Bayesian Support Vector Machines for Feature Ranking and Selection

Wei Chu[1], S. Sathiya Keerthi[2], Chong Jin Ong[3], and Zoubin Ghahramani[1]

[1] Gatsby Computational Neuroscience Unit, University College London, London, WC1N 3AR, UK. chuwei@gatsby.ucl.ac.uk, zoubin@gatsby.ucl.ac.uk
[2] Yahoo! Research Lab., Pasadena, CA 91105, USA. sathiya.keerthi@overture.com
[3] Department of Mechanical Engineering, National University of Singapore, Singapore, 119260. mpeongcj@nus.edu.sg

In this chapter, we develop and evaluate a feature selection algorithm for Bayesian support vector machines. The relevance level of features are represented by ARD (automatic relevance determination) parameters, which are optimized by maximizing the model evidence in the Bayesian framework. The features are ranked in descending order using the optimal ARD values, and then forward selection is carried out to determine the minimal set of relevant features. In the numerical experiments, our approach using ARD for feature ranking can achieve a more compact feature set than standard ranking techniques, along with better generalization performance.

1 Introduction

In the classical supervised learning task, we are given a training set of fixed-length feature vectors along with target values, from which to learn a mathematical model that represents the mapping function between the feature vectors and the target values. The model is then used to predict the target for previously unseen instances. The problem of feature selection can be defined as finding relevant features among the original feature vector, with the purpose of increasing the accuracy of the resulting model or reducing the computational load associated with high dimensional problems.

Many approaches have been proposed for feature selection. In general, they can be categorized along two lines as defined by John et al. (1994):

- Filters: the feature selector is independent of a learning algorithm and serves as a pre-processing step to modelling. There are two well-known filter methods, FOCUS and RELIEF. The FOCUS algorithm carries out an exhaustive search of all feature subsets to determine the minimal set

of features using a consistency criterion (Almuallim and Dietterich 1991). RELIEF (Kira and Rendell 1992) is a randomized algorithm, which attempts to give each feature a weighting indicating its level of relevance to the targets.

- Wrapper: this approach searches through the space of feature subsets using the estimated accuracy from a learning algorithm as the measure of the goodness for a particular feature subset (Langley and Sage 1994). This method is restricted by the time complexity of the learning algorithm, and when the number of features is large, it may become prohibitively expensive to run.

There are some learning algorithms which have built-in feature selection. Jebara and Jaakkola (2000) formalized a kind of feature weighting in the maximum entropy discrimination framework, and Weston et al. (2001) introduced a method of feature selection for support vector machines by minimizing the bounds on the leave-one-out error. MacKay (1994) and Neal (1996) proposed automatic relevance determination (ARD) as a hierarchical prior over the weights in Bayesian neural networks. The weights connected to an irrelevant input can be automatically punished with a tighter prior in model adaptation, which reduces the influence of such a weight towards zero effectively. In Gaussian processes, the ARD parameters can be directly embedded into the covariance function (Williams and Rasmussen 1996) which results in a type of feature weighting.

In this paper, we applied Bayesian support vector machines (BSVM) (Chu et al. 2003, 2004) with ARD techniques to select relevant features. BSVM, which is rooted in the probabilistic framework of Gaussian processes, can be regarded as a support vector variant of Gaussian processes. The sparseness in Bayesian computation helps us to tackle relatively large data sets. Bayesian techniques are used to carry out model adaptation. The optimal values of the ARD parameters can be inferred intrinsically in the modelling. Relevance variables are introduced to indicate the relevance level for features. The features can then be ranked from relevant to irrelevant accordingly. In our feature selection algorithm, a forward selection scheme is employed to determine the minimal set of relevant features.

The rest of this paper is organized as follows. Section 2 reviews the techniques of BSVM to estimate the optimal values for ARD parameters. Section 3 describes a forward selection scheme as post-processing to select the minimal set of relevant features. Section 4 presents the results of numerical experiments, followed by the conclusion in Section 5.

2 Bayesian Framework

As computationally powerful tools for supervised learning, support vector machines (SVMs) were introduced by Boser et al. (1992), and have been widely

used in classification and regression problems (Vapnik 1995). Let us suppose that a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, \ldots, m\}$ is given for training, where the feature vector $\mathbf{x}_i \in \mathbb{R}^n$ and y_i is the target value. In regression, the target is a real value, while in classification the target is the class label. SVMs map these feature vectors into a high dimensional reproducing kernel Hilbert space (RKHS), where the optimal values of the discriminant function $\{f(\mathbf{x}_i)|i = 1, 2, \ldots, m\}$ can be computed by minimizing a regularized functional. The regularized functional is defined as

$$\min_{\boldsymbol{f} \in \mathrm{RKHS}} \mathcal{R}(\boldsymbol{f}) = \sum_{i=1}^{m} \ell(y_i, f(\mathbf{x}_i)) + \frac{1}{C} \|\boldsymbol{f}\|^2_{\mathrm{RKHS}}, \tag{1}$$

where the regularization parameter C is positive, the stabilizer $\|\boldsymbol{f}\|^2_{\mathrm{RKHS}}$ is a norm in the RKHS and $\sum_{i=1}^{m} \ell\big(y_i, f(\mathbf{x}_i)\big)$ is the empirical loss term (Evgeniou et al. 1999). For various loss functions, the regularized functional (1) can be minimized by solving a convex quadratic programming optimization problem that guarantees a unique global minimum solution. In SVMs for classification (Burges 1998), hard margin, L_1 soft margin and L_2 soft margin loss functions are widely used. For regression, Smola and Schölkopf (1998) have discussed a lot of common loss functions, such as Laplacian, Huber's, ϵ-insensitive and Gaussian etc.

