Extended Bayesian Framework for Automatic Tuning of Kernel Data-Mining Methods

DMITRY KROPOTOV, VLADIMIR RYAZANOV, DMITRY VETROV
Situation Recognition Department
Dorodnicyn Computing Centre of the Russian Academy of Sciences
119991 Moscow, GSP-1, Valilova str., 40, CCAS
RUSSIAN FEDERATION

Abstract: Kernel methods like e.g. Support vector machines (SVM) and Relevance vector machines (RVM) are widely used as data-mining tools. The concept of Bayesian learning exploited in RVM leads to Automatic relevance determination (ARD) which provides sparsity in resulting decision rules. This concept also sets all regularization coefficients without involving computationally expensive cross-validation methods. In this paper we suggest an extension of Bayesian maximal evidence framework which allows to set kernel function most appropriate for the particular task. We propose a local evidence estimation method which establishes a compromise between accuracy and stability of algorithm. In the paper we first briefly describe maximal evidence principle, present model of kernel algorithms as well as our approximations for evidence estimation, and then give results of experimental evaluation. Both classification and regression cases are considered.

Key–Words: Data-mining, Kernel Methods, Bayesian Framework.

1 Introduction

Support Vector Machines (SVM) [1] has proved to be the state of the art technique for solving classification and regression problems. However, successful application of SVM needs choosing the particular kernel function as well as regularization coefficient C (or its analogue). Different values of C and forms of kernel functions lead to different behaviour of SVM for particular task.

Usually the parameters of kernel function and coefficient C are defined using cross-validation procedure. This may be too computationally expensive. Moreover the cross-validation estimates of performance, although unbiased [2], may have large variance due to the limited size of learning sample.

Several methods for model selection in SVM and SVM-like models were proposed, e.g. in [9, 10, 6, 7, 11, 8]. The popular way is application of Bayesian learning framework and maximal evidence principle [3]. Usually some probabilistic interpretation of SVM is provided which is then used for adaptation of maximal evidence principle [11, 8]. However, such probabilistic interpretation requires different approximations and changes in initial SVM training algorithm. Here we consider an SVM-like algorithm which is constructed directly from probabilistic model - Relevance Vector Machines (RVM), proposed by Tipping [4]. This approach doesn't require setting of coefficient C for restriction of weights' values as corresponding regularization coefficients are adjusted automatically during training. However, the problem of kernel selection still remains. We focus on the most popular RBF kernel functions $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}\right)$ and selection of parameter σ - width of Gaussian. We show that application of Bayesian framework for kernel selection requires extension of algorithms model - inclusion of kernel centers. Integration over posterior probability in the new model becomes intractable and hence point estimate of posterior probability is used. Laplace approximation for evidence estimation requires maximization of posterior probability as well as its Hessian computation. However, in the new model too high dimension of optimization parameters space and the fact that posterior probability is multi-modal function make the application of Laplace approximation impossible. Instead of this we propose a method of local evidence estimation which leads to a compromise between stability and training accuracy of algorithm.

The paper is organized as follows. Section 2 briefly summarizes ideas of Bayesian learning, maximal evidence principle and Relevance Vector Machines. Section 3 presents extended family of algorithms and our kernel selection procedure. In section 4 experimental results on toy problems and real data are provided, while the last section gives conclusion

and discussion.

2 Relevance Vector Machines

Let $D_{train} = \{\boldsymbol{x}, \boldsymbol{t}\} = \{x_i, t_i\}_{i=1}^m$ be a training sample where $x_i = (x_i^1, \ldots, x_i^n)$ are feature vectors in n-dimensional real space and t_i are hidden components either real (for regression) or from $\{-1, 1\}$ (for classification). Consider the family of algorithms $h(x_{new}, \boldsymbol{w}) = \sum_{i=1}^m w_i K(x_{new}, x_i) + w_0$, where $\{w_i\}_{i=0}^m$ are some real parameters or weights. Establish normal prior distribution on 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 over weights is looked for. For this model the evidence (or marginal likelihood) is given by the following equation:

$$P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{\alpha}) = \int_{W(\boldsymbol{\alpha})} P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\alpha}) P(\boldsymbol{w}|\boldsymbol{\alpha}) d\boldsymbol{w} \quad (1)$$

where $P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\alpha})$ is likelihood of training data (or more exactly likelihood of hidden components configuration) with respect to the given algorithm, $W(\boldsymbol{\alpha})$ - weights space in the model determined by $\boldsymbol{\alpha}$. Likelihood function is determined by expression $\prod_{i=1}^m \exp\left(-\frac{\|t_i - h(x_i, \boldsymbol{w})\|^2}{2\lambda^2}\right)$ in case of regression and calculated as $\prod_{i=1}^m \frac{1}{1+\exp(-t_i h(x_i, \boldsymbol{w}))}$ in case of classification.

