The Use of Bayesian Framework for Kernel Selection in Vector Machines Classifiers

Dmitry Kropotov[1], Nikita Ptashko[2], and Dmitry Vetrov[1]

[1] Dorodnicyn Computing Centre, Vavilova str. 40, Moscow, 119991, Russia
{DKropotov, VetrovD}@yandex.ru
http://dkropotov.narod.ru, vetrovd.narod.ru
[2] Moscow State University, Vorob'evy gory, Moscow, 119234, Russia
Ptashko@inbox.ru

Abstract. In the paper we propose a method based on Bayesian framework for selecting the best kernel function for supervised learning problem. The parameters of the kernel function are considered as model parameters and maximum evidence principle is applied for model selection. We describe a general scheme of Bayesian regularization, present model of kernel classifiers as well as our approximations for evidence estimation, and then give some results of experimental evaluation.

1 Introduction

Support Vector Machines [1] are one of the most popular algorithms for solving regression and classification problems. They have proved their good performance on numerous tasks. The main reasons for the success of SVM are the following. Vapnik's idea of optimal hyperplane construction led to maximal margin principle [2] which provides better generalization ability. Another useful property of SVM is the so-called "kernel trick" which allows linear methods of machine learning to build non-linear surfaces. However, there are some aspects which remain unclear when one starts using SVM. The concrete form of the kernel function should be defined by the user so as regularization coefficient C. As there are several parametric families of kernel functions it is not clear what family and what function from that family will lead to the best performance of SVM. Coefficient C limits the values of weights for the support vectors, thereby giving the algorithm different degrees of flexibility. Usually the parameters of kernel function and coefficient C are defined using a cross-validation procedure. This may be too expensive from computational point of view. Moreover the cross-validation estimates of performance, although unbiased [2], have large variance due to the limited size of the sample. Recently Tipping proposed an SVM-like algorithm, which used Bayesian regularization for best weights selection [4]. It was called Relevance Vector Machines (RVM). In this algorithm the weights of the so-called relevance vectors are interpreted as random values with gaussian prior distribution centered in zero. In this approach there is no need to set a regularization coefficient C to restrict the values of the weights. Large weights are penalized

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 252–261, 2005.

automatically during training. In the paper we propose an extension of this idea - Generalized Relevance Vector Machines (GRVM) which allows furthermore selecting the best kernel function from the given family for the particular problem. In the next section we give general scheme of Bayesian regularization of machine learning algorithms. Section 3 briefly describes the RVM concept and in section 4 we present the GRVM algorithm for classification tasks. Some numerical aspects of its realization are given in section 5. The last section contains experimental evaluation and discussion.

2 Bayesian Learning and Maximal Evidence Principle

The paradigm of Bayesian learning allows for choosing the most appropriate model for the given training data. The term model in this context means a set of classifiers with fixed number of parameters and their prior distributions. Suppose that we have a set of models (either finite, countable or continuum) $W(\boldsymbol{\alpha})$, $\boldsymbol{\alpha} \in A$. Here $\boldsymbol{\alpha}$ defines the family of classifiers, the structure of their parameters $\boldsymbol{w}$, and their prior distributions $P(\boldsymbol{w}|\boldsymbol{\alpha})$. Denote by $P(D_{train}|\boldsymbol{w})$ the likelihood of the training data description with given values of $\boldsymbol{w}$. As the hyperparameters $\boldsymbol{\alpha}$ do not have direct influence on the training data we may write

$$P(D_{train}|\boldsymbol{w}, \boldsymbol{\alpha}) = P(D_{train}|\boldsymbol{w}) \tag{1}$$

This means that $\boldsymbol{\alpha}$ affects the likelihood of the training data description only by means of its influence on $\boldsymbol{w}$. A classical way of classifier training is based on maximal likelihood principle, that is finding

$$\boldsymbol{w_{ML}} = \arg\max_{\boldsymbol{w}} P(D_{train}|\boldsymbol{w})$$

The probability of new data D_{test} given the training set is then just

$$P(D_{test}|D_{train}) = P(D_{test}|\boldsymbol{w_{ML}})$$

An alternative way of classifier training is to use Bayesian estimation of the posterior probability of $\boldsymbol{w}$

$$P(\boldsymbol{w}|D_{train}) = \frac{P(D_{train}|\boldsymbol{w})P(\boldsymbol{w})}{\int\limits_{W} P(D_{train}|\boldsymbol{w})P(\boldsymbol{w})d\boldsymbol{w}}$$

