

# MDL Based Model Selection for Relevance Vector Regression

Davide Anguita, and Matteo Gagliolo  
Dept. of Biophysical and Electronic Engineering  
University of Genova, Via Opera Pia 11a  
I-16145 Genova, Italy  
{anguita,gagliolo}@dibe.unige.it

## Abstract

Relevance Vector regression is a form of Support Vector regression, recently proposed by M.E.Tipping, which allows a sparse representation of the data. The Bayesian learning algorithm proposed by the author leaves the partially open question of how to automatically choose the optimal model.

In this paper we describe a model selection criterion inspired by the Minimum Description Length (MDL) principle. We show that our proposal is effective in finding the optimal kernel parameter both on an artificial dataset and a real-world application.

## 1 Introduction

In a *kernel model*, regression estimate at a value  $x$  is given by:

$$\hat{y}(x) = w_0 + \sum_i w_i K(x, x_i; \rho_k) \quad (1)$$

that is, a weighted sum of the kernel function  $K$ , with parameter  $\rho_k$ , and centered in  $x_i$ ,  $i = 1 \dots n$ .

Given a training set, represented by a set of points  $\{x_i\}$  and target values  $\{t_i\}$ , we may want to know which is the best interpretation of the set in terms of the family of models (1). This problem can be posed at different levels of abstraction. Namely, we may want to choose:

- a) a value of  $\{w_i\}$  for a given kernel  $K$  with fixed parameter  $\rho_k$ .
- b) a value for the kernel parameter  $\rho_k$ .
- c) a kernel function  $K$ .

The Minimum Description Length (MDL) principle [3, 2] can be applied to any of the problems above. This principle consists in picking, from a family of models for a data set, the one that gives the shortest description of the data.

In the present article we will show how MDL can be used to select the optimal kernel parameter (problem *b*) when the Relevance Vector kernel machine (RVM) [4, 1] is used to optimize  $\{w_i\}$  (problem *a*). The same framework presented here can be adopted to select between different kernel functions (problem *c*)

## 2 MDL Based Model Selection

### 2.1 The MDL Principle

Given a data vector  $\mathbf{t}$ , of length  $n$ , with elements discretized with precision  $\delta_t$ , we want to encode it using a parameterized model  $p(\mathbf{t} | \boldsymbol{\theta})$ ,  $p$  being a density function and  $\boldsymbol{\theta}$  a parameter with  $k$  elements discretized with precision  $\delta_\theta$ . The code length needed to describe the data will be at least [3, par. 3.1]:

$$L(\mathbf{t}) = L(\mathbf{t} | \boldsymbol{\theta}) + L(\boldsymbol{\theta}) \quad (2)$$

where  $L(\mathbf{t} | \boldsymbol{\theta}) = -\log p(\mathbf{t} | \boldsymbol{\theta}) - n \log \delta_t$  is the ideal code length paid to encode  $\mathbf{t}$ , and  $L(\boldsymbol{\theta}) = -k \log \delta_\theta$  “nats” are needed to encode the  $k$  elements of  $\boldsymbol{\theta}$ .

At this point, we can introduce a prior  $p(\boldsymbol{\theta})$  on the value of the parameter  $\boldsymbol{\theta}$ , as is usual in Bayesian methods; the difference here is that the prior does not reflect any *a priori* knowledge about  $\boldsymbol{\theta}$ , it is just a part of the coding scheme [3, pp. 54, 67, 84]. Thus, the second term in (2) becomes:

$$L(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta}) - k \log \delta_\theta \quad (3)$$

The “two-stage” implementation [2] of the MDL principle consists in choosing, for the data  $\mathbf{t}$ , the model  $p(\mathbf{t} | \boldsymbol{\theta})$  indexed by the value of  $\boldsymbol{\theta}$  which minimizes the data length (2).

