On One Method of Non-Diagonal Regularization in Sparse Bayesian Learning

Classification, Sparse Bayesian Learning, Automatic Relevance Determination, Laplacian priors

Abstract

In the paper we propose a new type of regularization procedure for training sparse Bayesian methods for classification. Transforming Hessian matrix of log-likelihood function to diagonal form with further regularization of its eigenvectors allows to optimize evidence explicitly as a product of one-dimensional integrals. The process of automatic regularization coefficients determination converges in one iteration. We show how to use the proposed approach with Gaussian and Laplacian priors. Both algorithms show comparable performance with the state-of-the-art Relevance Vector Machines (RVM) but require less time for training and produce more sparse decision rules (in terms of degrees of freedom).

1. Introduction

Bayesian methods have become very popular technique for classification during the last years (Bishop, 2006; Neal, 1996). Within this framework structural parameters (sometimes called model parameters) are considered to be the hyperparameters defining the family of possible classifiers. Conceptually there are two approaches to the determination of the hyperparameters. One approach is based on Automatic Relevance Determination (ARD) originally proposed by MacKay (MacKay, 1992) and leads to evidence (or type-II likelihood) maximization. Probably the most known algorithm which uses ARD is Relevance Vector Machine (RVM) (Tipping, 2000), where each weight has individual regularization coefficient that is adjusted iteratively during training. This algorithm is an example of sparse Bayesian classifier with the most of weights tend to zero. RVM may operate only with Gaussian priors over the weights. On the other hand it is known that Laplace priors are sparsity-promoting and may set a number of weights exactly to zero thus discovering irrelevant objects or features (Williams, 1995). However, direct application of Laplacian prior to RVM is impossible due to intractable integral which arises in the expression for evidence.

Alternative strategy is to integrate out hyperparameters obtaining parameter-free prior and then to maximize the product of this marginalized prior and likelihood function. It was first proposed by Williams (Williams, 1995) exactly for working with Laplacian priors and later was used successfully for processing large number of features in biomedical data (Cawley & Talbot, 2006) and for multi-class problems (Cawley et al., 2007). Unfortunately within this framework some useful properties of the problem may be lost. For example in the case of linear models the implementation of such prior with hyperparameters being integrated out results in the problem where criterion function is multi-modal, often extremely so (Tipping, 2001).

In this paper we propose an approach which allows to apply evidence framework for both types of priors. To achieve this we transform Hessian matrix of log-likelihood function to diagonal form, establish individual priors over each of eigenvectors and use ARD for estimating the values of the corresponding hyperparameters. In case of such priors the expression for evidence can be decomposed to the product of independent one-dimensional integrals each responsible for one degree of freedom. This approach is quite common since it does not depend on the particular form of prior and only requires that priors regularize each eigenvector independently of others. Besides that it seems more reasonable to assign individual regularization coefficients to the degrees of freedom defined by the eigenvectors of log-likelihood Hessian rather than to the weights which may contribute both to relevant and irrelevant eigenvectors.

Such transformation factorizes the evidence. It be-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

comes a product of one-dimensional integrals that can be optimized individually. This fact provides convergence of training process in one iteration. The number of relevant eigenvectors (degrees of freedom) becomes less than the number of zero-weights in RVM providing decision rules with fewer number of parameters.

The rest of the paper is organized as follows. Section 2 gives some notation, briefly describes evidence framework and presents problems in attempting to use Laplacian prior within the framework. In Section 3 we describe our approach and illustrate its application for the Gaussian and Laplacian priors. The comparative evaluation of accuracy, training time and sparsity with RVM is given in Section 4. Finally the work is summarized and conclusions are drawn in Section 5.

2. Evidence Framework

2.1. General Formulation

Suppose we are given a set of training objects $\{(\vec{x}_i, t_i)\}_{i=1}^n = (\mathcal{X}, \mathcal{T})$ that are described by d-dimensional real vector of features $\vec{x} \in \mathbb{R}^d$ and class label that may take one of two values $t \in \{-1, +1\}$. The classifier is determined by the vector of weights $\vec{w}$. Given the feature vector it returns posterior estimate for each class $P(-1|\vec{x}, \vec{w})$ and $P(+1|\vec{x}, \vec{w})$. The likelihood function of correct classification of training set is given by

$$P(\mathcal{T}|\mathcal{X}, \vec{w}) = \prod_{i=1}^{n} P(t_i|\vec{x}_i, \vec{w}).$$

The set of possible classifiers is defined by prior $P(\vec{w}|\vec{\alpha})$. Finding the weights according to maximum a posteriori rule $\vec{w}_{MP} = \arg\max P(\mathcal{T}|\mathcal{X}, \vec{w})P(\vec{w}|\vec{\alpha})$ is equivalent to the use of additive regularizer when optimizing logarithm of posterior. Hence the hyperparameters $\vec{\alpha}$ can be regarded as regularization coefficients.