Using known maximal evidence principle we should select $\boldsymbol{\alpha}$ by maximizing (1) and then get posterior distribution $P(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\alpha}) \propto P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\alpha}) P(\boldsymbol{w}|\boldsymbol{\alpha})$. For classification problems direct calculation of (1) is impossible due to intractable integral. Tipping used Laplace approximation for its estimation. Function $L_{\boldsymbol{\alpha}}(\boldsymbol{w}) = \log(Q_{\boldsymbol{\alpha}}(\boldsymbol{w})) = \log(P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\alpha}) P(\boldsymbol{w}|\boldsymbol{\alpha}))$ is approximated by quadratic function using its Taylor decomposition with respect to $\boldsymbol{w}$ at the point of maximum $\boldsymbol{w_{MP}}$. Such approximation can be then integrated yielding

$$P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{\alpha}) \approx Q_{\boldsymbol{\alpha}}(\boldsymbol{w_{MP}}) \mid \Sigma \mid^{1/2}, \quad (2)$$

$$\Sigma = (-\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{w}} L_{\boldsymbol{\alpha}}(\boldsymbol{w}) \mid_{\boldsymbol{w}=\boldsymbol{w_{MP}}})^{-1} = \\ = (-\nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{w}} \log(P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\alpha})) - A)^{-1} \quad (3)$$

where $A = \mathrm{diag}(\alpha_1, \ldots, \alpha_m)$. Note that for regression problems expression (2) comes to exact equation. Differentiating expression (2) with respect to $\boldsymbol{\alpha}$ and setting derivatives to zero leads to the following iterative re-estimation equations:

$$\alpha_i^{new} = \frac{\gamma_i}{w_{MP,i}^2} \quad (4)$$

$$\gamma_i = 1 - \alpha_i^{old} \Sigma_{ii} \quad (5)$$

Here γ_i is so-called effective weight of i^{th} parameter. It shows how much the corresponding weight is constrained by regularization term established by prior. It can be easily shown that $\gamma_i \in [0, 1]$. If α_i is close to zero, w_i is almost unconstrained and γ_i is close to one. On the contrary in case of large α_i the corresponding parameter w_i is close to zero and is not much affected by training information. So its effective weight tends to zero.

The training procedure consists of three iterative steps. At first we search for the maximum point $\boldsymbol{w_{MP}}$ of $L_{\boldsymbol{\alpha}}(\boldsymbol{w})$. Then we estimate Σ according to (3) and use (4), (5) to get the new $\boldsymbol{\alpha}$ values. The steps are repeated until the process converges.

In Bayesian framework decision is made by integrating throughout all algorithms within the model with respect to probabilistic measure derived by posterior probability $P(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\alpha})$:

$$P(t_{new}|x_{new}, \boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\alpha}) = \\ = \int_{W(\boldsymbol{\alpha})} P(t_{new}|x_{new}, \boldsymbol{w}, \boldsymbol{\alpha}) P(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\alpha}) d\boldsymbol{w} \quad (6)$$

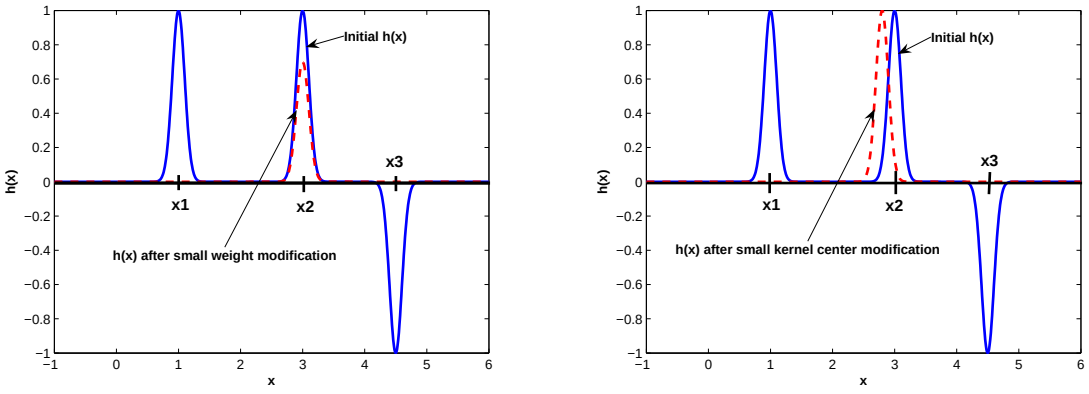
In RVM posterior distribution is approximated by setting $P(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\alpha}) \approx \delta(\boldsymbol{w} - \boldsymbol{w_{MP}})$ resulting in the expression:

