On Kernel Selection in Relevance Vector Machines Using Stability Principle

Kropotov Dmitry DKropotov@yandex.ru

Ptashko Nikita ptashko@inbox.ru

Vasiliev Oleg ovasiliev@inbox.ru

Vetrov Dmitry VetrovD@yandex.ru

Moscow State University, Russia, 119234, Moscow, Vorobyevy Gory,
Computation Mathematics and Cybernetics department

Abstract

In this paper we propose an alternative interpretation of Bayesian learning based on maximal evidence principle. We establish a notion of local evidence which can be viewed as a compromise between accuracy of obtained solution with respect to the training sample and its stability with respect to weight changes. The modification of traditional Bayesian approach allows selecting best solution among different models. This methodology was used successfully for choosing best kernel function in relevance vector machines algorithm. Both classification and regression cases are considered.

1 Introduction

Model selection problem is very important for practical implementation of machine learning algorithms. Many popular models of algorithms require some parameters (we call them model or structural parameters) to be set by user before training begins. Number of layers in multi-layer perceptron, number of clusters in cluster analysis, regularization coefficient can serve as examples of such parameters. One of the most known tasks of such kind is how to establish kernel function which is the best for particular problem. This task becomes more and more important as popularity of kernel methods increases every year. To cope with this problem several methods for particular families of algorithms were proposed [1], [2]. But the only general approach is still computationally expensive cross-validation whose estimates are known to have large variance.

A very interesting approach to model selection which became very popular recently is so-called Bayesian learning [3]. Its main idea is the following: choose the model with the highest rate of "good" algorithms (algorithms with large likelihood). This rate is called *evidence*. Also it is assumed that final decision is made by weighted voting among all algorithms in a model with weights proportional to posterior distribution of corresponding parameters. However this task is very difficult as it requires integration in high dimensional space. It can be shown that if an algorithm is linear with respect to all its parameters (e.g. SVM) then solution with the largest posterior probability can be taken as good approximation of weighted voting. In other words we may replace voting with only one solution which has the largest vote. This idea was used successfully in Relevance Vector Machines (RVM) [5] where regularization coefficients are adjusted automatically during training. Unfortunately the same idea cannot be used for more sophisticated tasks such as e.g. determination of the best kernel function. In this paper we propose a modification of Bayesian approach to model selection suggesting another interpretation of evidence notion.

The rest of the paper is organized as follows. In the next section we give a brief description of maximal evidence principle and introduce local evidence notion. Section 3 describes how this technique can be applied for kernel selection in RVM while section 4 gives some experimental results. Conclusions are given in the last section.

2 Model Selection Paradigm

2.1 Bayesian Approach

Let $\Omega(\vec{\alpha})$ be a set of algorithms (model) which is defined by model parameter $\vec{\alpha}$. Then model selection task is to choose the value of $\vec{\alpha}$ whose corresponding model is the best with respect to the training data $\{X,T\} = \{\vec{x_i}, t_i\}_{i=1}^n$, where $\vec{x_i} \in \mathbb{R}^d$ and $t \in \mathbb{R}$ or $t \in \{-1,1\}$ for regression and classification cases cor-

0-7695-2521-0/06/$20.00 (c) 2006 IEEE

respondingly. In other words the best model is determined as $\vec{\alpha^*} = \arg\max_{\vec{\alpha}} P(\vec{\alpha}|X,T)$. Assuming that all models are equally likely we may rewrite it in the following manner:

$$P(\vec{\alpha}|X,T) \sim P(T|X,\vec{\alpha}) = \\ = \int\limits_{\Omega(\vec{\alpha})} P(T|X,\vec{w},\vec{\alpha})P(\vec{w}|\vec{\alpha})d\vec{w} \quad (1)$$

Here $P(T|X,\vec{w},\vec{\alpha})$ is likelihood of training data according to the algorithm $\vec{w}$ from family $\vec{\alpha}$, while $P(\vec{w}|\vec{\alpha})$ is prior distribution of algorithms within the model. The value of this integral is called evidence of model which corresponds to the given value $\vec{\alpha}$. According to known principle of maximal evidence we should select the model parameter $\alpha_{\vec{M}E}$ which turns expression (1) into maximum. Decision about test data $\{X_{test}, T_{test}\}$ is made by voting over all algorithms of the model:

$$P(T_{test}|X_{test},X,T) = \\ = \int\limits_{\Omega(\alpha_{\vec{M}E})} P(T_{test}|X_{test},\vec{w},\alpha_{\vec{M}E})P(\vec{w}|\alpha_{\vec{M}E},X,T)d\vec{w} \quad (2)$$

where

$$P(\vec{w}|\vec{\alpha},X,T) = \frac{P(T|X,\vec{w},\vec{\alpha})P(\vec{w}|\vec{\alpha})}{\int\limits_{\Omega(\vec{\alpha})} P(T|X,\vec{w},\vec{\alpha})P(\vec{w}|\vec{\alpha})d\vec{w}}$$