If we assume that a prior $\mathcal{P}(\boldsymbol{f}) \propto e^{-\frac{1}{C}\|\boldsymbol{f}\|^2_{\mathrm{RKHS}}}$ and a likelihood $\mathcal{P}(\mathcal{D}|\boldsymbol{f}) \propto e^{-\sum_{i=1}^{m} \ell(y_i, f(\mathbf{x}_i))}$, the minimizer of regularized functional (1) can be directly interpreted as maximum a posteriori (MAP) estimate of the function $\boldsymbol{f}$ in the RKHS (Evgeniou et al. 1999). Due to the duality between RKHS and stochastic processes (Wahba 1990), the functions $f(\mathbf{x}_i)$ in the RKHS can also be explained as a family of random variables in a Gaussian process. Gaussian processes have provided a promising non-parametric Bayesian approach to classification problems (Williams and Barber 1998). The important advantage of Gaussian process models over other non-Bayesian models is the explicit probabilistic formulation. This not only provides probabilistic class prediction but also gives the ability to infer model parameters in Bayesian framework. We follow the standard Gaussian process to describe a Bayesian framework, in which we impose a Gaussian process prior distribution on these functions and employ particular loss functions in likelihood evaluation. Compared with standard Gaussian processes, the particular loss function results in a different convex programming problem for computing MAP estimates and leads to sparseness in computation.

2.1 Prior Probability

The functions in the RKHS (or latent functions) are usually assumed as the realizations of random variables indexed by the input vector $\mathbf{x}_i$ in a stationary zero-mean Gaussian process. The Gaussian process can then be specified by giving the covariance matrix for any finite set of zero-mean random variables

$\{f(\mathbf{x}_i)|i = 1, 2, \ldots, m\}$. The covariance between the outputs corresponding to the inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ can be defined by Mercer kernel functions, such as Gaussian kernel, polynomial kernels and spline kernels (Wahba 1990). We list two popular covariance functions with ARD parameters in the following:

- ARD Gaussian Kernel: this is a generalization of the popular Gaussian kernel defined as

$$Cov[f(\mathbf{x}_i), f(\mathbf{x}_j)] = \kappa_0 \exp\left(-\frac{1}{2}\sum_{l=1}^{n} \kappa_{a,l}(\mathbf{x}_{i,l} - \mathbf{x}_{j,l})^2\right) + \kappa_b, \qquad (2)$$

  where l is the feature index, $\kappa_0 > 0$, $\kappa_{a,l} > 0$ and $\kappa_b > 0$. κ_0 denotes the average power of $f(\mathbf{x})$ that reflects the noise level, while κ_b corresponds to the variance of the offset in the latent functions.[4] The ARD parameters $\kappa_{a,l}$ $\forall l$ are used to indicate the relevance level of the l-th feature to the targets. Note that a relatively large ARD parameter implies that the associated feature gives more contributions to the modelling, while a feature weighted with a very small ARD parameter implies that this feature is irrelevant to the targets.
- ARD Linear Kernel: this is a type of linear kernel parameterized with ARD parameters defined as

$$Cov[f(\mathbf{x}_i), f(\mathbf{x}_j)] = \sum_{l=1}^{n} \kappa_{a,l}\, \mathbf{x}_{i,l}\, \mathbf{x}_{j,l} + \kappa_b, \qquad (3)$$

  where $\kappa_b > 0$ and $\kappa_{a,l} > 0$.

We collect the parameters in the covariance function (2) or (3), as θ, the hyperparameter vector. Then, for a given hyperparameter vector θ, the prior probability of the random variables $\{f(\mathbf{x}_i)\}$ is a multivariate Gaussian, which can be simply written as

$$\mathcal{P}(\boldsymbol{f}\,|\theta) = \frac{1}{Z_{\boldsymbol{f}}}\exp(-\frac{1}{2}\boldsymbol{f}^T \Sigma^{-1} \boldsymbol{f}), \qquad (4)$$

where $\boldsymbol{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_m)]^T$, $Z_{\boldsymbol{f}} = (2\pi)^{\frac{m}{2}}|\Sigma|^{\frac{1}{2}}$, and Σ is the $m \times m$ covariance matrix whose ij-th element is $Cov[f(\mathbf{x}_i), f(\mathbf{x}_j)]$.

2.2 Likelihood

We usually assume that the training data are collected independently. The probability $\mathcal{P}(\mathcal{D}|\boldsymbol{f}, \theta)$, known as the likelihood, can be evaluated by:

[4] In classification settings, it is possible to insert a "jitter" term in the diagonal entries of the covariance matrix, that could reflect the uncertainty in the corresponding function value.

$$\mathcal{P}(\mathcal{D}|\boldsymbol{f},\theta) = \prod_{i=1}^{m} \mathcal{P}(y_i|f(\mathbf{x}_i)), \tag{5}$$

where $-\ln \mathcal{P}(y_i|f(\mathbf{x}_i))$ is usually referred to as the loss function $\ell(y_i, f(\mathbf{x}_i))$.

