Appears in Proceedings of the 15th International Conference on Machine Learning, ICML '98.

Ridge Regression Learning Algorithm in Dual Variables

C. Saunders, A. Gammerman and V. Vovk
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK
{craig,alex,vovk}@dcs.rhbnc.ac.uk

Abstract

In this paper we study a dual version of the Ridge Regression procedure. It allows us to perform non-linear regression by constructing a linear regression function in a high dimensional feature space. The feature space representation can result in a large increase in the number of parameters used by the algorithm. In order to combat this "curse of dimensionality", the algorithm allows the use of kernel functions, as used in Support Vector methods. We also discuss a powerful family of kernel functions which is constructed using the ANOVA decomposition method from the kernel corresponding to splines with an infinite number of nodes. This paper introduces a regression estimation algorithm which is a combination of these two elements: the dual version of Ridge Regression is applied to the ANOVA enhancement of the infinite-node splines. Experimental results are then presented (based on the Boston Housing data set) which indicate the performance of this algorithm relative to other algorithms.

1 INTRODUCTION

First of all, let us formulate regression estimation problem. Suppose we have a set of vectors[1] $x_1, \dots, x_T$, and we also have a supervisor which gives us a real value y_t, for each of the given vectors. Our problem is to construct a learning machine which when given a new set of examples, minimises some measure of discrepancy between its prediction $\hat{y}_t$ and the value of y_t. The measure of loss which we are using, average square loss (L), is defined by

$$L = \frac{1}{l} \sum_{t=1}^{l} (y_t - \hat{y}_t)^2,$$

where y_t are the supervisor's answers, $\hat{y}_t$ are the predicted values, and l is the number of vectors in the test set.

Least Squares and Ridge Regression are classical statistical algorithms which have been known for a long time. They have been widely used, and recently some papers such as Drucker *et al.* [2] have used regression in conjunction with a high dimensional feature space. That is the original input vectors are mapped into some feature space, and the algorithms are then used to construct a linear regression function in the feature space, which represents a non-linear regression in the original input space. There is, however, a problem encountered when using these algorithms within a feature space. Very often we have to deal with a very large number of parameters, and this leads to serious computational difficulties that can be impossible to overcome. In order to combat this "curse of dimensionality" problem, we describe a dual version of the Least Squares and Ridge Regression algorithms, which allows the use of kernel functions. This approach is closely related to Vapnik's kernel method as used in the Support Vector Machine. Kernel functions represent dot products in a feature space, which allows the algorithms to be used in a feature space without having to carry out computations within that space. Kernel functions themselves can take many forms and particular attention is paid to a family of kernel functions which are constructed using ANOVA decomposition (Vapnik [10]; see also Wahba [11, 12]). There are two

[1] We will use subscripts to indicate a particular vector (e.g. x_t is the tth vector), and superscripts to indicate a particular vector element (e.g x^i is the ith element of the vector x).

major objectives of this paper:

1. To show how to use kernel functions to overcome the curse of dimensionality in the above mentioned algorithms.
2. To demonstrate how ANOVA decomposition kernels can be constructed, and evaluate their performance compared to polynomial and spline kernels, on a real world data set.

Results from experiments performed on the well known Boston housing data set are then used to show that the Least Squares and Ridge Regression algorithms perform well in comparison with some other algorithms. The results also show that the ANOVA kernels, which only consider a subset of the input parameters, can improve on results obtained on the same kernel function without the ANOVA technique applied. In the next section we present the dual form of Least Squares and Ridge Regression.

2 RIDGE REGRESSION IN DUAL VARIABLES

Before presenting the algorithms in dual variables, the original formulation of Least Squares and Ridge Regression is stated here for clarity.

Suppose we have a training set $(x_1, y_1), \ldots, (x_T, y_T)$, where T is the number of examples, x_t are vectors in $\mathbb{R}^n$ (n is the number of attributes) and $y_t \in \mathbb{R}$, $t = 1, \ldots, T$. Our comparison class consists of the linear functions $y = w \cdot x$, where $w \in \mathbb{R}^n$.

The Least Squares method recommends computing $w = w_0$ which minimizes

$$L_T(w) = \sum_{t=1}^{T} (y_t - w \cdot x_t)^2$$

and using w_0 for labeling future examples: if a new example has attributes x, the predicted label is $w_0 \cdot x$.