Then

$$P(D_{test}|D_{train}) = \int\limits_{W} P(D_{test}|\boldsymbol{w})P(\boldsymbol{w}|D_{train})d\boldsymbol{w}$$

Such inference can be done within one model. Now suppose we have several (or even continuum) models $W(\boldsymbol{\alpha})$ of different nature, complexity etc. The question is what model is preferable. To answer it we should estimate the so-called evidence

$$P(D_{train}|\boldsymbol{\alpha}) = \int\limits_{W(\boldsymbol{\alpha})} P(D_{train}|\boldsymbol{w})P(\boldsymbol{w}|\boldsymbol{\alpha})d\boldsymbol{w} \tag{2}$$

The known principle of maximal evidence [3] states that we should choose that model which has the greatest value of evidence or, in other words, where the rate of "good" classifiers is the highest. This principle is a compromise between the complexity of a model and classifier's performance on the training sample. Taking into account (1) the likelihood of the test data is calculated in the following way:

$$P(D_{test}|D_{train}) = \int\limits_{A} \int\limits_{W(\boldsymbol{\alpha})} P(D_{test}|\boldsymbol{w},\boldsymbol{\alpha})P(\boldsymbol{w},\boldsymbol{\alpha}|D_{train})d\boldsymbol{w}d\boldsymbol{\alpha} = \tag{3}$$

$$\int\limits_{A} \int\limits_{W(\boldsymbol{\alpha})} P(D_{test}|\boldsymbol{w})P(\boldsymbol{w}|\boldsymbol{\alpha},D_{train})P(\boldsymbol{\alpha}|D_{train})d\boldsymbol{w}d\boldsymbol{\alpha},$$

where

$$P(\boldsymbol{\alpha}|D_{train}) \propto P(D_{train}|\boldsymbol{\alpha})P(\boldsymbol{\alpha}),$$

i.e. in case of absence of any prior assumptions on $\boldsymbol{\alpha}$, $P(\boldsymbol{\alpha}|D_{train})$ is proportional to evidence. Integration over A is often intractable that is why $P(\boldsymbol{\alpha}|D_{train})$ is usually approximated by $\delta(\boldsymbol{\alpha_{MP}})$ where $\boldsymbol{\alpha_{MP}} = \arg\max\limits_{\boldsymbol{\alpha}} P(D_{train}|\boldsymbol{\alpha})$. Then equation (3) turns into

$$P(D_{test}|D_{train}) \approx \int\limits_{W(\boldsymbol{\alpha_{MP}})} P(D_{test}|\boldsymbol{w})P(\boldsymbol{w}|\boldsymbol{\alpha_{MP}},D_{train})d\boldsymbol{w} \tag{4}$$

3 Relevance Vector Machines

Here we briefly consider the idea proposed by Tipping on using Bayesian framework in kernel methods [4]. Henceforth we consider the classification problem. Let $D_{train} = \{\boldsymbol{x},\boldsymbol{t}\} = \{x_i,t_i\}_{i=1}^m$ be training sample where x_i are feature vectors in an n-dimensional real space and t_i are class labels taking values in $\{-1,1\}$. Consider the family of classifiers $y(x) = \text{sign}(\sum_{i=1}^m w_i K(x,x_i) + w_0) = \text{sign}(h(x,\boldsymbol{w}))$. Establish prior distribution on the weights $P(w_i|\alpha_i) \sim N(0,\alpha_i^{-1})$. The set of parameters $\boldsymbol{\alpha}$ determines the model in which the posterior distribution is looked for. Define the likelihood of training sample as

$$P(D_{train}|\boldsymbol{w},\boldsymbol{\alpha}) = P(D_{train}|\boldsymbol{w}) = \prod_{i=1}^{m} \frac{1}{1+\exp(-t_i h(x_i,\boldsymbol{w}))}$$

