The Evidence Framework Applied to Support Vector Machines

James Tin-Yau Kwok

Abstract— **In this paper, we show that training of the support vector machine (SVM) can be interpreted as performing the level 1 inference of MacKay's evidence framework. We further on show that levels 2 and 3 of the evidence framework can also be applied to SVMs. This integration allows automatic adjustment of the regularization parameter and the kernel parameter to their near-optimal values. Moreover, it opens up a wealth of Bayesian tools for use with SVMs. Performance of this method is evaluated on both synthetic and real-world data sets.**

Keywords— **Support vector machine, Evidence framework, Bayesian inference**

I. Introduction

In recent years, there has been a lot of interest in studying the support vector machine (SVM) [1], [2], [3], [4], [5]. SVM is based on the idea of *structural risk minimization* (SRM) [4], which shows that the generalization error is bounded by the sum of the training set error and a term depending on the Vapnik-Chervonenkis dimension of the learning machine. By minimizing this upper bound, high generalization performance can be achieved. Moreover, unlike other machine learning methods, SVM's generalization error is related not to the input dimensionality of the problem, but to the margin with which it separates the data. This explains why SVMs can have good performance even in problems with a large number of inputs [6], [7]. To date, SVM has been successfully applied to a wide range of problems, including pattern recognition, regression, time series prediction and density estimation.

However, to obtain a high level of performance, some parameters in the SVM still have to be tuned. These include:

- a regularization parameter, which determines the tradeoff between minimizing training errors and minimizing model complexity; and
- a kernel parameter, which implicitly defines the high dimensional feature space to be used.

These parameters are sometimes just hand-picked by the user. A more disciplined approach is to use a validation set [8], or by data-resampling techniques such as cross-validation and bootstrapping. However, these methods can be very expensive in terms of computation time and/or training data. Alternatively, one can utilize an upper bound on the generalization error predicted by the theory of SRM [4]. Experiments in [1], [4], [7] indicated that the bounds are very loose, though the minimum of the bound seems to approximately coincide with the minimum of the generalization error.

James T. Kwok is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: jamesk@comp.hkbu.edu.hk.

On the other hand, this problem of finding good parameters also exists in the realm of feedforward neural networks. For example, one has to set a regularization parameter when using weight decay. Also, one needs to determine some model parameters (such as the number of hidden layers in the network and the number of hidden units in each layer) in order to obtain an optimal network architecture for a particular application. Recently, various researchers [9], [10], [11], [12], [13], [14] have applied Bayesian methods to tackle these problems. In general, the Bayesian approach is attractive in being logically consistent, simple and flexible. Compared with the traditional approach, the Bayesian methods mentioned above provide a rigorous framework for the automatic adjustment of the regularization parameters to their near-optimal values, without the need to set data aside in a validation set. Moreover, the Bayesian framework allows objective comparison among solutions using different network architectures. Bayesian techniques also offer some other important features. For example, in regression problems, error bars can be assigned to network predictions [11]. In classification problems, by using the moderated outputs [15], the tendency by conventional approaches of making over-confident predictions in regions of sparse data can be avoided. Among others, Bayesian techniques have also been used for active learning [16] and in forming a committee of networks [13]. It is thus promising to integrate SVMs with these Bayesian ideas.

A Bayesian interpretation of the SVM has been proposed by Smola *et al.* [17]. In particular, they showed that the use of different kernels in SVM can be regarded as defining different *prior* probability distributions on the function space, as $P[f] \propto \exp(-\beta\|\hat{P}f\|^2)$. Here, $\beta > 0$ is a constant and $\hat{P}$ is the regularization operator corresponding to the selected kernel. This prior, however, is based on a function-space view and cannot be readily incorporated into popular Bayesian techniques like [11], [13], whose priors are based on a weight-space view.

In this paper, we develop an alternate Bayesian interpretation of SVM from a weight-space view and then apply a well-known Bayesian approach, MacKay's evidence framework [11], to SVM. We will focus our attention on classification problems. The rest of this paper is organized as follows. Brief introductions to the SVM and the evidence framework are given in Sections II and III respectively. Section IV discusses how these two methods can be combined, and then be used to determine near-optimal values for the regularization parameter and the kernel parameter. Simulation results are presented in Section V, and the last section gives some concluding remarks.

II. Support Vector Machines for Classification

In this section, we briefly review the use of SVMs in classification problems. For more details and also on the use of SVMs in other kinds of problems, interested readers may consult [1], [4], [5].

Let the training set D be $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with each input $\mathbf{x}_i \in \Re^m$ and the output label $y_i \in \{\pm 1\}$. The SVM first maps $\mathbf{x}$ from the input space $\Re^m$ to $\mathbf{z} = \phi(\mathbf{x})$ in a feature space $\mathcal{F}$. Consider the case when the data is linearly separable in $\mathcal{F}$, i.e., there exists a vector $\mathbf{w} \in \mathcal{F}$ and a scalar $b \in \Re$ such that $y_i(\mathbf{w}^T\mathbf{z}_i + b) \geq 1$ for all patterns in the training set. The SVM constructs a hyperplane $(\mathbf{w}^T\mathbf{z} + b)$ for which the separation between the positive and negative examples is maximized. It can be shown [2] that the $\mathbf{w}$ for this "optimal" hyperplane can be found by minimizing $\|\mathbf{w}\|$, and the resultant solution can be written as $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{z}_i$ for some $\alpha_i \geq 0$. This vector of α_i's, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$, can be found by solving the following quadratic programming (QP) problem: maximize

