Kernels for the Relevance Vector Machine - An Empirical Study

David Ben-Shimon and Armin Shmilovici

Dept. of Information Systems Engineering, Ben-Gurion University, P.O.B 653,
84105 Beer-Sheva, Israel {dudibs,armin}@bgumail.bgu.ac.il

Abstract. The Relevance Vector Machine (RVM) is a generalized linear model that can use kernel functions as basis functions. Experiments with the Matérn kernel indicate that the kernel choice has a significant impact on the sparsity of the solution. Furthermore, not every kernel is suitable for the RVM. Our experiments indicate that the Matérn kernel of order 3 is a good initial choice for many types of data.

Keywords: Machine Learning, Relevance Vector Machine, Kernel Regression, Matérn Kernel

1 Introduction

Suppose that N noisy observations of an unknown function $f : R^d \to R$ are available:

$$Y_i = f(X_i) + \varepsilon_i \tag{1}$$

Suppose that f can be expressed in a form of some infinite expansion:

$$f(x) = \sum_{j=1}^{\infty} \theta_j^* g_j(x) \tag{2}$$

where $\{g_j(x)\}_0^\infty$ is an unknown family of basis functions. In the regression problem estimating f reduces to estimation of a suitable truncation of the vector of all parameters $\Theta = (\theta_0^* ... \theta_n^*)^T$, using the observations $\{X_i, Y_i\}_{i=1}^N$ and (2) is called a generalized linear model (GLM). To limit the number of parameters such that the coefficients θ_j^* decrease in a certain way as $j \to \infty$ some smoothness or regularity assumptions have to be stated about f. Generally speaking, smoothness conditions require that the unknown function f belongs to a particular restricted functional class. Otherwise, convergence can be arbitrary slow[1].

One commonly used implementation of (2) is the Parzen-Rosenblatt density estimator defined as:

$$\hat{f}_N(x) = \frac{1}{Nh_N^d} \sum_{i=1}^{N} k\left(\frac{x_i - x}{h_N}\right) \tag{3}$$

where the positive number h_N is called the bandwidth or scaling factor and the function k is called a kernel. A kernel function is a positive definite function [2] which decreases very fast outside the window $[x - h_N, x + h_N]$, thus, the estimator (3) is a moving average of the observations belonging to that window. The accuracy of the approximation (3) depends on how densely observation points fill the input space. Efficient uniform error bounds are available for kernel estimators when the function f is further restricted to the class of functions bounded by a polynomial of (unknown) orders [1]. Noisy observations introduce error in the estimation of the regression coefficients θ_j^*. The total mean square error of the estimates will be the sum of the stochastic part (because of the noise) and of the bias due to the approximation error. Thus, the optimal choice for the regression problem will depend more on the characteristics of the kernel function and less on the characteristics of the unknown f.

The use of kernels has received considerable attention in machine learning [2]. The kernel matrix is symmetric and positive definite matrix, thus, it can be defined as some kind of similarity between pairs of data points such as $k(x, x^{'}) = \left\langle \Psi(x), \Psi(x^{'} \right\rangle$. The transform $\Psi : X \rightarrow H$ from X, the input space to H, is often used to embed the training data into a high dimensional feature space. The assumption is that it could be easier to obtain the solution for a specific problem in the feature space. Using this technique, there is no restriction on the dimensionality of the data, since usually the number of examples N is much larger than the number of dimensions d.

The freedom in the choice of the mapping Ψ enables us to design a large variety of similarity measures adapted to the given problem [2], [10]. In practice, most applications of kernel methods just use the Gaussian kernel $k(x, x_i) = exp^{-||x - x_i||^2/2\sigma^2}$ where the $\|\|$ operator indicates the distance between the any two points, and σ is the width parameter of the Gaussian. The Gaussian kernel is also called the universal kernel. It is an infinitely smooth function, thus, may not be the best choice for noisy datasets.