Then the evidence of model is given by (2). Our goal is to find $\boldsymbol{\alpha}$ which maximizes evidence and then to get posterior distribution $P(\boldsymbol{w}|D_{train},\boldsymbol{\alpha})$. As direct calculation of (2) is impossible due to the intractable integral, Tipping used Laplace approximation for its estimation. He approximated $L_{\boldsymbol{\alpha}}(\boldsymbol{w}) = \log(P(D_{train}|\boldsymbol{w})P(\boldsymbol{w}|\boldsymbol{\alpha}))$ by quadratic function using its Taylor decomposition with respect to $\boldsymbol{w}$ at the point of maximum $\boldsymbol{w_{MP}}$. Such approximation can be integrated yielding

$$P(D_{train}|\boldsymbol{\alpha}) \approx \exp(L_{\boldsymbol{\alpha}}(\boldsymbol{w_{MP}})) \mid \Sigma \mid^{1/2}, \tag{5}$$

where $\Sigma = (\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{w}} L(\boldsymbol{w}) \mid_{\boldsymbol{w}=\boldsymbol{w}_{MP}})^{-1}$. Differentiating the last expression with respect to $\boldsymbol{\alpha}$ and setting the derivatives to zero gives the following iterative re-estimation equation

$$\alpha_i^{new} = \frac{1 - \alpha_i^{old} \Sigma_{ii}}{w_i^{MP}} \tag{6}$$

The training procedure consists of three iterative steps. First we search for the maximum point $\boldsymbol{w_{MP}}$ of $L(\boldsymbol{w})$. Then we make approximation according to (5) and use (6) to get the new values of $\boldsymbol{\alpha}$. The steps are repeated until the process converges.

After the training is finished the integral (4) can be approximated by setting $P(\boldsymbol{w}|D_{train}, \boldsymbol{\alpha}) \approx \delta_{(}\boldsymbol{w_{MP}})$ resulting in the expression

$$P(D_{test}|D_{train}) = P(D_{test}|\boldsymbol{w_{MP}})$$

It was shown [4] that RVM provides approximately the same quality as SVM with the same kernel function and best value of C selected by cross-validation but does not require the regularization coefficient C to be set by the user. Moreover it appeared that RVM is much more sparse, i.e. the rate of non-zero weights (relevance vectors) is significantly less than the rate of support vectors. This happens because most of the objects are treated as irrelevant and the corresponding α tend to infinity.

4 Generalized Relevance Vector Machines

Model selection via maximal evidence principle allows for avoiding the direct setting of weight constraints in RVM. Nevertheless, making a choice on a kernel function is still needed. The question is whether it is possible to use analogous approach and to treat the kernel function type as meta-parameter using Bayesian framework to define it. Henceforth we consider one of the most popular parametric kernels $K(x, z) = \exp(-\frac{||x-z||^2}{2\sigma^2})$. Our goal is to find the best σ value without cross-validation using maximal evidence principle.

It is easy to see that equation (5) presents a compromise between the accuracy on the training sample (the first item) and some kind of stability with respect to changes of the algorithm's weights (second item). Small values of σ lead to overfitting and hence to high accuracy on the training sample. On the other hand second item of formula (5) does not penalize such σ due to the following reason. Small σ means that almost all objects from the training set have non-zero weights and the influence from the neighboring objects can be neglected. But changes in object's weight just change the height of the corresponding gaussian still keeping its center in the object. The likelihood after such weight changes is still very high and the second term in (5) even encourages small σ. At the same time if we start changing the position of the gaussian center the likelihood of the training object changes dramatically (see fig.1). So small σ make classification unstable with respect to shifts of the kernel centers. Hence it is necessary to extend RVM model allowing kernels to be located at arbitrary point (relevant point) of objects space.

