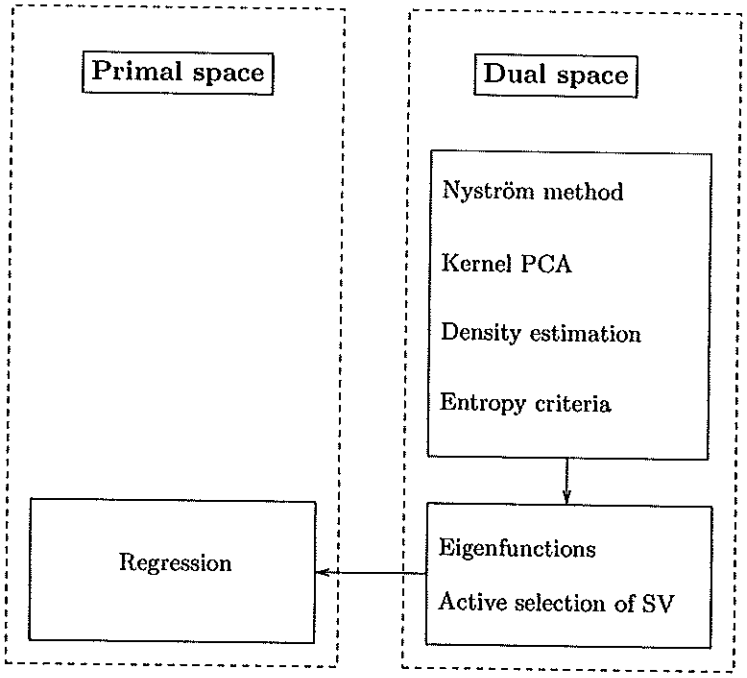


Figure 8.8: In the method of fixed size LS-SVM the Nyström method, kernel PCA and density estimation are linked to estimation of eigenfunctions and active selection of support vectors. Regression in the primal space is done in a next step.

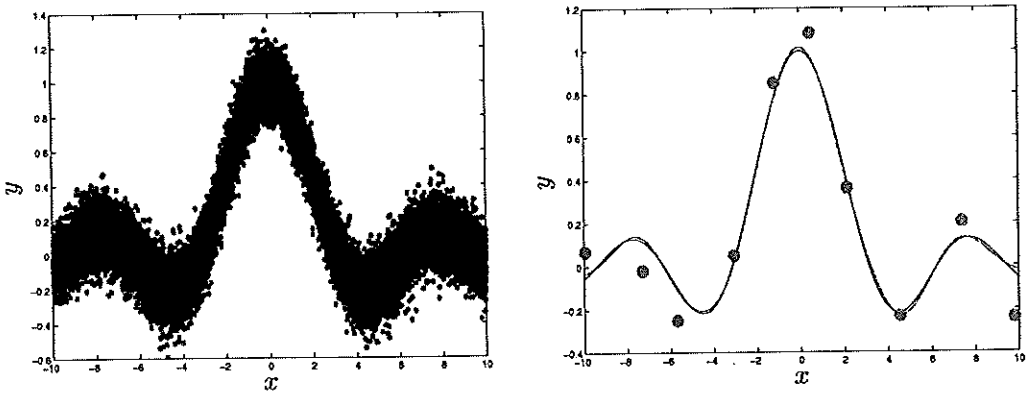


Figure 8.9: Fixed size LS-SVM for a noisy sinc function with $M = 10$ and $N = 20000$. (Left) given training set; (Right) true sinc function (solid line), estimate of Fixed Size LS-SVM (dashed line), and support vectors (grey dots).

