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

Dmitry Kropotov DKROPOTOV@YANDEX.RU
Dorodnicyn Computing Centre of the Russian Academy of Sciences, 119333, Russia, Moscow, Vavilov str. 40

Dmitry Vetrov VETROVD@YANDEX.RU
Moscow State University, 119992, Russia, Moscow, Leninskie Gory, MSU, 2-nd educational building, CMC Department

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 us to optimize evidence explicitly as a product of one-dimensional integrals. The process of automatic regularization coefficients determination then converges in one iteration. We show how to use the proposed approach for Gaussian and Laplace 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 Laplace 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 Laplace 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 us 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 general since it does not depend on the particular form of prior

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

and only requires that priors regularize each eigenvector independently. 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 becomes 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 arising when one tries to apply Laplace prior within the framework. In Section 3 we describe our approach and illustrate its application for the Gaussian and Laplace 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 expression (1) can be approximated in the following way

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

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). Such formulation allowed him to apply ARD by iteratively adjusting $\vec{\alpha}$ and led to very sparse decision rules with the most of weights set to zero. Alternative method suggested in (Bishop & Tipping, 2000) uses variational inference to get better approximation of subintegral function with a Gaussian.

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

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 $H = \nabla_{\vec{w}}\nabla_{\vec{w}} \log P(\mathcal{T}|\vec{w}, \mathcal{X})|_{\vec{w}=\vec{w}_{ML}}$.
3: Make eigenvalues decomposition of Hessian $H = Q^T \Lambda Q$, $\Lambda = \mathrm{diag}(-h_1, \ldots, -h_M)$ 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 $\vec{x}$: $f(\vec{x}) = \mathrm{sign}\left(\sum_{i=1}^M w_{MP,i}\phi_i(\vec{x})\right)$

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 Laplace 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.

3. Proposed approach

Without loss of generality hereinafter we assume 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 approximate likelihood function with a Gaussian, treat eigenvectors of its Hessian matrix as new axes and make regularization as in usual sparse Bayesian learning along these new axes[1]. After such approximation of likelihood function evidence (2) can be written as:

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

where $\vec{\mu}$ and $(-H)^{-1}$ are the mean and the covariance matrix of the Gaussian with which we approximate likelihood. Hereinafter we will use Laplace (or saddle-point) approximation of likelihood that yields

$$\vec{\mu} = \vec{w}_{ML} = \arg\max_{\vec{w}} P(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha}), \quad (6)$$
$$H = \nabla_{\vec{w}}\nabla_{\vec{w}} \log P(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha})|_{\vec{w}=\vec{w}_{ML}}. \quad (7)$$

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 $h_i = -\lambda_i \geq 0$. We propose to introduce independent regularization with respect to new variables $\vec{u}$. This means that prior function can be written as

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

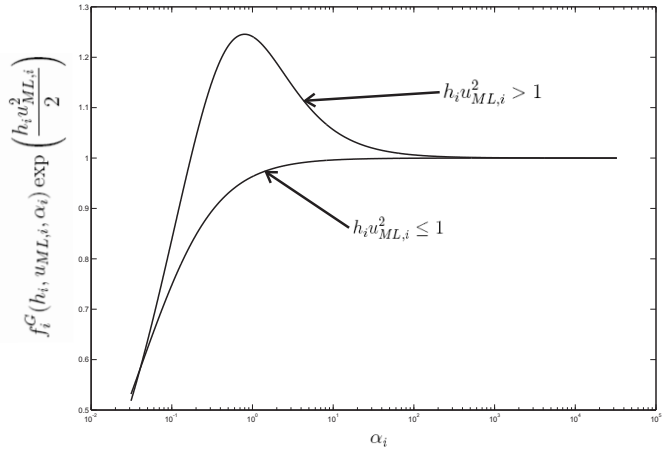

Figure 1. Behaviour of one-dimensional integral $f_i^G(h_i, u_{ML,i}, \alpha_i)$ depending on h_i and $u_{ML,i}$ in case of Gaussian prior. Function f_i^G is multiplied by exponent just for normalizing reason (both curves have the same limit).