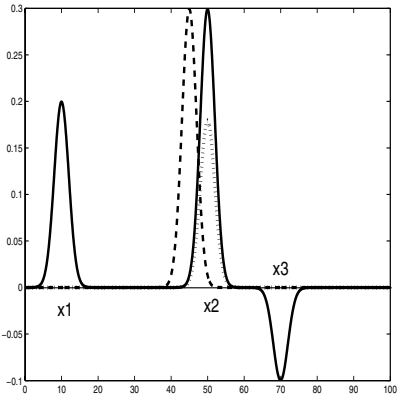

Fig. 1. The likelihood of the training sample is a product of likelihoods in each training object $x1, x2, x3$. Narrow Gaussians centered in training objects have nearly no influence on the other objects from the training set. Small weight change still keeps the likelihood of the corresponding object high enough (dotted line) while small shifts of a relevant point (gaussian center) make likelihood catastorphically low in case of small σ (dashed line).

Let $M(\boldsymbol{\alpha}, \sigma)$ be the model that defines the family of classifiers $y(x) = \text{sign}(\sum_{i=1}^{p} w_i K(x, z_i) + b) = \text{sign}(h(x, \boldsymbol{w}, \boldsymbol{z}))$. Here z_i is the center of i^{th} kernel function (in our case this function is a gaussian). We call it a relevant point. Then the likelihood of the training sample is given by

$$P(D_{train}|\boldsymbol{w}, \boldsymbol{z}) = \prod_{j=1}^{m} \frac{1}{(1 + \exp(-t_j h(x_j, \boldsymbol{w}, \boldsymbol{z})))}$$

We have no prior knowledge about $\boldsymbol{z}$ so that we assume improper uniform distribution across the whole space of objects. Then the evidence is expressed by

$$P(D_{train}|\boldsymbol{\alpha}, \sigma) \propto \int_W \int_{R^n} P(D_{train}|\boldsymbol{w}, \boldsymbol{z}) P(\boldsymbol{w}|\boldsymbol{\alpha}) d\boldsymbol{w} d\boldsymbol{z} \tag{7}$$

Again we will use Laplace approximation for the expression under the integral. Denote $L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z}) = \log(P(D_{train}|\boldsymbol{w}, \boldsymbol{z}) P(\boldsymbol{w}|\boldsymbol{\alpha}))$. Then the integral (7) can be evaluated analytically yielding

$$P(D_{train}|\boldsymbol{\alpha}, \sigma) \approx \exp(L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w_{MP}}, \boldsymbol{z_{MP}})) det(\nabla_{\boldsymbol{w},\boldsymbol{z}} \nabla_{\boldsymbol{w},\boldsymbol{z}} L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z}) \,|\, _{\substack{\boldsymbol{w}=\boldsymbol{w_{MP}} \\ \boldsymbol{z}=\boldsymbol{z_{MP}}}})^{-1/2}$$

Since σ is a scalar we may use direct search methods for its estimation by evaluating

$$E(\sigma) = \max_{\boldsymbol{\alpha}} P(D_{train}|\boldsymbol{\alpha}, \sigma) \tag{8}$$

Then the training process can be presented in the following way:

1. Start with some initial values of $\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\alpha}, \sigma$.
2. Maximize $P(D_{train}|\boldsymbol{w}, \boldsymbol{z})P(\boldsymbol{w}|\boldsymbol{\alpha})$ with respect to $\boldsymbol{w}$.
3. Re-estimate $\boldsymbol{\alpha}$ according to formula (6).
4. Go to step 2 until the process converges. Otherwise go to step 5.
5. Maximize $P(D_{train}|\boldsymbol{w}, \boldsymbol{z})$ with respect to $\boldsymbol{z}$.
6. Go to step 2 until process converges. Otherwise get $E(\sigma)$ according to (8).
7. Change σ in order to maximize $E(\sigma)$.

Note that there is no need to make additional optimization with respect to $\boldsymbol{\alpha}$ in step 6 as it has been already optimized during steps 2-5. We may use $\boldsymbol{z} = \boldsymbol{x}$ as initial estimation. As the most of α will tend to infinity to the step 5, the number of relevant points to be optimized will be relatively small and we may utilize a gradient descent method.