In regression problems, the discrepancy between the target value y_i and the associated latent function $f(\mathbf{x}_i)$ at the input $\mathbf{x}_i$ is evaluate by a specific noise model. Various loss functions can be used depending on the assumption on the distribution of the additive noise (Chu et al. 2004). In this paper, we focus on binary classification only.

For binary classifier designs, we measure the probability of the class label y_i for a given latent function $f(\mathbf{x}_i)$ at $\mathbf{x}_i$ as the likelihood, which is a conditional probability $\mathcal{P}(y_i|f(\mathbf{x}_i))$.[5] The logistic and probit functions are widely used in likelihood evaluation (Williams and Barber 1998; Neal 1997b). However these do not result in sparse solutions to the optimization problem. In order to introduce sparseness into this Bayesian framework, Chu et al. (2003) proposed a trigonometric loss function. The trigonometric loss function is defined as

$$\ell_t(y_i, f(\mathbf{x}_i)) = \begin{cases} +\infty & \text{if } \ y_i \cdot f(\mathbf{x}_i) \in (-\infty, -1]; \\ 2\ln\sec(\frac{\pi}{4}(1 - y_i \cdot f(\mathbf{x}_i))) & \text{if } \ y_i \cdot f(\mathbf{x}_i) \in (-1, +1); \\ 0 & \text{if } \ y_i \cdot f(\mathbf{x}_i) \in [+1, +\infty), \end{cases} \tag{6}$$

The trigonometric likelihood function is therefore written as

$$\mathcal{P}_t(y_i|f(\mathbf{x}_i)) = \begin{cases} 0 & \text{if } \ y_i \cdot f(\mathbf{x}_i) \in (-\infty, -1]; \\ \cos^2(\frac{\pi}{4}(1 - y_i \cdot f(\mathbf{x}_i))) & \text{if } \ y_i \cdot f(\mathbf{x}_i) \in (-1, +1); \\ 1 & \text{if } \ y_i \cdot f(\mathbf{x}_i) \in [+1, +\infty). \end{cases} \tag{7}$$

Note that $\mathcal{P}_t(y_i = +1|f(\mathbf{x}_i)) + \mathcal{P}_t(y_i = -1|f(\mathbf{x}_i)) = 1$ always holds for any value of $f(\mathbf{x}_i)$, and the trigonometric loss function possesses a flat zero region that is the same as the L_1 and L_2 loss functions in support vector machines.

2.3 Posterior Probability

Based on Bayes' theorem, the posterior probability of $\boldsymbol{f}$ can then be written as

$$\mathcal{P}(\boldsymbol{f}\,|\mathcal{D},\theta) = \frac{1}{Z_S}\exp\left(-S(\boldsymbol{f})\right), \tag{8}$$

where $S(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^T\Sigma^{-1}\boldsymbol{f} + \sum_{i=1}^{n}\ell(y_i, f(\mathbf{x}_i))$, $\ell(\cdot)$ is the loss function we used and $Z_S = \int \exp(-S(\boldsymbol{f}))\,d\boldsymbol{f}$. Since $\mathcal{P}(\boldsymbol{f}|\mathcal{D},\theta) \propto \exp(-S(\boldsymbol{f}))$, the Maximum A Posteriori (MAP) estimate on the values of $\boldsymbol{f}$ is therefore the minimizer of the following optimization problem:

[5] Here, y_i is a discrete random variable, and the sum of the probabilities for all possible cases of y_i should be equal to 1, i.e. $\sum_{y_i}\mathcal{P}(y_i|f(\mathbf{x}_i)) = 1$, which is referred to as the *normalization requirement*.

$$\min_{\boldsymbol{f}} S(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^T \Sigma^{-1} \boldsymbol{f} + \sum_{i=1}^{m} \ell(y_i, f(\mathbf{x}_i)). \tag{9}$$

This is a regularized functional. For any differentiable loss function, the solution of the regularization functional $S(\boldsymbol{f})$, is always a linear superposition of covariance functions, one for each data point. This elegant form of a minimizer of (9) is also known as the representer theorem (Kimeldorf and Wahba 1971). A generalized representer theorem can be found in Schölkopf et al. (2001), in which the loss function is merely required to be a strictly monotonically increasing function $\ell : \mathbb{R} \to [0, +\infty)$.

2.4 MAP Estimate

Introducing the trigonometric loss function (6) into the regularized functional of (9), the optimization problem (9) can be then restated as the following equivalent optimization problem, which we refer to as the *primal* problem:

$$\min_{\boldsymbol{f},\boldsymbol{\xi}} \frac{1}{2}\boldsymbol{f}^T \Sigma^{-1} \boldsymbol{f} + 2\sum_{i=1}^{m} \ln \sec\left(\frac{\pi}{4}\xi_i\right) \tag{10}$$

subject to $y_i \cdot f(\mathbf{x}_i) \geq 1 - \xi_i$ and $0 \leq \xi_i < 2, \forall i$. Standard Lagrangian techniques (Fletcher 1987) are used to derive the *dual* problem. The *dual* problem can be finally simplified as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(y_i\alpha_i)(y_j\alpha_j)Cov\left[f(\mathbf{x}_i), f(\mathbf{x}_j)\right] - \sum_{i=1}^{m}\alpha_i \\ + \sum_{i=1}^{m} n\left[\frac{4}{\pi}\alpha_i \arctan\left(\frac{2\alpha_i}{\pi}\right) - \ln\left(1 + \left(\frac{2\alpha_i}{\pi}\right)^2\right)\right] \end{aligned} \tag{11}$$

subject to $\alpha_i \geq 0, \forall i$. Refer to Chu et al. (2003) for the derivation details.

