# Machine learning regularization based on instability penalty

Vetrov Dmitry
Computing Center of the Russian Academy of Sciences
Russia, 119991, Moscow
Vavilova str. 40
VetrovD@yandex.ru

## Abstract

*The procedure of machine learning is an example of ill-posed problem and hence has to be regularized. The idea of procedure described below is to consider properly classified objects with big gradient of posterior probabilities as wrong answers of the algorithm. After modifying usual quality functional it becomes possible to use only the training sample for finding algorithms, which classify the independent sample as well.*

## 1. Introduction

It is well known, that a classical pattern recognition task is ill-posed. There is only a finite number of objects in the training sample while the algorithmic family usually depends on parameters with continuous values. This leads to continuum of possible solutions and, what's worse, to unstable solutions. The latter means that after changing a little the values of features, one can't be sure that the algorithm's answer will not change dramatically. Variability of algorithms among the parametric family results in so-called overfitting which is one of the most severe problems in machine learning theory nowadays. Quite obvious fact is that the more variable classification algorithm is, the more probably it suffers from overfitting. Theory of structural risk minimization by Vapnik [6], [7] offered a way of defining the best algorithmic family as a compromise between its performance on the training sample and some penalty value which depends on algorithm's complexness expressed in VC-dimension. Although excellent in theory, this method is quite difficult to implement in practice because VC-dimension for a large number of algorithms is still unknown. Another way is to assume that the simplest models are more probable (Occam's razor). This brings us to Bayes regularization [3] and quite close to the minimal description length principle [2]. The best algorithmic family for solving the particular task, are those with minimal values of information criteria (AIC, BIC etc.). These methods allow choosing the best in some sense family, but can't be used directly during the training in the general case. However, in some special kinds of classifiers (like Relevance Vector Machines [5]) these concepts appeared to be possible to use and showed good results [1].

The aim of this article is to establish a possible way of building classifier with good performance on the test sample based on the regularization of quality functional. In the next section there given some conceptual aspects of the overfitting phenomena. Section 3 describes the way of regularizing the quality functional with convergence theorem and in section 4 there are some practical results.

## 2. The reason of overfitting

During the construction of pattern recognition algorithm we usually want it to work "well" on the objects for which the answers are unknown. This is achieved by showing it some objects with known answers (training sample). Good algorithm should satisfy two demands: high memorizing capability and high generalizing capability. The first means high quality of the work on the training sample and can be measured directly during the process of learning for example the following way:

$$MC = \frac{m_{errors}}{m} \tag{1}$$

where $m_{errors}$ is the number of errors made on the training sample and $m$ is total number of objects in the training sample. Another way is to measure some deviations of algorithm's output from the desired output.

As for the generalizing capability, it can be interpreted as closeness of work on the training sample and

on the arbitrary object. The example of it is given by the following formula:

$$GC = (P_{tr.err.} - P_{arb.err.})^2 \quad (2)$$

where $P_{tr.err.}$ is error probability on the training sample and $P_{arb.err.}$ is error probability on arbitrary object.

Unfortunately the probability of error on arbitrary object can't be measured directly as we have just a limited number of objects. Indirect measuring of generalizing capability involves the independent control sample (which hence stops being independent and can't be used for estimating the quality of obtained algorithm afterwards) or cross-validation procedures. The first means that some part of objects must be excluded from the training and is unacceptable in the cases of small samples. The second way requires numerous retraining thus becoming too expensive from the computational point of view. Moreover, not all methods allow to consider the control sample or to perform cross-validation during the training . That is why the most methods of training pattern recognition algorithms are based on the maximization of just memorizing capability. This would be enough if the capability of generalization were the same during the whole process of training. Unfortunately it becomes lower and lower while training continues (see Figure 1). It's obvious that just minimizing the error on the training sample we will loose generalization capability catching nothing but noise. From