It can be shown that if algorithms are linear with respect to their parameters $\vec{w}$ then subintergral function $Q_{\vec{\alpha}}(\vec{w}) = P(T|X,\vec{w},\vec{\alpha})P(\vec{w}|\vec{\alpha})$ is unimodal and it can be approximated by Gaussian function. In this case one may replace equation (2) with a simpler one

$$P(T_{test}|X_{test},X,T) \approx P(T_{test}|X_{test},w_{\vec{M}P}(\alpha_{\vec{M}E}),\alpha_{\vec{M}E}) \quad (3)$$

where $w_{\vec{M}P}(\vec{\alpha}) = \arg\max_{\vec{w}} Q_{\vec{\alpha}}(\vec{w})$. This approximation is no more valid if an algorithm is not linear with respect to its parameters. Indeed $Q_{\vec{\alpha}}(\vec{w})$ is then a multimodal function and can't be approximated by Gaussian and hence equation (2) can't be approximated by (3). Moreover it can be too difficult even to find maximum of $Q_{\vec{\alpha}}(\vec{w})$.

2.2 Local Evidence

Now consider a bit closer evidence expression for linear models i.e. for the cases when we may approximate $Q_{\vec{\alpha}}(\vec{w})$ by Gaussian function. Then evidence integral (1) can be taken analytically yielding

$$P(T|X,\vec{\alpha}) \approx (2\pi)^{n/2} Q_{\vec{\alpha}}(w_{\vec{M}P}(\vec{\alpha}))|\Sigma|^{-1/2} \quad (4)$$

where $\Sigma = -\frac{\partial^2}{\partial\vec{w}^2}\log Q_{\vec{\alpha}}(\vec{w}(\vec{\alpha}))|_{\vec{w}=w_{\vec{M}P}(\vec{\alpha})}$. This equation can be viewed as a compromise between accuracy on the training sample (the first multiplier is regularized likelihood at the point of maximum) and *stability* of performance with respect to changes of classifier parameters (the last multiplier is squared root of inverse Hessian of log-likelihood function).

If function $Q_{\vec{\alpha}}(\vec{w})$ is multi-modal then approximation (4) is incorrect and hence equation (2) can't be replaced by (3). Moreover the most of known methods find only one algorithm from model rather than posterior distribution of algorithms within the model. Hence it is necessary to estimate the "quality" of algorithm obtained as a result of training rather than the whole model. Following the interpretation of evidence as combination of accuracy and stability, we may introduce a local analogue of evidence. Let $\vec{w_0}$ be a vector of algorithm's parameters received in training process. Generally speaking it is not a local extremum point of $Q_{\vec{\alpha}}(\vec{w})$. Consider the following value:

$$LE(\vec{w_0},\vec{\alpha}) = Q_{\vec{\alpha}}(\vec{w_0})\prod_{i=1}^{n} A_i \quad (5)$$

Here n is number of algorithm's parameters and

$$A_i = \begin{cases} |a_i|^{-1}, & b_i \le 0 \\ \frac{1}{2}\sqrt{\frac{2\pi}{b_i}}\exp\left(\frac{a_i^2}{2b_i}\right)\left(1 - erf\left(\frac{|a_i|}{\sqrt{2b_i}}\right)\right), & b_i > 0 \end{cases} \quad (6)$$

where $a_i = \frac{\partial}{\partial w_i}\log Q_{\vec{\alpha}}(\vec{w})|_{\vec{w}=w_{\vec{M}P}(\vec{\alpha})}$ and $b_i = -\frac{\partial^2}{\partial w_i^2}\log Q_{\vec{\alpha}}(\vec{w})|_{\vec{w}=w_{\vec{M}P}(\vec{\alpha})}$ are first and negative second derivatives of regularized likelihood at the point of solution correspondingly. Geometrically (5) and (6) means that we integrate over the tail of gaussian (or exponent in case of positive curvature) approximation of likelihood behavior in the vicinity of our solution (see fig.1). Note that in case when w_0 is the only maximum point and $n = 1$ equation (5) turns to (4). Such generalization allows to estimate the "quality" of single solution which will be used for processing test data. If we had calculated the "quality" of model according to equation (1) we should have used integration (2) for making further predictions rather than single solution obtained via training procedure.