The *dual* problem (11) is a convex programming problem. The popular SMO algorithm for classical SVMs (Platt 1999; Keerthi et al. 2001) can be easily adapted to solve the optimization problem (Chu et al. 2003). The MAP estimate on the values of the random variables $\boldsymbol{f}$ can be written in column vector form as

$$\boldsymbol{f}_{\mathrm{MP}} = \Sigma \cdot \boldsymbol{v} \tag{12}$$

where $\boldsymbol{v} = [y_1\alpha_1, y_2\alpha_2, \ldots, y_m\alpha_m]^T$. The training samples $(\mathbf{x}_i, y_i)$ associated with non-zero Lagrange multiplier α_i are called *support vectors* (SVs). The other samples associated with zero α_i do not involve in the solution representation and the following Bayesian computation. This property is usually referred to as sparseness, and it reduces the computational cost significantly.

2.5 Hyperparameter Inference

The hyperparameter vector θ contains the ARD parameters and other parameters in the covariance function. The optimal values of hyperparameters θ can be inferred by maximizing the posterior probability $\mathcal{P}(\theta|\mathcal{D})$, using $\mathcal{P}(\theta|\mathcal{D}) = \mathcal{P}(\mathcal{D}|\theta)\mathcal{P}(\theta)/\mathcal{P}(\mathcal{D})$. The prior distribution on the hyperparameters $\mathcal{P}(\theta)$ can be specified by domain knowledge. As we typically have little idea about the suitable values of θ before training data are available, we usually assume a flat distribution for $\mathcal{P}(\theta)$, i.e., $\mathcal{P}(\theta)$ is greatly insensitive to the values of θ. Therefore, $\mathcal{P}(\mathcal{D}|\theta)$ (which is known as the evidence of θ) can be used to assign a preference to alternative values of the hyperparameters θ (MacKay 1992).

The evidence is given by an integral over all $\boldsymbol{f}$: $\mathcal{P}(\mathcal{D}|\theta) = \int \mathcal{P}(\mathcal{D}|\boldsymbol{f},\theta)\mathcal{P}(\boldsymbol{f}|\theta)\, d\boldsymbol{f}$. Using the definitions in (4) and (5), the evidence can also be written as

$$\mathcal{P}(\mathcal{D}|\theta) = \frac{1}{Z_{\boldsymbol{f}}} \int \exp(-S(\boldsymbol{f}))\, d\boldsymbol{f}, \tag{13}$$

where $S(\boldsymbol{f})$ is defined as in (8). A major technical difficulty is the difficulty of computing the evidence as a high dimensional integral. So far, a variety of approximation techniques have been discussed: Monte Carlo sampling (Neal 1997a), the MAP approach (Williams and Barber 1998), bounds on the likelihood (Gibbs 1997) and mean field approaches (Opper and Winther 2000; Csató et al. 2000). Recently, Kim and Ghahramani (2003) coupled the Expectation Propagation algorithm (Minka 2001) with variational methods (Seeger 1999) for evidence maximization. To maintain the sparseness in Bayesian computation, we apply Laplace approximation at the MAP estimate. The evidence (13) could be calculated by an explicit formula, and then hyperparameter inference can be done by gradient-based optimization methods.

The marginalization can be done analytically by considering the Taylor expansion of $S(\boldsymbol{f})$ around its minimum $S(\boldsymbol{f}_{\mathrm{MP}})$, and retaining terms up to the second order. Since the first order derivative with respect to $\boldsymbol{f}$ at the MAP point $\boldsymbol{f}_{\mathrm{MP}}$ is zero, $S(\boldsymbol{f})$ can be written as

$$S(\boldsymbol{f}) \approx S(\boldsymbol{f}_{\mathrm{MP}}) + \frac{1}{2}(\boldsymbol{f} - \boldsymbol{f}_{\mathrm{MP}})^T \left(\frac{\partial^2 S(\boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^T}\bigg|_{\boldsymbol{f}=\boldsymbol{f}_{\mathrm{MP}}} \right) (\boldsymbol{f} - \boldsymbol{f}_{\mathrm{MP}}), \tag{14}$$

where $\frac{\partial^2 S(\boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^T} = \Sigma^{-1} + \Lambda$, and Λ is a diagonal matrix coming from the second order derivative of the loss function we used. Introducing (14) into (13) yields

$$\mathcal{P}(\mathcal{D}|\theta) = \exp(-S(\boldsymbol{f}_{\mathrm{MP}})) \cdot |\mathbf{I} + \Sigma \cdot \Lambda|^{-\frac{1}{2}},$$

where $\mathbf{I}$ is the $m \times m$ identity matrix. Note that, when the trigonometric loss function (6) is employed, only a sub-matrix of Σ plays a role in the determinant $|\mathbf{I} + \Sigma \cdot \Lambda|$ due to the sparseness of the diagonal matrix Λ in which only the entries associated with SVs are non-zero. We denote their submatrices as Σ_{M} and Λ_{M} respectively by keeping their non-zero entries. The

