Moderating the Outputs of Support Vector Machine Classifiers

James Tin-Yau Kwok, *Member, IEEE*

***Abstract*— In this paper, we extend the use of moderated outputs to the support vector machine (SVM) by making use of a relationship between SVM and the evidence framework. The moderated output is more in line with the Bayesian idea that the posterior weight distribution should be taken into account upon prediction, and it also alleviates the usual tendency of assigning overly high confidence to the estimated class memberships of the test patterns. Moreover, the moderated output derived here can be taken as an approximation to the posterior class probability. Hence, meaningful rejection thresholds can be assigned and outputs from several networks can be directly compared. Experimental results on both artificial and real-world data are also discussed.**

***Index Terms*—Bayesian, evidence framework, moderated output, support vector machine.**

I. Introduction

In RECENT years, there has been a lot of interest in studying the support vector machine (SVM) [1]–[7]. SVM is based on the idea of *structural risk minimization* (SRM) [6], 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 SVM can have good performance even in problems with a large number of inputs [8]–[11]. To date, SVM has been applied successfully to a wide range of problems, such as classification (e.g., [3], [12]–[14]), regression [15], [16], time series prediction [17] and density estimation [18]. In this paper, we will focus on classification problems.

Despite its many successes, SVM relies on only one weight solution (indirectly represented by the set of Lagrangian multipliers) in making predictions. However, from a Bayesian perspective, the weights of any machine, even after learning, still take a certain posterior distribution. Using just one weight solution as the sole representative thus neglects posterior uncertainty in the weights. This often leads to more extreme predicted outputs during testing, and in turn indicates an overly high confidence that the pattern belongs to a particular class. Under the Bayesian framework, the proper way to handle these weight parameters is by marginalization, which involves integrating them out from the conditional distribution. MacKay called the resultant marginalized output the *moderated output*, and this has been shown to be better in the context of neural networks [19].

Another concern is that the output from a SVM classifier has no clear relationship to posterior class probabilities. Such a connection, if exists, would offer a number of advantages [20], such as the meaningful assignment of rejection thresholds and a direct comparison or combination of outputs from several networks. Hastie and Tibshirani [21] proposed a crude estimate by assuming that the posterior class probability distribution to be a single Gaussian. Dumais *et al.* [22] used regularized maximum likelihood fitting.[1] Other *ad hoc* calibration schemes such as simple binning [24] may also be used. Notice that these methods, in contrast to the previous discussion on moderated output, are not Bayesian in nature and hence do not take the posterior uncertainty of the weights into account.

In this paper, we extend the use of moderated outputs to SVM by making use of a relationship[2] [27], [28] between the evidence framework [29] and the SVM. The evidence framework is a Bayesian framework proposed by MacKay and has been applied successfully to the learning of feedforward neural networks in both classification and regression problems [19], [30]–[33]. In general, such a Bayesian approach is attractive in being logically consistent, simple and flexible [34], [35]. Moreover, because of the Bayesian nature of our approach, the resultant moderated output also serves as an approximation to the posterior class probability. The rest of this paper is organized as follows. A brief summary of the SVM and its relationship to the evidence framework will be described in Sections II and III, respectively. The moderated output for SVM will be derived in Section IV. Simulation results are then presented in Section V, and the last section gives some concluding remarks.

II. Support Vector Machine for Classification

In this section, we briefly review the use of SVM in classification problems. Interested readers may consult [1]–[3], [6], and [7] for details.

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

Manuscript received January 24, 1999; revised May 1, 1999. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region under Grant HKBU2063/98E.

The author is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

Publisher Item Identifier S 1045-9227(99)07266-5.

[1] Details of this method are expected to be in a forthcoming paper [23].

[2] Connections between the SVM and other Bayesian ideas have also been discussed in [25] and [26].