$$W(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}, \tag{1}$$

with respect to $\boldsymbol{\alpha}$, under the constraints $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\boldsymbol{\alpha}^T\mathbf{y} = 0$, where $\mathbf{y}^T = (y_1, \ldots, y_N)$ and $\mathbf{Q}$ is a symmetric $N \times N$ matrix with elements $Q_{ij} = y_i y_j \mathbf{z}_i^T \mathbf{z}_j$. To obtain Q_{ij}, one does not need to use the mapping ϕ to explicitly get $\mathbf{z}_i$ and $\mathbf{z}_j$. Instead, under certain conditions, one can find a *kernel* $K(\cdot,\cdot)$ such that $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_i^T\mathbf{z}_j$. Moreover, notice that $\mathbf{Q}$ is always positive semi-definite and so there is no local optima while maximizing (1). For those α_i's greater than zero, the corresponding training examples must lie along the margins of the decision boundary (by the Kuhn-Tucker theorem), and these are called the *support vectors*.

During testing, for a test vector $\mathbf{x} \in \Re^m$, we first compute the activation

$$a(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T\mathbf{z} + b = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b. \tag{2}$$

The class label $o(\mathbf{x})$ for $\mathbf{x}$ is then assigned by the following rule:

$$o(\mathbf{x}) = \begin{cases} 1 & a(\mathbf{x}; \mathbf{w}) > 0, \\ -1 & \text{otherwise.} \end{cases}$$

When the training set is not separable in $\mathcal{F}$, the SVM algorithm introduces non-negative slack variables $\xi_i \geq 0,\ i = 1, \ldots, N$ [2]. The resultant problem becomes

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^N \xi_i, \tag{3}$$

subject to $y_i a(\mathbf{x}_i, \mathbf{w}) \geq 1 - \xi_i,\ i = 1, \ldots, N$. Here, C is a regularization parameter controlling the tradeoff between model complexity (the first term in (3)) and training error (the second term) in order to ensure good generalization performance. The variable ξ_i, whenever it is nonzero, measures the (absolute) difference between y_i and $a_i = a(\mathbf{x}_i, \mathbf{w})$, and may be written concisely as

$$\xi_i = [1 - y_i a_i]_+, \tag{4}$$

where $[u]_+ = uI_{\{u>0\}}$. Again, minimization of (3) can be transformed to a QP problem: maximize (1) subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}$ and $\boldsymbol{\alpha}^T\mathbf{y} = 0$.

III. The Evidence Framework

The evidence framework is a Bayesian framework proposed by MacKay [11], [15]. Computationally, it is equivalent to the *type II maximum likelihood* method in Bayesian statistics [18]. The evidence framework has been applied successfully to the learning of feedforward neural networks in both classification and regression problems. In this Section, we review the evidence framework as applied to classification problems [15].

First, we introduce some notations used in the evidence framework. A model $\mathcal{H}$, with a k-dimensional parameter vector $\mathbf{w}$, consists of its functional form f, the distribution $p(D|\mathbf{w}, \mathcal{H})$ that the model makes about the data D, and a prior parameter distribution $p(\mathbf{w}|\mathcal{H}, \lambda)$, which is usually written in the form:

$$p(\mathbf{w}|\mathcal{H}, \lambda) = \frac{\exp(-\lambda E_W(\mathbf{w}|\mathcal{H}))}{Z_W(\lambda)}. \tag{5}$$

Here, λ is a regularization parameter and $Z_W(\lambda) = \int d^k\mathbf{w} \exp(-\lambda E_W)$ is for normalization.

A. Level 1 Inference

The evidence framework is divided into three levels of inference. For a given value of λ, the first level of inference infers the posterior distribution of $\mathbf{w}$ by the Bayes rule:

$$p(\mathbf{w}|D, \lambda, \mathcal{H}) \propto p(D|\mathbf{w}, \mathcal{H}) p(\mathbf{w}|\lambda, \mathcal{H}). \tag{6}$$

Substituting in (5), the posterior distribution of $\mathbf{w}$ then becomes

$$p(\mathbf{w}|D, \lambda, \mathcal{H}) = \frac{\exp(-M(\mathbf{w}))}{Z_M(\lambda)}, \tag{7}$$

where $M(\mathbf{w}) \stackrel{\text{def}}{=} \lambda E_W(\mathbf{w}) - \log p(D|\mathbf{w}, \mathcal{H})$, and $Z_M(\lambda) = \int d^k\mathbf{w} \exp(-M)$. Minimizing M is thus the same as finding the *maximum a posteriori* (MAP) estimate $\mathbf{w}_{MP}$ of $\mathbf{w}$. In the sequel, we will denote $\log p(D|\mathbf{w}, \mathcal{H})$ simply as $G(\mathbf{w})$.