5 Numerical Realization and Approximations

To implement the algorithm described in the previous section we have to deal with problems connected with high dimensionality of the $(\boldsymbol{w}, \boldsymbol{z})$ space. Its dimension is $p(n+1)+1$. Large number of relevance points (large value of p) is typical in case of small σ. We have to make some assumptions to reduce the computation time. First of all we will set all mixed derivatives $\frac{\partial^2 L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w},\boldsymbol{z})}{\partial w_i \partial x_{jk}}$ to zero. Then the Taylor decomposition of $L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z})$ at the point of maximum $(\boldsymbol{w_{MP}}, \boldsymbol{z_{MP}})$ will turn to

$$L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z}) \approx L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w_{MP}}, \boldsymbol{z_{MP}}) + \frac{1}{2}(\triangle w^T H_w \triangle w + \triangle z^T H_z \triangle z)$$

Here Hessian H_w is responsible for the selection of $\boldsymbol{\alpha}$ i.e. for stability with respect to weight changes and Hessian H_z is responsible for the selection of σ i.e. for stability with respect to shifts of relevant points.

Hessian H_z is still difficult to compute as its size is $pn \times pn$. So another approximation is to interpret each relevance point z_k as a single variable. Our goal is to estimate the measure of unsteadiness at the point, not its direction. Differentiating formally with respect to z_k as a single variable we get

$$\frac{\partial}{\partial z_k} L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z}) = \frac{\partial}{\partial z_k} \sum_{i=1}^{p} \log(1 + exp(-t_i h(x_i, \boldsymbol{w}, \boldsymbol{z}))) = - \sum_{i=1}^{p} \frac{t_i w_k}{1 + exp(t_i h(x_i, \boldsymbol{w}, \boldsymbol{z}))} \frac{\partial h(x_i, \boldsymbol{w}, \boldsymbol{z})}{\partial z_k}$$

$$\frac{\partial^2}{\partial {z_k}^2} L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z}) = \sum_{i=1}^{p} \left[- \frac{exp(t_i h(x_i, \boldsymbol{w}, \boldsymbol{z}))}{(1 + exp(t_i h(x_i, \boldsymbol{w}, \boldsymbol{z})))^2} \left(\frac{\partial h(x_i, \boldsymbol{w}, \boldsymbol{z})}{\partial z_k} \right)^2 + \frac{t_i}{1 + exp(t_i h(x_i, \boldsymbol{w}, \boldsymbol{z}))} \frac{\partial^2 h(x_i, \boldsymbol{w}, \boldsymbol{z})}{\partial {z_k}^2} \right]$$

where

$$\frac{\partial h(x_i, \boldsymbol{w}, \boldsymbol{z})}{\partial z_k} = w_k \frac{||z_k - x_i||}{\sigma^2} K(z_k, x_i);$$
$$\frac{\partial^2 h(x_i, \boldsymbol{w}, \boldsymbol{z})}{\partial {z_k}^2} = w_k \left(\frac{||z_k - x_i||^2}{\sigma^4} - \frac{1}{\sigma^2} \right) K(z_k, x_i)$$

In this Hessian the off-diagonal elements are several orders smaller than the diagonal elements, so we may neglect them getting a diagonal Hessian

$$\hat{H}_z = \text{diag}(\frac{\partial^2 L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z})}{\partial z_1^2}, \ldots, \frac{\partial^2 L_{\boldsymbol{\alpha},\sigma}(\boldsymbol{w}, \boldsymbol{z})}{\partial z_p^2})$$

6 Experimental Evaluation and Discussion

To illustrate the performance of GRVM we made several experiments using datasets taken from the UCI repository [5]. Each data table were split randomly into training (67% of objects) and test sets. We used gaussian kernel function and selected its width via leave-one-out procedure both for SVM and RVM as well as regularization coefficient C in SVM. The use of cross-validation methods is typical for searching the best kernels for the given task and widely used both for SVMs and RVMs. Our task was to check whether our approach can lead to better kernels. So we used GRVM for evidence estimation. The value of σ which corresponded to the maximum of evidence was then used for training usual RVM. Actually we could continue using GRVMs with obtained kernel but our experiments showed that RVM had slightly better performance on all tasks. The results of experiments are shown in table 1. The first three columns