### 2.2 Relevance Vector Machines

In Relevance Vector regression [4, 1], the value of  $\mathbf{w}$  in (1) is obtained using a *type-II Maximum Likelihood* method, consisting in the introduction of Bayesian prior probabilities of the form:

$$p(w_i | \alpha_i) = \mathcal{N}(w_i | 0, \alpha_i^{-1}) \quad (4)$$

and consequent optimization of a vector of hyperparameters  $\boldsymbol{\alpha}$ ; for each  $\alpha_i$ , a corresponding hyperprior  $p(\alpha_i)$  is chosen uniform over a logarithmic scale. The errors on the training set  $\mathbf{t}$  are modeled as additive Gaussian noise:

$$p(t_i | \hat{y}(x_i)) = \mathcal{N}(t_i | \hat{y}(x_i), \sigma_N^2) \quad (5)$$

where  $\hat{\mathbf{y}}(\mathbf{x}) = \boldsymbol{\Phi} \mathbf{w}$  is the output of the RVM,  $\boldsymbol{\Phi}$  being the design matrix, and  $\sigma_N^2$  represents an automatic estimate of noise variance, whose value is optimized as an additional hyperparameter.

The value of the parameters ( $\mathbf{w}$ ,  $\alpha$ ,  $\sigma_N^2$ ) can be found using Tipping’s algorithm [1, 4]. Obviously, the obtained values change depending on the choice of the kernel parameter  $\rho_k$ , and the machine outputs different estimates  $\hat{y}(x_i)$ , thus giving different interpretations of training data. We suggest the use of MDL to select among them. Hence we shall evaluate the data length (2) for the RVM, and then choose the value of  $\rho_k$  for which a minimum of this quantity is achieved.

The use of a different prior for each parameter does not fit well within the MDL framework, in which the introduced hyperparameters should have a small impact on the data length (2) [2]. Nonetheless, we can use the quantity (2) to compare the compression of information achieved by the RVM using different kernels, or kernel parameters, in a relative manner, thus obtaining a quantitative criterion for model selection, even if absolute performance may be poor. We must also note that, during learning, many  $\alpha_i$  overflow the machine precision, so the corresponding weights can be “pruned” [1] by setting them to 0, and do not need to be transmitted nor taken into account in (2).

### 2.3 Data Length Evaluation

We have to choose between two possible ways of describing the data in the training set using the RVM parameters: in fact we can see the output  $\hat{\mathbf{y}}(\mathbf{x})$  as a function of  $\mathbf{w}$ , thus using  $\boldsymbol{\theta} \equiv \mathbf{w}$  in (2), or we can integrate out the weights and use  $\alpha$  instead ( $\boldsymbol{\theta} \equiv \alpha$ ). In the first case we obtain:

$$p(\mathbf{t}|\mathbf{w}) = \mathcal{N}(\mathbf{t}|\Phi\mathbf{w}, \sigma_N^2) = (2\pi\sigma_N^2)^{-n/2} \exp\{-\|\mathbf{t} - \Phi\mathbf{w}\|^2/2\sigma_N^2\} \quad (6)$$

$$p(\mathbf{w}) = \prod_i \int p(w_i | \alpha_i) p(\alpha_i) d\alpha_i \quad (7)$$

With these quantities, the two terms in Eq. (2) become respectively:

$$L(\mathbf{t}|\mathbf{w}) = -\log p(\mathbf{t}|\mathbf{w}) - n \log \delta_t = \frac{n}{2} \log(2\pi\sigma_N^2) + \frac{\|\mathbf{t} - \Phi\mathbf{w}\|^2}{2\sigma^2} - n \log \delta_t$$

$$L(\mathbf{w}) = -\log p(\mathbf{w}) - k \log \delta_w = \sum_i \log |w_i| - k \log \delta_w + O(1) \quad (9)$$

the second term obtained evaluating the integrals in (7) over a logarithmic scale on  $\alpha_i$  [1]. Omitting the constant terms we can write:

$$L(\mathbf{t}) = \frac{n}{2} \log(\sigma_N^2) + \frac{\|\mathbf{t} - \Phi\mathbf{w}\|^2}{2\sigma_N^2} + \sum_i \log |w_i| - k \log \delta_w + L(\gamma) \quad (10)$$

where  $k$  is the number of relevant weights,  $1 < k < k_0$ , and  $L(\gamma)$  is the number of nats needed to transmit *which* of the initial  $k_0$  weights are being used. We will label the quantity (10) *MDL-w*.