B. Level 2 Inference

The second level of inference determines the value of λ, by maximizing $p(\lambda|D, \mathcal{H}) \propto p(D|\lambda, \mathcal{H})p(\lambda|\mathcal{H})$. When $p(\lambda|\mathcal{H})$ is a flat prior, the evidence for λ, $p(D|\lambda, \mathcal{H})$, can be used to assign a preference to alternative values of λ. By approximating[1] the posterior distribution of $\mathbf{w}$ in (7) by a single Gaussian at $\mathbf{w}_{MP}$, as $M(\mathbf{w}) = M(\mathbf{w}_{MP}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{MP})^T\mathbf{A}(\mathbf{w} - \mathbf{w}_{MP})$, where $\mathbf{A} = \nabla^2 M$, the evidence for λ can be obtained by integrating out $\mathbf{w}$ as [11]:

$$\log p(D|\lambda, \mathcal{H}) = -\lambda E_W^{MP} + G^{MP} - \frac{1}{2}\log\det\mathbf{A} + \frac{k}{2}\log\lambda, \tag{8}$$

[1]Notice that while we need to take the Gaussian approximation to the posterior distribution of $\mathbf{w}$, no such approximation is needed for $p(D|\mathbf{w}, \mathcal{H})$.

where E_W^{MP} and G^{MP} are the values of E_W and G evaluated at $\mathbf{w}_{MP}$.

One can also obtain an near-optimal value of λ in an iterative manner. First, by setting the derivative of (8) to zero, the following condition for the most probable value of λ, λ_{MP}, can be obtained:

$$2\lambda_{MP}E_W^{MP} = \gamma, \tag{9}$$

where

$$\gamma \stackrel{\text{def}}{=} k - \lambda \operatorname{trace}\mathbf{A}^{-1} \tag{10}$$

is called the *effective number of parameters*. An near-optimal value of λ is then obtained by iterating the process of finding $\mathbf{w}_{MP}$ and re-estimating λ by (9).

C. Level 3 Inference

The third level of inference in the evidence framework ranks different models by examining their posterior probabilities $p(\mathcal{H}|D) \propto p(D|\mathcal{H})p(\mathcal{H})$. Assuming a flat prior $p(\mathcal{H})$ for all models, different models can then be rated by their evidence $p(D|\mathcal{H})$. Again, this is obtained by integrating out λ, as $p(D|\mathcal{H}) = \int p(D|\lambda, \mathcal{H})p(\lambda|\mathcal{H})d\lambda$. Using a Gaussian approximation for $p(D|\lambda, \mathcal{H})$, it can be shown that [11]:

$$p(D|\mathcal{H}) \propto p(D|\lambda_{MP}, \mathcal{H})/\sqrt{\gamma}. \tag{11}$$

IV. Applying the Evidence Framework to SVM

In Section IV-A, we develop a Bayesian interpretation for the SVM from a weight-space view and show that minimizing (3) during SVM training can be interpreted as performing the level 1 inference of the evidence framework. Then in Sections IV-B and IV-C, we proceed further and use levels 2 and 3 of the evidence framework to determine the regularization parameter and the kernel parameter.

A. A Bayesian Interpretation for SVM

Assuming that the patterns are i.i.d., then $p(D|\mathbf{w}, \mathcal{H}) = \prod_i p(\mathbf{x}_i, y_i|\mathbf{w}, \mathcal{H}) = \prod_i p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H})p(\mathbf{x}_i)$, and (6) becomes

$$p(\mathbf{w}|D, \lambda, \mathcal{H}) \propto p(\mathbf{w}|\lambda, \mathcal{H}) \prod_i p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H})p(\mathbf{x}_i). \tag{12}$$

Consider the following probability model:

- The prior over $\mathbf{w}$ is the Gaussian prior $p(\mathbf{w}|\lambda, \mathcal{H}) \propto \exp(-\frac{\lambda}{2}\|\mathbf{w}\|^2)$, and
- The probability density function $p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H})$ for $y_i = \pm 1$ is given by

$$p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H}) = \frac{\exp(-[1 - y_i a_i]_+)}{\exp(-[1 - a_i]_+) + \exp(-[1 + a_i]_+)}.$$

Substituting these probabilities into (12), we obtain

$$\begin{aligned} -\log p(\mathbf{w}|D, \lambda, \mathcal{H}) &= \frac{\lambda}{2}\|\mathbf{w}\|^2 \\ &- \sum_i \log\left(\frac{\exp(-[1 - y_i a_i]_+)}{\exp(-[1 - a_i]_+) + \exp(-[1 + a_i]_+)}\right) \\ &- \sum_i \log p(\mathbf{x}_i) + \text{constant}. \end{aligned}$$

This cannot be cast readily under the SVM framework. However, if we take the approximation that

$$p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H}) \simeq \exp(-[1 - y_i a_i]_+) = \exp(-\xi_i),$$

using (4). Then, on substituting this approximated probability model back into (12), we get

$$\begin{aligned} -\log p(\mathbf{w}|D, \lambda, \mathcal{H}) &= \\ \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i \xi_i - \sum_i \log p(\mathbf{x}_i) &+ \text{constant}. \end{aligned}$$

The last two terms on the right do not depend on $\mathbf{w}$. Hence, by setting $C = 1/\lambda$, optimizing (3) can be regarded[2] as finding the MAP estimate $\mathbf{w}_{MP}$ of $\mathbf{w}$. In other words, training of the SVM can be regarded as approximately performing the first level of inference in the evidence framework. A comparison of the true and approximated probability distributions is shown in Figure 1.