$\mathcal{F}$. Consider the case when the data is linearly separable in $\mathcal{F}$. The SVM constructs a hyperplane $\mathbf{w}^T\mathbf{z}+b$ in $\mathcal{F}$ for which the separation between the positive and negative examples is maximized. The $\mathbf{w}$ for this "optimal" hyperplane can be written as $\mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i \mathbf{z}_i$, where $\boldsymbol{\alpha}=(\alpha_1, \cdots, \alpha_N)$ can be found by solving the following quadratic programming (QP) problem: maximize

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

with respect to $\boldsymbol{\alpha}$, under the constraints $\boldsymbol{\alpha}\geq\mathbf{0}$ and $\boldsymbol{\alpha}^T\mathbf{Y}=0$, where $\mathbf{Y}^T=(y_1, \cdots, y_N)$ 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*[3] $K(\cdot,\cdot)$ such that $K(\mathbf{x}_i, \mathbf{x}_j)=\mathbf{z}_i^T\mathbf{z}_j$, and Q_{ij} is then computed as $Q_{ij}=y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. For example, the kernel for a polynomial classifier of degree d is $K(\mathbf{x}_i, \mathbf{x}_j)=(\mathbf{x}_i^T\mathbf{x}_j+1)^d$. Moreover, notice that $\mathbf{Q}$ is always positive semidefinite and so there is no local optima for the QP problem. 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

$$a(\mathbf{x}, \mathbf{w})=\mathbf{w}^T\mathbf{z}+b=\sum_{\boldsymbol{i}}\alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)+b \tag{2}$$

and then its class label $o(\mathbf{x}, \mathbf{w})$ is given by

$$o(\mathbf{x}, \mathbf{w})=\begin{cases}1, & a(\mathbf{x}, \mathbf{w})>0\\ -1, & \text{otherwise.}\end{cases} \tag{3}$$

When the training set is not separable in $\mathcal{F}$, the SVM algorithm introduces nonnegative slack variables $\xi_i\geq 0$, $i=1, \cdots, N$ [3]. The resultant problem becomes

$$\text{minimize } \tfrac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{N}\xi_i \tag{4}$$

subject to $y_i a(\mathbf{x}_i, \mathbf{w})\geq 1-\xi_i$, $i=1, \cdots, N$. Here, C is a user-defined regularization parameter controlling the tradeoff between model complexity and training error, and ξ_i measures the (absolute) difference between $a(\mathbf{x}_i, \mathbf{w})$ and y_i. Again, minimization of (4) 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. Relationship Between the Evidence Framework and the SVM

The evidence framework is divided into three levels of inference. In this section, we recapitulate the relationship between the first level of inference and the SVM as described in [27] and [28].

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 $p(\mathbf{w}|\lambda, \mathcal{H})$ with a regularization parameter λ. The first level of inference infers the posterior distribution of $\mathbf{w}$ for a given value of λ by using the Baye's rule

$$p(\mathbf{w}|D, \lambda, \mathcal{H})\propto p(\mathbf{w}|\lambda, \mathcal{H})p(D|\mathbf{w}, \mathcal{H}). \tag{5}$$

Assuming that the patterns are independently identically distributed (i.i.d.), then

$$\begin{aligned}p(D|\mathbf{w}, \mathcal{H})&=\prod_i p(\mathbf{x}_i, y_i|\mathbf{w}, \mathcal{H})\\ &=\prod_i p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H})p(\mathbf{x}_i|\mathbf{w}, \mathcal{H})\\ &=\prod_i p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H})p(\mathbf{x}_i)\end{aligned}$$

and (5) becomes

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

Now, consider the following probability model.

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

$$p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H})=\frac{\exp(-[1-y_i a_i]_+)}{\exp(-[1-a_i]_+)+\exp(-[1+a_i]_+)} \tag{7}$$

where $a_i=a(\mathbf{x}_i, \mathbf{w})$ as defined in (2) and $[u]_+=uI_{\{u>0\}}$.

Substituting these probabilities into (6), we obtain

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

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

$$p(y_i|\mathbf{x}_i, \mathbf{w}, \mathcal{H})\simeq\exp(-[1-y_i a_i]_+)=\exp(-\xi_i) \tag{8}$$

where ξ_i is the slack variable in (4). Then, on substituting this approximated probability model back into (6), we get

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

The last two terms on the right do not depend on $\mathbf{w}$. Hence, by setting $C=1/\lambda$, optimizing (4) can be regarded as finding the *maximum a posteriori* (MAP) estimate $\mathbf{w}_{\mathrm{MP}}$ 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. Moreover, in the light of this correspondence, traditional SVM can be considered as using $\mathbf{w}_{\mathrm{MP}}$ as the sole representative of the whole posterior distribution

[3]For a full discussion on the possible choices of the kernels, interested readers are referred to [25].