The same can be done choosing to transmit  $\alpha$  instead of  $\mathbf{w}$ , whose value, and hence the output  $\hat{\mathbf{y}}(\mathbf{x})$ , can be easily reconstructed. Also in this case we

use the logarithmic scale on  $\alpha_i$  [1]. Combination of (4) and (5) yields [1]:

$$p(\mathbf{t}|\boldsymbol{\alpha}) = \mathcal{N}(\mathbf{t}|0, \mathbf{C}) = (2\pi)^{-n/2} |\mathbf{C}|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{t}^T \mathbf{C}^{-1} \mathbf{t}\right\} \quad (11)$$

where  $\mathbf{C} = \sigma_N^2 \mathbf{I}_n + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T$  and  $\mathbf{A} = \text{diag}(\boldsymbol{\alpha})$ . Then:

$$\begin{aligned} L(\mathbf{t}|\boldsymbol{\alpha}) = -\log p(\mathbf{t}|\boldsymbol{\alpha}) - n \log \delta_t &= \frac{1}{2} \log[(2\pi)^n |\mathbf{C}|] + \frac{1}{2} \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t} - n \log \delta_t \\ L(\boldsymbol{\alpha}) &= -k \log \delta_e \end{aligned} \quad (12)$$

where  $\delta_e$  is the precision used to describe each  $\log \alpha_i$ , and we eventually obtain the formula for *MDL- $\alpha$* :

$$L(\mathbf{t}) = \frac{1}{2} \log |\mathbf{C}| + \frac{1}{2} \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t} - k \log \delta_e + L(\boldsymbol{\gamma}) \quad (14)$$

In both methods only relevant parameters are transmitted, so the “relevance flags”  $\boldsymbol{\gamma}$  are needed to reconstruct the output. These can be transmitted in various ways: as a bit string of length  $k_0$ , which can be compressed using an adaptive Bernoulli model, with  $p(1) = k/k_0$  (this yields to  $L(\boldsymbol{\gamma}) = \log k_0 - k \log \frac{k}{k_0} - (k_0 - k) \log(1 - \frac{k}{k_0})$  [2]); or as a sequence of  $k + 1$  natural numbers from the interval  $[1, k_0]$  ( $L(\boldsymbol{\gamma}) = (k + 1) \log k_0$  [2]). The first method yields better compression, with  $L(\boldsymbol{\gamma})$  always bounded by  $k_0$ , but the obtained values are symmetric around  $k_0/2$ , so large models are not penalized: this is why we adopted the second method, which gives similar results for small values of  $k$  but exhibits a linear growth, thus penalizing non-sparse models. A more rigorous approach would have required the use of a Bernoulli model with fixed  $p$ , whose value should also have been optimized to minimize the resulting data length [2]. The quantities  $\sigma_N^2$  and  $\rho_k$  should also be transmitted, but their contribution to code length is negligible and can be considered constant.

### 3 Experiments

To validate the proposed method, a Relevance Vector Machine with Gaussian kernel  $K(x_i, x_j, \rho_k) = \exp\{-\|x_i - x_j\|^2 / 2\rho_k^2\}$  has been trained on different data sets, using a logarithmic grid of values for its parameter ( $\rho_k$ ), and the MDL quantities described above were evaluated on another (coarser) logarithmic grid of values for  $\delta_w$  and  $\delta_e$ . We present two examples here, an artificial problem and a real-world application.