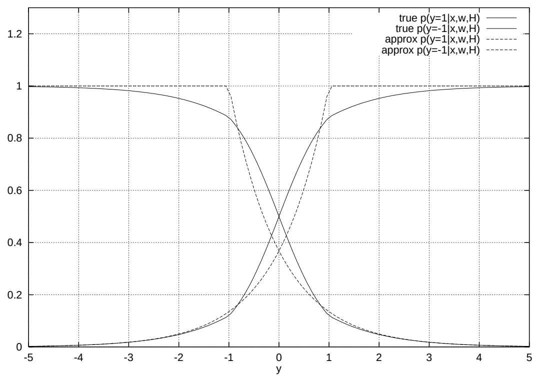


Fig. 1. True and approximated probabilities.

Note that this Bayesian interpretation can also be applied to regression problems, with ξ_i being replaced by a different loss function (such as the ϵ-insensitive loss function). Again, each loss function corresponds to a different noise model. For an overview of some common loss functions and their corresponding noise models, interested readers may consult [3].

Considering the case when the training set is separable in $\mathcal{F}$, let α_{max} be the largest Lagrangian multiplier in the set of support vectors. We can view the training process as minimizing (3) with $C = \alpha_{max}$. With a larger $C > \alpha_{max}$, both $\mathbf{w}$ and ξ_i will remain unchanged. Hence, effectively, we can take $C = \alpha_{max}$ in (3).

B. Computing the Hessian

To proceed under the evidence framework, we next have to determine the hessian

$$\mathbf{A} = \nabla^2 M = \nabla^2(\lambda E_W + \sum_{i=1}^{N} \xi_i). \tag{13}$$

[2]Notice that the constraints $y_i a_i \geq 1 - \xi_i,\ i = 1, \ldots, N$ in (3) have been implicitly taken care of by the equation $[1 - y_i a_i]_+ = \xi_i$.

Recall that $\xi_i = [1 - y_i a_i]_+$ with $[u]_+ = uI_{\{u>0\}}$. However, $I_{\{u>0\}}$ is not smooth and does not have second derivative. Hence, we replace it by the sigmoid function

$$s(u) = \frac{1}{1+e^{-\eta u}}, \quad \eta > 0, \tag{14}$$

and ξ_i becomes $(1-y_i a_i)s(1-y_i a_i)$. Differentiating with respect to $\mathbf{w}$, we obtain $\nabla \xi_i = -y_i(1-y_i a_i)s'(1-y_i a_i)\nabla a_i - y_i s(1-y_i a_i)\nabla a_i$, where the prime denotes the derivative with respect to the argument of $s(\cdot)$. From (2), $\nabla a_i = \mathbf{z}_i$ and $\nabla^2 a_i = 0$. Differentiating $\nabla \xi_i$ once more, we obtain $\nabla^2 \xi_i = r(|y_i - a_i|)\mathbf{z}_i \mathbf{z}_i^T \stackrel{\text{def}}{=} r_i \mathbf{z}_i \mathbf{z}_i^T$, where $r(u) \stackrel{\text{def}}{=} us''(u) + 2s'(u)$. Figure 2 shows a plot of $r(u)$, which has the shape of a Mexican hat and is concentrated around $u = 0$.

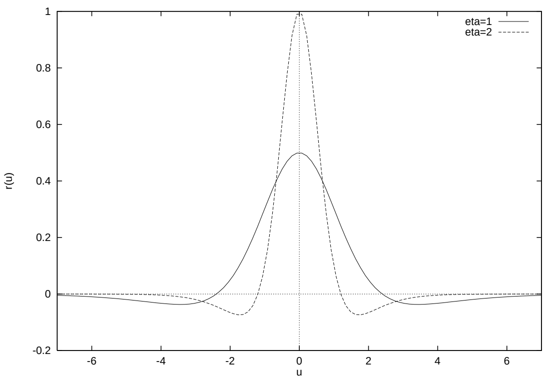


Fig. 2. $r(u)$.

Noting that $\nabla^2 E_W = \mathbf{I}$, (13) thus becomes

$$\mathbf{A} = \lambda \mathbf{I} + \mathbf{B}, \tag{15}$$

where

$$\mathbf{B} = \sum_{i=1}^{N} r_i \mathbf{z}_i \mathbf{z}_i^T. \tag{16}$$

Denote the eigenvalues and eigenvectors of $\mathbf{B}$ by ρ_l and $\mathbf{v}_l$ respectively. As $\rho_l \mathbf{v}_l = \mathbf{B}\mathbf{v}_l = \sum_{i=1}^{N} r_i(\mathbf{z}_i^T \mathbf{v}_l)\mathbf{z}_i$, all $\mathbf{v}_l$'s with $\rho_l \neq 0$ must lie in the span of $\mathbf{z}_i, \ldots, \mathbf{z}_N$, i.e.