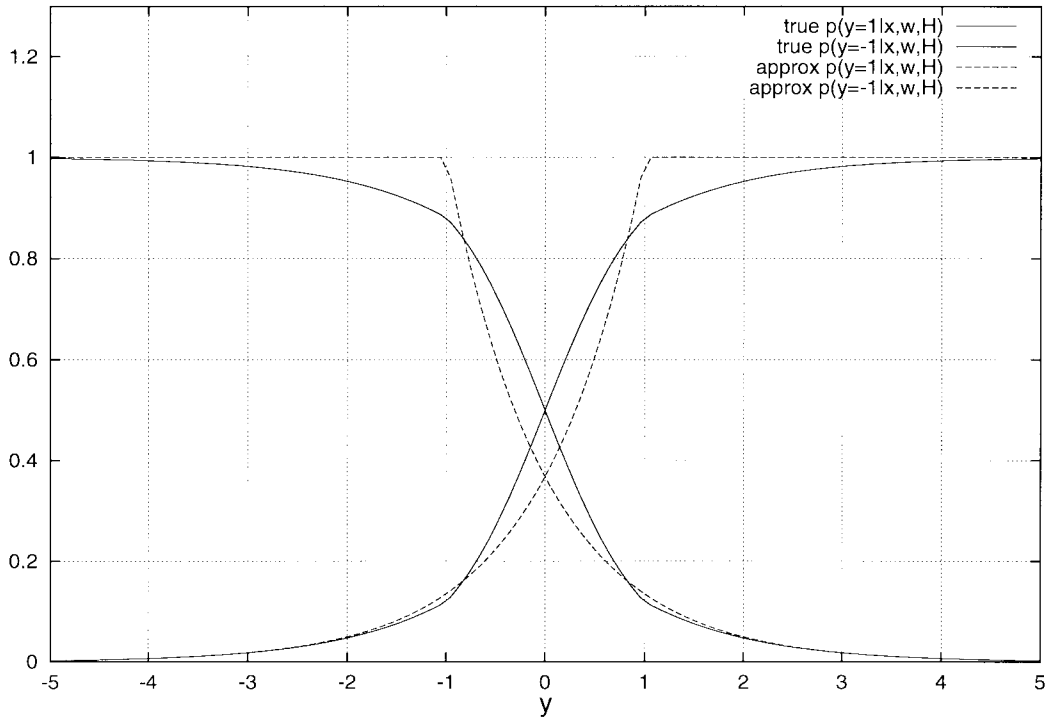


Fig. 1. True and approximated probabilities.

upon prediction. A comparison of the true and approximated probability distributions is shown in Fig. 1.

The second and third levels of inference in the evidence framework can also be applied to SVM [27], [28]. The second level of inference determines the value of λ by maximizing $p(\lambda|D, \mathcal{H}) \propto p(D|\lambda, \mathcal{H})p(\lambda|\mathcal{H})$, whereas the third level of inference ranks different models by examining their posterior probabilities $p(\mathcal{H}|D) \propto p(D|\mathcal{H})p(\mathcal{H})$. As discussed in [27] and [28], applying these two levels of inference to SVM allows 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.

IV. Moderating the SVM Output

The approximated class probabilities (8) in Section III can be written out more explicitly as[4]

$$p(y=1|\mathbf{x}, \mathbf{w}) = \begin{cases} \exp(-1+a(\mathbf{x}, \mathbf{w})), & a(\mathbf{x}, \mathbf{w}) < 1 \\ 1, & \text{otherwise} \end{cases}$$

and

$$p(y=-1|\mathbf{x}, \mathbf{w}) = \begin{cases} \exp(-1-a(\mathbf{x}, \mathbf{w})), & a(\mathbf{x}, \mathbf{w}) > -1 \\ 1, & \text{otherwise.} \end{cases}$$

As mentioned in Section I, the Bayesian approach of handling the unwanted parameter $\mathbf{w}$ is to integrate it out from the conditional distribution. Here, as in the evidence framework, we assume that the posterior distribution of $\mathbf{w}$ can be approximated by a single Gaussian at $\mathbf{w}_{\mathrm{MP}}$. Since $a(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{z} + b$, therefore the posterior distribution of $a(\mathbf{x}, \mathbf{w})$ will also be Gaussian $N(a_{\mathrm{MP}}, s^2)$, with mean $a_{\mathrm{MP}}(\mathbf{x}) = a(\mathbf{x}, \mathbf{w}_{\mathrm{MP}})$ and variance