MAP estimate of $\boldsymbol{f}$ (12) on support vectors can also be simplified as $\Sigma_{\mathrm{M}} \cdot \boldsymbol{\upsilon}_{\mathrm{M}}$, where $\boldsymbol{\upsilon}_{\mathrm{M}}$ denotes the sub-vector of $\boldsymbol{\upsilon}$ by keeping entries associated with SVs. Because of these sparseness properties, the negative log of the evidence can then be simplified as follows

$$-\ln \mathcal{P}(\mathcal{D}|\theta) = \frac{1}{2}\boldsymbol{\upsilon}_{\mathrm{M}}^T \cdot \Sigma_{\mathrm{M}} \cdot \boldsymbol{\upsilon}_{\mathrm{M}} + 2 \sum_{v \in \mathrm{SVs}} \ln \sec\left(\frac{\pi}{4}\xi_v\right) + \frac{1}{2}\ln|\mathbf{I} + \Sigma_{\mathrm{M}} \cdot \Lambda_{\mathrm{M}}|, \quad (15)$$

where $\mathbf{I}$ is the identity matrix with the size of SVs, "$v \in$ SVs" denotes that v is varied over the index set of SVs, and $\xi_v = 1 - y_v \cdot f_{\mathrm{MP}}(\mathbf{x}_v)$, $\forall v$. The evidence evaluation is a convenient yardstick for model selection. It is straightforward to consider the posterior distribution $\mathcal{P}(\theta|\mathcal{D}) \propto \mathcal{P}(\mathcal{D}|\theta)\mathcal{P}(\theta)$ by specifying a particular prior distribution $\mathcal{P}(\theta)$. The gradient of $-\ln \mathcal{P}(\theta|\mathcal{D})$ with respect to the variables in the hyperparameter vector θ can be explicitly derived (see Chu et al. (2003) for the detailed derivatives), and then gradient-based optimization methods can be used to find the minimum locally.

The optimization method usually requests evidence evaluation at tens of different θ before the minimum is found. For each θ, a quadratic programming problem should be solved first to find MAP estimate, and then the approximate evidence (15) is calculated along with its gradients with respect to the hyperparameters. Due to the sparseness, the quadratic programming problem costs almost the same time as SVMs at the scale about $\mathcal{O}(m^{2.2})$, where m is the size of training data. In gradient evaluations for ARD parameters, the inversion of the matrix Σ_{M}, corresponding to SVs at the MAP estimate, is required that costs time at $\mathcal{O}(s^3)$ for each feature, where s is the number of SVs that is usually much less than m.

3 Post-processing for Feature Selection

The generalization performance of BSVM with ARD techniques are very competitive (Chu et al. 2003). In practical applications, it might be desirable to further select a minimal subset of relevant features for modelling while keeping the accuracy of the resulting model and reducing the computational load. In this section, we describe our method for feature selection based on the techniques described in the previous section. The task of feature selection can be tackled in two steps:

1. The original features can be ranked in descending order using the optimal values of the ARD parameters $\{\kappa_a^l\}_{l=1}^m$ we inferred (see Section 3.1).
2. Then a subset of the top features in the rank is used as the relevant features for modelling. The minimal subset can be determined by the validation performance of the learning machine (see Section 3.2).

3.1 Feature Ranking

We first introduce a set of relevance variables $\{r^i\}_{i=1}^n$ for the features we are given, which are extracted from the ARD parameters by normalizing them:

$$r^i = \frac{\kappa_{a,i}}{\sum_{j=1}^n \kappa_{a,j}}. \tag{16}$$

The relevance variable indicates the relevance level of the feature to the targets, which is independent of the overall scale of ARD parameters.

Since there might be several local minima on the curve of $-\ln \mathcal{P}(\theta|\mathcal{D})$, it is possible that the optimization problem is stuck at local minima in the determination of θ.[6] We may reduce the impact of this problem by minimizing (15) several times starting from several different initial states, and simply choosing the one with the highest evidence as our preferred choice for θ. We can also organize these candidates together to represent the evidence distribution that might reduce the uncertainty with respect to the hyperparameters. An approximation scheme is described in the following.

Suppose we started from several different initial states, and reached several minima θ_τ^* of the optimization problem (15). We simply assume that the underlying distribution is a superposition of individual distributions with mean θ_τ^*. The underlying distribution $\mathcal{P}(\theta|\mathcal{D})$ is roughly reconstructed as

$$\mathcal{P}(\theta|\mathcal{D}) \approx \sum_{\tau=1}^{t} w_\tau \mathcal{P}_\tau(\theta; \theta_\tau^*) \tag{17}$$

with

$$w_\tau = \frac{\mathcal{P}(\theta_\tau^*|\mathcal{D})}{\sum_{\iota=1}^{t} \mathcal{P}(\theta_\iota^*|\mathcal{D})}, \tag{18}$$

where t is the number of the minima we have discovered by gradient descent methods, and $\mathcal{P}_\tau(\theta; \theta_\tau^*)$ denotes the individual distribution which can be any distribution with the mean θ_τ^*. The values of $\mathcal{P}(\theta_\tau^*|\mathcal{D})$ have been obtained from the functional (15) already. In Figure 1, we present a simple case as an illustration.

We can evaluate the expected values of the relevance variables based on the approximated $\mathcal{P}(\theta|\mathcal{D})$ (17) as follows

$$\hat{r^i} \approx \sum_{\tau=1}^{t} w_\tau r_\tau^i, \tag{19}$$

where r_τ^i is defined as in (16) at the minimum θ_τ^*. This is not an approximation designed for the posterior mean of the relevance variables, but helpful to overcome the difficulty caused by the poor minima. Using these values $\{\hat{r^i}\}_{i=1}^d$, we can sort the features in descending order from relevant to irrelevant.