Bayesian inference assumes that decision is made by weighted voting throughout the whole set of possible classifiers within a model and, in case of multiple possible models, throughout the whole set of models as well. Then the posterior for the classification of new object $\vec{x}$ can be written as

$$P(t|\vec{x}, \mathcal{T}, \mathcal{X}) = \int_{\mathcal{A}} \int_{\mathcal{W}(\vec{\alpha})} P(t|\vec{x}, \vec{w}, \vec{\alpha}) P(\vec{w}, \vec{\alpha}|\mathcal{T}, \mathcal{X}) d\vec{w} d\vec{\alpha} = \int_{\mathcal{A}} \int_{\mathcal{W}(\vec{\alpha})} P(t|\vec{x}, \vec{w}, \vec{\alpha}) P(\vec{w}|\mathcal{T}, \mathcal{X}, \vec{\alpha}) P(\vec{\alpha}|\mathcal{T}, \mathcal{X}) d\vec{w} d\vec{\alpha}. \quad (1)$$

MacKay has proposed to approximate $P(\vec{\alpha}|\mathcal{T}, \mathcal{X})$ with $\delta(\vec{\alpha} - \vec{\alpha}_{MP})$ where $\vec{\alpha}_{MP}$ is maximum evidence estimate

$$\vec{\alpha}_{MP} = \arg\max E(\vec{\alpha}),$$

where evidence is computed as likelihood of model

$$E(\vec{\alpha}) = P(\mathcal{T}|\mathcal{X}, \vec{\alpha}) = \int_{\mathcal{W}(\vec{\alpha})} P(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha}) P(\vec{w}|\mathcal{T}, \mathcal{X}, \vec{\alpha}) d\vec{w}. \quad (2)$$

Then equation (1) can be approximated in the following way

$$P(t|\vec{x}, \mathcal{T}, \mathcal{X}) \approx \int_{\mathcal{W}(\vec{\alpha}_{MP})} P(t|\vec{x}, \vec{w}, \vec{\alpha}_{MP}) P(\vec{w}|\mathcal{T}, \mathcal{X}, \vec{\alpha}_{MP}) d\vec{w}. \quad (3)$$

2.2. Relevance Vector Machine

In 2000 Tipping applied evidence framework for automatically adjusting individual regularization coefficients in generalized linear models

$$y(\vec{x}, \vec{w}) = \sum_{i=1}^{M} w_i \phi_i(\vec{x}).$$

The likelihood function is given by

$$P(t|\vec{x}, \vec{w}, \vec{\alpha}) = \frac{1}{1 + \exp(-ty(\vec{x}, \vec{w}))} \quad (4)$$

with normal priors on each weight $w_i \sim \mathcal{N}(0, \alpha_i^{-1})$. For evidence estimation Tipping used Laplace approximation of subintegral function in (2)[1]. Such formulation allowed to apply ARD by iteratively adjusting $\vec{\alpha}$ and led to very sparse decision rules with the most of weights set to zero.

In case of generalized linear models integration (3) can be reasonably well approximated by taking only the most probable weights $\vec{w}_{MP} = \arg\max_{\vec{w}} P(\vec{w}|\mathcal{T}, \mathcal{X}, \vec{\alpha}_{MP})$.

It should be noted, however, that the weights of irrelevant basis functions $\phi_i(\vec{x})$ only tend to zero with α_i going to infinity. On the contrary the use of Laplacian priors makes some weights equal exactly zero. But the approximation of evidence becomes then intractable problem since subintegral function is no longer smooth and should be decomposed to 2^M parts to be estimated.

[1] Alternative methods were suggested in (Bishop & Tipping, 2000).

