Monograph:
Drezet, P. and Harrison, R.F. (1998) Directly Optimised Support Vector Machines for Classification and Regression. Research Report. ACSE Research Report 715 . Department of Automatic Control and Systems Engineering

---

Directly Optimized Support Vector Machines for Classification and Regression

P. Drezet and R. F. Harrison

Department of Automatic Control & Systems Engineering
The University of Sheffield, Sheffield, S1 3JD

Research Report No. 715

Keywords: Support Vector Machines, Support Vector Regression, Sparse approximation.

Abstract

A new method of implementing Support Vector learning algorithms for classification and regression is presented which deals with problems of over-defined solutions and excessive complexity. Classification problems are solved with the minimum number of support vectors, irrespective of over-laping training data. Support vector regression can be solved as a sparse solution, without requiring an ϵ-insensitive zone. The optimisation method is generalised to include control of sparsity for both support vector classification and regression.

1 Introduction

Support Vector Machines (SVMs) are an implementation of Vapnik's theory of Structural Risk Minimisation (SRM). SRM is intended to improve generalisation performance for small sample size learning problems, where Empirical Risk Minimisation (ERM) is likely to overfit the training data [6]. The idea of *capacity* [6] in Vapnik's learning theory is the cornerstone of SRM and is used to evaluate the confidence interval between the empirical risk and the *actual* risk of test errors. Capacity is quantified by VC dimension [5], which can be estimated for a set of real, or indicator functions. Appropriate control of a learning machine's capacity implements SRM.

SVMs can be constructed for classification and regression purposes based on a similar optimisation method. Both methods attempt to produce the *flattest* function in feature space under the constraints of training set errors. This approach approximates SRM minimisation by minimising the estimated VC dimension. In the following text, the discussions of classification SVMs also applies to support vector regression if the idea of non-separable data is associated with noisy regression data. It is also worth stressing that the use of the word 'complexity' is intended to mean practical complexity, rather than the theoretical quantity, capacity.

SVMs are based on dual optimisation techniques and are a mathematically well defined implementation of SRM for separable classification problems. This type of classifier is called an *optimal margin* (OM) classifier, because a decision boundary is found which maximises the margin between the vectors of each class.

The generalised SVM or *soft margin* (SM) classifier for non-separable classification problems, deals with training errors by increasing machine complexity. Complexity is allowed to increase freely, while estimated VC dimension is effectively controlled. The inherent dilemma between complexity minimisation and training error minimisation is thus difficult to balance because practical complexity increases with increasing frequency of training errors, rather than vice-versa. The soft margin algorithm ideally minimises the number of mis-classified data points, but in practical implementations the magnitude of projections on the decision boundary from miss-classified data vectors are minimised. For separable data and where high cost is assigned to training errors, SM classifiers closely approximate OM classifiers, but where training data overlaps, soft margin classifiers simultaneously attempt to minimise training errors and gradient of the weights. The practical problem associated with SVM classifiers based on dual optimisation is that decreasing the cost of training errors often increases the population of support vectors and therefore increases machine complexity.

In noisy regression applications this problem is most evident when small ϵ-insensitive zones [6] are used to maintain accuracy. This results in more outlying data points being considered for error minimisation [6], which decreases sparsity in the solution. In an extreme case where ϵ-insensitivity is reduced to zero, in order to minise the total error magnitude, the full set of training examples are included in the support vector set.

In this paper some alternative methods of solving support vector and similar sparse approximation methods are presented which cope with non-separable classification, or noisy regression data. The cause of SVM's uncontrolled complexity problems lies in the *dual* optimisation method [6]. The dual method has the property that every vector inside the error margin is automatically included in the support vector set, owing to Kuhn-Tucker boundary conditions. If the constrained optimisation functions are solved directly, however, a more flexible basis for the solution can be obtained. SVMs using direct optimisation can deal with error magnitude minimisation without unnecessarily increasing the complexity of the solution.