$$\mathbf{v}_l = \sum_{i=1}^{N} \mu_{li} \mathbf{z}_i. \tag{17}$$

Moreover, (17) implies that there are at most N independent $\mathbf{v}_l$'s (and at most N nonzero eigenvalues). For a particular $\mathbf{z}_k = \phi(\mathbf{x}_k) \in \mathcal{F}$, consider $\rho_l \mathbf{z}_k^T \mathbf{v}_l = \mathbf{z}_k^T \mathbf{B} \mathbf{v}_l$. Using (16) and (17), we have

$$\begin{aligned} \rho_l \sum_{i=1}^{N} \mu_{li} \mathbf{z}_k^T \mathbf{z}_i &= \mathbf{z}_k^T (\sum_{j=1}^{N} r_j \mathbf{z}_j \mathbf{z}_j^T) \sum_{i=1}^{N} \mu_{li} \mathbf{z}_i \\ &= \sum_{i=1}^{N} \mu_{li} \sum_{j=1}^{N} r_j \mathbf{z}_k^T \mathbf{z}_j \mathbf{z}_j^T \mathbf{z}_i, \quad k = 1, \ldots, N. \end{aligned}$$

Or, in matrix form, $\rho_l \mathbf{K} \boldsymbol{\mu}_l = \mathbf{K}\tilde{\mathbf{K}} \boldsymbol{\mu}_l$, where $\boldsymbol{\mu}_l$ is the N-vector $(\mu_{l1}, \ldots, \mu_{lN})^T$, $\mathbf{K}$ is the $N \times N$ matrix with entries $\mathbf{z}_i^T \mathbf{z}_j = K(\mathbf{x}_i, \mathbf{x}_j)$, and $\tilde{\mathbf{K}}$ is another $N \times N$ matrix with entries $r_i \mathbf{z}_i^T \mathbf{z}_j = r_i K(\mathbf{x}_i, \mathbf{x}_j)$. Assuming that $\mathbf{K}$ is invertible, we have

$$\rho_l \boldsymbol{\mu}_l = \tilde{\mathbf{K}} \boldsymbol{\mu}_l. \tag{18}$$

Solving this eigen system and using (15), we can obtain the eigenvalues $\hat{\rho}_l$ of $\mathbf{A}$ as

$$\hat{\rho}_l = \begin{cases} \lambda + \rho_l & l = 1, \ldots, N, \\ \lambda & l = N+1, N+2, \ldots \end{cases} \tag{19}$$

The solving of the eigen system in (18) has $O(N^3)$ time complexity, which can be computationally expensive for large N. However, as can be seen from Figure 2, the value of r_i is very small when $|y_i - a_i|$ is large, and so $\mathbf{B}$ is dominated by patterns whose a_i is almost the same as y_i. We can thus reduce the complexity significantly by only including those patterns in the computation.

C. Levels 2 and 3 Inference for SVM

Level 2 inference determines the value of λ by maximizing $p(D|\lambda, \mathcal{H})$ in (8). In the following, denote the number of nonzero eigenvalues of $\tilde{\mathbf{K}}$ by $n \leq N$. Then, using (19), we obtain

$$\begin{aligned} &\log p(D|\lambda, \mathcal{H}) \\ &= -\lambda E_W^{MP} + G^{MP} + \frac{1}{2}(k \log \lambda - \log \det \mathbf{A}) \\ &= -\lambda E_W^{MP} + G^{MP} \\ &\quad + \frac{1}{2}(k \log \lambda - \log(\underbrace{\lambda \cdots \lambda}_{k-n} \cdot (\lambda + \rho_1) \cdots (\lambda + \rho_n))) \\ &= -\lambda E_W^{MP} + G^{MP} - \frac{1}{2}\sum_{i=1}^{n} \log(\lambda + \rho_i) + \frac{n}{2}\log \lambda \end{aligned} \tag{20}$$

To obtain the model evidence $p(D|\mathcal{H})$ in level 3 inference or to obtain iterates of λ in level 2 inference, the effective number of parameters γ in (10) has to be computed. This involves the calculation of trace$\mathbf{A}^{-1}$, which, again, can be computed readily from the eigenvalues of $\mathbf{A}$ in (19), as

$$\begin{aligned} \gamma &= k - \lambda \text{trace} \mathbf{A}^{-1} \\ &= k - \lambda(\underbrace{\frac{1}{\lambda} + \cdots + \frac{1}{\lambda}}_{k-n} + \frac{1}{\lambda + \rho_1} + \cdots + \frac{1}{\lambda + \rho_n}) \\ &= \sum_{i=1}^{n} \frac{\rho_i}{\lambda + \rho_i}. \end{aligned}$$

V. Simulation

In this section, we report results on applying the evidence framework to SVMs, with polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d$ and Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$. Three data sets are used in the experiments and they are described in Section V-A. Section V-B reports results on choosing the regularization parameter λ (or, equivalently, C) using the level 2 inference of the evidence framework. Section V-C presents results on choosing the kernel parameters (d or σ) using the level 3 inference.

In the experiments, η in (14) is set to 100. Moreover, sometimes when the regularization parameter is set to an extreme value, all the eigenvalues of $\mathbf{B}$ are numerically very close to zero. In this case, we suspect that the parameter is poorly matched to the problem, and we set the corresponding evidence value to negative infinity.

A. Data Sets

Simulation is performed on three data sets. The first one is a toy problem, with the data generated from 5 Gaussians (Figure 3). It is not separable even with a degree 3 polynomial decision surface. The training set has 500 patterns and the test set has 10,000 patterns.