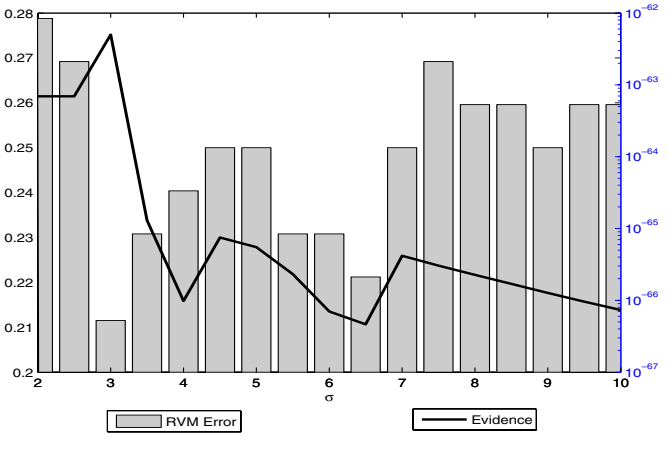

Fig. 2. Maximum of evidence most often corresponds to the minimum of test error

Table 1. Testing results of kernel function selection procedure. Column N contains number of objects in learning sample, other columns contains error rate and number of support[relevant] vectors for RVM and SVM with Leave-One-Out and Maximal Evidence kernel parameter selection procedure (RVM LOO, SVM LOO and RVM ME correspondingly).

		Errors			Vectors		
Data set	**N**	**RVM LOO**	**SVM LOO**	**RVM ME**	**RVM LOO**	**SVM LOO**	**RVM ME**
AUSTRALIAN	482	14.9%	11.54%	10.58%	37	188	19
BUPA	241	25%	26.92%	21.15%	6	179	7
CREDIT	482	16.35%	15.38%	15.87%	57	217	36
HEPATITIS	108	36.17%	31.91%	31.91%	34	102	11
PIMA	537	22.08%	21.65%	21.21%	29	309	13

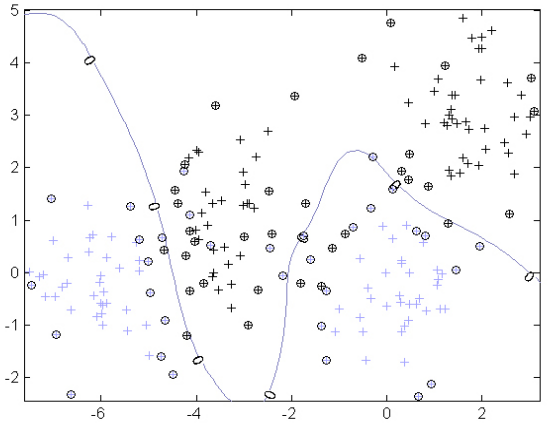

Fig. 3. Example of SVM classifier performance. Data consists of 200 objects of two classes generated from mixture of Gaussian distributions. Parameters of the classifier are: $C = 1, \sigma = 1$. Encircled objects are support vectors. In this case there are 65 support vectors.

(Errors) contain test errors of RVM with kernel selected via leave-one-out procedure, SVM with kernel obtained in the similar way and RVM with kernel which maximizes evidence respectively. The last three columns (Vectors) present number of non-zero weights. It is easy to see that maximum evidence principle selects generally better kernels than leave-one-out procedure. RVM with kernel that was selected by the proposed method outperforms state-of-art SVM classifier with kernel selected by cross-validation. Another important aspect is sparseness of obtained RVM. It is illustrated by last columns of the table. It is known that RVM is more sparse than SVM as extra non-zero weights are penalized through training. Application of maximum evidence principle to kernel determination leads to even more sparse models. A typical relation between test error and evidence value with respect to different σ values is shown in Fig. 2. Evidence is