Algorithm 1 Gaussian REVM (GREVM)

input Training data $(\mathcal{X}, \mathcal{T}) = \{\vec{x}_i, t_i\}_{i=1}^n$, $\vec{x}_i \in \mathbb{R}^d, t_i \in \{-1, 1\}$, a set of basis functions $\{\phi_i(\vec{x})\}_{i=1}^M$.
1: Find maximum of log-likelihood function $\vec{w}_{ML} = \arg\max_{\vec{w}} \log P(\mathcal{T}|\mathcal{X}, \vec{w})$.
2: Take Hessian matrix at maximum point $H = \nabla_{\vec{w}}\nabla_{\vec{w}} P(\mathcal{T}|\vec{w}, \mathcal{X})|_{\vec{w}=\vec{w}_{ML}}$.
3: Make eigenvalues decomposition of Hessian $H = Q^T \Lambda Q$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_M)$ and calculate $\vec{u}_{ML} = Q\vec{w}_{ML}$.
4:
for $i = 1$ **to** M **do**
  if $h_i u_{ML,i}^2 > 1$ **then**
    $\alpha_i^* = h_i/(h_i u_{ML,i}^2 - 1)$
  else
    $\alpha_i^* = +\infty$
  end if
end for
5: Find maximum of regularized log-likelihood function $\vec{w}_{MP} = \arg\max_{\vec{w}} \log P(\mathcal{T}|\mathcal{X}, \vec{w}) P(Q\vec{w}|\vec{\alpha}^*)$.

output Decision rule for classification of new object $\vec{x}$: $f(\vec{x}) = \mathrm{sign}\left(\sum_{i=1}^M w_{MP,i}\phi_i(\vec{x})\right)$

3. Proposed approach

Without loss of generality hereinafter we suggest likelihood function (4) to be the product of sigmoids that is used in RVM. Hence log-likelihood can be written as

$$L(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha}) = -\sum_{i=1}^{n} \log(1 + \exp(-t_i y(\vec{x_i}, \vec{w}))). \quad (5)$$

The main idea of the proposed approach is to make Laplace approximation of likelihood function, treat eigenvectors of Hessian matrix of likelihood function as new axes and make regularization as in usual sparse Bayesian learning along these new axes[2]. After Laplace approximation of likelihood function evidence (2) can be written as:

$$E(\vec{\alpha}) = \int_{\mathcal{W}(\alpha)} P(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha}) P(\vec{w}|\vec{\alpha}) d\vec{w} \approx$$

$$P(\mathcal{T}|\mathcal{X}, \vec{w}_{ML}, \vec{\alpha}) \int_{\mathcal{W}(\alpha)} \exp\left(\frac{1}{2}(\vec{w} - \vec{w}_{ML})^T H (\vec{w} - \vec{w}_{ML})\right) P(\vec{w}|\vec{\alpha}) d\vec{w},$$

[2]Quite similar "diagonalizing trick" for constructing Bayesian formulations of sparse kernel methods is given in (Cawley & Talbot, 2005).

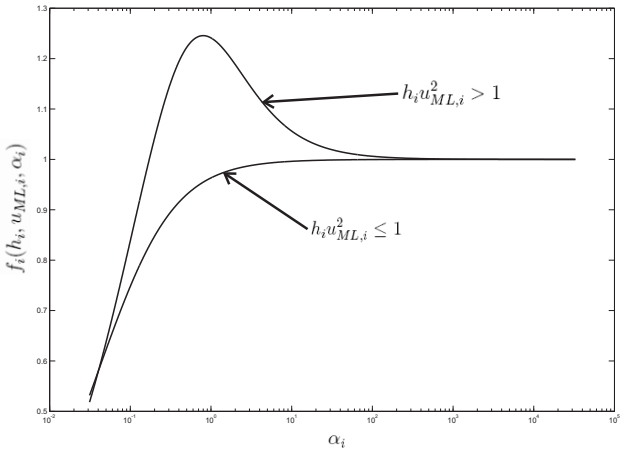

Figure 1. Behaviour of one-dimensional integral $f_i(h_i, u_{ML,i}, \alpha_i)$ depending on h_i and $u_{ML,i}$ in case of Gaussian prior.

where

$$\vec{w}_{ML} = \arg\max_{\vec{w}} P(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha}),$$
$$H = \nabla_{\vec{w}}\nabla_{\vec{w}} P(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha})|_{\vec{w}=\vec{w}_{ML}}.$$