$$s^2(\mathbf{x}) = \mathbf{z}^T\mathbf{A}^{-1}\mathbf{z} \tag{9}$$

where $\mathbf{A} = \nabla^2 M = \nabla^2(\lambda E_W + \sum_{i=1}^{N} \xi_i)$ is the Hessian. Thus

$$\begin{aligned} p(y=1|D, \mathbf{x}) &= \int_{-\infty}^{1} e^{-1+a}\mathrm{N}(a_{\mathrm{MP}}, s^2)\,da + \int_{1}^{\infty} \mathrm{N}(a_{\mathrm{MP}}, s^2)\,da \\ &= \frac{1}{2}\exp\left(\frac{s^2}{2} + a_{\mathrm{MP}} - 1\right)\mathrm{erfc}\left(\frac{s^2 + a_{\mathrm{MP}} - 1}{\sqrt{2}\,s}\right) \\ &\quad + \frac{1}{2}\mathrm{erfc}\left(\frac{1 - a_{\mathrm{MP}}}{\sqrt{2}\,s}\right) \end{aligned} \tag{10}$$

and

$$\begin{aligned} p(y=-1|D, \mathbf{x}) &= \int_{-\infty}^{-1} \mathrm{N}(a_{\mathrm{MP}}, s^2)\,da + \int_{-1}^{\infty} e^{-1-a}\mathrm{N}(a_{\mathrm{MP}}, s^2)\,da \\ &= \frac{1}{2}\exp\left(\frac{s^2}{2} - a_{\mathrm{MP}} - 1\right)\mathrm{erfc}\left(\frac{s^2 - a_{\mathrm{MP}} - 1}{\sqrt{2}\,s}\right) \\ &\quad + \frac{1}{2}\mathrm{erfc}\left(\frac{1 + a_{\mathrm{MP}}}{\sqrt{2}\,s}\right). \end{aligned} \tag{11}$$

Here, $\mathrm{erfc}(x) = (2/\sqrt{\pi})\int_x^\infty e^{-t^2}\,dt$ is the complementary error function [36] and can be readily computed. Recall that (8) is only an approximation of (7), consequently (10) and

[4] For simplicity of notations, we will drop $\mathcal{H}$ in the sequel.

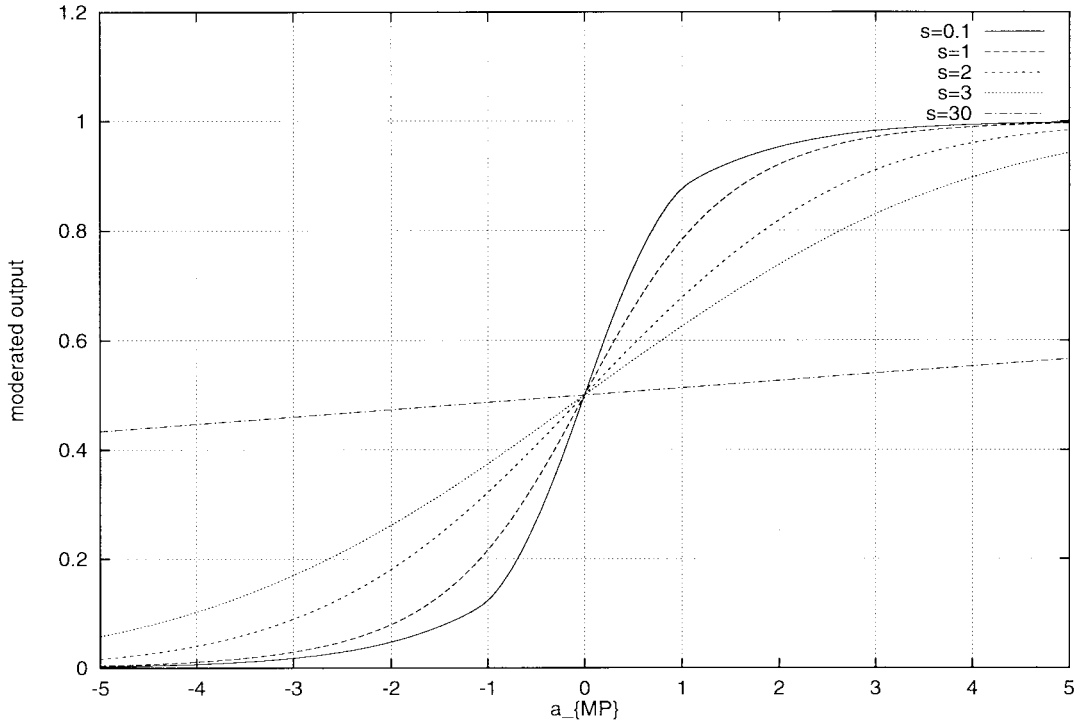


Fig. 2. Moderated output as a function of $a_{\rm MP}$ and s.