2 Structural Risk Minimisation

SRM is achieved by minimising the VC dimension of the set of real, or indicator functions. The VC dimension of a linear neural network is bounded by the dimension of the feature space, n, and can be estimated from a trained network's weights, $\vec{w}$, by

$$\min(|\vec{w}|^2 R^2, n) \tag{1}$$

where R is the radius of the smallest sphere in feature space which contains the training vectors. To minimise VC dimension it is therefore evident that

the magnitude of the weights, $\vec{w}$, must be minimised. Geometrically this is described as the optimal margin because the decision boundary seperates training vectors with a maximum separation.

3 Support Vector Classifier Optimisation

The generalised optimal margin classifier is called the soft margin classifier because the constraint of seperating training vectors by a margin is softened. Slack variables, ξ_i, are incorporated to accommodate vectors in and beyond the optimal margin, which gives the following cost function.

$$C(\vec{w}, b) = ||\vec{w}||^{n_w} + C||\vec{\xi}||^{n_\xi} \tag{2}$$

subject to

$$\begin{aligned} y_i(\vec{w}.\vec{x}_i + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \\ \text{for} \quad i &= 1, \ldots, l \end{aligned} \tag{3}$$

$\vec{w}$ is the weight vector and, b, a bias value.

The cost function, (2) and (3), can be solved either as a linear, ($n_w = n_\xi = 1$) or quadtratic ($n_w = 2, n_\xi \in \{1, 2\}$) optimisation, usually this is solved as a quadratic function as this has benefits when non-linear kernels are used to expand the feature space [6]. The quadratic form also lends itself to the dual functional, in parameters of Lagrange multipliers only, the derivation of which can be found in [2]. During this derivation it is shown that

$$w_o = \sum_{i=1}^{l} \alpha_i y_i \vec{x}_i \tag{4}$$

which can also be derived in general from linear feedforward neural networks. It is this property which allows non-linear decision boundaries to be calculated by kernel functions. In the quadratic optimisation function, $||w||^2$ can be written $\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j < \vec{x}_i, \vec{x}_j >$, the dot product $< \vec{x}_i, \vec{x}_j >$ can then be substituted by a reproducing kernel Hilbert space function, $K_H(x_i, x_j)$, which implicitly calculates the scalar dot product in a non-linearly expanded Hilbert space [6].

The dual function has the property that all training vectors which define the constraint boundaries, (3), are included in the support vector set because the associated Lagrange multipliers can only be non-zero for these vectors owing to Kuhn-Tucker conditions. This is an unnecessary restriction in achieving an optimal margin classifier, but is not detrimental in terms of estimated VC dimension, because the minimum set of independant vectors required to construct an optimal boundary always includes the vectors defining the boundary.

By solving the constrained optimisation directly using equations (2) & (3) and substituting equation (4) for $\vec{w}$, no such conditions on boundary vectors are imposed. To guarantee convexity the sum of coefficients, α_i, multiplied by some small constant should be included, which leads to sparse solutions. The resulting set of support vectors is linearly independent in feature space and therefore bounded by the maximum VC dimension of the learning machine and the number of training vectors. This holds when training errors are allowed.

By setting $n_w = 2$ and $n_\xi = 1$, and including a sparsity cost term with coefficient D, the cost functional of the direct approach, with weights substituted by kernel functions, is

$$\begin{aligned} C(\vec{\alpha}) &= \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K_H(\vec{x}_i, \vec{x}_j) \\ &+ C \sum_{i=1}^{l} \xi_i + D \sum_{i=1}^{l} \alpha_i \end{aligned} \tag{5}$$

subject to

$$\begin{aligned} y_i \left(\sum_{j=1}^{l} \alpha_j y_j K_H(\vec{x}_j, \vec{x}_i) + b \right) &\geq 1 - \xi_i, \\ \alpha_i, \xi_i &\geq 0 \\ \text{for} \quad i &= 1, \ldots, l \end{aligned} \tag{6}$$

Suitable kernel functions,K_H, obeying Mercer's condition [3] include polynomial, Gaussian, hyperbolic tan and B-splines. The sparseness weighting, D, is present to avoid multiple solution problems, and has the property that the minimum number of training vectors with the largest magnitudes are selected as support vectors, because this minimise, α, for a particular solution solution. If D is small, the value for C is approaches that of dual optimised SVM.