3 Kernel Validity Index

Consider popular model of kernel methods $y(\vec{x}) = \sum_{i=1}^{m} w_i K(\vec{x},\vec{z_i}) + w_0$. Here $\vec{x}$ is a vector of d real features. In case of classification task the output of algorithm is $t(\vec{x}) = \text{sign}[y(\vec{x})]$. This model includes such known algorithms like e.g. SVM, RBF networks

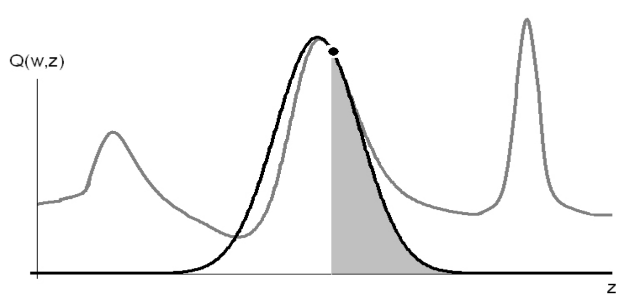

Figure 1. Local evidence notion. Grey curve is subintegral function $Q_{\vec{\alpha}}(\vec{w})$ represented along selected variable. Black curve is Gaussian function obtained as approximation at point of our solution. Grey figure corresponds to value A_i.

with one hidden layer, etc. Formally these algorithms are linear with their parameters $\vec{w}$. In relevance vector machines [5] the methods of Bayesian learning were applied to find best $\vec{w}$. The weights are supposed to have priors equal to normal distributions with zero mean and α^{-1} variance $P(w_i|\alpha_i) = N(0, \alpha_i^{-1})$. Model parameters $\vec{\alpha}$ are adjusted during training by evidence maximization. Likelihood function is calculated as $P(D_{train}|\vec{w}, \vec{\alpha}) = \prod_{i=1}^{q} \frac{1}{1+\exp(-t_i y(\vec{x_i}))}$ for classification task and $P(D_{train}|\vec{w}, \vec{\alpha}) = \exp\left(\sum_{i=1}^{q} \frac{1}{2\lambda^2}(t_i - y(\vec{x_i}))^2\right)$ for regression task.

One of the most popular kernel functions is $K(\vec{x_1}, \vec{x_2}) = \exp(-\frac{1}{2\sigma^2}\|\vec{x_1} - \vec{x_2}\|^2)$. To select best value of σ we should extend our model adding kernel centers $\vec{z_i}$. But subintegral function $Q_{\vec{\alpha},\sigma}(\vec{w}, \vec{z})$ now becomes multi-modal as dependence $y(\vec{x})$ from $\vec{z} = (\vec{z_1}, \ldots, \vec{z_m})$ is non-linear. Moreover optimization of kernel centers location in $m \times d$ dimensional space is computationally very difficult problem. Actually, these centers are usually not optimized at all and are set in the objects of the training sample $\vec{z_0}$. In this case we may use local evidence to estimate quality of algorithm obtained by optimizing the weights w_i of kernels located in the training objects with given σ. The kernel validity index is calculated as follows:

$$KV(\sigma) = Q_{\alpha_{\vec{ME}},\sigma}(w_{\vec{MP}}(\alpha_{ME}), \vec{z})|\Sigma|^{-1/2} \prod_{i=1}^{m} \left(\prod_{j=1}^{d} A_{ij} \right)^{\gamma_i} \tag{7}$$

here $\Sigma = -\frac{\partial^2}{\partial \vec{w}^2} \log Q_{\vec{\alpha},\sigma}(\vec{w}(\vec{\alpha}), \vec{z})|_{\vec{w}=w_{\vec{MP}}(\vec{\alpha})}$, A_{ij} is stability component with respect to the shift of j^{th} coordinate of i^{th} object calculated according to (6). The stability components with respect to $\vec{z_i}$ should be considered with their effective weights $\gamma_i = 1 - \alpha_i \Sigma_{ii}$ calculated according to [5]. Indeed if kernel has zero weight its stability with respect to the center shift should be ignored as if there were no kernel at all. It is easy to prove that $\gamma_i = 0$ if and only if $w_i = 0$.

Function $KV(\sigma)$ has a typical bell-shaped form. The value of $\sigma = \arg\max KV(\sigma)$ is recommended for further use.

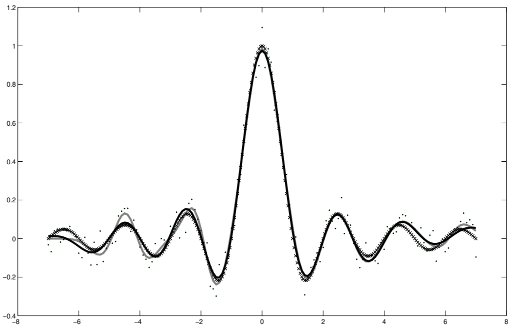

Figure 2. Sinc data, 141 train objects, 281 test objects, noise level 0.1. RVR performance with kernel function obtained by cross-validation and kernel validity index are shown by black and grey curves correspondingly.