$$P(t_{new}|x_{new}, \boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\alpha}) = P(t_{new}|x_{new}, \boldsymbol{w_{MP}}, \boldsymbol{\alpha}) \quad (7)$$

It was shown [4] that RVM provides approximately the same quality as SVM. Moreover RVM appeared to be much more sparse, i.e. the rate of non-zero weights (relevance vectors) is significantly less than the rate of support vectors.

3 Kernel Selection

Although maximal evidence principle is fully given in probabilistic terms we may suggest its another interpretation. Equation (2) can be viewed as a compromise between accuracy of algorithm on the training sample (the value of $Q_{\boldsymbol{\alpha}}(\boldsymbol{w_{MP}})$) and its stability with respect to small changes of parameters (expressed by squared root of inverse Hessian determinant). Then we may formulate *stability principle*. The more "stable" the algorithm is, the better its generalization ability becomes. The notion of stability is quite informal.



(a) (b)

Figure 1: The likelihood of the training sample is a product of likelihoods in each training object $x1, x2, x3$. In case of small σ values small change of weight still keeps the likelihood of the corresponding object high enough (a) while small shifts of relevant point (gaussian center) make likelihood significantly lower (b).

Different definitions of stability and their relation to generalization ability were investigated [13, 14]. Here we understand stability as ability to keep large likelihood (or more exactly the values of $Q_{\boldsymbol{\alpha}}(\boldsymbol{w})$) as long as possible moving from the point of maximum in algorithms parameter space. Such view allows to modify the concept of Bayesian regularization for the cases where its direct application is impossible or not reasonable.

In straightforward approach kernel parameter σ can be treated as one more meta-parameter (like $\boldsymbol{\alpha}$) and evidence maximization procedure can be used for its determination [11, 17]. However, in this way too small values of σ can be chosen. Indeed, small σ values lead to overfitting and high accuracy on the training sample (high value of the first term in (2)). At that almost all objects from the training set have non-zero weights and the influence from the neighboring objects can be neglected. Small variations of object's weight just change the height of the corresponding kernel function, but doesn't change classification of object in the kernel center (Fig. 1 (a)). This means that small weight's modification cannot change $L_{\boldsymbol{\alpha}}(\boldsymbol{w})$ much and the likelihood after modification is still very high. At the same time the second term in (2) even encourages small σ as the algorithm becomes more stable with respect to the changes of weights. However, if we start moving the position of the kernel center, the likelihood of the training object changes dramatically (Fig.1 (b)). So small σ makes classification unstable with respect to shifts of the kernel centers.

Actually stability with respect to weight changes is important for selection of regularization coefficients $\boldsymbol{\alpha}$. Parameter of kernel function σ is responsible for stability with respect to kernel shifts. Hence kernel selection requires inclusion of kernel centers into decision model resulting in $h_E(x_{new}, \boldsymbol{w}, \boldsymbol{z}) = \sum_{i=1}^{m} w_i K(x_{new}, z_i) + w_0$. In the extended model direct calculation of evidence (1) becomes impossible even for regression case. Laplace approximation for evidence requires additional optimization w.r.t. kernels locations $\boldsymbol{z}$ maximizing

$$L_{\sigma,\boldsymbol{\alpha}}(\boldsymbol{w}, \boldsymbol{z}) = \log(P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\alpha})P(\boldsymbol{w}|\boldsymbol{\alpha})P(\boldsymbol{z})) \tag{8}$$

Unfortunately optimization of $L_{\sigma,\boldsymbol{\alpha}}(\boldsymbol{w}, \boldsymbol{z})$ with respect to $\boldsymbol{z}$ is too difficult due to large amount of dimensions as $\boldsymbol{z} \in R^{mn}$. Moreover unlike $h(x, \boldsymbol{w})$ function $h_E(x, \boldsymbol{w}, \boldsymbol{z})$ is non-linear with respect to kernel centers $\boldsymbol{z}$ and hence $L_{\sigma,\boldsymbol{\alpha}}(\boldsymbol{w}, \boldsymbol{z})$ is multi-modal function. This hardens optimization even more.