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

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 (13) using one-dimensional optimization procedure:
$\alpha_i^* = \arg\max_{\alpha_i} f_i^L(h_i, u_{ML,i}, \alpha_i)$
end for
5: Find maximum of regularized log-likelihood function with respect to $\vec{u}$:
$\vec{u}_{MP} = \arg\max_{\vec{u}} \log P(\mathcal{T}|\mathcal{X}, Q^T\vec{u}) P(\vec{u}|\vec{\alpha}^*)$
under constraints $u_{ML,i} u_i \geq 0$ for all i.
6: Calculate the weights $\vec{w}_{MP} = Q^T \vec{u}_{MP}$.
output Decision rule for classification of new object $\vec{x}$: $f(\vec{x}) = \text{sign}\left(\sum_{i=1}^M w_{MP,i}\phi_i(\vec{x})\right)$

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

$$E(\vec{\alpha}) = P(\mathcal{T}|\mathcal{X}, \vec{u}_{ML}, \vec{\alpha}) \prod_{i=1}^M f_i(h_i, u_{ML,i}, \alpha_i) = P(\mathcal{T}|\mathcal{X}, \vec{u}_{ML}, \vec{\alpha}) \prod_{i=1}^M \int \exp\left(-\frac{h_i}{2}(u_i - u_{ML,i})^2\right) P(u_i|\alpha_i) du_i, \quad (8)$$

and then perform ARD procedure for setting hyperparameters $\vec{\alpha}$. We call this procedure Relevance Eigen Vector Machine (REVM). Note that in spite of Laplace approximation (6)-(7) we may use any other method which provides the approximation of likelihood or regularized likelihood with a Gaussian, e.g. variational bounds (Jaakkola & Jordan, 2000) or expectation propagation (Minka, 2001). The further regularization procedure remains unchanged.

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

3.1. Gaussian prior

Gaussian prior is given by the following expression

$$P(u_i|\alpha_i) = \sqrt{\frac{\alpha_i}{2\pi}} \exp\left(-\frac{\alpha_i {u_i}^2}{2}\right). \quad (9)$$

Consider one-dimensional integral $f_i(h_i, u_{ML,i}, \alpha_i)$ in expression (8) with prior (9). It can be computed analytically yielding:

$$f_i^G(h_i, u_{ML,i}, \alpha_i) = \sqrt{\frac{\alpha_i}{2\pi}} \int \exp\left(-\frac{h_i}{2}(u_i - u_{ML,i})^2 - \frac{\alpha_i}{2}u_i^2\right) du_i = \sqrt{\frac{\alpha_i}{h_i + \alpha_i}} \exp\left(-\frac{h_i \alpha_i u_{ML,i}^2}{2(h_i + \alpha_i)}\right) \quad (10)$$

Depending on h_i and $u_{ML,i}$ integral (10) has unique maximum or grows continuously as α_i tends to infinity (see fig. 1). Setting derivative of (10) with respect to α_i to zero, we obtain optimal value of α_i:

$$\alpha_i^* = \begin{cases} \frac{h_i}{h_i u_{ML,i}^2 - 1} & \text{if } h_i u_{ML,i}^2 > 1 \\ +\infty & \text{otherwise} \end{cases} \quad (11)$$

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.

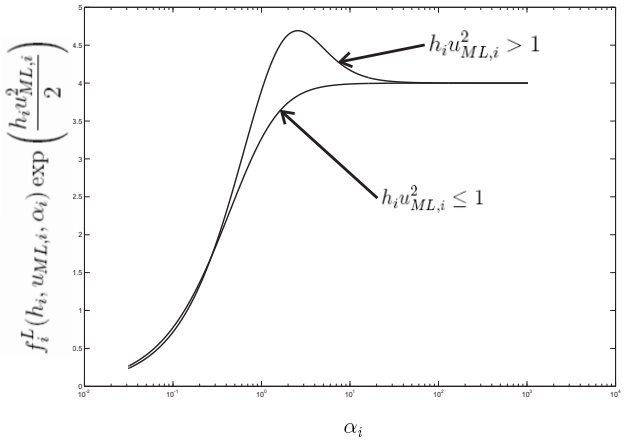

Figure 2. Behaviour of one-dimensional integral $f_i^L(h_i, u_{ML,i}, \alpha_i) \exp\left(\frac{h_i u_{ML,i}^2}{2}\right)$ depending on h_i and $u_{ML,i}$ in case of Laplace prior. Function f_i^L is multiplied by exponent just for normalizing reason (both curves have the same limit).