Representing Hessian as $H = Q^T \Lambda Q$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_M)$, $\{\lambda_i\}_{i=1}^M$ - Hessian eigenvalues, we come to new variables $\vec{u} = Q\vec{w}$. Since log-likelihood function (5) is concave, Hessian H is non-positively defined and all eigenvalues $\{\lambda_i\}_{i=1}^M$ are non-positive. Denote $h_i = -\lambda_i \ge 0$. We propose to introduce independent regularization with respect to new variables $\vec{u}$. This means that prior function can be written as

$$P(\vec{u}|\vec{\alpha}) = \prod_{i=1}^{M} P(u_i|\alpha_i).$$

The main goal of such regularization is to present evidence as a product of one-dimensional integrals

$$E(\vec{\alpha}) = P(\mathcal{T}|\mathcal{X}, \vec{u}_{ML}, \vec{\alpha}) \prod_{i=1}^{M} f_i(h_i, u_{ML,i}, \alpha_i) =$$

$$P(\mathcal{T}|\mathcal{X}, \vec{u}_{ML}, \vec{\alpha}) \prod_{i=1}^{M} \int \exp\left(-\frac{h_i}{2}(u_i - u_{ML,i})^2\right) P(u_i|\alpha_i) du_i, \quad (6)$$

and then perform ARD procedure for setting hyperparameters $\vec{\alpha}$. We call this procedure Relevance Eigen Vector Machine (REVM).

In the following we consider two cases of regularization: with Gaussian and Laplace prior functions.

Algorithm 2 Laplacian REVM (LREVM)

input Training data $(\mathcal{X}, \mathcal{T}) = \{\vec{x}_i, t_i\}_{i=1}^n$, $\vec{x}_i \in \mathbb{R}^d, t_i \in \{-1, 1\}$, a set of basis functions $\{\phi_i(\vec{x})\}_{i=1}^M$.
1-3: The same as in Algorithm 1.
4:
for $i = 1$ **to** M **do**
Find maximum of (11) using one-dimensional optimization procedure:
$\alpha_i^* = \arg\max_{\alpha_i} f_i(h_i, u_{ML,i}, \alpha_i)$
end for
5: Find maximum of regularized log-likelihood function using coordinate-descend method proposed by (Shevade & Keerthi, 2003):
$\vec{w}_{MP} = \arg\max_{\vec{w}} \log P(\mathcal{T}|\vec{w}, \mathcal{X}) P(Q\vec{w}|\vec{\alpha}^*)$.
output Decision rule for classification of new object $\vec{x}$: $f(\vec{x}) = \text{sign}\left(\sum_{i=1}^M w_{MP,i}\phi_i(\vec{x})\right)$

3.1. Gaussian prior

Gaussian prior is given by the following expression

$$P(u_i|\alpha_i) = \sqrt{\frac{\alpha_i}{2\pi}} \exp\left(-\frac{\alpha_i {u_i}^2}{2}\right). \quad (7)$$

Consider one-dimensional integral $f_i(h_i, u_{ML,i}, \alpha_i)$ in expression (6) with prior (7). It can be computed analytically yielding:

$$f_i(h_i, u_{ML,i}, \alpha_i) = \sqrt{\frac{h_i\alpha_i}{2\pi}} \int \exp\left(-\frac{h_i}{2}(u_i - u_{ML,i})^2 - \frac{\alpha_i}{2}u_i^2\right) du_i = C \exp\left(\frac{h_i^2 u_{ML,i}^2}{2(h_i + \alpha_i)}\right) \sqrt{\frac{\alpha_i}{h_i + \alpha_i}} \quad (8)$$

Here C is some positive constant. Depending on h_i and $u_{ML,i}$ integral (8) has unique maximum or grows continuously as α_i tends to infinity (see fig. 1). Setting derivative of (8) with respect to α_i to zero, we obtain optimal value of α_i:

$$\alpha_i^* = \begin{cases} \frac{h_i}{h_i u_{ML,i}^2 - 1} & \text{if } h_i u_{ML,i}^2 > 1 \\ +\infty & \text{otherwise} \end{cases} \quad (9)$$

Analogous equations for training RVM using coordinate-descend method are obtained in (Tipping & Faul, 2003).