In Bayesian framework decision rule is constructed with the aid of equation (6). But in our case (6) is intractable integral and hence we would prefer using only algorithm which was obtained via maximization of $L_{\sigma,\boldsymbol{\alpha}}(\boldsymbol{w}, \boldsymbol{z})$. If function $L_{\sigma,\boldsymbol{\alpha}}(\boldsymbol{w}, \boldsymbol{z})$ were unimodal then it could be approximated by its local behaviour at the maximum point $(\boldsymbol{w_{MP}}, \boldsymbol{z_{MP}})$. Now consider the following situation. Our solution is located in narrow peak at point $(\boldsymbol{w_{MP}}, \boldsymbol{z_{MP}})$ but there is a good stable algorithm somewhere else within the model. The evidence of obtained answer will be high, but the generalization ability of this single algorithm is poor (see fig. 2). Exact evidence calculation makes sense in case when we are able to make integration (6). However, we can use only point estimate (7). In

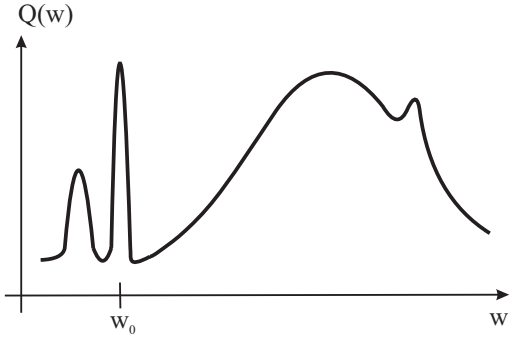


Figure 2: Example of model which has large evidence value with quite poor point estimate. There is no profit of large evidence value if we use only algorithm with $\boldsymbol{w} = \boldsymbol{w_0}$. At the same time local characteristics of point $\boldsymbol{w_0}$ such as $\nabla_w \nabla_w Q(\boldsymbol{w}) \mid_{\boldsymbol{w}=\boldsymbol{w_0}}$ penalize the obtained algorithm belonging to the model.

stability approach only local characteristics of point taken as final solution should be considered. Such characteristics are the value of function $L_{\sigma,\boldsymbol{\alpha}}(\boldsymbol{w}, \boldsymbol{z})$ and its derivatives which represent instability measure. The analogies with Bayesian framework can be used to unite these values into one equation.

Optimization of kernel locations is very difficult and time consuming task. Moreover, our experiments show that such optimization gives nearly no profit in accuracy while training time increases significantly. So we propose keeping kernel centers in training objects estimating at the same time algorithm's stability with respect to hypothetical kernel shifts. Then $\boldsymbol{w_{MP}}$ can be treated as constant which does not depend on $\boldsymbol{z}$. Assuming that there are no prior constrains on centers location establish improper uniform prior $P(\boldsymbol{z}) = const$.

Denote A_i the stability of $P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w_{MP}}, \boldsymbol{z})$ with respect to kernel located in z_i. We assume that it may be decomposed as if the stabilities with respect to different coordinates were independent

$$A_i = \prod_{j=1}^{n} A_{ij}$$

A_{ij} is determined by integrating the approximation of $\log(P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w_{MP}}, \boldsymbol{z}, \boldsymbol{\alpha}))$ with parabolic function using its Taylor decomposition at the point $\boldsymbol{z} = \boldsymbol{x}$ with respect to z_i^j:

$$A_{ij} = \begin{cases} \frac{1}{2}\sqrt{\frac{2\pi}{b}} \exp\left(\frac{a^2}{2b}\right)\left(1 - erf\left(\frac{|a|}{\sqrt{2b}}\right)\right), & b > 0 \\ |a|^{-1}, & b \leq 0 \end{cases} \quad (9)$$

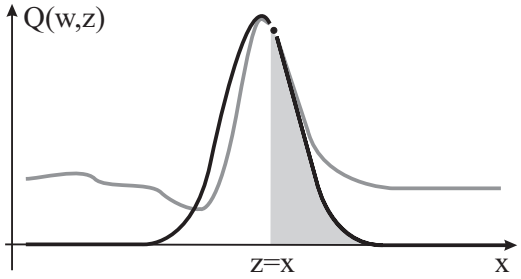


Figure 3: Algorithm stability (grey area) is expressed as integration of tail in Laplace approximation of $Q(\boldsymbol{w}, \boldsymbol{z})$ for each z_i^j.