(11) have to be normalized

$$\tilde{p}(y=1|D,\mathbf{x})=\frac{p(y=1|D,\mathbf{x})}{p(y=1|D,\mathbf{x})+p(y=-1|D,\mathbf{x})}$$
$$\tilde{p}(y=-1|D,\mathbf{x})=1-\tilde{p}(y=1|D,\mathbf{x}).$$

We call $\tilde{p}(y=1|D,\mathbf{x})$ the *moderated output* of the SVM for pattern $\mathbf{x}$. Fig. 2 shows the variation of the moderated output as a function of $a_{\rm MP}$ and s. As can be seen, when s increases, changes in the moderated output becomes more gradual to changes in $a_{\rm MP}$. And when s becomes very large, the moderated output stays near 0.5. As the moderated output estimates the posterior class probability, this indicates maximum uncertainty in predicting the class membership of $\mathbf{x}$. Moreover, notice that the moderated output passes through 0.5 at $a_{\rm MP}=0$. Hence, if the class label of $\mathbf{x}$ is obtained by thresholding the moderated output at 0.5, then it will be the same as the MAP-based decision rule in (3).

To compute $s^2(\mathbf{x})$ in (9), we have to determine the Hessian $\mathbf{A}$. First, define the sigmoid function $\sigma(u)=1/(1+e^{-\eta u})$ for some $\eta>0$, and $r(u)=u\sigma''(u)+2\sigma'(u)$. Using results in [27], $\mathbf{A}$ can be approximated as

$$\mathbf{A}\simeq\mathbf{P}\mathbf{\Lambda}\mathbf{P}^T \tag{12}$$

where

$$\mathbf{P}=(\mathbf{v}_1,\cdots,\mathbf{v}_n)=\left(\sum_{i=1}^N\mu_{1i}\mathbf{z}_i,\cdots,\sum_{i=1}^N\mu_{ni}\mathbf{z}_i\right) \tag{13}$$

and

$$\mathbf{\Lambda}=\mathrm{diag}(\lambda+\rho_1,\cdots,\lambda+\rho_n). \tag{14}$$

Here, the $\mathbf{v}_l$'s are orthogonal, n is the number of significant eigenvalues in the $N\times N$ matrix $\tilde{\mathbf{K}}$ with entries $\tilde{K}_{ij}=r(|y_i-a(\mathbf{x}_i,\mathbf{w})|)K(\mathbf{x}_i,\mathbf{x}_j)\equiv r_iK(\mathbf{x}_i,\mathbf{x}_j)$, while ρ_l and $\boldsymbol{\mu}_l=(\mu_{l1},\cdots,\mu_{lN})^T$ are obtained from the eigensystem $\rho_l\boldsymbol{\mu}_l=\tilde{\mathbf{K}}\boldsymbol{\mu}_l$.

Later on, it will be more convenient to have $\mathbf{P}^{-1}=\mathbf{P}^T$. This is satisfied by ensuring that the set $\{\mathbf{v}_l\}_l$ is orthonormal. Let the required normalizing factor for $\mathbf{v}_l$ be c_l and define $\tilde{\boldsymbol{\mu}}_l\stackrel{\rm def}{=}(\mu_{l1}/r_1,\cdots,\mu_{lN}/r_N)^T$. Then,

$$\begin{aligned}1&=\left\|c_l\sum_{i=1}^N\mu_{li}\mathbf{z}_i\right\|^2=c_l^2\sum_{i,j=1}^N\mu_{li}\mu_{lj}K(\mathbf{x}_i,\mathbf{x}_j)\\&=c_l^2\sum_{i,j=1}^N\frac{\mu_{li}}{r_i}\mu_{lj}\tilde{K}_{ij}=c_l^2\tilde{\boldsymbol{\mu}}_l^T\tilde{\mathbf{K}}\boldsymbol{\mu}_l\\&=c_l^2\rho_l\tilde{\boldsymbol{\mu}}_l^T\boldsymbol{\mu}_l=c_l^2\rho_l\sum_{i=1}^N\frac{\mu_{li}^2}{r_i}.\end{aligned}$$