Algorithm 1 presents training procedure for sparse Bayesian model using Gaussian prior. Note that in contrast to RVM, where iterative process is needed for training, in Gaussian REVM (GREVM) optimal $\vec{\alpha}$ values can be found in one step. Experimental results (see section 4) show that GREVM is much faster and produces more sparse solutions comparing to RVM.

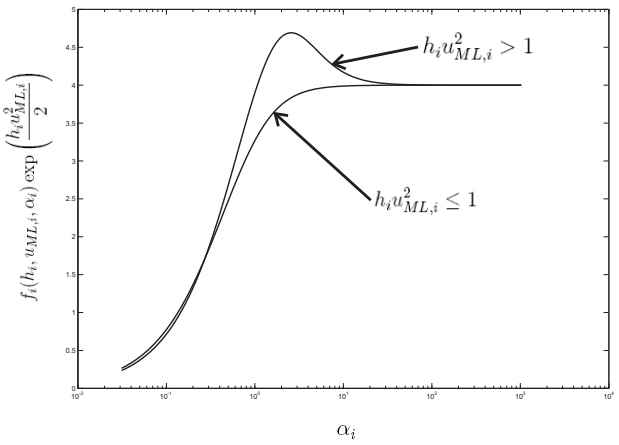

Figure 2. Behaviour of one-dimensional integral $f_i(h_i, u_{ML,i}, \alpha_i) \exp\left(\frac{h_i u_{ML,i}^2}{2}\right)$ depending on h_i and $u_{ML,i}$ in case of Laplace prior. Function f_i is multiplied by exponent just for normalizing reason (both curves have the same limit).

3.2. Laplace prior

Laplace prior function can be written as

$$P(u_i|\alpha_i) = \frac{\alpha_i}{4} \exp\left(-\frac{\alpha_i|u_i|}{2}\right). \quad (10)$$

Substituting (10) to (6) one-dimensional integral becomes

$$f_i(h_i, u_{ML,i}, \alpha_i) = \sqrt{\frac{h_i}{2\pi}}\frac{\alpha_i}{4} \int \exp\left(-\frac{h_i}{2}(u_i - u_{ML,i})^2 - \frac{\alpha_i}{2}|u_i|\right) du_i = C\alpha_i \exp\left(-\frac{h_i u_{ML,i}^2}{2}\right) \left[\text{erfcx}\left(\sqrt{\frac{h_i}{2}}\left(\frac{\alpha_i}{2h_i} - u_{ML,i}\right)\right) + \text{erfcx}\left(\sqrt{\frac{h_i}{2}}\left(\frac{\alpha_i}{2h_i} + u_{ML,i}\right)\right)\right], \quad (11)$$

where $\text{erfcx}(x) = \frac{2}{\sqrt{\pi}} \exp(x^2) \int_x^{+\infty} \exp\left(-\frac{t^2}{2}\right) dt$ - the scaled complementary error function[3] and C is some positive constant. Representation (11) involving function $\text{erfcx}(x)$ is useful due to numeric stability reasons. Expression (11) is unimodal function with respect to α_i (see fig. 2) and optimal value can be found efficiently using one-dimensional optimization methods. Algorithm 2 presents the training procedure for the case of Laplace prior. Similar to GREVM in Laplacian REVM (LREVM) optimization of $\vec{\alpha}$ values can

[3] which is implemented, e.g. in MATLAB

Table 1. Error rates together with standard deviations (in percents).

Data set	GREVM CV	LREVM CV	RVM CV
WBCD	3.55 ± 0.33	3.75 ± 0.28	3.72 ± 0.45
Bupa liver disorders	33.45 ± 2.20	34.38 ± 2.02	33.62 ± 3.73
Echocardiogram	8.33 ± 0.00	8.33 ± 0.00	8.33 ± 0.00
Heart	17.70 ± 1.21	16.22 ± 1.49	17.33 ± 1.32
Hepatitis	17.94 ± 1.06	18.45 ± 2.12	13.94 ± 0.35
Pima	23.49 ± 0.44	23.83 ± 0.69	23.75 ± 0.36
Votes	5.10 ± 0.57	5.43 ± 0.53	5.93 ± 0.64
WPBC	22.02 ± 2.07	22.42 ± 2.98	22.93 ± 1.27
Rank	**12.00**	**19.00**	**17.00**
Color	Place 1	Place 2	Place 3

Table 2. Training time together with standard deviations (in seconds).