The first consists of 200 equally spaced samples of the function  $f(x) = \cos(x^2)$  on the interval  $[-3, 3]$ , with added uniform  $[-0.1, 0.1]$  noise. A good choice of the parameter is crucial here, as small values would overfit the noise, while large values would cut the high frequency ends. The real data set represents the flux of vehicles on a highway, measured at a constant rate during a whole day, for a total of 720 samples.

We must remark that the MDL indicators (10) and (14) may assume negative values, which may sound strange for a length measure; this is because they are

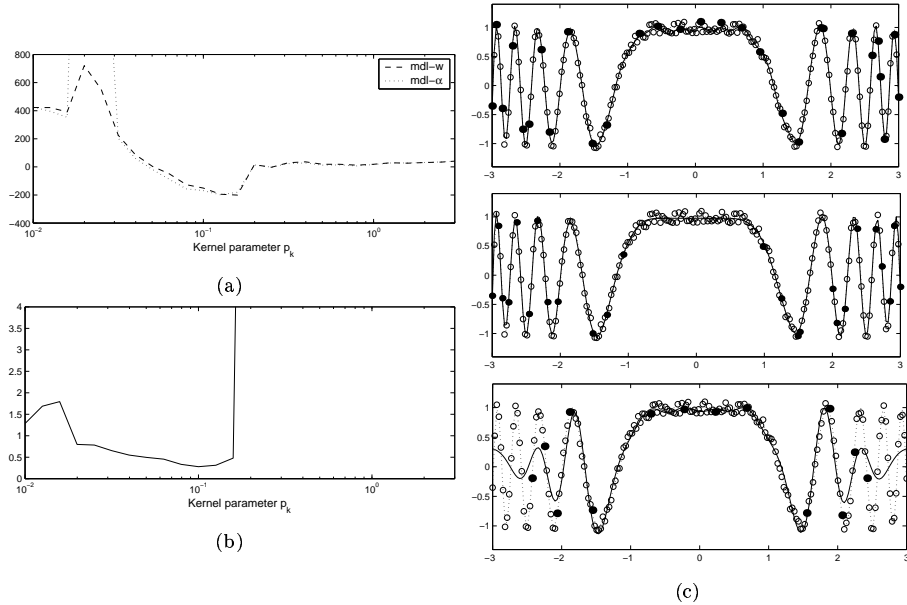


Figure 1:  $\cos(x^2)$  with added uniform noise. a) MDL criteria: MDL- $w$ ,  $\delta_w = 0.1$  (slashed line) and MDL- $\alpha$ ,  $\delta_\alpha = 1$  (dotted line). b) Squared error on noiseless set. c) RVM output for  $\rho_k = 10^{-0.9}$  (up),  $\rho_k = 10^{-0.8}$  (middle),  $\rho_k = 10^{-0.7}$  (low); circles are training set points, black circles are RVs, dotted line is the noiseless  $\cos(x^2)$  function.

evaluated omitting all constant terms. This is not an issue for our purposes because we are interested in the *location* of the minima and not in the overall compression performance [3, p. 56].

In the first experiment (Fig. 1), both MDL criteria select a single value,  $\rho_k = 10^{-0.8}$ , which allows the RVM to track the narrow waves without overfitting the noise and corresponds, with good accuracy, to the minimum error with respect to the noiseless original function. The result is obtained using only 25 Relevance Vectors (RVs) out of the original 200 samples. Note that the RVs are placed at the extremes of the interval, while the bias alone is used in the flat zone around the origin (Fig. 1.c, middle).

We have depicted the behavior of the RVM for the two adjacent values of the parameter  $\rho_k$ , for comparison. As expected, at a smaller value ( $\rho_k = 10^{-0.9}$ ) the RVM overfits the noise and the number of RVs increases, while at  $\rho_k = 10^{-0.7}$  or greater we obtain a sparser model (15 RVs), but the data is underfitted.