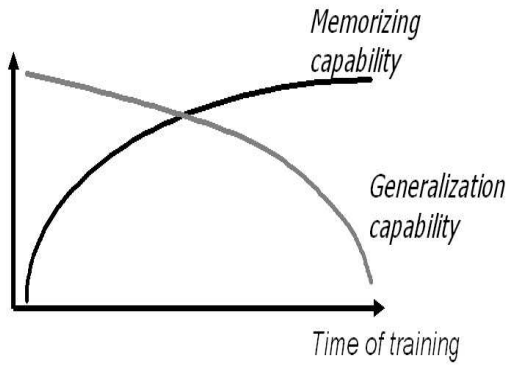


**Figure 1. Changes of memorizing and generalizing capabilities during training.**

the other side stopping training too early we will receive "undertraining" thus catching not all regularities and associations in the data. The question is where we should stop to achieve the minimum error on the arbitrary object.

## 3. Regularization of functional

As was noted in the introduction, pattern recognition task is ill-posed problem. In mathematics there are methods for solving such applied tasks by removing them with new ones, which are quite close in some sense to the initial, but are well-posed and hence more useful [4]. The main trouble in pattern recognition is overfitting on the training sample, which is connected with wrong quality functional. If the training sample were infinitely long, minimizing the error rate would be enough to build good algorithm. But in real life, there should be some considerations about its generalizing capability.

In this work we assume that this capability is connected with stability of the algorithm on the training sample. Consider recognition task with $l$ classes, where each object is represented as a vector of $n$ real features. Below there reviewed the classifiers which return posterior probabilities of belonging the object to be recognized to each class. In this case the algorithm can be considered as a vector function $A : R^n \to P^l$, where $P^l = \{(p_1, \ldots, p_l) | \sum_{j=1}^{l} p_j = 1, p_j \geq 0\}$ . Note that the most of classifiers, even those, which construct some kinds of hypersurfaces, can be represented in this way (see for example [5]). The classical quality functional, which is connected with the error rate on the training sample, can be scaled in order to have values from zero to one. Then consider the following regularizator:

$$R(\lambda) = \frac{1}{m} \sum_{j=1}^{m} \exp\left( -\frac{P^2\{\omega_k | \vec{x_j}\}}{2\|\lambda \nabla P\{\omega_k | \vec{x_j}\}\|^2} \right) \quad (3)$$

where $P\{\omega_k|\vec{x}\}$ is an estimate of posterior probability that the object belongs to class $\omega_k$ and

$$k = \arg \max_{1 \leq i \leq l} P\{\omega_i | \vec{x}\}$$

Each item under the sign of summation shows the degree of instability on $j$-th element of the training sample. As regularizator varies from zero to one, it can be added to the quality functional. Then the regularized functional to be minimized is:

$$\Psi(\vec{w}, \lambda) = \Phi(\vec{w}) + R(\lambda) \quad (4)$$

$\vec{w}$ is a set of algorithm's parameters, while $\Phi(\vec{w})$ is traditional quality functional, which we interpret as a degree of falseness on the training sample.

Regularization parameter $\lambda$ shows the strength of instability penalty, while the whole regularizator is the average degree of instability on the training sample. As

in regularization theory, we may easily prove *(Convergence theorem)* that

$$\lim_{\lambda \to 0} \Psi(\vec{w}, \lambda) = \Phi(\vec{w}) \tag{5}$$

The aim of training procedure now is to minimize the regularized functional. Note, that we sacrifice the correct classification of some objects from the training sample in order to achieve more stability and hence higher generalization capability.

## 4. Experimental results

To check the use of such regularization, an easy model classifier was built. The posterior probabilities are given by the following formula:

$$P\{\omega_i|\vec{x}_j\} = \frac{\sum_{\vec{x}_k \in \omega_i} \exp(-\frac{\rho^2(\vec{x}_j,\vec{x}_k)}{2\sigma^2})}{\sum_{k=1}^{m} \exp(-\frac{\rho^2(\vec{x}_j,\vec{x}_k)}{2\sigma^2})} \tag{6}$$

This classifier depends on only one parameter $\sigma$ which defines the wideness of kernel function. The less it is, the better algorithm works on the training sample, but the higher is overfitting. The use of regularized functional helps to find parameter value, which gives the best results on the independent test sample. The results of two experiments are shown on Figure 2 and Figure 3. The first task was cancer diagnostics and the