Hence, the required normalizing factor for $\mathbf{v}_l$ is

$$c_l=1/\sqrt{\rho_l\sum_{i=1}^N\mu_{li}^2/r_i}.$$

Utilizing (12)–(14) and the fact that $\mathbf{P}^{-1}=\mathbf{P}^T$, $s^2(\mathbf{x})$ in (9) can then be computed as

$$\begin{aligned}s^2&=\mathbf{z}^T\mathbf{A}^{-1}\mathbf{z}\\&=\mathbf{z}^T\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T\mathbf{z}\\&=(\mathbf{P}^T\mathbf{z})^T\mathbf{\Lambda}^{-1}(\mathbf{P}^T\mathbf{z})\\&=\sum_{i=1}^n\frac{1}{\lambda+\rho_i}\left(\sum_{j=1}^N\mu_{ij}K(\mathbf{x}_j,\mathbf{x})\right)^2.\end{aligned} \tag{15}$$

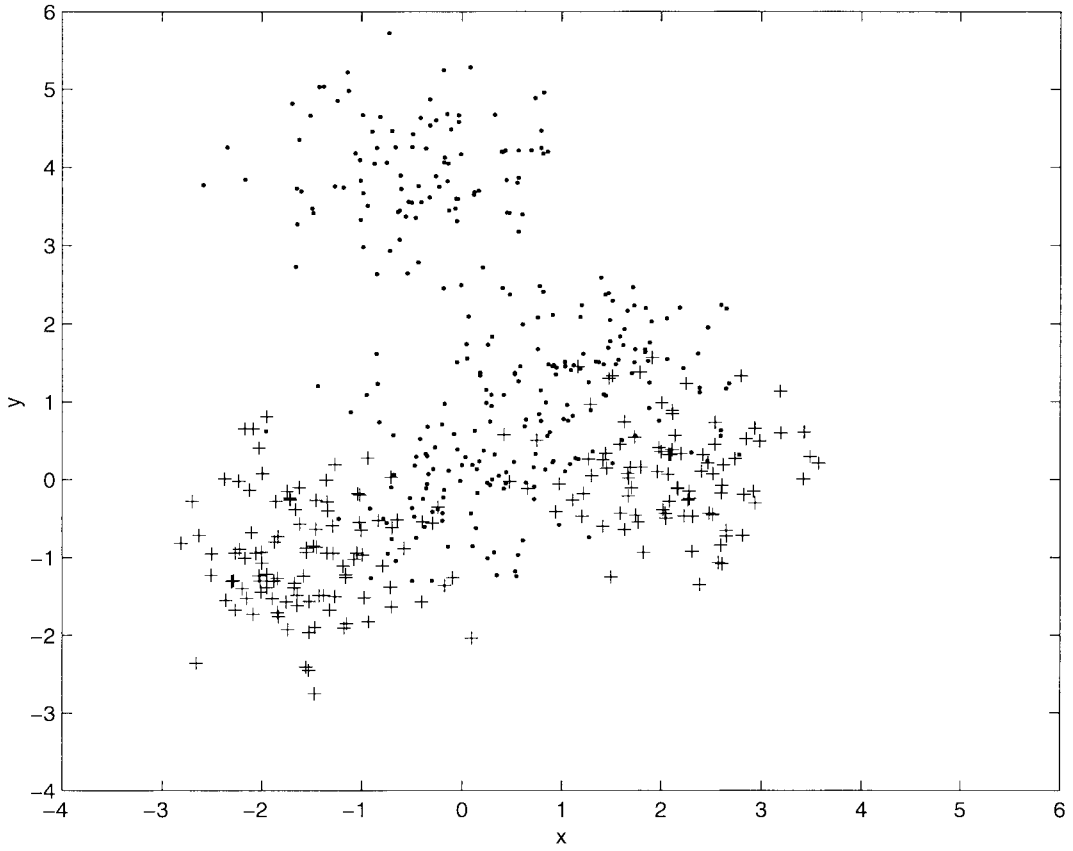


Fig. 3. Training data for the toy problem.