here

$$a = \frac{\partial \log(P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w_{MP}}, \boldsymbol{z}, \boldsymbol{\alpha}))}{\partial z_i^j}$$

$$b = -\frac{\partial^2 \log(P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w_{MP}}, \boldsymbol{z}, \boldsymbol{\alpha}))}{(\partial z_i^j)^2}$$

The sense of equation (9) is shown on figure 3. Estimating algorithm's stability in the first place we would like to insure ourselves against accuracy degrade on the test sample. So $f(z_i^j) = \log P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w_{MP}}, \boldsymbol{z}, \boldsymbol{\alpha})$ is approximated with negative parabola or with a line (if second derivative is non-negative) at point $z_i^j = x_i^j$ and decreasing tail of approximation is integrated yielding stability measure A_{ij}. If x_i^j were an extremum point of $f(z_i^j)$ then A_{ij} would be proportional to the result of Laplace approximation taken along x_i^j coordinate.

For uniting stability and accuracy in one expression we should consider the weight of each kernel. Actually if the weight of kernel is close to zero its stability doesn't play important role. Taking into consideration the effective weights (5) of each kernel γ_i which vary from 0 to 1 we get the expression for total stability of likelihood with respect to all kernels

$$Z = \prod_{i=1}^{m} A_i^{\gamma_i} = \prod_{i=1}^{m} (\prod_{j=1}^{n} A_{ij})^{\gamma_i} \quad (10)$$

Multiplying Z and the value of likelihood at the point $\boldsymbol{w_{MP}}$ we get *kernel validity* value

$$KV = P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w_{MP}}, \boldsymbol{z}, \boldsymbol{\alpha}) Z \quad (11)$$

The kernel function which corresponds to the largest validity value is supposed to be the best one for the particular task.

Thus the procedure for selection of width parameter σ in gaussian parametric family of kernel functions becomes the following:

1. Choose some σ value.
2. Put $\boldsymbol{z} = \boldsymbol{x}$.
3. Train RVM algorithm with selected σ.
4. At the point $\boldsymbol{w_{MP}}$ calculate kernel validity (11), where components A_{ij} are taken from (9), while effective weights γ_i are determined by (5).

The σ value corresponding to the largest validity value is considered to be the optimal one.

4 Experimental Results

We compared kernel selection performance of kernel validity index vs. cross-validation using 9 classification problems from UCI repository. For each 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 and with best cross-validation estimate 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. Finally MinTestError 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.

Experimental results show that RVM and SVM have competitive performance 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. But our approach has two advantages. The algorithm should be trained only once thus requesting significantly less time for training. Another good property of the proposed index is its unimodality. Unlike cross-validation measure which has lots of local extrema $KV(\sigma)$ may be optimized using gradient or quasi-gradient methods. 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 Discussion and Conclusion

Unlike structural risk minimization [2] which restricts too flexible classifiers and minimum description length approach [16] 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. maximizing 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 $\boldsymbol{w} = \boldsymbol{w_{MP}}$) but consider only local stability of $Q_{\boldsymbol{\alpha}}(\boldsymbol{w})$ at point $\boldsymbol{w_{MP}}$.

The idea to take into consideration both the model of algorithms and particular training procedure (our ability to find good algorithm inside the model) for estimation of algorithm's quality is not novel. For example, Vapnik proposed so-called effective VC dimension [2]. Unlike traditional VC dimension new notion suggests consideration of training sample and considers only those algorithms which can be obtained inside the model using particular training sample. As a result error bounds become more accurate. Popular boosting and bagging techniques are said to increase both training accuracy and generalization ability of algorithms. These methods make algorithm's model sufficiently more complex. Nevertheless, effective way of choosing particular algorithm inside the extended model avoids drawbacks of such complication. Explicit consideration of training procedure together with model's properties led to new theory of algorithms quality estimates, based on combinatorial approach [15]. In our case we are not able to consider all possible algorithms inside the model (to integrate over posterior probability $P(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\alpha})$). However, consideration of local stability of $Q(\boldsymbol{w}, \boldsymbol{z})$ at point $\boldsymbol{w_{MP}}$ (our ability to find good algorithm inside the model) gives us appropriate technique for kernel selection task.