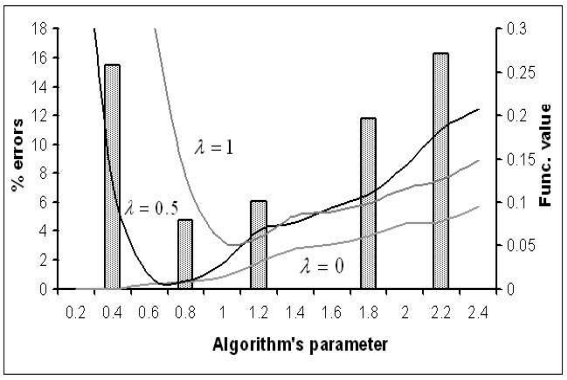


**Figure 2. Cancer diagnostics (344 objects, 9 features). There showed the values of three types of functional (curves) and the percent of errors on the test sample (histogram)**

second task was to define drug intoxication according to the reaction of human eye on the light flash. In both cases, minimum of regularized functional (with $\lambda = 0.5$ and $\lambda = 1$) is in accord with minimum of errors on the test sample (histogram). There is also shown the curve

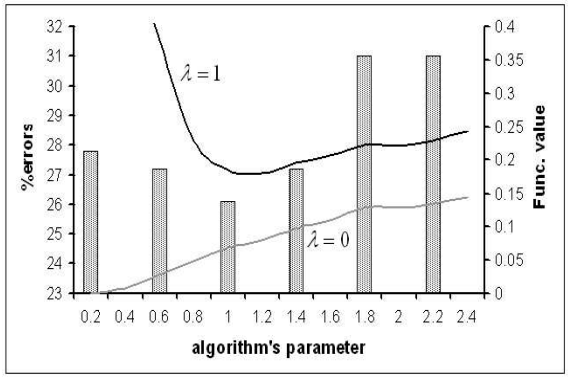


**Figure 3. Drug intoxication (500 objects, 12 features). There showed the value of classical and traditional quality functional (curves) and the percent of errors on the test sample (histogram)**

of classical quality functional $\Phi(\vec{w}) = \Psi(\vec{w}, \lambda)$, which depends on the training errors. During the numerous experiments a curious fact was noted. With necessary preprocessing of feature table (shifting and scaling), the best value of $\lambda$ does not depend on the specific task and is about one.

## 5. Conclusion and directions for future work

The results of experiments showed that such regularization could be used during training to avoid the overfitting and achieve the best proportion between memorizing and generalizing capabilities. Despite the other regularization techniques there is not so necessary to solve special optimization task for searching the best regularization parameter as it is connected with geometrical properties of the feature space and is nearly invariant after the preprocessing. During the next months it is planned to check how such regularization affects other families of algorithms (e.g. Neural Networks [8]). It seems clear that training algorithms can be easily updated to the regularized functional. Also it is quite interesting to compare this concept with the ones described in the introduction.

## 6. Acknowledgements

The work was partially supported by the Russian Foundation for Fundamental Research (grants 02-07-90134, 02-01-00558, 02-01-08007).

## References


[1] C. Bishop and M. Tipping. Variational relevance vector machines. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53, 2000.

[2] M. Hansen and B. Yu. Model selection and minimum description length principle. *Journ. Amer. Statist. Assoc*, 96:746–774, 2001.

[3] S. Shumskii. Bayes regularization of learning. *IV All-Russian Science-Technical Conference 'Neuroinformatika 2002': Lectures on Neuroinformatics*, 2:30–93, 2002.

[4] A. Tikhonov and V. Arsenin. *Solutions of Ill-posed Problems*. W.H. Winston, Washington D.C., 1977.

[5] M. Tipping. The relevance vector machine. *Advances in Neural Information Processing Systems*, 12:652–658, 2000.

[6] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

[7] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., New York, 1998.

[8] P. Wasserman. *Neural Computing: Theory and Practice*. Van Nostrand Reinhold, New York, 1989.