V. Simulation

In this section, we report simulation results on three data sets. The polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d$ is used throughout. Here, we do not address the issue of model selection. This, in general, can be handled by the use of a separate validation set [17], or by utilizing an upper bound on the generalization error predicted from the SRM theory [6]. As mentioned in Section III, the evidence framework can also be used to address this issue. Interested readers may refer to [27], [28], and the references therein.

A. Toy Problem

This is an artificial two-class classification problem, with data generated from five Gaussians (Fig. 3). The training set has 500 patterns and the test set has 10 201 patterns. Polynomial kernel with $d = 3$ is used.

Fig. 4 compares the following outputs obtained from the SVM:

1) MAP output $a_{\mathrm{MP}} = a(\mathbf{x}, \mathbf{w}_{\mathrm{MP}})$;
2) a_{MP} clipped at ± 1;
3) unmoderated probability estimate

$$p(y = 1|\mathbf{x}, \mathbf{w}_{\mathrm{MP}}) = \frac{\exp(-[1 - a_{\mathrm{MP}}]_+)}{\exp(-[1 - a_{\mathrm{MP}}]_+) + \exp(-[1 + a_{\mathrm{MP}}]_+)}; \quad (16)$$

4) another unmoderated probability estimate produced by regularized maximized likelihood fitting[5] [22];
5) moderated output $\tilde{p}(y = 1|D, \mathbf{x})$.

The MAP output does not show any meaningful pattern in general, while the clipped MAP output and both unmoderated probability outputs show high confidence (with output values near ± 1 and $0/1$, respectively) in most parts of the input space, even far beyond the decision boundary and in regions of low data density. In comparison, the moderated output becomes less certain as the pattern moves away from the training data.

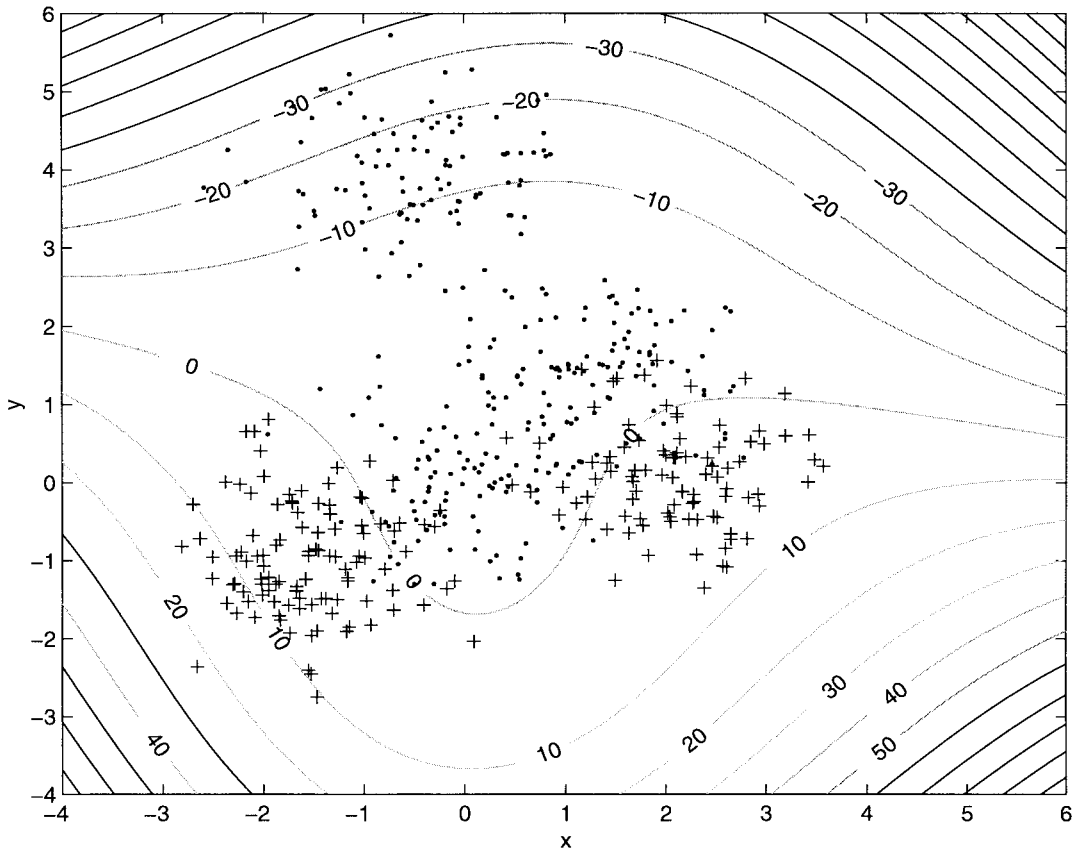
As discussed in Section IV, if the classifier output is obtained by thresholding the moderated output at 0.5, then it will be the same as the MAP-based decision rule in (3). In order to obtain a numerical comparison of the performance from various SVM outputs, we use

$$G = -\left(\sum_{i=1}^{N_t} \frac{1 + y_i}{2} \log o_i + \frac{1 - y_i}{2} \log(1 - o_i) \right) \quad (17)$$