[6] Monte Carlo sampling methods can provide a good approximation on the posterior distribution, but might be prohibitively expensive to use for high-dimensional problems.

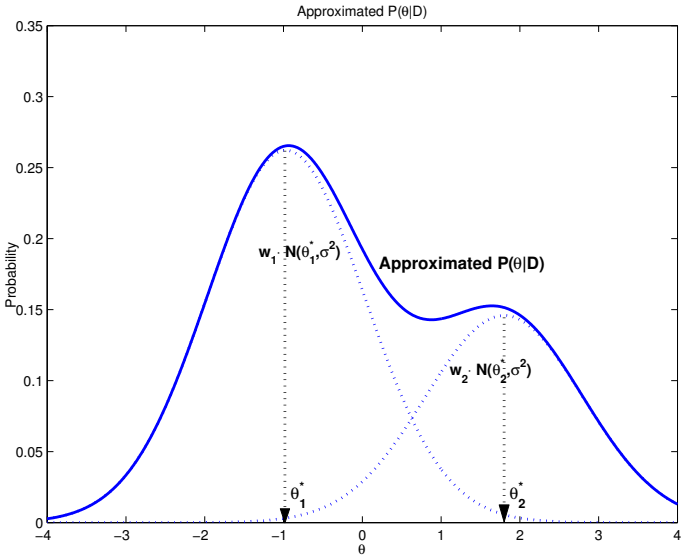

Fig. 1. An illustration on the reconstructed curve for the posterior distribution $\mathcal{P}(\theta|\mathcal{D})$. θ_1^* and θ_2^* are the two maxima we found by gradient descent methods. The underlying distribution $\mathcal{P}(\theta|\mathcal{D})$ can be simply approximated as the superposition of two individual distributions (dotted curves) with mean θ_1^* and θ_2^* respectively. The weights w_1 and w_2 for the two distributions are defined as in (18). In this graph, Gaussian distributions with same variance were used as individual distribution.

3.2 Feature Selection

Given any learning algorithm, we can select a minimal subset of relevant features by carrying out cross validation experiments on progressively larger sets of features, adding one feature at a time ordered by their relevance variables $\hat{r_i}$.[7] Let S denote the set of relevant features being investigated. The features top-ranked according to $\hat{r^i}$ defined as in (19) are added into the set S one by one, and the validation error is calculated. This procedure is repeated as long as adding the next top-ranked feature into S does not increase the validation error significantly. This feature set S is then used along with all the training data for modelling.

3.3 Discussion

RELIEF (Kira and Rendell 1992) also attempts to specify relevance level for features, but the ARD techniques (MacKay 1994; Neal 1996) can carry out Bayesian inference systematically. In our approach, the performance of

[7] The relevance variables of useless features are usually much less than the average level $1/n$, where n is the total number of features.

Table 1. The outline of our algorithm for feature selection.

Ranking	employ a kind of ARD kernel randomly select starting points θ_0 for Optimization Package while Optimization Package requests evidence/gradient evaluation at θ solve (11) by convex programming to find MAP estimate evaluate the approximate evidence (13) calculate the gradients with respect to θ return evidence/gradient to the optimization package Optimization Package returns the optimal θ^* compute the relevance variables defined as in (19) ranking the features in descending order
Selection	initialize validation error to infinity, and $k = 0$ do $k = k + 1$ use the top k features as input vector to a learning algorithm carry out cross validation via grid searching on model parameters pick up the best validation error while validation error is not increasing significantly
Exit	return the top $k - 1$ features as the minimal subset.

a learning algorithm is used as the criterion to decide the minimal subset; this is analogous to the wrapper approach (Langley and Sage 1994). The key difference is that we only check the subsets including the top-ranked feature sequentially rather than search through the huge space of feature subsets.

A potential problem may be caused by correlations between features. Such correlations introduce dependencies into the ARD variables. More exactly, at the minima, correlated features may share their ARD values randomly. This makes it difficult to distinguish relevant features based on their ARD values alone. In this case, we suggest a backward elimination process (Guyon et al. 2002). This process begins with the full set and remove the most irrelevant feature one by one. At each step, we carry out inference on the reduced dataset to update their relevance variables. This procedure is computationally expensive, since it requires performing hyperparameter inference m times.

4 Numerical Experiments

We give an outline of the algorithm for feature selection in Table 1.[8] The feature vectors with continuous elements were normalized to have zero mean and unit variance coordinate-wise. The ARD parameters were used for feature weighting. The initial value of the hyperparameters were chosen as $\kappa_0 = 1.0$

[8]The source code of Bayesian support vector machines in ANSI C can be accessed at http://guppy.mpe.nus.edu.sg/∼chuwei/btsvc.htm.

Table 2. NIPS 2003 challenge results we submitted. "Score" denotes the score used to rank the results by the organizers (times 100). "BER" denotes balanced error rate (in percent). "AUC" is the area under the ROC curve (times 100). "Feat" is percent of features used. "Probe" is the percent of probes found in the subset selected. "Test" is the result of the comparison with the best entry using the MacNemar test.