4 Support Vector Regression Optimisation

SVMs based on dual optimisation are dependent on boundary vectors to construct a solution which is independent of a large part of the training set outside the boundary. This has benefits in some classification problems, but causes problems for regression. The ϵ-insensitive zone in support vector regression has the

purpose of defining a space where redundant training vectors can exist inside a bounded area. It is equivalent to the area outside the optimal margin in SVM classification.

The purpose of ϵ-insensitivity is to prevent the entire training set meeting boundary conditions, and so enables the possibilty of sparsity in dual optimisation of SVM. For unbiased estimation, its size should be less than or equal to the maximum symmetrical deviations expected from the true value, but for the sparsest solution, should be greater than the maximum deviations.

A result of $\epsilon > 0$ is that the solution for noise free training data is guaranteed to be biased. However where amplitude limited noise is present, setting the value of ϵ equal to the maximum noise amplitude yields an unbiased estimate. Even under these rare conditions the solution is prone to error owing to the small proportion of training vectors contributing to error minimisation

The direct optimisation method as described for classification can also be applied to SV regression and solves the sparsity problem without requiring an ϵ-insensitive zone. This allows unbiased and sparse solutions to be found without finding a suitable value for ϵ. The solution can be found as a quadratic optimisation problem implementing either least squares, or least modulus loss for training errors.

The SV regression cost function is

$$C(\vec{w}, b) = ||\vec{w}||^{n_w} + C(||\vec{\xi_i} + \vec{\xi_i^*}||^{n_\epsilon}) \qquad (7)$$

subject to

$$\begin{aligned} \vec{w}.\vec{x}_i + b &\geq y - \epsilon - \xi_i \qquad (8) \\ \vec{w}.\vec{x}_i + b &\leq y + \epsilon + \xi_i^*. \\ \xi_i, \xi_i^* &\geq 0 \\ \text{for} \quad i &= 1, \ldots, l \end{aligned}$$

Introducing lagrange multipliers and deriving the dual functional provides a function which gives weights in terms of input vectors and multipliers as for SVM classifiers. For regression, two non-negative multipliers α, α^* are associated with each training vector to cope with both upper and lower accuracy constraints, the weights are given by $\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\vec{x}_i$.

Regression optimisation can be solved directly from equations (7)-(9). As for classification the weights vector can be substitution for kernel functions K_H. Again, support vectors can be selected arbitrarily from the training set by the direct optimisation of equation (7), and so a sparsity term is included in the cost function. For generality the ϵ-insensitivity option is included, and can be set to zero, in which case the weighting D is singularly responsible for controlling. The direct optimisation cost function with, $n_w = 2$, and, $n_\xi = 1$, is

$$\begin{aligned} C(\vec{w}, b) &= \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K_H(\vec{x}_i, \vec{x}_j) \qquad (9) \\ &+ C \sum_{i=1}^{l} \xi_i + D \sum_{i=1}^{l} \alpha_i \end{aligned}$$

subject to

$$\begin{aligned} \sum_{j=1}^{L} (\alpha_j - \alpha_j^*) K_H(\vec{x}_j, \vec{x}_i) + b &\geq y - \epsilon - \xi_i \qquad (10) \\ \sum_{j=1}^{L} (\alpha_j - \alpha_j^*) K_H(\vec{x}_j, \vec{x}_i) + b &\leq y + \epsilon + \xi_i \\ \alpha_i, \alpha_i^*, \xi_i &\geq 0 \\ \text{for} \quad i &= 1, \ldots, l \end{aligned}$$

5 Sparse Approximation

It is interesting to examine the directly optimised support vector machine with large values of D, or ones which simply minimise the multipliers, α_i, ignoring the quadratic form necessary for SRM. This approach creates a sparse approximation function but does not necessarily implement SRM. Regression does not have the same degree of freedom as classification and so SRM has a weaker theoretical foundation in this area, as it does for highly overlapping classification problems. SRM still benefits these situations, in that contribution from redundant Hilbert sub-spaces are minimised in the solution because feature space weights are minimised, rather than the *hidden layer* weights minimised by sparse sparse approximation (The multipiers α_i can be considered weights in a neural network's hidden layer).