The Matérn kernel [2], [3] is unique because it has an extra parameter v to explicitly control the smoothness of the kernel. The Matérn function of order v belongs to the class of functions bounded by a polynomial of order v. One formulation of the Matérn kernel takes the following form:

$$M(x, x_i) = \frac{2(\frac{\sqrt{v}}{\sigma}||x - x_i||)^v}{\Gamma(v)} K_v(2\frac{\sqrt{v}}{\sigma}||x - x_i||) \tag{4}$$

where $\Gamma(v)$ is the gamma function and $K_v(x)$ is the modified Bessel function of the second kind[1]of order v, and σ in this case is the width scaling parameter of the Matérn function.

When $v \to \infty$ the Matérn kernel degenerates to the Gaussian kernel and when v=0.5 it degenerates to the exponential kernel . Thus, the Matérn kernel is able to define a wide range of kernel functions. Figure 1 illustrates the Matérn kernel with different degrees of smoothness and its ability to behave as the Gaussian and the exponential kernels.

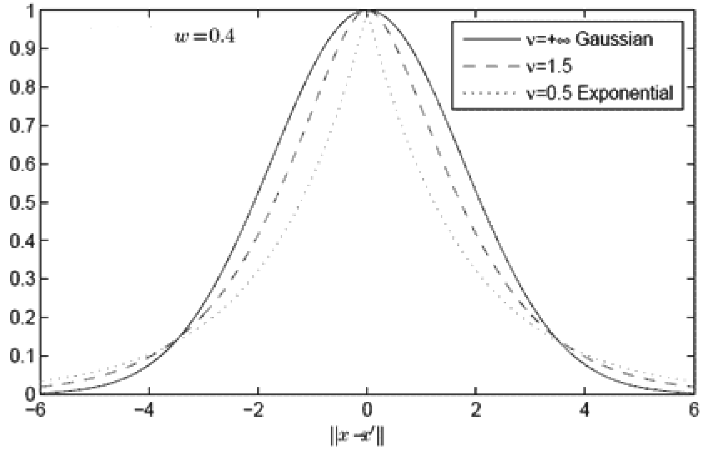

Fig. 1. The Matérn Kernel; w denotes the standard deviation (the width) and v is the smoothness parameter

Sparsity is generally considered a desirable feature of a machine learning algorithm. Sparse algorithms prefer a simple solution. In the context of GLM, as sparse solution will have a small number of non-zero coefficients. The Relevance Vector Machine (RVM) is a method for training a GLM such as (2). In the literature, it was presented as a method for sparse kernel regression.

$$y(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{N} w_i \cdot k(x, x_i) + \epsilon \tag{5}$$

$k(\mathbf{x}, x_i)$ is a bi-variate kernel function centered on each one of the N training data points x_i, $\mathbf{w} = [w_i ... w_N]^T$ is a vector of regression coefficients, and ϵ is the noise. This means that it will select a subset, often a small subset, of the kernel functions in the final model.

Though it is stated that the RVM can use any basis functions [4], the examples in the literature apply only the Gaussian Kernel. The novelty in this

[1] $K_v(z) = \frac{\Gamma(v+\frac{1}{2})(2z)^v}{\sqrt{\pi}} \int_0^\infty \frac{\cos(t)}{(t^2+z^2)^{v+\frac{1}{2}}} dt$

paper is the first investigation of the RVM for kernels other than the Gaussian, and specifically the Matérn kernel. We study for the first time the effect of the smoothness of the kernel function on the convergence and sparseness of the RVM solution for datasets with various attributes such as non linearity and noise.

The rest of this paper is as follows: section 2 introduces the RVM algorithm; section 3 presents experiments with different kernels; and section 4 concludes with a discussion.

2 The RVM Algorithm

2.1 The Regression RVM

Consider a dataset of input-target pairs $\{X_i, t_i\}_{i=1}^{N}$. Each target t_i is assumed Normally distributed with mean $y(x_i)$ and uniform variance σ^2 of the noise ϵ so $p(\mathbf{t}|\mathbf{x}) = N(t|y(\mathbf{x}), \sigma^2)$. The targets are also assumed *jointly* Normal distributed as $N(\mu, \Sigma)$, where (μ, Σ) are the unknowns to be determined by the algorithm. The conditional probability of the targets given the parameters and the data can now be expressed as (6).