4 Experimental Results

We compare kernel selection performance of kernel validity index vs. cross-validation using 9 classification problems from UCI repository and data generated from sinc function for regression (see fig.2). For each classification task we randomly split 20 times the data into train (33%) and test (67%) sets and use RVM with kernels of different width ($\sigma = 0.01, 0.1, 0.3, 1, 2, 3, 4, 5, 7, 10$). Test errors corresponded to the kernels with maximum validity or with best cross-validation estimate and averaged by 20 pairs of train/test tables together with their standard deviations are shown in table 1. Columns RVM CV and SVM CV show the averaged test error with kernel selection according to 5-fold cross-validation for RVM and SVM. RVM MV shows averaged test errors corresponded to maximum kernel validity index. Column SVM MV shows how SVM performs *with the same kernels* as in RVM MV. This column helps us to check whether the optimal kernel width is defined only by the problem itself or also by the training algorithm.

Table 1. Experimental results.

Sample Name	# obj.	# feat.	RVM CV	SVM CV	RVM MV	SVM MV	MinTestError
AUSTRALIAN	690	14	15.5 ± 1.2	16.5 ± 1.9	18.6 ± 1.35	21 ± 3.6	13.4
BUPA	345	6	41 ± 0.4	37.5 ± 2.5	39 ± 0.6	37.6 ± 3.8	31
CLEVELAND	303	13	18.6 ± 1.8	21 ± 2.7	20 ± 2.5	28 ± 5.6	17
CREDIT	690	15	17.3 ± 2.7	18 ± 1.6	16.9 ± 2.4	20 ± 2.9	14.5
HEPATITIS	155	19	43 ± 5.6	39.17 ± 3.8	39 ± 3.9	39.21 ± 4.6	36
HUNGARY	294	13	22 ± 4.4	20 ± 2.3	24 ± 5.3	26 ± 4	18
LONG BEACH	200	13	25.25 ± 0.5	25.18 ± 0.9	27 ± 1.7	26 ± 4.6	24.5
PIMA	768	8	34 ± 2.7	30 ± 2	27 ± 2.5	29.6 ± 2.9	23
SWITZERLAND	123	13	6.4 ± 1.6	8 ± 1.8	7 ± 2	7.6 ± 2.3	5.8
Total			**21**	**20**	**20**	**29**	

Finally the last column contains minimal possible test error.

The results from table 1 were rated in the following way. The least test error was given one point, while the second two points, etc. The worst result was assigned four points. Total results are shown in the last line of the table. According to it we may say that RVM and SVM show competitive results although RVM generated 5-8 times less kernels than the corresponding SVM. Also our kernel validity measure works at least not worse than cross-validation alternative. Moreover it requires only one cycle of training and hence works significantly faster. Very interesting effect is poor quality of SVM performance using the kernels which were considered to be the best (in sense of our validity measure) for RVM. This proves that kernel validity depends much on the method of training vector machine classifier. Also we should mention that neither cross-validation nor maximum validity index lead to minimum possible test error. This can be connected both with peculiarities of training sample and with the fact that test sample may be biased with respect to the universal set.

5 Conclusion

Unlike structural risk minimization [6] which restricts too flexible classifiers and minimum description length approach [4] which penalizes algorithmic complexity, the concept of Bayesian regularization (and its modification described above) tries to establish the model where the solution is stable with respect to changes of classifier parameters. We decided to move from probabilistic approach and concentrate directly on idea of stability rather than on applying maximal likelihood principle to models (i.e. estimating evidence). The proposed characteristic of kernel validity does not show how good is the kernel for particular task. It only can serve for estimation of kernel utility in case of fixed training procedure (in our case this is RVM). This happens because we do not estimate the validity of whole model (as we use only one classifier with $\vec{w} = w_{\vec{M}P}$) but consider only local stability of $Q_{\vec{\alpha}}(\vec{w})$ at point $w_{\vec{M}P}$. This method seems to be quite general and probably could be applied to other complex machine learning algorithms for tuning their model parameters.

The work was supported by the Russian Foundation for Basic Research (grants 06-01-00492, 06-07-89120, 05-07-90333, 05-01-00332, 04-01-00161) and INTAS (projects YS 04-83-2942, 04-77-7036).

References

[1] O. Chapelle and V. Vapnik. Model selection for support vector machines. *Advances in Neural Information Processing Systems*, 12, 2000.

[2] J.-Y. Kwok. The evidence framework applied to support vector machines. *IEEE Transactions on Neural Networks*, 11(5), 2000.

[3] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[4] J. Rissanen. Modelling by the shortest data description. *Automatica*, 14, 1978.

[5] M. Tipping. Sparse bayesian learning and the relevance vector machines. *Journal of Machine Learning Research*, 1:211–244, 2001.

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