SV regression has been shown to be equivalent to sparse approximation methods under conditions of zero training errors by Girosi [4]. Sparse approximation methods [1] are derived in a regularisation framework and result in a cost function similar to SV regression, except it attempts to minimise the number of non-zero multipliers simultaneously with least squares error cost. The equivelence between direct optimisation of support vector regression machines and sparse approximation methods holds for noisy training problems. The direct optimisation cost can be seen to be a superset of both SRM and sparse approximation. Large values of D tend to implement sparse approximation rather than SRM. If sparse approximation is all that is required, then minimising

multipliers without quadratic coefficients allows implementation through linear programming.

6 Experiments

To demonstrate the differences and similarities between direct and dual optimisation, some typical scenarios are considered.

6.1 Classification

6.1.1 Separable Data

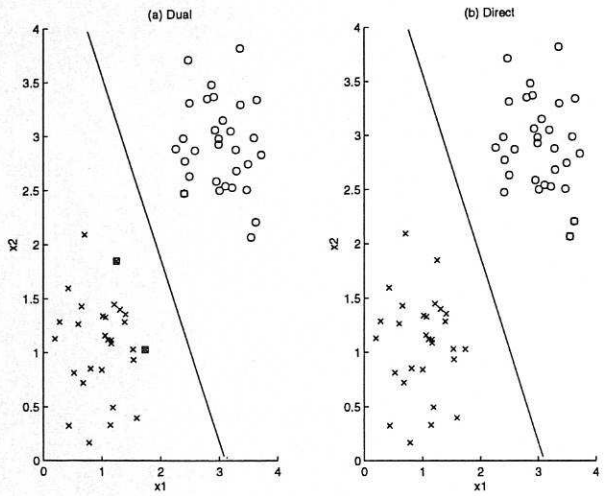

Figure 1: Linear classifiers, $C = 1000$, $D = 10^{-6}$.

The linearly seperable data and the decision boundaries from linear SVM shown in figure 1 are seen to be identical, however different support vectors are selected as the basis of the solution. The weights and bias are identical for both types of classifier. Dual optimisation selects support vectors to be those closest to the decision boundary, figure 1a, while directly optimised networks favor large magnitude vectors, figure 1b.

Figure 2 shows the same data used to train SVMs with RBF kernels. The solutions are identical, including the support vectors selected, the estimated VC dimensions, h_est, are also identical. This differs from the linear case in that identical support vectors are chosen, which is because the training vectors are linearly independent in feature space, and so a single solution exists.

Increasing D, further sparsifies the support vector set until a minimum structure is achieved (figure 3). It is of note that in this example of normally distributed training data that the RBF centres are chosen close the means of the distributions ([2,2] & [3,3]) as can be seen in figure 3b.

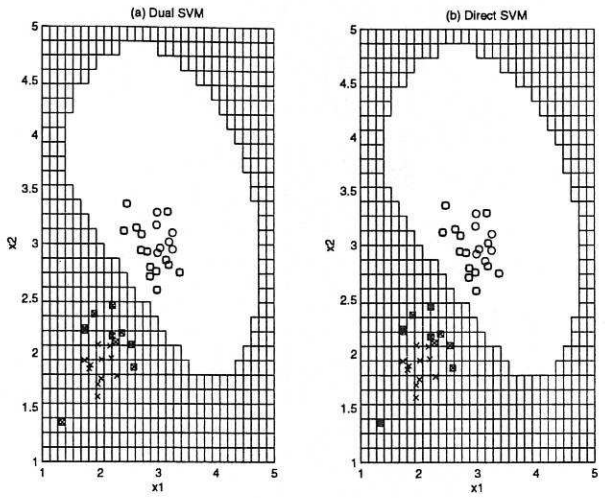

Figure 2: RBF classifiers, $C = 1000$, $D = 10^{-9}$, RBF kernel $K(\vec{x}_i, \vec{x}_j) = exp(\vec{x}_i - \vec{x}_j)^2$, $C = 1000$, $D = 10^{-6}$, $h_{est} = 17$ for both types of SVM. #SVs 24 for both.