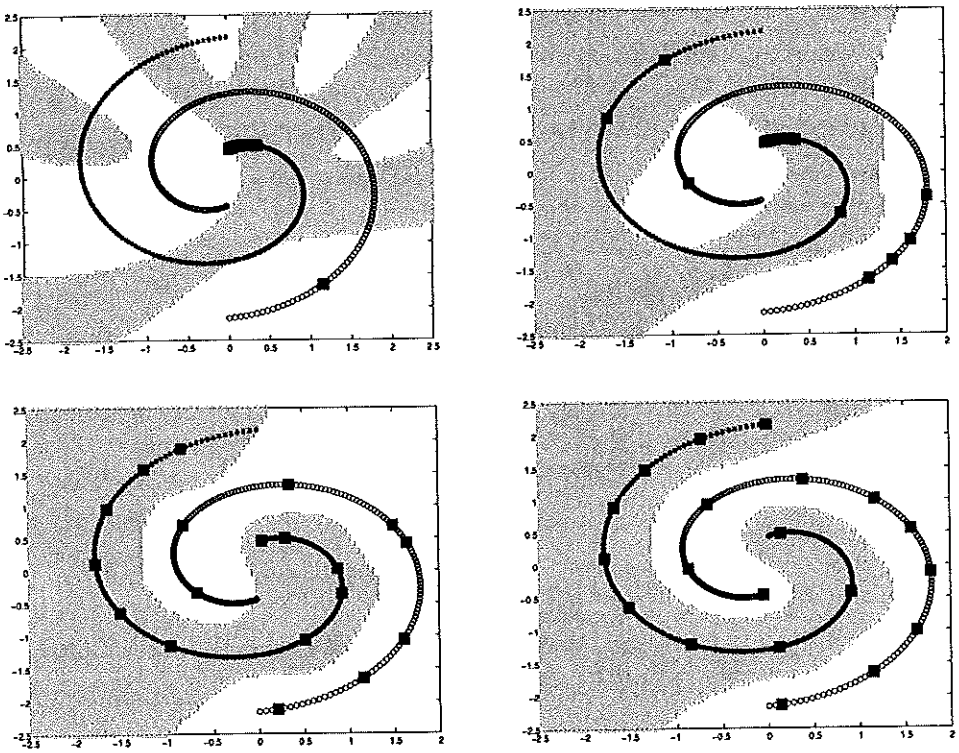


Figure 8.10: Illustration of fixed size LS-SVM on a double spiral classification problem with $M = 20$ support vectors and $N = 400$ given training data. The figures show different stages in the learning process.

8.5 Recurrent Networks and Control

For LS-SVM models also extensions have been made towards recurrent networks and use for optimal control. From a traditional systems identification point of view, one usually first considers a certain model structure (such as NARX, NARMAX, NOE,

nonlinear state space etc.) and then one parameterizes the model structure and solves an optimization problem in the unknown parameter vector [31]. While for classical parametric models such as MLPs and RBF networks one can parameterize both NARX (Nonlinear ARX) and NOE (Nonlinear Output Error) models in a straightforward way (which leads to feedforward and recurrent networks, respectively), the situation is more difficult in the SVM context, due to the non-parametric nature of the method.

In [36] it has been a recurrent LS-SVM model has been formulated, which preserves certain primal-dual interpretations and uses of the kernel trick. However, due to the fact that the equality constraints of the problem formulation are nonlinear in the unknowns instead of linear, the recurrent LS-SVM problem becomes non-convex. The approach has been successfully illustrated on trajectory learning and prediction of chaotic systems.

The fact that SVM theory makes use of elements of optimization theory and optimal control theory is closely related to optimization theory (in the area of model predictive control these links are even more explicit), the study and use of SVM methodology within control theory becomes a natural open question. In [37] first attempts and results are made in this direction, where LS-SVM formulations for nonlinear state feedback controllers are merged with finite time horizon optimal control problems. After formulating the primal problems and taking conditions for optimality from the Lagrangian, the solution is characterized by a set of nonlinear equations for which the kernel trick can be applied.

8.6 Conclusions

In this chapter a brief overview is given of least squares approaches to SVMs. Several extensions to present SVM formulations in classifications and static nonlinear function estimation can be made in terms of least squares and equality constraints based formulations. In this way it becomes straightforward to extend the methods to nonlinear kernel versions of existing linear techniques in pattern recognition and statistics (such as FDA, PCA, CCA, PLS) and make extensions to areas as recurrent networks and control. In addition the links between various related kernel based methods as SVMs, regularization networks, Gaussian processes, kriging and kernel ridge regression become often more explicit. For LS-SVMs the primal-dual optimization problem formulations are emphasized and exploited. It leads for example to a fixed-size version which is capable of handling very large data sets, being related to the Nyström method, and is at the same time very natural towards adaptive signal processing and transductive inference. This fixed-size LS-SVM technique enables to consider modelling in the primal space (parametric) as well as the dual space (non-parametric). The former representation is suitable for handling large data sets while the latter is convenient for problems with large dimensional inputs. It also offers a framework for studying such problems and its many related applications in an highly interdisciplinary manner[2].

[2]For a more extensive discussion of this chapter we refer to the book [33]. A related Matlab/C toolbox LS-SVMlab is available at **http://www.esat.kuleuven.ac.be/sista/lssvmlab/.**

Bibliography

[1] F.R. Bach, M.I. Jordan, Kernel independent component analysis, *Journal of Machine Learning Research* **3** (2002) 1–48.

[2] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, *Neural Computation* **12** (2000) 2385–2404.

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

[4] N. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, New York (1993)

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

[6] L. Csató, M. Opper, Sparse on-line Gaussian processes, *Neural Computation* **14**(3) (2002) 641–668.

[7] F. Cucker, S. Smale, On the mathematical foundations of learning theory, *Bulletin of the AMS* **39** (2002) 1–49.

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

[9] M. Girolami, Orthogonal series density estimation and the kernel eigenvalue problem, *Neural Computation* **14**(3) (2002) 669–688.

[10] F. Girosi, An equivalence between sparse approximation and support vector machines, *Neural Computation* **10**(6) (1998) 1455–1480.