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

Data set	GREVM	LREVM	RVM	VRVM	VGREVM
Bupa	33.33 ± 3.26	30.67 ± 1.11	32.41 ± 2.56	30.78 ± 2.08	31.88 ± 0.65
Heart	18.15 ± 2.05	17.41 ± 0.79	17.26 ± 1.67	17.56 ± 2.72	23.11 ± 1.75
Hepatitis	14.32 ± 0.71	19.35 ± 0.79	20.26 ± 1.34	13.03 ± 1.06	17.81 ± 1.49
Votes	5.56 ± 0.57	5.93 ± 0.30	6.44 ± 2.04	5.61 ± 0.55	4.92 ± 0.76
WPBC	23.64 ± 1.15	23.13 ± 1.36	23.74 ± 0.36	24.04 ± 0.85	23.43 ± 0.28
Contractions	19.18 ± 2.21	17.96 ± 1.37	18.78 ± 1.85	17.35 ± 2.04	14.08 ± 3.18
Laryngeal1	16.81 ± 0.61	17.46 ± 1.12	17.46 ± 0.61	17.37 ± 0.57	18.31 ± 2.78
Respiratory	7.06 ± 1.18	7.53 ± 1.58	9.41 ± 2.50	8.24 ± 1.86	5.88 ± 1.18
Weaning	15.70 ± 4.76	14.83 ± 1.29	16.36 ± 1.36	13.64 ± 1.20	13.31 ± 2.06
Rank	**28.00**	**24.50**	**36.50**	**24.00**	**22.00**
Color	Place 1	Place 2	Place 3	Place 4	Place 5

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

Data set	GREVM	LREVM	RVM	VRVM	VGREVM
Bupa	138.72 ± 30.54	19.82 ± 0.61	64.14 ± 0.80	62.92 ± 3.03	310.30 ± 22.20
Heart	39.93 ± 5.52	11.64 ± 0.38	46.07 ± 0.43	21.47 ± 0.45	115.56 ± 2.60
Hepatitis	8.92 ± 0.35	4.88 ± 0.14	25.17 ± 0.37	6.88 ± 0.33	37.19 ± 1.03
Votes	123.76 ± 10.01	31.46 ± 0.47	87.38 ± 0.77	82.09 ± 3.57	433.28 ± 14.60
WPBC	19.67 ± 0.78	7.47 ± 0.14	33.06 ± 0.17	16.30 ± 0.52	89.72 ± 3.30
Contractions	7.49 ± 0.65	2.99 ± 0.05	16.10 ± 0.06	3.80 ± 0.27	22.26 ± 0.57
Laryngeal1	24.16 ± 4.04	8.73 ± 0.37	36.33 ± 0.54	18.59 ± 0.85	100.47 ± 4.30
Respiratory	4.51 ± 0.22	2.44 ± 0.03	13.75 ± 0.15	2.45 ± 0.05	14.93 ± 0.09
Weaning	53.16 ± 1.29	14.14 ± 0.18	53.17 ± 0.21	32.48 ± 0.87	177.16 ± 4.88

3.2. Laplace prior

Laplace prior function can be written as

$$P(u_i|\alpha_i) = \frac{\alpha_i}{4} \exp\left(-\frac{\alpha_i|u_i|}{2}\right). \tag{12}$$