6.1.2 Non-Separable Data

Figures 4 &5 show linearly inseperable data and how both types of SVM find similar decision boundaries. The RBF SVMs require a low value of C to prevent overfitting, which causes all but two training examples to be included as support vectors for the SVM optimised with dual formulation (figure 4a). The directly optimised solution selects a smaller set of support vectors, both for linear and RBF constructions.

Where RBF kernels are used the direct approach has also further decreased the estimated VC dimension, h_{est}. Increasing D has decreased ,h_{est}, in this noisy situation because ERM, is traded for hidden layer minimisation, which has decreased the magnitude of feature space weights. Non-seperable problems such as this mean that the extent to which weights can be minimised is not solely determined by the training vectors, therefore weight optimality for SRM is not intrinsically bounded by the problem. In this situation decreasing multipliers, α, may decrease feature space weight magnitude for a given value of, C, but this is not the general case. The decrease in VC dimension is because decreasing $|\alpha|$ decreases $|w|$ locally, but the global minimisation of $|\alpha|$ and $|w|$ subject to error constraints is not similar.

The implication of this is that for a value of estimated VC dimensions, there *may* be several solutions depending on the weightings of C and D in the SM classifier. For the OM classifier, or where $C \to \infty$, there is only one optimal value for h_{est}, and increasing, D, can only maintain or increase this value.

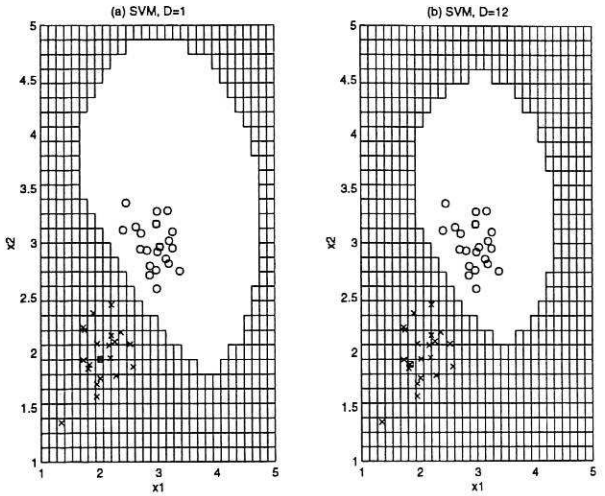

Figure 3: Effects of Increasing D, $C = 1$, RBF kernel $K(\vec{x}_i, \vec{x}_j) = exp((\vec{x}_i - \vec{x}_j)^2$, $h_{est} = 11$ and 1, for D=1 and 11 respecively.

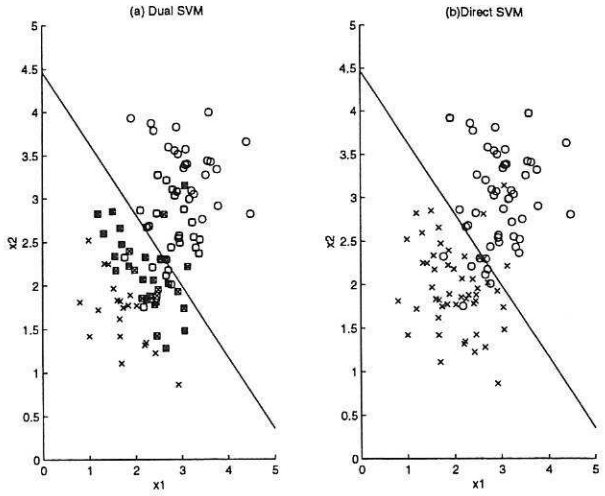

Figure 4: Linear classifiers, $C = 10$,$D = 10^{-6}$. #SVs=48 for dual SVM and 2 for direct SVM.