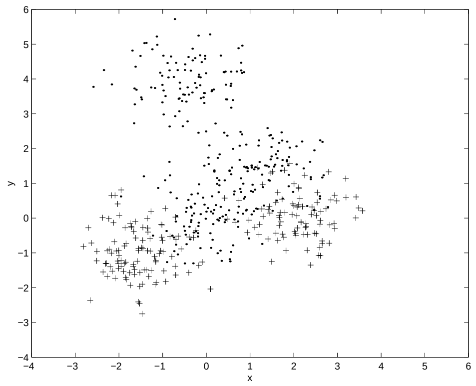


Fig. 3. The data set used in the toy problem.

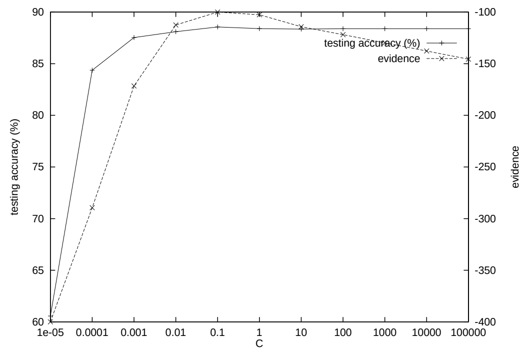
The second data set is the image segmentation data from the UCI machine learning repository[19]. Each pattern has 19 continuous attributes and corresponds to a 3×3 region of an outdoor image. There are 210 patterns in the training set and 2,100 patterns in the test set. The original problem is to classify the pattern into one of the seven classes (brickface, sky, foliage, cement, window, path and grass). In our experiments, we will only concentrate on determining if a particular pattern belongs to the class brickface or not.

The third data set is the Wisconsin breast cancer data, also from the UCI machine learning repository. Each pattern has 9 attributes. First, we remove patterns with missing attributes. Then inconsistent patterns that share the same set of input attribute values but with different output labels are also removed. The resultant data set is randomly partitioned into a training set of 150 patterns and a test set of 299 patterns.

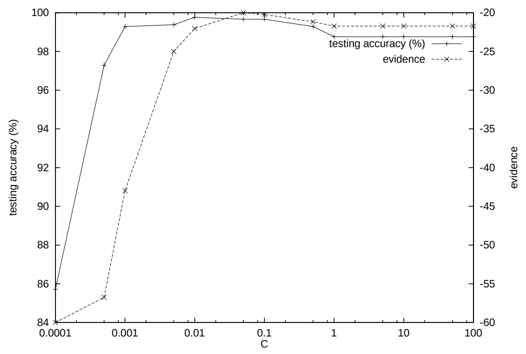
B. Choosing the Regularization Parameter

In this section, we use the evidence for λ, $p(D|\lambda, \mathcal{H})$, computed in (20) to rank the different values of λ (or, equivalently, C in (3)). Figure 4 plots $p(D|\lambda, \mathcal{H})$ and the percentage of correct classifications on the test set at different values of C ($= \frac{1}{\lambda}$) when a polynomial kernel is used. The corresponding graph for the Gaussian kernel is shown in Figure 5. In most cases, the evidence for λ follows the testing accuracy closely.

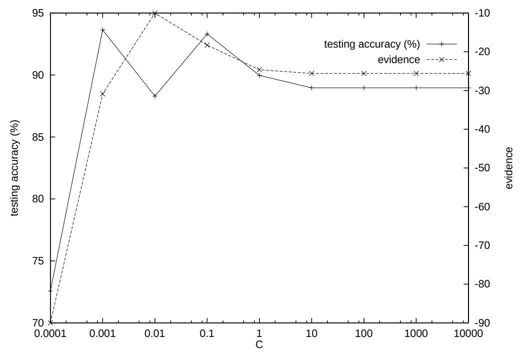
As mentioned in Section III-B, an near-optimal value of λ can be obtained by iterating the process of finding $\mathbf{w}_{MP}$



(a) toy problem

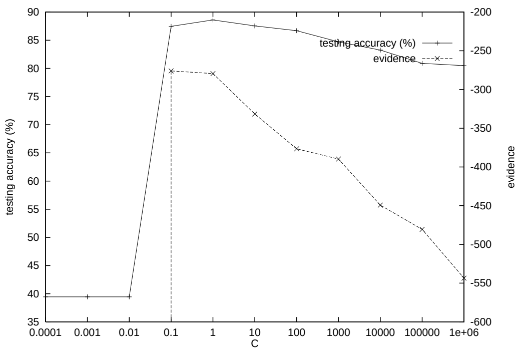


(b) image segmentation

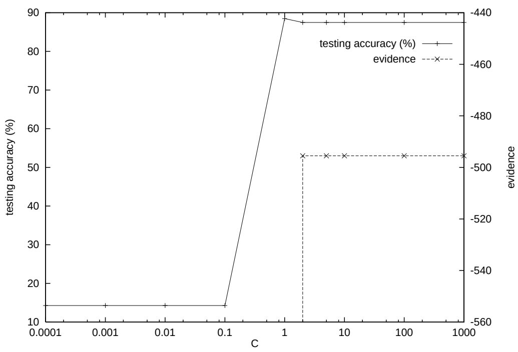


(c) breast cancer

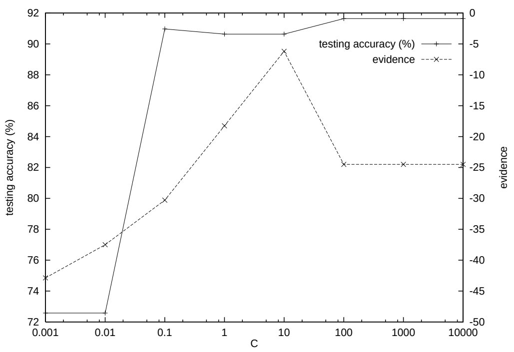
Fig. 4. Results on using different values of C (polynomial kernel).