This method seems to be quite general and proba-

Table 1: Experimental results for classification problems (error rates and standard deviations).

Sample Name	# obj.	RVM CV	SVM CV	RVM MV	SVM MV	MinTestError
AUSTRALIAN	690	15.5 ± 1.2	16.5 ± 1.9	18.6 ± 1.35	21 ± 3.6	13.4
BUPA	345	41 ± 0.4	37.5 ± 2.5	39 ± 0.6	37.6 ± 3.8	31
CLEVELAND	343	18.6 ± 1.8	21 ± 2.7	20 ± 2.5	28 ± 5.6	17
CREDIT	690	17.3 ± 2.7	18 ± 1.6	16.9 ± 2.4	20 ± 2.9	14.5
HEPATITIS	155	43 ± 5.6	39.17 ± 3.8	39 ± 3.9	39.21 ± 4.6	36
HUNGARY	294	22 ± 4.4	20 ± 2.3	24 ± 5.3	26 ± 4	18
LONG BEACH	200	25.25 ± 0.5	25.18 ± 0.9	27 ± 1.7	26 ± 4.6	24.5
PIMA	768	34 ± 2.7	30 ± 2	27 ± 2.5	29.6 ± 2.9	23
SWITZERLAND	123	6.4 ± 1.6	8 ± 1.8	7 ± 2	7.6 ± 2.3	5.8
Total rank		**21**	**20**	**20**	**29**	

bly could be applied to other complex machine learning algorithms for tuning their model parameters.

Acknowledgements: This work was supported by the Russian Foundation for Basic Research (projects No. 06-01-00492, 05-07-90333, 04-01-00161, 06-01-08045) and INTAS (YS 04-83-2942, 04-77-7036).

References:

[1] C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery,* 2, 1998, pp. 121–167.

[2] V. Vapnik, *The Nature of Statistical Learning Theory,* Springer–Verlag, New York, 1995

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

[4] M. Tipping, Sparse Bayesian Learning and the Relevance Vector Machines, *Journal of Machine Learning Research* 1, 2001, pp. 211–244.

[5] P. Murphy and D. Aha, UCI Repository of Machine Learning Databases [Machine Readable Data Repository], *Univ. of California, Dept. of Information and Computer Science, Irvine, Calif.*, 1996

[6] N. Ayat, M. Cheriet and C. Suen, Optimization of SVM Kernels using an Empirical Error Minimization Scheme, *Proc. of the First International Workshop on Pattern Recognition with Support Vector Machines*, 2002.

[7] F. Friedrichs and C. Igel, Evolutionary Tuning of Multiple SVM Parameters, *Neurocomputing,* 64, 2005, pp. 107–117.

[8] C. Gold and P. Sollich, Model Selection for Support Vector Machine Classification, *Neurocomputing,* 55(1-2), 2003, pp. 221–249.

[9] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, Feature Selection for Support Vector Machines, *Proc. of 15th International Conference on Pattern Recognition*, 2000.

[10] O. Chapelle and V. Vapnik, Model Selection for Support Vector Machines, *Advances in Neural Information Processing Systems 12, ed. S.A. Solla, T.K. Leen and K.-R. Muller*, MIT Press, 2000

[11] J. Kwok, The Evidence Framework Applied to Support Vector Machines, *IEEE-NN,* 11(5), 2000.

[12] J. Friedman, T. Hastie and R. Tibshirani, *The Elements of Statistical Learning*, Springer–Verlag, 2001.

[13] S. Kutin and P. Niyogi, Almost-everywhere algorithmic stability and generalization error, *Tech. Rep. TR-2002-03: University of Chicago*, 2002.

[14] O. Bousquet and A. Elisseeff, Algorithmic stability and generalization performance, *Advances in Neural Information Processing Systems 13*, 2001.

[15] K. Vorontsov, Combinatorial substantiation of learning algorithms, *Journal of Comp. Maths Math. Phys.* 44(11), 2004, pp. 1997-2009, *http://www.ccas.ru/frc/papers/voron04jvm-eng.pdf*

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

[17] T. Van Gestel, J. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor and J. Vandewalle, Bayesian Framework for Least Squares Support Vector Machine Classifiers, Gaussian Processes and Kernel Fisher Discriminant Analysis, *Neural Computation,* 15(5), 2002, pp. 1115–1148.