The Ridge Regression procedure is a slight modification on the least squares method and replaces the objective function $L_T(w)$ by

$$a\|w\|^2 + \sum_{t=1}^{T} (y_t - w \cdot x_t)^2,$$

where a is a fixed positive constant.

We now derive a "dual version" for Ridge Regression (RR); since we allow $a = 0$, this includes Least Squares (LS) as a special case. In this derivation we partially follow Vapnik [8]. We start with re-expressing our problem as: minimize the expression

$$a\|w\|^2 + \sum_{t=1}^{T} \xi_t^2 \tag{1}$$

under the constraints

$$y_t - w \cdot x_t = \xi_t, \quad t = 1, \ldots, T. \tag{2}$$

Introducing Lagrange multipliers α_t, $t = 1, \ldots, T$, we can replace our constrained optimization problem by the problem of finding the saddle point of the function

$$a\|w\|^2 + \sum_{t=1}^{T} \xi_t^2 + \sum_{t=1}^{T} \alpha_t \left(y_t - w \cdot x_t - \xi_t \right). \tag{3}$$

In accordance with the Kuhn—Tucker theorem, there exist values of Lagrange multipliers $\alpha = \alpha^{\mathrm{KT}}$ for which the minimum of (3) equals the minimum of (1), under constraints (2). To find the optimal w and ξ, we will do the following; first, minimize (3) in w and ξ and then maximize it in α. Notice that for any fixed values of α the minimum of (3) (in w and ξ) is less than or equal to the value of the optimization problem (1)–(2), and equality is attained when $\alpha = \alpha^{\mathrm{KT}}$. By doing this, we will therefore find the solution to our original constrained minimization problem (1)–(2).

Differentiating (3) in w, we obtain the condition

$$2aw - \sum_{t=1}^{T} \alpha_t x_t = 0,$$

i.e.,

$$w = \frac{1}{2a} \sum_{t=1}^{T} \alpha_t x_t. \tag{4}$$

(Lagrange multipliers are usually interpreted as reflecting the importance of the corresponding constraints, and equation (4) shows that w is proportional to the linear combination of x_t, each of which is taken with a weight proportional to its importance.) Substituting this into (3), we obtain

$$\frac{1}{4a} \sum_{s,t=1}^{T} \alpha_s \alpha_t (x_s \cdot x_t) + \sum_{t=1}^{T} \xi_t^2$$

$$+ \frac{1}{2a} \left(\sum_{t=1}^{T} \alpha_t x_t \right) \cdot \left(- \sum_{t=1}^{T} \alpha_t x_t \right) + \sum_{t=1}^{T} y_t \alpha_t - \sum_{t=1}^{T} \alpha_t \xi_t$$

$$= -\frac{1}{4a}\sum_{s,t=1}^{T}\alpha_s\alpha_t(x_s\cdot x_t)+\sum_{t=1}^{T}\xi_t^2+\sum_{t=1}^{T}y_t\alpha_t-\sum_{t=1}^{T}\alpha_t\xi_t. \tag{5}$$

Differentiating (5) in ξ_t, we obtain

$$\xi_t = \frac{\alpha_t}{2}, \quad t = 1,\dots,T \tag{6}$$

(i.e., the importance of the tth constraint is proportional to the corresponding residual); substitution into (5) gives

$$-\frac{1}{4a}\sum_{s,t=1}^{T}\alpha_s\alpha_t(x_s\cdot x_t)-\frac{1}{4}\sum_{t=1}^{T}\alpha_t^2+\sum_{t=1}^{T}y_t\alpha_t. \tag{7}$$

Denoting K as the $T\times T$ matrix of dot products

$$K_{s,t} = x_s\cdot x_t,$$

and differentiating in α_t, we obtain the condition

$$-\frac{1}{2a}K\alpha-\frac{1}{2}\alpha+y=0,$$

which is equivalent to

$$\alpha = 2a(K+aI)^{-1}y.$$

Recalling (4), we obtain that the prediction y given by the Ridge Regression procedure on the new unlabeled example x is

$$w\cdot x = \left(\frac{1}{2a}\sum_{t=1}^{T}\alpha_t x_t\right)\cdot x = \frac{1}{2a}\alpha\cdot k = y'(K+aI)^{-1}k,$$

where $k=(k_1,\dots,k_T)'$ is the vector of the dot products:

$$k_t := x_t\cdot x,\ \ t=1,\dots,T.$$

Lemma 1 *RR's prediction of the label y of a new unlabeled example x is*

$$y'(K+aI)^{-1}k, \tag{8}$$

where K is the matrix of dot products of the vectors $x_1,\dots,x_T$ in the training set,

$$K_{s,t}=\mathcal{K}(x_s,x_t),\ \ s=1,\dots,T,\ \ t=1,\dots,T,$$

k is the vector of dot products of x and the vectors in the training set,

$$k_t := \mathcal{K}(x_t,x),\ \ t=1,\dots,T,$$

and $\mathcal{K}(x,x')=x\cdot x'$ is simply a function which returns the dot product of the two vectors, x and x'.

3 LINEAR REGRESSION IN FEATURE SPACE

When $\mathcal{K}(x_i,x_j)$ is simply a function which returns the dot product of the given vectors, formula (8) corresponds to performing linear regression within the input space $\mathbb{R}^n$ defined by the examples. If we want to construct a linear regression in some feature space, we first have to choose a mapping from the original space X to a higher dimensional feature space F ($\phi : X\to F$). In order to use Lemma 1 to construct the regression in the feature space, the function $\mathcal{K}$ must now correspond to the dot product $\phi(x_i)\cdot\phi(x_j)$. It is not necessary to know $\phi(x)$ as long as we know $\mathcal{K}(x_i,x_j)=\phi(x_i)\cdot\phi(x_j)$. The question of which functions $\mathcal{K}$ correspond to a dot product in some feature space F is answered by Mercer's theorem and addressed by Vapnik [9] in his discussion of support vector methods. As an illustration of the idea, an example of a simple kernel function is presented here. (See Girosi [4].) Suppose there is a mapping function ϕ which maps a two-dimensional vector into 6 dimensions:

$$\phi : (x^1,x^2)\mapsto((x^1)^2,(x^2)^2,\sqrt{2}x^1,\sqrt{2}x^2,\sqrt{2}x^1x^2,1),$$

then dot products in F take the form

$$\begin{aligned}&(\phi(x)\cdot\phi(y))\\&=(x^1)^2(y^1)^2+(x^2)^2(y^2)^2+2x^1y^1\\&\quad+2x^2y^2+2x^1y^1x^2y^2+1\\&=((x\cdot y)+1)^2.\end{aligned}$$

One possible kernel function is therefore $((x\cdot y)+1)^2$. This can be generalised into a kernel function of the form

$$\mathcal{K}(x,y)=((x\cdot y)+1)^d,$$

and more than 2 dimensions.

The use of kernel functions allows us to construct a linear regression function in a high dimensional feature space (which corresponds to a non-linear regression in the input space) avoiding the curse of having to carry out computations in the high dimensional space. In particular, kernel functions are a way to combat the curse of dimensionality problems such as those faced in Drucker *et al.* [2], where a regression function was also constructed in a feature space, but computations were carried out in the high dimensional space, leading to huge number of parameters for non-trivial problems.

For more information on the kernel technique, see Vapnik [8, 10, 9] and Wahba [11].

4 MULTIPLICATIVE KERNELS

Before indicating how ANOVA decomposition can be used to form kernels, a brief description is needed of the family of kernels to which the ANOVA decomposition can be applied, this being the family of multiplicative kernels. This refers to the set of kernels where the multi-dimensional case is calculated as the product of the one-dimensional case. That is, if the one-dimensional case is $k(x^i, y^i)$, then the n-dimensional case is

$$\mathcal{K}_n(x,y) = \prod_{i=1}^{n} k(x^i, y^i).$$

One such kernel (to which the ANOVA decomposition is applied here) is the spline kernel with an infinite number of nodes (see Vapnik [8, 10] and Kimeldorf and Wahba [5]). A spline approximation which has an infinite number of nodes can be defined on the interval $(0, a)$, $0 < a < \infty$, as the expansion

$$f(x) = \int_0^a a(t)(x-t)_+^d dt + \sum_{i=0}^{d} a_i x^i,$$

where a_i, $i = 0, \ldots, d$, are unknown values, and $a(t)$ is an unknown function which defines the expansion. This can be considered as an inner product, and the kernel which generates splines of dimension d with an infinite number of nodes can be expressed as