6.2 Regression

6.2.1 Noise Free Data

Figures 6 and 7 show linear and sinusoidal functions, respectively, estimated by both direct and dual SVM with ϵ set to zero. Both approximations are unbiased, but sparsity is maintained for the direct SVM, while the estimated VC dimensions remain equal. Both direct and dual SVM behave similarly for small values of D.

Figure 8 shows ϵ-insensitivity giving a sparse solution for both machines, but both also giving biased results, however, sparsity and accuracy can be obtained simultaneously by direct SVR by using the sparsity factor, D, as shown in figure 9. Increasing D effects the precision of ERM as does ϵ-insensitivity, but includes all data points in calculating training error. A large value of D in this infinite dimensional feature space example results in a further decrease in h_{est}, because ERM is traded for hidden layer weights, as is the case for non-seperable classification problems.

6.2.2 Noisy Data

Where noise is present in the training data it can be seen from figures 10 and 11 that the direct approach yeilds a more accurate and sparse approximation for similar estimates of VC dimension. The direct formulation has found a minimum error magnitude solution including all training vectors, while the dual method has only considered 10 out of 20 training vectors as noisey. Sparsity in directly optimised SVR is also superior because the mimimum number of hidden layer units are slected to achieve a level of ERM and SRM.

7 Conclusions

Direct SVMs differ from SVM in the following ways:

- The number of SVs are less than or equal to the dimension of feature space.
- SVs are not necessarily training vectors meeting boundary conditions.
- Sparsity can be increased arbitrarily from a lower bound, but at the expense of ERM and SRM.

The similarities in the solution are the following:

- Decision boundaries are similar for a similar training parameter, C.
- Estimated VC dimensions are similar for a given value of C.

The direct SV method is a flexible method of approximating SRM which has practical and theoretical advantages. It can allow a better balance between complexity and ERM, which in the case of regression provides a method of achieving sparsity and SRM, without ϵ-insensitivity and its associated precision problems. It has been shown that VC dimension can be reduced, though not in an optimal sense, by sparsifying methods, rather than weight space flattening.

Including sparseness in the soft margin cost function may lead to multiple weight solutions for a given VC-dimension, which will have different decision boundaries (or approximating properties, in the case of regression). This last point has little theoretical foundation at present, but suggests that in some cases, for a fixed confidence interval, an improvement

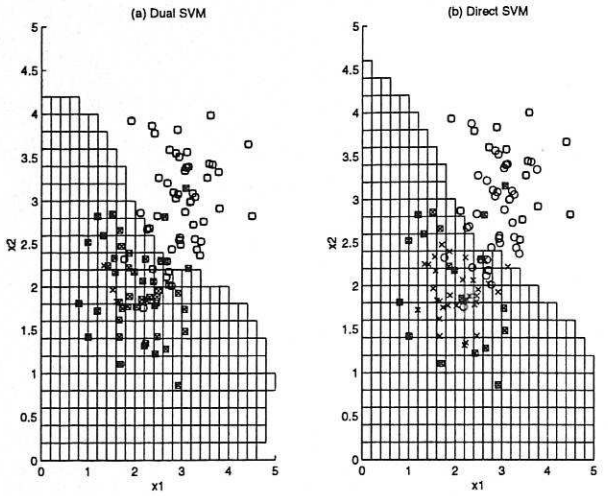

Figure 5: $C = 1$,$D = 10^{-6}$, RBF kernel $K(\vec{x}_i, \vec{x}_j) = exp(\vec{x}_i - \vec{x}_j)^2$, $C = 1000$, $D = 0.01$, #SV =99, 25 ,and $h_{est} = 36, 33$ for dual and direct classifiers, respectively.

in ERM can be attained by additionally controlling sparsity.