Data set	GREVM CV	LREVM CV	RVM CV
WBCD	145.05 ± 19.66	412.18 ± 45.49	571.69 ± 83.50
Bupa	19.48 ± 0.80	88.06 ± 2.88	91.42 ± 20.51
Echocardiogram	2.76 ± 0.15	16.90 ± 0.29	3.97 ± 0.17
Heart	11.08 ± 0.22	58.67 ± 1.35	29.75 ± 3.79
Hepatitis	4.90 ± 0.22	27.83 ± 0.52	6.92 ± 0.26
Pima	160.19 ± 2.27	470.76 ± 7.36	796.36 ± 59.63
Votes	33.69 ± 1.94	161.62 ± 11.27	84.87 ± 4.04
WPBC	7.34 ± 0.43	40.19 ± 0.50	14.28 ± 0.66

be done in one step thus speeding up training procedure. However, the last step of the algorithm LREVM - optimization of regularized log-likelihood function - becomes non-trivial as this function is non-smooth at the points where at least one of the weights equals zero. For solving this problem we use algorithm proposed in (Shevade & Keerthi, 2003).

4. Experiments

In this section we compare RVM with GREVM and LREVM measuring their error rates, training time and obtained sparsity (for REVM methods sparsity means number of non-zero values in $\vec{u}_{MP}$) on a set of data taken from UCI repository (Newman et al., 1998). For each data set nominal features were transformed into a set of binary ones, unknown values were changed to mean values for each feature and then each sample was normalized in a way that each feature had zero mean and unit variance. In all classifiers being compared number of basis functions $M = n + 1$, $\phi_i(\vec{x}) = \exp(-\|\vec{x} - \vec{x}_i\|/(2\sigma^2))$, $i = 1, \ldots, n$ and $\phi_{n+1}(\vec{x}) \equiv 1$. An optimal value of width σ was chosen from the set $\{0.01, 0.1, 0.3, 0.6, 1, 2, 3, 5, 7, 10\}$ using 5x2-fold cross validation strategy (Dietterich, 1998). For each data set error rates, training time and sparsity were measured using 5x2-fold cross validation strategy as well. Tables 1, 2 and 3 report about experimental results. Rank was calculated in a usual way: for each data set the winner gets one point, the second winner - two points and the loser - three points, and then points are summed for all data sets.

These results allow to make the following conclusions. All three algorithms show comparable performance in terms of error rates. However, the sparsity of GREVM and especially LREVM is significantly less than corresponding value in RVM. This fact indirectly indicates that it is more reasonable to assign individual regularization coefficients to the degrees of freedom $\vec{u}$ defined by the eigenvectors of log-likelihood Hessian rather than to the weights $\vec{w}$ which may contribute both to relevant and irrelevant eigenvectors.

GREVM seems to be faster than RVM as optimization

Table 3. Sparsity together with standard deviations

Data set	#Obj./2	GREVM CV	LREVM CV	RVM CV
WBCD	349	4.90 ± 0.65	3.10 ± 0.55	14.40 ± 8.09
Bupa	172	4.80 ± 0.45	3.60 ± 0.55	23.10 ± 14.06
Echocardiogram	48	2.20 ± 0.45	1.30 ± 0.45	4.00 ± 0.00
Heart	135	8.30 ± 0.76	5.20 ± 0.76	12.10 ± 4.02
Hepatitis	77	5.50 ± 0.28	3.80 ± 0.27	10.20 ± 2.68
Pima	384	8.40 ± 0.96	7.10 ± 0.42	10.10 ± 3.27
Votes	435	8.60 ± 0.65	7.10 ± 0.55	14.60 ± 3.66
WPBC	99	6.80 ± 0.76	4.80 ± 0.76	20.80 ± 2.49

of regularization coefficients $\vec{\alpha}$ in GREVM requires only one step comparing to iterative process in RVM. LREVM is faster than RVM for datasets with many objects and a little amount of features and slower for other datasets. LREVM benefits in training time as it has one-step optimization of regularization coefficients but requires additional sophisticated optimization for getting $\vec{u}_{MP}$, where optimization speed depends on number of features. However, the LREVM training procedure can be probably improved by using Newton methods under some constrains.