$$k_d(x,y) = \int_0^a (x-t)_+^d (y-t)_+^d dt + \sum_{r=0}^{d} x^r y^r.$$

Note that when $t > \min(x,y)$ the function under the integral sign will have value zero. It is therefore sufficient only to consider the interval $(0, \min(x,y))$, which makes the formula above equivalent to

$$k_d(x,y) = \sum_{r=0}^{d} \frac{\begin{pmatrix} d \\ r \end{pmatrix}}{2d-r+1} \min(x,y)^{2d-r+1} |x-y|^r$$

$$+ \sum_{r=0}^{d} x^r y^r.$$

In particular, for the case of linear splines ($d = 1$) we have :

$$k_1(x,y) = 1 + xy + \frac{1}{2}|y-x| \min(x,y)^2 + \frac{\min(x,y)^3}{3}.$$

5 ANOVA DECOMPOSITION KERNELS

The ANOVA decomposition kernels are inspired by their namesake in statistics, which analyses different subsets of variables. The actual decomposition can be adapted to form kernels (as in, e.g., Vapnik [10]) which involve different subsets of the attributes of the examples up to a certain size. There are two main reasons for choosing to use ANOVA decomposition. Firstly, the different subsets which are considered may group together like variables, which can lead to greater predictive power. Also, by only considering some subsets of the input parameters, ANOVA decomposition reduces the VC dimension of the set of functions that you are considering, which can avoid overfitting your training data.

Given a one-dimensional kernel k, the ANOVA kernels are defined as follows:

$$\begin{aligned} \mathcal{K}_1(x,y) &= \sum_{1 \le k \le n} k(x^k, y^k), \\ \mathcal{K}_2(x,y) &= \sum_{1 \le k_1 < k_2 \le n} k(x^{k_1}, y^{k_1}) k(x^{k_2}, y^{k_2}), \\ &\ldots, \\ \mathcal{K}_n(x,y) &= k(x^{k_1}, y^{k_1}) \ldots k(x^{k_n}, y^{k_n}). \end{aligned}$$

From Vapnik [10] the following recurrent procedure can be used when calculating the value of $\mathcal{K}_n(x,y)$. Let

$$\mathcal{K}^s(x,y) = \sum_{i=1}^{n} (k(x^i, y^i))^s$$

and $\mathcal{K}_0(x,y) = 1$; then

$$\mathcal{K}_p(x,y) = \sum_{1 \le k_1 < k_2 < \cdots < k_p \le n} k(x^{k_1}, y^{k_1}) \ldots k(x^{k_p}, y^{k_p}),$$

$$\mathcal{K}_p(x,y) = \frac{1}{p} \sum_{s=1}^{p} (-1)^{s+1} \mathcal{K}_{p-s}(x,y) \mathcal{K}^s(x,y).$$

For the purposes of this paper, when using kernels produced by ANOVA decomposition, only the order p is considered:

$$\mathcal{K}(x,y) = \mathcal{K}_p(x,y).$$

An alternative method of using ANOVA decomposition would be to consider order p and all lower orders (as in Stitson [7]), i.e.,

$$\mathcal{K}(x,y) = \sum_{i=1}^{p} \mathcal{K}_i(x,y).$$

6 EXPERIMENTAL RESULTS

Experiments were conducted on the Boston Housing data set[2]. This is a well known data set for testing non-linear regression methods; see, e.g., Breiman [1] and Saunders [6]. The data set consists of 506 cases in which 12 continuous variables and 1 binary variable determine the median house price in a certain area of Boston in thousands of dollars. The continuous variables represent various values pertaining to different locational, economic and structural features of the house. The prices lie between \$5000 and \$50,000 in units of \$1000. Following the method used by Drucker *et al.* [2], the data set was partitioned into a training set of 401 cases, a validation set of 80 cases and a test set of 25 cases. This partitioning was carried out randomly 100 times, in order to carry out 100 trials on the data. For each trial the Ridge Regression algorithm was applied using:

- a kernel which corresponds to a spline approximation with an infinite number of nodes,
- the same kernel but with the ANOVA decomposition technique applied,
- and polynomial kernels.