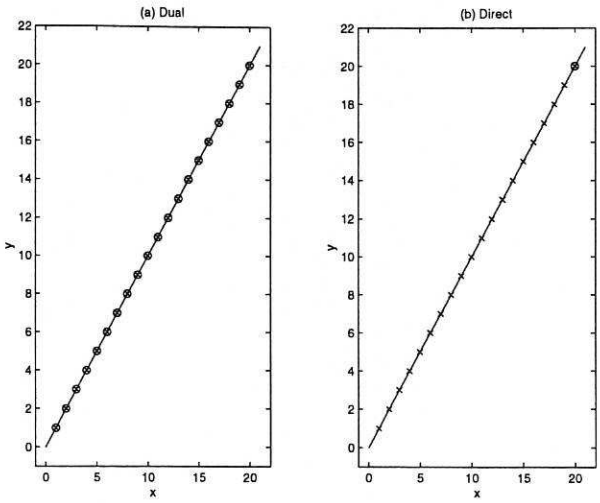

Figure 6: Data from a linear function $y = x$. learning parameters: $C = 1000$, $\epsilon = 0$, $D = 10^{-6}$.

References

[1] M. Bertero. *Regularization methods for linear inverse problems. In C.G. Talenti, editor, Inverse Problems.* Springer-Verlag New York, 1986.

[2] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[3] R. Courant and D. Hilbert. *Methods of Mathematical Physics.* J. Whiley, New York, 1953.

[4] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 1998.

[5] V. Vapnik and Z. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Doklady Akademii Nauk USS*, 4(181), 1968.

[6] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, 1995.

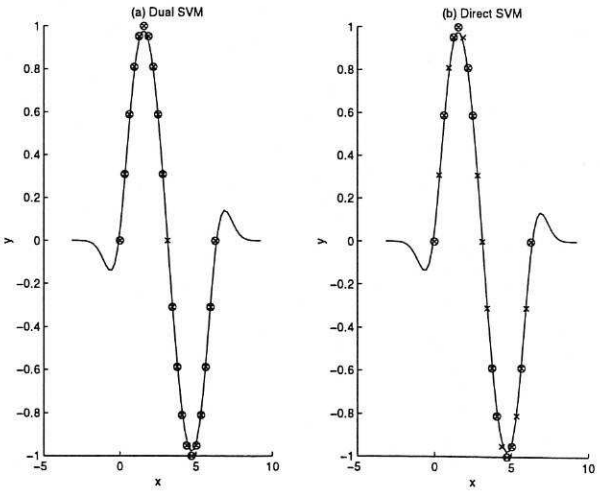

Figure 7: Data from sinusoidal function, learning parameters: $\epsilon = 0$ $C = 10$, $D = 10^{-3}$, $K(\vec{x}_i, \vec{x}_j) = \exp(\vec{x}_i - \vec{x}_j)^2$, $h_{est} = 23$ for both.

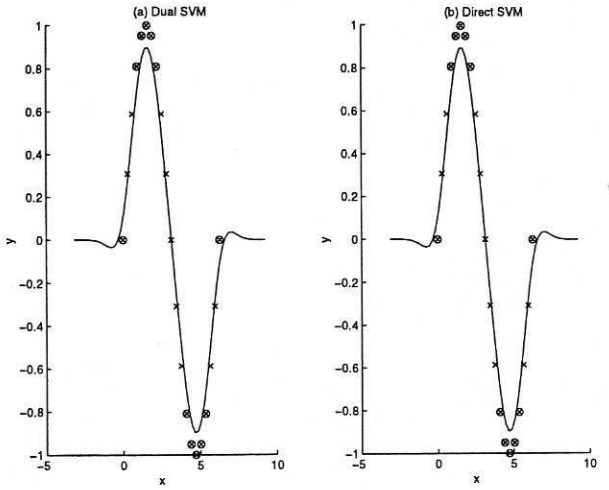

Figure 8: Data from sinusoidal function, learning parameters: $\epsilon = 0.1$, $C = 100$, $D = 10^{-6}$, $K(\vec{x}_i, \vec{x}_j) = \exp(\vec{x}_i - \vec{x}_j)^2$, $h_{est} = 18$ for both.