Dec. 1^{st}	Our challenge entry					The winning challenge entry					
Dataset	Score	BER	AUC	Feat	Probe	Score	BER	AUC	Feat	Probe	Test
Overall	15.27	9.43	95.70	67.53	38.03	88.00	6.84	97.22	80.3	47.8	0.6
Arcene	78.18	15.17	91.52	100	30	98.18	13.30	93.48	100.0	30.0	0
Dexter	-21.82	6.35	98.57	36.04	60.15	96.36	3.90	99.01	1.5	12.9	1
Dorothea	-25.45	15.47	92.06	100	50	98.18	8.54	95.92	100.0	50.0	1
Gisette	-47.27	2.62	99.67	100	50	98.18	1.37	98.63	18.3	0.0	1
Madelon	100.00	7.17	96.95	1.6	0.0	100.00	7.17	96.95	1.6	0.0	0

and $\kappa_b = 10.0$. We tried ten times starting from different values of κ_a^l to maximize the evidence by gradient descent methods.[9] The ten maxima we found are used to estimate the values of the relevance variables as in (19). In the forward feature selection, SVMs with Gaussian kernel, $\exp(-\frac{\kappa}{2}\|x_i - x_j\|^2)$, was used as the learning algorithm. Grid search in the parameter space spanned by the κ in the Gaussian kernel and the regularization factor C as in (1), was carried out to locate the best validation output. The primary grid search was done on a 7×7 coarse grid linearly spaced in the region $\{(\log_{10} C, \log_{10} \kappa) | -0.5 \leq \log_{10} C \leq 2.5, -3 \leq \log_{10} \kappa \leq 0\}$, followed by a fine search on a 9×9 uniform grid linearly spaced by 0.1 in the $(\log_{10} C, \log_{10} \kappa)$ space. At each node in this grid, 5-fold cross validation was repeated ten times to reduce the bias in fold generations, and the validation errors were averaged over all the trials.

The NIPS 2003 challenge results we submitted are reported in Table 2. On the Madelon data set, we carried out feature selection as described in Table 1. For other datasets, we submitted the predictive results of linear SVMs with L_1 loss function without strict feature selection at that time.[10] The "AUC" performance of our entry is competitive with that of the winning entry.

The Madelon task[11] is to categorize random data into two classes. There are 2000 samples for training and 1800 samples for test. Each sample has 500 features. In Figure 2, we present the top 13 features ranked by the estimated values of relevance variables. The performance of 5-fold cross validation (SVMs with L_1 loss function and Gaussian kernel was used) is presented in the right

[9] The number of the minima should be large enough to reproduce the results stably.

[10] On the Dexter dataset, we simply removed the features with zero weight in the optimal linear SVMs, and then retrained linear SVMs on the reduced data.

[11] The data set can be found at http://clopinet.com/isabelle/Projects/NIPS2003/.

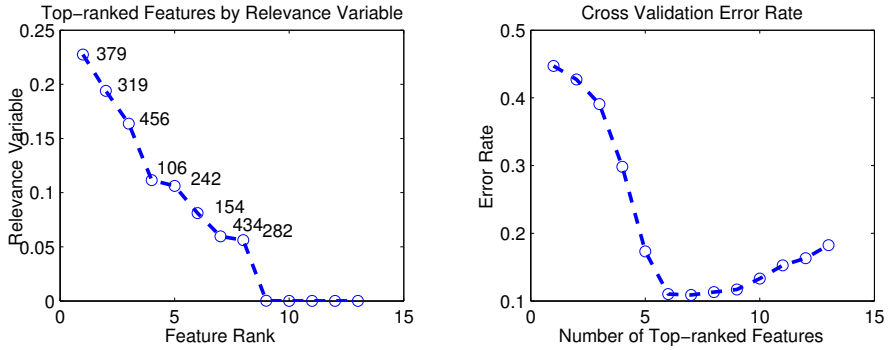

Fig. 2. The result on Madelon data set for our algorithm. The values of relevance variables $\hat{r}^i$ for the top 13 features are presented in the left part along with the feature indices, while the validation error rates are plotted in the right part.

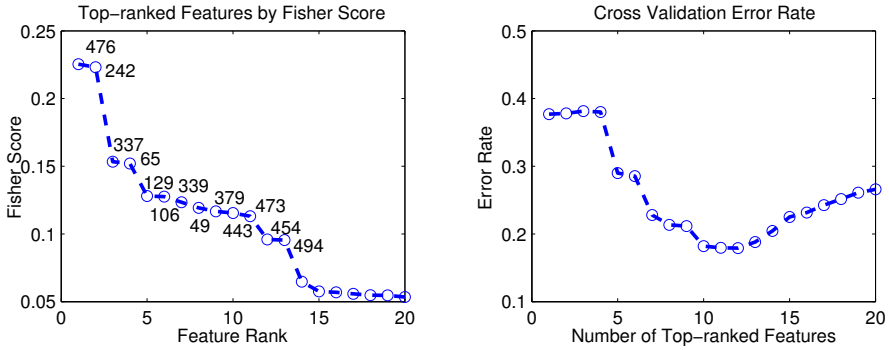

Fig. 3. The result on Madelon data set using fisher score for feature ranking. The fisher scores of the 13 top-ranked features are presented in the left part along with the feature indices, while the validation error rates are plotted in the right part.

part of Figure 2. In the challenge, we selected the top 8 features for modelling, and built up SVMs using ARD Gaussian kernel with fixed relevance variables shown in the left part of Figure 2. Cross validation was carried out to decide the optimal regularization factor and the common scale parameter in the Gaussian kernel. The blind test on the 1800 samples got 7.17% error rate. This entry was assigned the highest score by the organizers.