For each kernel the set of parameters (the order of spline/degree of polynomial and the value of coefficient a) was selected which gave the smallest error on the validation set, and then the error on the test set was measured. This experiment was then repeated using a support vector machine (SVM), with the same kernels and exactly the same 100 training files (see Stitson [7] for full details). As an illustration of the number of parameters which were considered by the Ridge Regression Algorithm (and the SVM), consider the polynomial kernel which was outlined earlier, using a degree of 5. This maps the input vectors into a high dimensional feature space which is equivalent to evaluating $13^5 = 371,293$ different parameters.

The results obtained from the experiments are shown in Table 1. The measure of error used for the tests was the average squared error. For each of the 100 test files, the algorithm was run and the square of the difference between the predicted and actual value was taken. This was then averaged over the 25 test cases. This produces an average error for each of the 100 test files, and an average of these were taken, which produces the final error which is quoted in the 3rd column of the table. The variance measure in the table is the average squared difference, between the squared error measured on each sample and the average squared error.

There are two additional results which should be noted here. One is from Breiman [1] using bagging with average squared error of 11.7, and one from Drucker *et al.* [2] using Support Vector regression with polynomial kernels with average squared error of 7.2. The result obtained by Drucker et al. is slightly better than the one obtained here using a similar machine; this may be, however, due to the random selection of the training, validation and testing sets.

7 COMPARISONS

In this section we will give a comparison of the results of this paper with the known results.

7.1 SV MACHINES

In this subsection we describe in more detail the connection of the approach of this paper with the Support Vector Machine.

Our optimization problem (minimizing (1) under constraints (2)) is essentially a special case of the following general optimization problem: minimize the expression

$$\frac{1}{2}\|w\|^2 + \frac{C}{k}\left(\sum_{t=1}^{T}(\xi_t^*)^k + \sum_{t=1}^{T}(\xi_t)^k\right) \tag{9}$$

under the constraints

$$y_t - w \cdot x_t \le \epsilon + \xi_t^*, \quad t = 1, \ldots, T, \tag{10}$$

$$w \cdot x_t - y_t \le \epsilon + \xi_t, \quad t = 1, \ldots, T; \tag{11}$$

$\epsilon > 0$ and $k \in \{1, 2\}$ are some constants. This optimization problem (along with a similar problem corresponding to Huber's loss function) is considered in Vapnik [10], Chapter 11 (Vapnik, however, considers more general regression functions of the form $w \cdot x + b$ rather than $w \cdot x$; the difference is minor because we can always add an extra attribute which is always 1 to all examples).

Our problem (1)–(2) corresponds to the problem (9)–(11) with $k = 2$, $\epsilon = 0$ and $C = 1/a$. Vapnik [10] gives a dual statement of his, and *a fortiori* our, problem; he does not reach, however, the closed-form expression (8)

[2] Available by anonymous FTP from:
ftp://ftp.ics.uci.com/pub/
machine-learning-databases/housing.

Table 1: Experimental Results on the Boston Housing Data

METHOD	KERNEL	SQUARED ERROR	VARIANCE
Ridge Regression	Polynomial	10.44	18.34
Ridge Regression	Splines	8.51	11.19
Ridge Regression	ANOVA Splines	7.69	8.27
SVM [7]	Polynomial	8.14	15.13
SVM	Splines	7.87	12.67
SVM	Anova Splines	7.72	9.44

(because he was mainly interested in positive values of ϵ).

As we mentioned before, our derivation of formula (8) follows [8]. The dual Ridge Regression is also known in traditional statistics, but statisticians usually use some clever matrix manipulations rather than the Lagrange method. Our derivation (modelled on Vapnik's) gives some extra insight: see, e.g., equations (4) and (6). For an excellent survey of connections between Support Vector Machine and the work done in statistics we refer the reader to Wahba [11, 12] and Girosi [4].

7.2 KRIEGING

Formula (8) is well known in the theory of Krieging; in this subsection we will explain the connection for readers who are familiar with Krieging. Consider the Bayesian setting where:

- the vector w of weights is distributed according to the normal distribution with mean 0 and covariance matrix $\frac{1}{2a}I$;
- $y_t = w \cdot x_t + \epsilon_t$, $t = 1, \ldots, T$, where ϵ_t are random variables distributed normally with mean 0 and variance $\frac{1}{2}$.