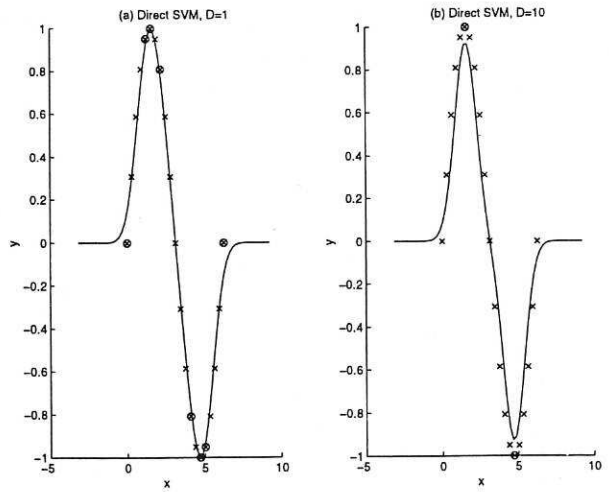

Figure 9: Effects of increasing sparsity factor D. Learning parameters: $\epsilon = 0$, $C = 15$, $K(\vec{x}_i, \vec{x}_j) = \exp(\vec{x}_i - \vec{x}_j)^2$,$h_{est} = 23$, and 2, respectively.

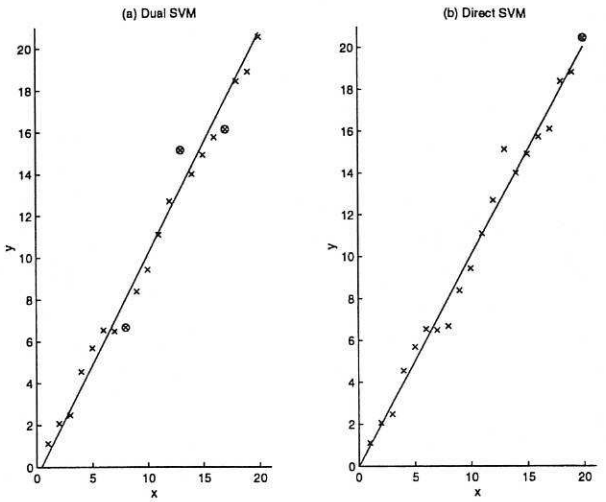

Figure 10: Generator function $y = x+$ Gaussian noise ($\sigma = 0.1$). $C = 100$, $D = 10^{-6}$, $\epsilon = 1$ for dual SVR and zero for direct SVR

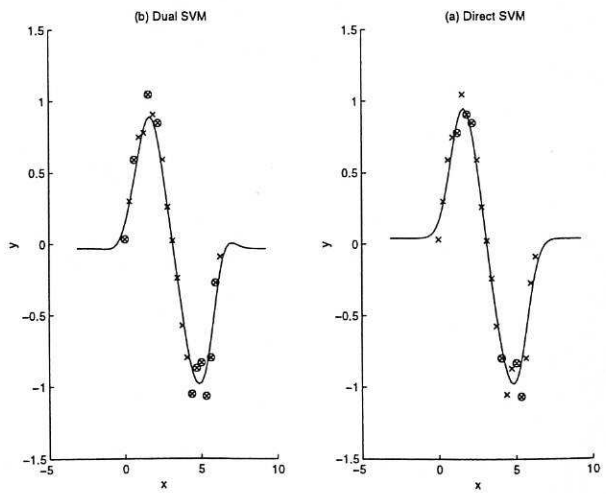

Figure 11: Generator function $y = sin(x)+$ Gaussian noise ($\sigma = 0.1$), $0 < x < 2\pi$, $C = 10$, $D = 1$, ϵ=0.1 for dual SVM. RBF parameters as for noise free example. $h_{est} = 20$ for both SVMs.