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{\frac{N}{2}} exp\left\{-\frac{1}{2\sigma^2}||\mathbf{t} - \mathbf{\Phi w}||\right\} \tag{6}$$

where the data is hidden in the NxN kernel function matrix $\mathbf{\Phi}$ representing all the pairs $\Phi_{i,j} = k(x_i, x_j), i, j \in [1...N]$.($\mathbf{\Phi}$ could be extended to include a possible bias term).

The goal of the RVM is to accurately predict the target function, while retaining as few basis functions as possible in (5). Sparseness is achieved via the framework of sparse Bayesian learning and the introduction of an additional vector of hyper parameters α_i that controls the width of a Normal prior distribution over the precision of each element of w_i.

$$p(w_i|\alpha_i) = \sqrt{\frac{\alpha_i}{2\pi}} exp(1 - \frac{1}{2}\alpha_j w_j^2) \tag{7}$$

A large parameter α_i indicates a prior distribution sharply peaked around zero. For a sufficiently large α_i, the basis function is deemed irrelevant and w_i is set to zero, maximizing the posterior probability of the parameters' model (7). As an analogy to the Support Vector Machine [5], the non-zero elements of $\mathbf{w}$ are called Relevance Values, and their corresponding data-points are called Relevance Vectors (RVs).

The solution is derived via the following iterative type II maximization of the marginal likelihood $p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)$ with respect to $\boldsymbol{\alpha}$ and σ^2.

$$\alpha_i^{new} = \frac{1 - \alpha_i \Sigma_{ii}}{\mu_i^2} \tag{8}$$

$$(\sigma^2)^{new} = \frac{||\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu}||^2}{N - \Sigma_{i=1}^{N}(1 - \alpha_i \Sigma_{ii})} \tag{9}$$

The unknowns $(\boldsymbol{\mu}, \mathbf{\Sigma})$ are computed as:

$$\mathbf{\Sigma} = (\mathbf{\Phi^T B \Phi} + \mathbf{A})^{-1} \tag{10}$$

$$\boldsymbol{\mu} = \mathbf{\Sigma \Phi^T B t} \tag{11}$$

where $\mathbf{B} \equiv \sigma^{-2}\mathbf{I}_{NxN}$. The basic RVM algorithm cycles between (8),(9),(10),(11) reducing the dimensionality of the problem when any α_i larger than a preset threshold. The algorithm stops when the likelihood $p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)$ ceases to increase. Further information about the algorithm, as well as priors for $(\boldsymbol{\alpha}, \sigma^2)$ is presented in [4].

2.2 The Classification RVM

In the classification problem each target t_i is Binary: $t_i \in \{0, 1\}$. The model (5) is assumed to be noise-free. That is $\sigma^2 \equiv 0$. Note that equation (5) can not produce a binary function by itself without an additional rounding to the closest value $\{0, 1\}$. The sigmoid function $\rho(y) = 1/(1 + e^{-y})$ is used to generalize the linear model. The main idea of the sigmoid function is to make an approximation of the regression case to the two-class classification problem. With the sigmoid link function we can adopt the Bernoulli distribution $P(t|x)$ and rewrite the likelihood as:

$$p(t|w) = \prod_{i=1}^{N} \rho\left(\Phi_i^T w\right)^{t_i} \left(1 - \rho(\Phi_i^T w)\right)^{1-t_i} \tag{12}$$

In the classification RVM framework, we need to find two solutions to two different coupled problems, an optimization problem and a regression RVM problem [4]. The multi-class problem is solved with an assembly of binary classifiers. The classification RVM will not be considered in this paper

2.3 Attributes of the RVM

The RVM is an approximate Bayesian method, thus it can generate not only a predicted values, but also the probability distribution of the values [6]. For further details of the basic RVM look in [4]. For a discussion of convergence and sparseness look in [7].

The matrix inversion operation in (10), which requires $O(N^3)$ operations is the computationally intensive part of the algorithm. The matrices $\mathbf{\Phi}$ and $\mathbf{\Sigma}$ are full rank, thus require initially $O(N^2)$ space complexity. Furthermore, it is common that the inversion of a large matrix becomes ill-conditioned after several cycles even for positive definite matrices unless the parameters of the kernel function are optimized. These problems limit the practicality of the basic RVM algorithm for moderately sized problems. Fortunately, practical approaches were developed for reducing the runtime complexity to $O(N^2)$ [8].