Then the optimization problem (1) under the constraints (2) becomes the problem of finding the posterior mode (which, because of our normality assumption, coincides with the posterior mean) of w; therefore, formula (8) gives the mean value of the random variable $w \cdot x$ (which is the "clean version" of the label $y = w \cdot x + \epsilon$ of the next example). Notice that the random variables $y_1, \ldots, y_T, w \cdot x$ are jointly normal and the covariances between them are

$$\mathrm{cov}(y_s, y_t) = \mathrm{cov}(w \cdot x_s + \epsilon_s, w \cdot x_t + \epsilon_t) = \frac{1}{2a}(x_s \cdot x_t) + \frac{1}{2}$$

and

$$\mathrm{cov}(y_t, w \cdot x) = \mathrm{cov}(w \cdot x_t + \epsilon_t, w \cdot x) = \frac{1}{2a}(x_t \cdot x).$$

In accordance with the Krieging formula the best prediction for $w \cdot x$ will be

$$y' \left(\frac{1}{2a}K + \frac{1}{2}I \right)^{-1} \left(\frac{1}{2a}k \right) = y'(K + aI)^{-1}k,$$

which coincides with (8).

8 CONCLUSIONS

A formula for Ridge Regression (which included Least Squares as a special case) in dual variables was derived using the method of Lagrange multipliers. This was then used to perform linear regression in a feature space. Therefore, we once more showed how the problem of learning in a very high dimensional space can be solved by using kernel functions. This allowed the algorithm to overcome the "curse of dimensionality" and run efficiently, even though a very large number of parameters were being considered. Experimental results show that Ridge Regression performs well. The results also indicate that applying ANOVA decomposition to a kernel can achieve better results than using the same kernel without the technique applied. Both Ridge Regression and the Support Vector method gave a smaller error when using ANOVA splines compared to the other spline kernel.

A weak part of our experimental section is that, though the Boston housing data is a useful benchmark, we have not applied our algorithm to a wider range of practical problems. This is what we plan to do next.

In order to confirm that ANOVA kernels can outperform kernels in their orginal form, the ANOVA decomposition technique should be applied to other multiplicative kernels. The technique of applying kernel functions to overcome problems of high dimensionality should also be investigated futher, to see if it can be applied to any other algorithms which prove computationally difficult or impossible when faced with a large number of parameters.

We feel that a very interesting direction of developing the results of this paper would be to combine the dual version of Ridge Regression with the ideas of Gammerman *et al.* [3] to obtain a measure of confidence for predictions output by our algorithms. We expect that in this case simple closed-form formulas can be obtained.

Acknowledgments

We thank EPSRC for providing financial support through grant GR/L35812 ("Support Vector and Bayesian Learning Algorithms"). The referees' thoughtful comments are gratefully appreciated.

References

[1] L. Breiman. Bagging predictors. Technical Report 421, Department of Statistics, University of California, Berkley, 1994. Also at: ftp://ftp.stat.berkely.edu/ pub/tech-reports/421.ps.Z.

[2] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. N. Vapnik. Support Vector regression machines. In *Advances in Neural Information Processing Systems 9*, volume 9, page 155. The MIT Press, 1996.

[3] A. Gammerman, V. Vapnik, and V. Vovk. Learning by transduction. In *Uncertainty in Artificial Intelligence*, 1998. To appear.

[4] F. Girosi. An equivalence between sparce approximations and Support Vector Machines. Technical Report A. I. Memo No. 1606, C. B. C. L. Paper No. 147, Massachusetts Institute of Technology Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Department of Brain and Cognitive Sciences, May 1997.

[5] G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.*, 33:82–95, 1971.

[6] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression in dual variables. Technical report, Royal Holloway, University of London, 1998.

[7] M. O. Stitson, A. Gammerman, V. N. Vapnik, V. Vovk, C. Watkins, and J. Weston. Support Vector regression with ANOVA decomposition kernels. Technical report, Royal Holloway, University of London, 1997.

[8] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[9] V. N. Vapnik. Statistical learning theory. In A. Gammerman, editor, *Computational Learning and Probabilistic Reasoning.* Wiley, 1996.

[10] V. N. Vapnik. *Statistical Learning Theory.* Wiley, Forthcoming.

[11] G. Wahba. *Spline models for observational data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, 1990.

[12] G. Wahba. Support Vector machines, reproducing kernel Hilbert spaces and the randomized GACV. Technical Report 984, Department of Statistics, University of Wisconsin, USA, 1997.