[11] G.H. Golub, C.F. Van Loan, *Matrix Computations*, Baltimore MD: Johns Hopkins University Press (1989)

[12] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W.A. Stahel, *Robust Statistics, the Approach Based on Influence Functions*, John Wiley & Sons, New York (1986)

[13] B. Hassibi, D.G. Stork, Second order derivatives for network pruning: optimal brain surgeon, In Hanson, Cowan, Giles (Eds.) *Advances in Neural Information Processing Systems*, 5, San Mateo, CA: Morgan Kaufmann (1993) 164–171.

[14] L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, B. De Moor, Kernel PLS variants for regression, *European Symposium Artificial Neural Networks* (ESANN 2003) (2003) to appear.

[15] H. Hotelling, Relations between two sets of variates, *Biometrica* **28** (1936) 321–377.

[16] P.J. Huber, *Robust Statistics*, John Wiley & Sons, New York (1981)

[17] I.T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer-Verlag (1986)

[18] S.S. Keerthi, S.K. Shevade, SMO algorithm for Least Squares SVM formulations, *Neural Computation*, to appear (2003)

[19] G.S. Kimeldorf, G. Wahba, A correspondence between Bayesian estimation on stochastic processes and smoothing by splines, *Ann. Math. Statist.* **2** (1971) 495–502.

[20] D.G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, *J. Chem. Metall. Mining Soc. S. Africa* **52**(6) (1951) 119–139.

[21] Y. Le Cun, J.S. Denker, S.A. Solla, Optimal brain damage, In Touretzky (Ed.) *Advances in Neural Information Processing Systems* San Mateo, CA: Morgan Kaufmann 2 (1990) 598–605.

[22] D.J.C. MacKay, Bayesian interpolation, *Neural Computation* 4(3) (1992) 415–447.

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

[24] T. Poggio, F. Girosi, Networks for approximation and learning, *Proceedings of the IEEE* **78**(9) (1990) 1481–1497.

[25] J. Principe, J. Fisher, D. Xu, Information theoretic learning, in S. Haykin (Ed.), *Unsupervised Adaptive Filtering*, John Wiley & Sons, New York (2000).

[26] R. Rosipal, L.J. Trejo, Kernel partial least squares regression in reproducing kernel Hilbert space, *Journal of Machine Learning Research* **2** (2001) 97–123.

[27] C. Saunders, A. Gammerman, V. Vovk, Ridge regression learning algorithm in dual variables, *Proc. of the 15th Int. Conf. on Machine Learning (ICML-98)*, Madison-Wisconsin (1998) 515–521.

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

[29] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, A. Smola, Input space vs. feature space in kernel-based methods, *IEEE Transactions on Neural Networks* 10(5) (1999) 1000–1017.

[30] B. Schölkopf, A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA (2002).

[31] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, *Automatica* **31**(12) (1995) 1691–1724.

[32] A.J. Smola, B. Schölkopf, Sparse greedy matrix approximation for machine learning, in P. Langley (Ed.) *Proc. 17th International Conference on Machine Learning*, 911–918, San Francisco, Morgan Kaufman (2000) 911–918,

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

[34] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, *Neural Processing Letters* **9**(3) (1999) 293–300.

[35] J.A.K. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, *Neurocomputing*, Special issue on fundamental and information processing aspects of neurocomputing **48**(1-4) (2002) 85–105.

[36] J.A.K. Suykens, J. Vandewalle, Recurrent least squares support vector machines, *IEEE Transactions on Circuits and Systems-I* **47**(7) (2000) 1109–1114.

[37] J.A.K. Suykens, J. Vandewalle, B. De Moor, Optimal control by least squares support vector machines, *Neural Networks* **14**(1) (2001) 23–35.

[38] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, B. De Moor, A support vector machine formulation to PCA analysis and its kernel version, *IEEE Transactions on Neural Networks* (2003) in press.

[39] T. Van Gestel, J.A.K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, J. Vandewalle, Benchmarking least squares support vector machine classifiers, *Machine Learning* (2003) in press.

[40] T. Van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, J. Vandewalle, A Bayesian framework for least squares support vector machine classifiers, Gaussian processes and kernel Fisher discriminant analysis, *Neural Computation* **14**(5) (2002) 1115–1147.

[41] V. Vapnik, *Estimation of Dependencies based on Empirical Data*, Springer-Verlag, New York (1982).

[42] V. Vapnik, *The Nature of Statistical Learning Theory,* Springer-Verlag, New York (1995).

[43] V. Vapnik, *Statistical Learning Theory,* John Wiley & Sons, New York (1998)

[44] G. Wahba, *Spline Models for Observational Data*, Series in Applied Mathematics **59**, SIAM, Philadelphia (1990)

[45] C.K.I. Williams, C.E. Rasmussen, Gaussian processes for regression, In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems* 8 MIT Press (1996) 514–520.

[46] C.K.I. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, In T.K. Leen, T.G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems*, 13, MIT Press (2001) 682–688.