5. Conclusions

In the paper we presented a new approach to regularization of classifiers' training procedure. Our suggestion is to regularize degrees of freedom (expressed in terms of log-likelihood Hessian eigenvectors) rather than the weights of classifier. In the weight space it corresponds to the use of non-diagonal regularizer of specific form. This regularizer is given by

$$P(\vec{w}|\vec{\alpha}) = \frac{\sqrt{|A|}}{\sqrt{2\pi}^M} \exp\left(-\frac{1}{2}\vec{w}^T Q^T A Q \vec{w}\right)$$

for Gaussian prior and

$$P(\vec{w}|\vec{\alpha}) = \frac{|A|}{4^M} \exp\left(-\frac{1}{2}\sum_{i=1}^{M}\left|\sum_{j=1}^{M} q_{ij} w_j\right|\right)$$

for Laplacian prior. Here $A = \mathrm{diag}(\alpha_1, \ldots, \alpha_M)$ and $Q = \{q_{ij}\}_{i,j=1}^{M}$. We claim that the number of freedom degrees is more natural measure of complexity. Besides that such approach provides decomposition of evidence to the product of one-dimensional integrals that can be optimized independently. The latter means that evidence framework can be used effectively for automatic relevance determination with different types of priors. This was demonstrated on the example of Laplace prior whose application to classical RVM leads to the integral which is too complicated for direct estimation.

Another interesting property of such regularization is the potential of using information criterions such as AIC (Akaike, 1974) and BIC (Schwarz, 1978) for selecting optimal set of basis functions, e.g. in case of using RBFs for setting the value of σ. Instead of the number of free parameters one may use the number of freedom degrees computed as

$$k = \sum_{i=1}^{M} \frac{h_i}{h_i + \alpha_i}.$$

Then the best set of basis functions can be found by optimizing information criterions.

It seems very promising to continue the regularization by considering main axes of regularized likelihood Hessian taken at the point $\vec{w}_{MP}$ and repeat it iteratively until the process converges. The value of evidence is expected to increase with each iteration. It can be shown easily that the sparseness in terms of freedom degrees can't be less than it was on the previous iteration. Such training presumably leads to even more sparse decision rules preserving generalization ability. This can be viewed as a search of optimal non-negative regularization matrix with respect to the weights, i.e. regularizer which provides maximum value of evidence. We consider it as one of the directions for the future work.

References

Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control*, *25*, 461–464.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bishop, C. M., & Tipping, M. E. (2000). Variational relevance vector machines. In C. Boutilier and M. Goldszmidt (Eds.), *Uncertainty in artificial intelligence 2000*, 46–53. Morgan Kaufmann.

Cawley, G. C., & Talbot, N. L. C. (2005). A simple trick for constructing bayesian formulations of sparse kernel learning methods. *Proceedings of International Joint Conference on Neural Networks (IJCNN-2005)* (pp. 1425–1430). Montreal, Canada, July 31 - August 4.

Cawley, G. C., & Talbot, N. L. C. (2006). Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *Bioinformatics*, *22*, 2348–2355.

Cawley, G. C., Talbot, N. L. C., & Girolami, M. (2007). Sparse multinomial logistic regression via bayesian l1 regularisation. In B. Scholkopf, J. C. Platt and T. Hoffmann (Eds.), *Advances in neural information processing systems 19*. Cambridge MA USA: MIT Press.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*, 1895–1924.

MacKay, D. J. C. (1992). The evidence framework applied to classification networks. *Neural Computation*, *4*, 720–736.

Neal, R. M. (1996). *Bayesian learning for neural networks*. New York: Springer.

Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

Shevade, S. K., & Keerthi, S. S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, *19*, 2246–2253.

Tipping, M. E. (2000). The relevance vector machine. In S. A. Solla, T. K. Leen and K. R. Mueller (Eds.), *Advances in neural information processing systems 12*, 652–658. MIT Press.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, *1*, 211–244.

Tipping, M. E., & Faul, A. C. (2003). Fast marginal likelihood maximisation for sparse bayesian models. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. Key West, FL, Jan 3-6 2003.

Williams, P. M. (1995). Bayesian regularization and pruning using a laplace prior. *Neural Computation*, *7*, 117–143.