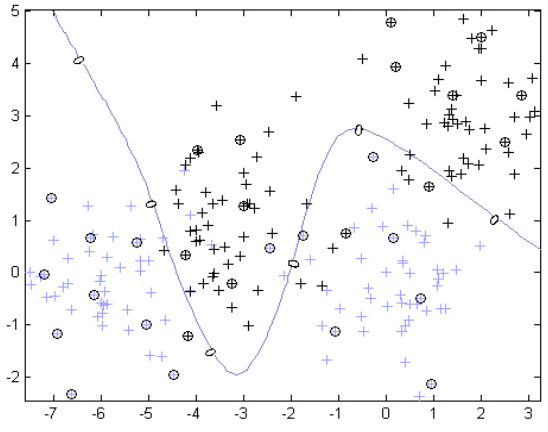

Fig. 4. Example of RVM classifier performance. Data consists of 200 objects of two classes generated from mixture of Gaussian distributions. Encircled objects are relevant vectors.

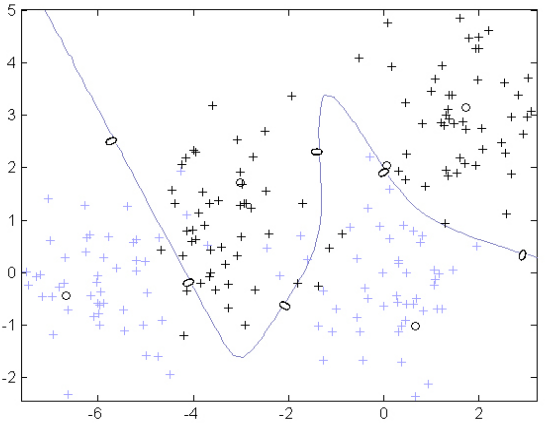

Fig. 5. Example of GRVM classifier performance. Data consists of 200 objects of two classes generated from mixture of Gaussian distributions. Black circles are relevant vectors. There are only 5 relevant vectors.

shown in logarithmic scale. It can be seen that it reaches its maximum on the same σ where test error has minimum. Although it is not necessarily so for all samples, in general this approach works better than traditional cross validation methodology.

GRVM classifier tends to be even more sparse in comparison with RVM and SVM. Figures 3, 4 and 5 illustrate performance of three classifiers on simple data. Data consists of 200 objects of two classes generated from mixture of Gaussian

distributions. In all cases σ equals 1. It can be seen that decision surface of SVM is determined by more than 60 support vectors while for RVM the correspondent parameter - number of relevance vectors - is near to 30. Accuracy of GRVM is comparable with those of SVM and RVM, but number of relevant points is only 5. Extension of RVM model allowing kernels to be located at arbitrary points of feature space and optimization of log-likelihood function with respect to kernels centres provide simplier decision models with little amount of relevant points.

7 Conclusions

Success of RVM classifiers shows that Bayesian regularization can be effectively used for optimal determination of models parameters. But simple application of this approach for kernel selection task is not reasonable since classifiers with narrower kernels are more stable with respect to weights variances. Inclusion of kernels centres to model parameters leads to sophisticated optimization procedures which nevertheless can be rather effectively implemented using some approximations.

Series of experiments using data from UCI repository show that maximum of evidence generally better corresponds to the minimum of test error than leave-one-out error. GRVM as well as RVM with kernel parameters selected according to maximal evidence tends to be more sparse. Maximization of evidence can improve the performance in many cases for both RVMs and SVMs. Moreover it allows us to carry out more sophisticated optimization, e.g. setting different σ_i for different features. Creation of effective procedure of evidence gradient estimation is still open question but seems to be a solvable task.

Acknowledgements

This work was supported by the Russian Foundation for Basic Research (projects No. 05-07-90333, 04-01-00161, 04-01-08045, 03-01-00580) and INTAS (YS 04-83-2942).

References

1. Burges, C.J.S: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery **2** (1998) 121–167
2. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag New York (1995)
3. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003)
4. Tipping, M.E.: Sparse Bayesian Learning and the Relevance Vector Machines. Journal of Machine Learning Research **1** (2001) 211–244
5. Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning Databases [Machine Readable Data Repository]. Univ. of California, Dept. of Information and Computer Science, Irvine, Calif. (1996)