An important step in GLM learning is to find a feature space - a projection of the data on highly dimensional space - where the data is linear for regression problems and linearly separable for classification problems. The choice of projection (the kernel function) is important for the accuracy and the convergence of the RVM. Note that the RVM typically produces very sparse solutions compared to the SVM, when its kernel is as the SVM kernel [4].

3 Comparative Experiments

3.1 The Matérn Kernel

The purpose of the following experiments is to check the sensitivity of the RVM to the kernel choice, the smoothness of the kernel function and various attributes of the dataset. In the Matérn kernel it is possible to control the smoothness of the kernel function. Thus, the following types of the kernel functions were considered: Gaussian kernel, Matérn of orders $v = 1, 2, 3, 4$ (respectively Matérn1, Matérn2, Matérn3 and Matérn4). Higher orders Matérn are not that different than the Gaussian. Moreover, we also test the finitely supported kernel function.

Instead of taking a set of unrelated benchmark data sets, we used the *Pumadyn* family of datasets[2]. These are eight synthetic datasets generated from a realistic simulation of the dynamics of the Puma 560 robotic manipulator. The regression problem is to predict the angular acceleration of one of the robotic links. Each dataset has a unique combination of three attributes: dimensionality (8 or 32 attributes), non-linearity (fairly linear or non-linear), and output noise (moderate or high). Table 1 present the details about the datasets and the split between the training set and the test set. We selected this specific set of benchmark datasets in order to study for the first time the dependency of the RVM solution on both the selected kernel and the attributes of the dataset (such as non-linearity).

We used a MATLAB implementation of the working set RVM from [8]. The Training set was used for the learning phase, and the error was measured on the testing set. Each kernel was simulated for 10 different repetitions and the same randomizations were used for testing each kernel. The width parameter for each kernel was optimized manually via cross-validation experiments on the training set. Table 2 presents the width parameters found for each combination of kernel and dataset.

[2] www.cs.toronto.edu/ delve/data/pumadyn/desc.htmwww.cs.toronto.edu/ delve /data/pumadyn/desc.html

Table 1. Details of the Pumadyn family of datasets

Name	Size	# of Attributes	Level of noise	Level of non linearity	Training set/test set
8fh	8192	8	high	fairly linear	6144/2048
8fm	8192	8	moderate	fairly linear	6144/2048
8nh	8192	8	high	non-linear	6144/2048
8nm	8192	8	moderate	non-linear	6144/2048
32fh	8192	32	high	fairly linear	6144/2048
32fm	8192	32	moderate	fairly linear	6144/2048
32nh	8192	32	high	non-linear	6144/2048
32nm	8192	32	moderate	non-linear	6144/2048

Table 2. The selected width parameters

Name	Matérn1	Matérn2	Matérn3	Matérn4	Gauss
8fh	3	4	8	18	2
8fm	5	4	8	25	2
8nh	7	8	16	25	2
8nm	12	14	18	32	10
32fh	50	50	50	90	25
32fm	20	25	125	175	25
32nh	180	190	135	135	135
32nm	135	140	150	160	60

We used two measures in these experiments, the number of RVs and the accuracy (RMSE). Tables 3 and 4 presents the comparative RMSE and the comparative number of RVs respectively. The standard deviation of each measure is also presented in the tables.

Table 3. Comparative RMSE

Name	Matérn1	Matérn2	Matérn3	Matérn4	Gauss
8fh	3.14±0.04	3.14±0.05	3.16±0.06	3.17±0.04	3.14±0.05
8fm	1.07±0.02	1.05±0.02	1.05±0.02	1.23±0.03	1.05±0.02
8nh	3.24±0.06	3.22±0.04	3.25±0.03	4.25±0.06	3.25±0.04
8nm	1.18±0.02	1.14±0.02	1.20±0.02	3.54±0.03	1.22±0.02
32fh	$.021 \pm 3*10^{-4}$	$.02 \pm 3*10^{-4}$	$.02 \pm 3*10^{-4}$	$.02 \pm 3*10^{-5}$	$.02 \pm 3*10^{-4}$
32fm	$.005 \pm 7*10^{-5}$	$.005 \pm 7*10^{-5}$	$.005 \pm 9*10^{-5}$	$.005 \pm 1*10^{-5}$	$.005 \pm 6*10^{-5}$
32nh	$.034 \pm 7*10^{-4}$	$.033 \pm 5*10^{-4}$	$.033 \pm 4*10^{-4}$	$.033 \pm 3*10^{-5}$	$.033 \pm 5*10^{-4}$
32nm	$.028 \pm 5*10^{-4}$	$.027 \pm 5*10^{-4}$	$.027 \pm 3*10^{-4}$	$.027 \pm 5*10^{-5}$	$.027 \pm 5*10^{-4}$