With the traffic example (Fig. 2), the MDL curves present two local minima: the lowest one (Fig. 2.b) is located at  $\rho_k \approx 50$  (11 RVs), and the second one (Fig. 2.c) at a slightly higher value, in  $\rho_k \approx 200$  (5 RVs). The interesting feature of this result is that both outputs provide a valuable interpretation of the physical phenomena underlying the data: the first one allows to spot

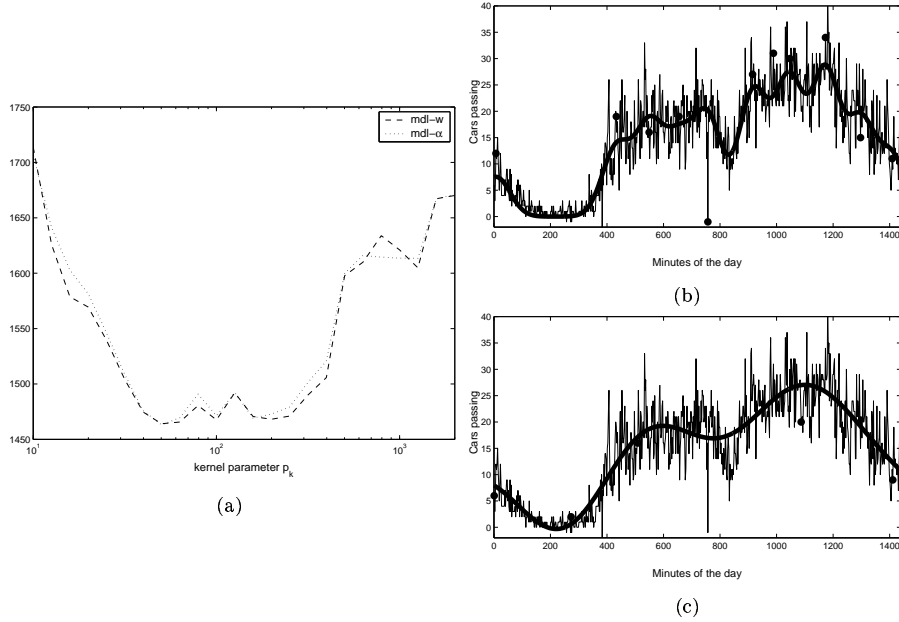


Figure 2: Traffic data. a) MDL criteria: MDL- $w$ ,  $\delta_w = 1$  (slashed line) and MDL- $\alpha$ ,  $\delta_\alpha = 1$  (dotted line). b) Best fit (thick line) of data (thin line):  $\rho_k = 10^{1.7}$  (black circles are RVs). c) Sub-optimal fit:  $\rho_k = 10^{2.3}$ .

temporary variations in traffic flow, so is useful for accident detection, while the second one gives an overall idea of the main traffic fluxes during the day.

In these and other experiments, not presented here due to space constraints, different choices of discretization steps  $\delta_w$  and  $\delta_\alpha$  did not affect the final outcome of the model selection, resulting just in small vertical shifts of equally shaped curves. In addition, both criteria produced similar curves and selected the same value, or adjacent ones. It must be also noted that MDL- $\alpha$  in some occasions did attribute relevance to a smaller number of vectors, giving coarser results.

## 4 Conclusions

Both implementations of the MDL criterion, as presented in this work, do perform well, achieving the minimum of the test error, when available, while maintaining a strong preference for sparse models. A major drawback of the presented method is, obviously, the large computational requirement. The “brute force” technique adopted, which requires the exhaustive training of the RVM on the entire grid of different values for the parameter, could be replaced by heuristic methods. These may allow the extension of this method to multi-parameter kernels.

## References

- [1] Tipping, M.E.: Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 1 (2001) 211–244.
- [2] Hansen, M., Yu, B.: Model Selection and the Principle of Minimum Description Length. Technical Memorandum, Bell Labs, Murray Hill, N.J. (1998), <http://cm.bell-labs.com/who/cocteau/papers/index.html>
- [3] Rissanen, J.: *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co., Singapore (1989).
- [4] Tipping, M.E.: The relevance vector machine. *Advances in Neural Information Processing Systems 12*, The MIT Press (2000) 652–658.