Substituting (12) to (8) one-dimensional integral becomes

$$f_i^L(h_i, u_{ML,i}, \alpha_i) = \frac{\alpha_i}{4} \int \exp\left(-\frac{h_i}{2}(u_i - u_{ML,i})^2 - \frac{\alpha_i}{2}|u_i|\right) du_i = \frac{\alpha_i}{4}\sqrt{\frac{\pi}{2h_i}} \times \exp\left(-\frac{h_i u_{ML,i}^2}{2}\right) \left[\operatorname{erfcx}\left(\sqrt{\frac{h_i}{2}}\left(\frac{\alpha_i}{2h_i} - u_{ML,i}\right)\right) + \operatorname{erfcx}\left(\sqrt{\frac{h_i}{2}}\left(\frac{\alpha_i}{2h_i} + u_{ML,i}\right)\right)\right], \tag{13}$$

where $\operatorname{erfcx}(x) = \frac{2}{\sqrt{\pi}} \exp(x^2) \int_x^{+\infty} \exp\left(-t^2\right) dt$ - the scaled complementary error function[2]. See Appendix A for further considerations of stable calculation of expression (13). Last equation defines 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 Laplace REVM (LREVM) optimization of $\vec{\alpha}$ values can 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. However taking into account the fact that the point $\vec{u}_{MP}$ is located within the same hyperoctant (with respect to $\vec{u}$ variable) as the point $\vec{u}_{ML}$ we may reduce the

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

Table 3. Rate of non-zero parameters together with standard deviations

Data set	#Obj./2	GREVM	LREVM	RVM	VRVM	VGREVM
Bupa	172	23.10 ± 14.06	8.20 ± 1.44	5.80 ± 0.97	172.50 ± 0.00	17.20 ± 1.35
Heart	135	12.10 ± 4.02	9.00 ± 0.94	6.00 ± 1.06	135.00 ± 0.00	86.60 ± 13.53
Hepatitis	77	10.20 ± 2.68	4.60 ± 1.92	3.10 ± 1.08	77.50 ± 0.00	39.40 ± 17.26
Votes	217	14.60 ± 3.66	6.60 ± 1.14	4.70 ± 0.57	217.50 ± 0.00	58.70 ± 21.76
WPBC	99	20.80 ± 2.49	2.20 ± 1.04	2.40 ± 1.39	99.00 ± 0.00	27.00 ± 22.84
Contractions	49	14.50 ± 12.34	3.30 ± 1.35	2.30 ± 0.76	49.00 ± 0.00	34.20 ± 3.35
Laryngeal1	106	7.20 ± 1.79	3.90 ± 1.39	1.90 ± 0.55	106.40 ± 0.22	57.10 ± 22.58
Respiratory	42	5.70 ± 1.25	2.50 ± 0.61	1.60 ± 0.42	42.50 ± 0.00	22.40 ± 1.71
Weaning	151	12.90 ± 4.22	10.10 ± 2.07	6.80 ± 1.57	151.00 ± 0.00	120.20 ± 10.22

problem to an optimization of smooth function under constraints which define the borders of hyperoctant containing the point $\vec{u}_{ML}$.

4. Experiments

In this section we compared original 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) and Pattern Recognition Research Group at University of Wales website[3]. We used known algorithm proposed by Tipping and Faul for training RVM (Tipping & Faul, 2003). Also we added variational variants for both RVM and GREVM denoted by VRVM and VGREVM respectively. 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 so on, the loser gets five points, and then points are summed for all data sets.

These results allow us to make the following conclusions. All algorithms show comparable performance in terms of error rates. However, the rate of non-zero parameters in GREVM and especially LREVM is significantly less than corresponding value in RVM. Note however that in REVM the number of non-zero parameters corresponds to eigenvectors (or degrees of freedom) and hence the obtained sparsity is implicit. All the original weights $\vec{w}_{MP}$ demanded to classify new objects generally differ from zero. So this approach is not applicable if one needs to select a subset of relevant features or training objects.

GREVM seems to be faster than RVM as optimization 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 slower for other datasets. On the one hand, LREVM benefits in training time since it has one-step optimization of regularization coefficients. On the other hand LREVM requires M one-dimensional optimizations for estimation of $\vec{\alpha}$ as unlike Gaussian prior case here we cannot get explicit equations, and additional constrained optimization for finding $\vec{u}_{MP}$.