Table 4. Comparative number of RVs

Name	Matérn1	Matérn2	Matérn3	Matérn4	Gauss
8fh	46.7±5.3	36.5±2.5	34.2±1.7	12.3±1.4	38.5±2.9
8fm	136.1±10.3	79.8±3.2	57.2±2.6	16.1±1	70.6±3.1
8nh	160.7±8.5	93±2.47	45.9±5.8	19.3±1.4	142.4±8
8nm	530.4±85.4	173.3±6	82.8±3.9	17.7±1.3	82.8±2.5
32fh	240.8±37.1	41±11	38.6±3.4	11.3±2.5	79.1±3.7
32fm	265.1±14.3	64±7	34.5±4.3	17±4.6	140±13.4
32nh	268.3±19.5	28±21.5	7.3±1.6	7.3±1.4	9.9±1.7
32nm	324±10	31.1±12.5	16.9±8.6	8.8±2.6	67.7±5.2

Analysis of the results in tables 2, 3, 4 indicates that:

- The Gaussian and Matérn of orders 1,2,3,4 achieved a similar accuracy. Thus, from an accuracy point of view there is no difference among them. The Matérn4 demonstrates a significantly lower accuracy for three of the datasets.
- The Matérn4 achieved the sparsest results (less than 0.3%) for all the data sets. The Matérn3 is also typically sparser than the Gaussian, however, it has a similar accuracy as the Gaussian. It seems that the Matérn4 presents a danger of under-fitting the data.
- Regarding which kernel is more suitable for a given problem (non-linearity, noise) it is hard to decide. While the Matérn4 obtained the sparsest results for all the datasets, the Matérn3 also retained the accuracy, thus, it is recommended.

The main contribution of these experiments is that we found a kernel which is better than the Gaussian - we suggest using the Matérn of order 3.

We can also analyze for the first time the sensitivity of the RVM to different attributes of the dataset, and as expected, a decrease in the noise level (e.g. from puma8fh to puma8fm) results in an increase in the number of RVs, since less features of the function are now masked by the noise.

3.2 The Finitely Supported Matérn Kernel

The typical kernel measures the distance/similarity between any two points in the data. In a finitely supported kernel, we set the corresponding value in the kernel ma-trix to zero whenever the similarity between two points x_i and x_j is below a certain threshold. In a typical dataset, data is distributed among separate clusters in the multidimensional space. Effectively, a data is similar only to data in the same cluster, and will not be considered similar to data from different clusters. Thus, a finitely supported kernel matrix is expected to containing a majority of zero values (a sparse matrix). The advantage in a sparse matrix is that there exist efficient sparse linear algebra and sparse matrix computation techniques [9] that reorder the nonzero values to be around the diagonal of the kernel matrix and invert a sparse matrix in $O(N^2)$- effectively accelerating the RVM.

Simply truncating the kernel below a certain threshold does not result in a positive definite matrix in general. However, any kernel can easily become a compactly supported kernel by multiplying it with the "hyper-triangular" kernel (13).

$$max\left\{\left(1-\frac{||x_i-x_j||}{\sigma}\right)^v,0\right\} \tag{13}$$

where $\sigma > 0$ is a width parameter (same as the one used in the regular Matérn kernel) and $v \geq (d+1)/2$ in order to insure positive definiteness (d is the dimensionality of the data which generated the kernel matrix).