(a) toy problem



(b) image segmentation



(c) breast cancer

Fig. 5. Results on using different values of C (Gaussian kernel).

from SVM training and re-estimating λ by (9). Experimental results are shown in Tables I and II. The testing accuracy for the SVM obtained in the iterative manner is very close to the testing accuracy for the best SVM in the range of values tested. Results are especially favorable for Gaussian kernels. Moreover, the number of iterations required to find the near-optimal value is quite small.

C. Choosing the Kernel Parameter

This section discusses results on using the model evidence $p(D|\mathcal{H})$ in (11) to estimate the kernel parameters, which include the polynomial degree d for polynomial kernels and the width σ for Gaussian kernels. The regularization parameter C (for a fixed d or σ) is estimated iteratively as mentioned in Section III-B.

Figure 6 plots $p(D|\mathcal{H})$ and the percentage of correct classifications on the test set, at different values of d when a polynomial kernel is used. The corresponding graph for the Gaussian kernel at different values of σ is shown in Figure 7. Though not perfect, the evidence still follows the testing accuracy closely. There are several reasons for this imperfectness. For example, the testing accuracy we measured is based on one SVM with weights set to $\mathbf{w}_{MP}$. The evidence, however, takes account of the complete posterior distribution around this most probable value. For a more complete discussion on this, interested readers may consult [20].

VI. Conclusion

In this paper, we introduce an alternate Bayesian interpretation of SVM based on the weight-space view. By relating the learning of SVM to the level 1 inference in MacKay's evidence framework, we further on show that levels 2 and 3 of the evidence framework can also be applied to SVM. In particular, we have investigated some of the benefits from such an integration, namely, the automatic adjustment of the regularization parameter and the kernel parameter to their near-optimal values, without the need to set data aside in a validation set.

A number of issues need to be addressed in the future. First, an extensive study is needed to compare the use of evidence advocated here and the more traditional method based on an upper bound of the generalization error. Nevertheless, we want to emphasize that the Bayesian framework is not confined to this automatic adjustment of parameters. In fact, as mentioned in Section I, a lot more benefits can possibly be reaped. For example, we have obtained some encouraging results on the use of moderated outputs in SVM [21]. Extending this integration from classification problems to regression problems should also be straight-forward.

Moreover, as the focus of this paper is to investigate the feasibility of applying the evidence framework to SVM, comparison with other approaches (such as decision trees, feedforward neural networks) has not been performed. Lastly, a central question in the evidence framework is the validity of the Gaussian approximation used for the posterior weight distribution. This issue will be addressed in

TABLE I
Results obtained by iterating levels 1 and 2 (polynomial kernel).

	C obtained	testing accuracy (%)	number of iterations	best C in range tested	best testing accuracy (%)
toy problem	6.39	88.4	2	0.1	88.6
image segmentation	0.91	98.8	7	0.01	99.8
breast cancer	0.125	92.3	5	0.001	93.6

TABLE II
Results obtained by iterating levels 1 and 2 (Gaussian kernel).

	C obtained	testing accuracy (%)	number of iterations	best C in range tested	best testing accuracy (%)
toy problem	0.55	88.7	5	1	88.6
image segmentation	1.71	87.5	2	1.71	87.5
breast cancer	80.7	91.6	5	80.7	91.6

the future and the application of other Bayesian techniques like [12], [14] to SVM will also be considered.

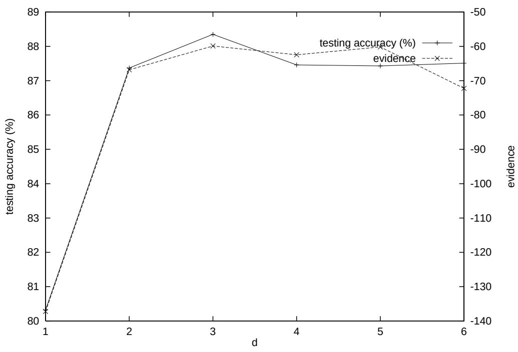
Acknowledgments

This research has been partially supported by the Research Grants Council of the Hong Kong Special Administrative Region under grant HKBU2063/98E. The author would also like to thank Robert Vanderbei for the LOQO quadratic programming solver used in the experiments, and also to thank the anonymous reviewers for their constructive comments on an earlier version of this paper.