Variational approximation requires significantly more time for training. In VGREVM we first find Gaussian variational approximation of an unregularized likelihood by minimizing $KL(\mathcal{N}(\vec{\mu}, -H^{-1}) \| P(\mathcal{T}|\mathcal{X}, \vec{w}, \vec{\alpha}))$. Then we regularize eigenvectors of H just as in case of Laplace approximation (6)-(7). Note that VGREVM is even slower than VRVM due to the fact that it takes quite a long time to approximate unregularized likelihood with a Gaussian. Probably the process can be accelerated using some small initial regularizer over $\|\vec{w}\|$. On the other hand the accuracy of variational methods seems to be a little bit better although further careful research is necessary to make any conclusions. The rate of non-zero weights is greater than in case of Laplace approximation but it's general property of variational approach. Analogously we may use

[3] http://www.informatics.bangor.ac.uk/∼kuncheva/activities/patrec1.html

variational approximation in LREVM.

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 Laplace prior. Here $A = \mathrm{diag}(\alpha_1, \ldots, \alpha_M)$ and $Q = \{q_{ij}\}_{i,j=1}^{M}$. We claim that the number of the degrees of freedom 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.

More sparse classifiers in REVM indirectly indicate 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. We suppose that eigenvalues and directions of eigenvectors present characteristic features of likelihood function and they are responsible for generalization ability of a classifier. If our suggestion is true then the weights of classifier are secondary with respect to generalization and it is eigenvectors that are to be regularized directly rather than the weights.

It seems very promising to consider the regularization of generalized linear models by using arbitrary non-negative symmetric regularization matrix R_0. The coefficients of R_0 then can be found by optimizing evidence

$$R_0 = \arg\max_{\mathcal{R}} P(\mathcal{T}|\mathcal{X}, \vec{w}) P(\vec{w}|R),$$

where $\mathcal{R} = \{R \in \mathbb{R}^{M\times M} \mid R^T = R,\ R \geq 0\}$ and $P(\vec{w}|R) = \sqrt{\frac{\det(R)}{2\pi^M}} \exp\left(-\frac{1}{2}\vec{w}^T R \vec{w}\right)$. Such matrix can be found analytically. We consider it as one of the directions for the future work.

6. Acknowledgements

We would like to thank the reviewers for their useful remarks and valuable suggestions. The work was supported by Russian Foundation for Basic Research (grants no. 06-01-08045, 05-01-00332, 05-07-90333, 07-01-00211).

References

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.

Jaakkola, T., & Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, *10*, 25–37.

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

Minka, T. (2001). Expectation propagation for approximate bayesian inference. *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence* (pp. 362–369). Morgan Kaufmann.

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.

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.

A. Evidence efficient calculations for LREVM

Consider calculation of evidence (13) for the case of Laplace prior. Expression (13) can be written as

$$C\exp\left(-\frac{h_i u_{ML,i}^2}{2}\right)\times \left[\exp\left(x_1^2\right)\operatorname{erfc}(x_1)+\exp\left(x_2^2\right)\operatorname{erfc}(x_2)\right], \quad (14)$$

where C is some positive constant, $\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty \exp(-t^2)dt$ – complementary error function and $x_{1,2} = \sqrt{\frac{h_i}{2}}\left(\frac{\alpha_i}{2h_i} \mp u_{ML,i}\right)$. For large positive values of x the product $\exp(x^2)\operatorname{erfc}(x)$ is convenient to write as scaled complementary error function

$$\operatorname{erfcx}(x) = \exp(x^2)\operatorname{erfc}(x) \approx 1/(\sqrt{\pi}x).$$

For large negative values of x it is reasonable to unite $\exp(-h_i u_{ML,i}^2/2)$ and $\exp(x^2)$ in one expression yielding

$$\exp(-h_i u_{ML,i}^2/2)\exp(x^2) = \exp(y),$$

where

$$y_{1,2} = \frac{\alpha_i^2}{8h_i} \mp \frac{\alpha_i u_{ML,i}}{2}.$$