For the experiments with the finitely supported Matérn kernel, we used only the first four Pumadyn datasets from table 1 which have $d = 8$, thus we choose $v = 5$ (for a higher smoothness than that, the Matérn is quite similar to the Gaussian). We used the same experimental setting as before. Table 5 presents the results of the experiments with the order 5 finitely supported hyper-triangular kernel. The finitely supported Matérn kernel was generated by the multiplication of (13) with (4). Unfortunately, for three of the datasets we failed to find an appropriate kernel width that leads to convergence of the RVM. Maybe the high noise and relative linearity of the first dataset facilitates finding a single appropriate width parameter.

Table 5. Experiments with finitely supported kernels

Name	Matérn5fs				Triangular5			
	width	Time	RV	Rmse	width	Time	RV	Rmse
8fh	50	394± 140	128.4± 13.8	3.17± 0.04	45	363± 160	129± 20.4	3.2± 0.05
8fm		no convergence				no convergence		
8nh		no convergence				no convergence		
8nm		no convergence				no convergence		

Comparing the results in table 5 to the results in tables 2, 3, 4, we see that:

- The hyper-triangular kernel of order 5 behaved fairly similar to the finitely supported Matérn kernel of order 5. When the effective width of the Matérn is larger than the effective width of the Triangular kernel, this could be expected, since in this case the Matérn will have a fairly constant value within the effective support of the Triangular kernel, and the multiplication of the kernels will not differ much from the values of the Triangular kernel.

- The accuracy of the finitely supported kernels is fairly similar to that of the regular kernels, while the number of RVs is much larger.
- A careful analysis of the two kernels that did converge, indicate 0% sparsity for the width parameters selected.

While these experiments are not conclusive, it indicates that finite supported kernels are not a good choice for the RVM - the algorithm does not converge for sparse kernels.

4 Discussion

The problem of selecting the best kernel and tuning its parameters lies in the core of all the kernel based methods. In this paper we investigated for the first time the sensitivity of the RVM to the kernel choice. We found that Matérn of order 3 provides a fair trade-off between sparsity and error. Considering that cubic splines (splines of order 3) are well known to provide good approximations for many types of functions, this is not surprising.

It turns out - unlike conjectured by [4] - that not every kernel is suitable for the RVM. Surprisingly, when we trained the RVM using the finitely supported Matérn kernels, the results were very poor convergence if any, and a very long training time relatively to the ordinary Matérn kernels. One possible explanation to the poor results could be the existence of regions of zero derivative of the finitely support kernels which causes the RVM to slow down, or stop its convergence. This phenomena merits further investigation, since sparse linear algebra can potentially accelerate the RVM.

We did not manage to answer the question if there is a kernel that is best suited for a given problem such as non-linearity, noise or whether kernels for classification should be different than kernels for regression. However, this is the first time (to the best of our knowledge) that a kernel different than the Gaussian, or specifically the Matérn family, was ever used for the RVM.

References

1. A. Juditsky, H. Hjalmarsson, A. Beneviste, B. Delyon, L. Ljung, J. Sjoberg, Q. Zhang (1995), Nonlinear Black-box Models in System Identification: Mathematical Foundations, Automatica, 31(12), pp. 1725-1750.
2. M.G.Genton (2001). Classes of Kernels for Machine Learning: A Statistics Perspective. Journal of Machine Learning Research 2, pages 299-312.
3. B. Matérn (1960). Spatial Variation. New York, Springer.
4. M.E. Tipping (2001). Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research 1, 211-244.

5. A. Shmilovici (2005). Support Vector Machines. In O. Maimon and L. Rokach (editors), Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Springer.
6. C.E. Rasmussen, J. Quinonero-Candela, (2005). Healing the Relevance Vector Machine through Augmentation. Proceedings of the 22nd International Conference on Machine Learning, August 7-11, Bonn, Germany.
7. D. Wipf, J. Palmer, B. Rao (2004). Perspectives on Sparse Bayesian Learning. Advances in Neural Information Processing systems, 16. Cambridge, Massachussettes, MIT Press.
8. D. Ben-Shimon, A. Shmilovici (2006). Accelerating the Relevance Vector Machine via Data Partitioning, Journal of Computing and Decision Sciences, forthcoming.
9. J.R.Gilbert, C. Moler, R. Schreiber (1992). Sparse matrices in MATLAB design and implementation. SIAM Journal on Matrix Analysis, 13(1), pages 333-356.
10. N. Cristianini, J. Shawe-Taylor (2003). An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press.