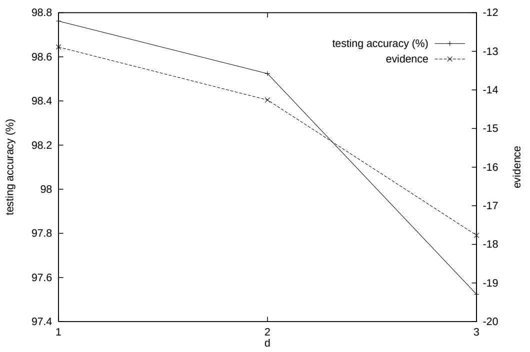
References

[1] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.
[2] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
[3] A.J. Smola and B. Schölkopf, "A tutorial on support vector regression," NeuroCOLT2 Technical Report NC2-TR-1998-030, Royal Holloway College, 1998.
[4] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.
[5] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
[6] J.T. Kwok, "Automated text categorization using support vector machine," in *Proceedings of the International Conference on Neural Information Processing*, Kitakyushu, Japan, Oct. 1998, pp. 347–351.
[7] B. Schölkopf, *Support Vector Learning*, Ph.D. thesis, Technische Universität Berlin, 1997.
[8] K.R. Müller, A.J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Proceedings of the International Conference on Artificial Neural Networks*, 1997.
[9] W.L. Buntine and A.S. Weigend, "Bayesian back-propagation," *Complex Systems*, vol. 5, pp. 603–643, 1991.
[10] T.Y. Kwok and D.Y. Yeung, "Bayesian regularization in constructive neural networks," in *Proceedings of the International Conference on Artificial Neural Networks*, Bochum, Germany, July 1996.
[11] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, May 1992.
[12] R.M. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics. Springer, 1996.
[13] H.H. Thodberg, "A review of Bayesian neural networks with an application to near infrared spectroscopy," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 56–72, 1996.
[14] C.K.I. Williams, "Computing with infinite networks," in *Advances in Neural Information Processing Systems 9*, M.C. Mozer, M.I. Jordan, and T. Petsche, Eds., San Mateo, CA, 1997, pp. 295–301, Morgan Kaufmann.
[15] D.J.C. MacKay, "The evidence framework applied to classification networks," *Neural Computation*, vol. 4, no. 5, pp. 720–736, Sept. 1992.
[16] D.J.C. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.
[17] A. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, pp. 637–649, 1998.
[18] J.O. Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer Series in Statistics. Springer-Verlag, New York, 2nd edition, 1985.
[19] C. Blake, E. Keogh, and C.J. Merz, "UCI repository of machine learning databases," 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html, University of California, Irvine, Department of Information and Computer Sciences.
[20] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford, 1995.
[21] J.T. Kwok, "Moderating the outputs of support vector machine classifiers," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1018–1031, 1999.

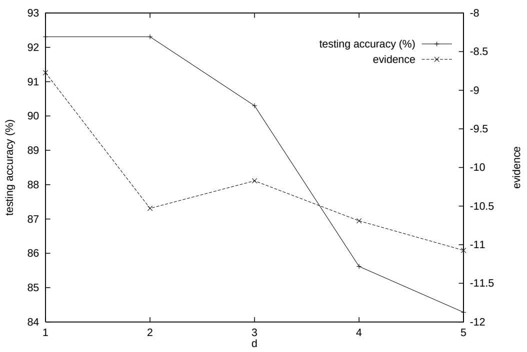
James Tin-Yau Kwok received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 1996. He is currently an Assistant Professor in the Department of Computer Science, Hong Kong Baptist University. His research interests include support vector machine, artificial neural networks and information retrieval.



(a) toy problem

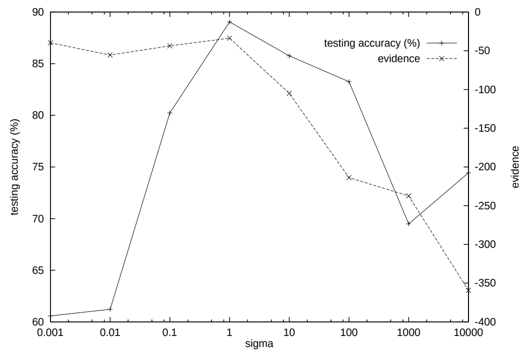


(b) image segmentation

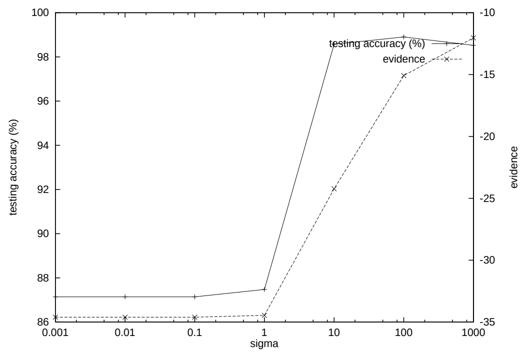


(c) breast cancer

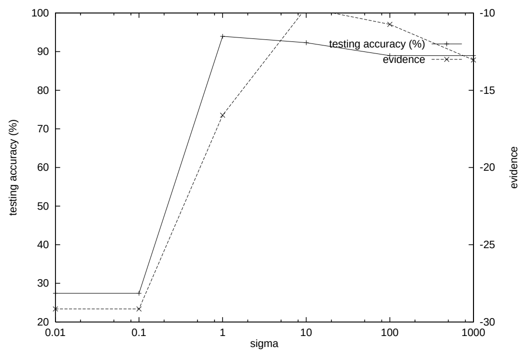
Fig. 6. Results on using different values of d (polynomial kernel).



(a) toy problem



(b) image segmentation



(c) breast cancer

Fig. 7. Results on using different values of σ (Gaussian kernel).