In the next experiment, the popular Fisher score was used in feature ranking for comparison purpose. The Fisher score (Golub et al. 1999) is defined as

$$s_i = \frac{|\mu_{i,+} - \mu_{i,-}|}{\sigma_{i,+} + \sigma_{i,-}}, \tag{20}$$

where $\mu_{i,+}$ and $\sigma_{i,+}$ are the mean and standard deviation of the i-th feature on the positive samples, while $\mu_{i,-}$ and $\sigma_{i,-}$ are of the negative samples. In

Figure 3, the top-ranked features by their Fisher score (20) are presented in the left graph, and the validation error rate using the top-ranked features incrementally are presented in the right graph. The best validation result is about 0.18 using 12 features, which is much worse than the result of our algorithm as shown in Figure 2.

5 Conclusion

In this chapter, we embedded automatic relevance determination in Bayesian support vector machines to evaluate feature relevance. The Bayesian framework provides various computational procedures for hyperparameter inference which can be used for feature selection. The sparseness property introduced by our loss function (6) helps us to tackle relatively large data sets. A forward selection method is used to determine the minimal subset of informative features. Overall we have a probabilistic learning algorithm with built-in feature selection that selects informative features automatically. The results of numerical experiments show that this approach can achieve quite compact feature sets, and achieve good generalization performance.

Acknowledgment

This work was supported by the National Institutes of Health and its National Institute of General Medical Sciences division under Grant Number 1 P01 GM63208 (NIH/NIGMS grant title: Tools and Data Resources in Support of Structural Genomics).

References

Almuallim, H. and T. G. Dietterich. Learning with many irrelevant features. In *Proc. AAAI-91*, pages 547–552. MIT Press, 1991.

Boser, B., I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifier. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, USA, 1992.

Burges, C. J. C. A tutorial on support vector machines for pattern recogintion. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Chu, W., S. S. Keerthi, and C. J. Ong. Bayesian trigonometric support vector classifier. *Neural Computation*, 15(9):2227–2254, 2003.

Chu, W., S. S. Keerthi, and C. J. Ong. Bayesian support vector regression using a unified loss function. *IEEE transactions on neural networks*, 15(1):29–44, 2004.

Csató, L., E. Fokoué, M. Opper, B. Schottky, and O. Winther. Efficient approaches to Gaussian process classification. In *Advances in Neural Information Processing Systems*, volume 12, pages 251–257, 2000.

Evgeniou, T., M. Pontil, and T. Poggio. A unified framework for regularization networks and support vector machines. A.I. Memo 1654, MIT, 1999.

Fletcher, R. *Practical methods of optimization.* John Wiley and Sons, 1987.

Gibbs, M. N. *Bayesian Gaussian Processes for Regression and Classification.* Ph.D. thesis, University of Cambridge, 1997.

Golub, T., D. Slonim, P. Tamaya, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

Guyon, I., J. Weston, and S. Barnhill. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

Jebara, T. S. and T. S. Jaakkola. Feature selection and dualities in maximum entropy discrimination. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, pages 291–300, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

John, G., R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proc. ML-94*, pages 121–129. Morgan Kaufmann Publishers, 1994.

Keerthi, S. S., S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13 (3):637–649, March 2001.

Kim, H. and Z. Ghahramani. The EM-EP algorithm for Gaussian process classification. In *Proc. of the Workshop on Probabilistic Graphical Models for Classification (at ECML)*, 2003.

Kimeldorf, G. S. and G. Wahba. Some results on Tchebycheffian spline function. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

Kira, K. and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proc. AAAI-92*, pages 129–134. MIT Press, 1992.

Langley, P. and S. Sage. Induction of selective Bayesian classifiers. In *Proc. UAI-94*, pages 399–406. Morgan Kaufmann, 1994.

MacKay, D. J. C. A practical Bayesian framework for back propagation networks. *Neural Computation*, 4(3):448–472, 1992.

MacKay, D. J. C. Bayesian methods for backpropagation networks. *Models of Neural Networks III*, pages 211–254, 1994.

Minka, T. P. *A family of algorithm for approximate Bayesian inference.* Ph.D. thesis, Massachusetts Institute of Technology, January 2001.

Neal, R. M. *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics. Springer, 1996.

Neal, R. M. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report No. 9702, Department of Statistics, University of Toronto, 1997a.

Neal, R. M. Regression and classification using Gaussian process priors (with discussion). In Bernerdo, J. M., J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics*, volume 6, 1997b.

Opper, M. and O. Winther. Gaussian processes for classification: Mean field algorithm. *Neural Computation*, 12(11):2655–2684, 2000.

Platt, J. C. Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, 1999.

Schölkopf, B., R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Proc. of the Annual Conference on Computational Learning Theory*, 2001.

Seeger, M. Bayesian model selection for support vector machines, Gaussian processes and other kernel classifiers. In *Advances in Neural Information Processing Systems*, volume 12, 1999.

Smola, A. J. and B. Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, GMD First, October 1998.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

Wahba, G. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1990.

Weston, J., S. Mukherjee, O. Chapelle, M. Pontil, and T. Poggio. Feature selection in SVMs. In Leen, T., T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, 2001. MIT Press.

Williams, C. K. I. and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

Williams, C. K. I. and C. E. Rasmussen. Gaussian processes for regression. In Touretzky, D. S., M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 598–604, 1996. MIT Press.