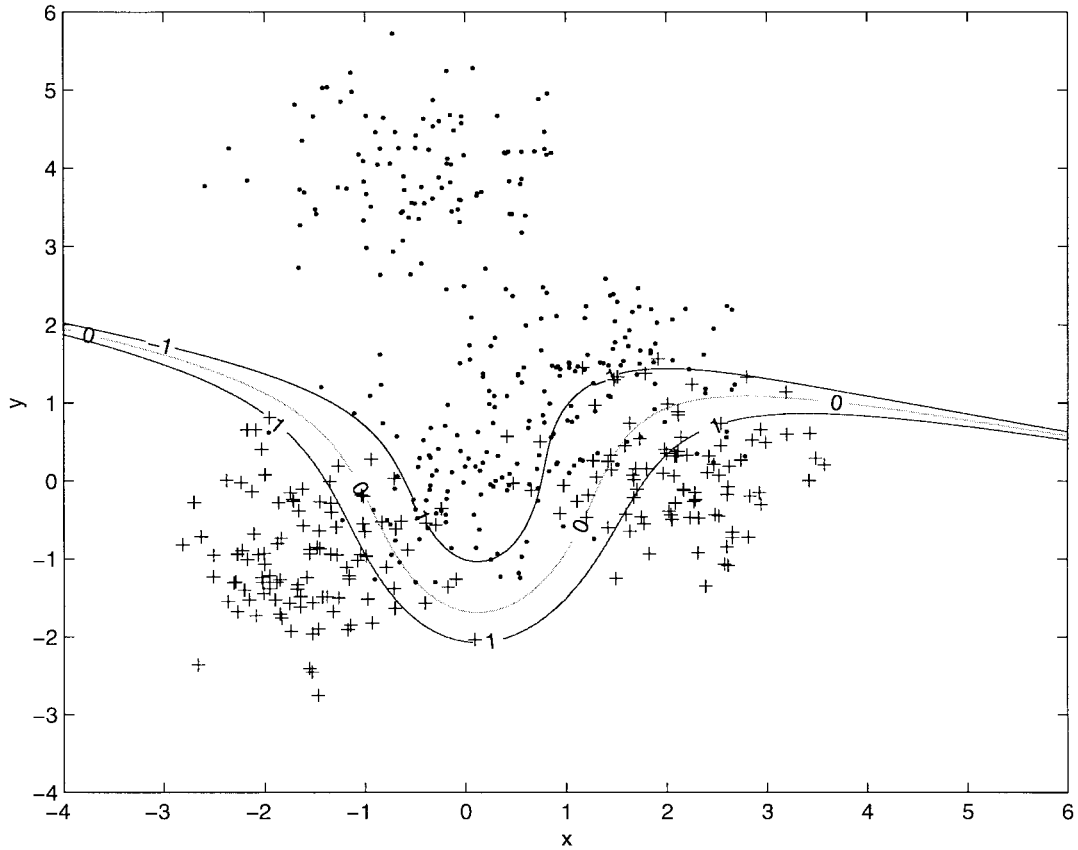
[5]Our implementation of regularized maximum likelihood fitting is as follows. After training the SVM, we fit a sigmoid to a_{MP} to produce $\bar{o}(\mathbf{x}_i, \mathbf{w}_{\mathrm{MP}}) = 1/(1 + \exp(-\omega(a_{\mathrm{MP}} + c)))$. For a given regularization parameter $\kappa > 0$, parameters ω and c are determined by minimizing

$$\frac{1}{2} \sum_i \left(\bar{o}(\mathbf{x}_i, \mathbf{w}_{\mathrm{MP}}) - \frac{1 + y_i}{2} \right)^2 + \kappa \omega^2.$$

An appropriate value for κ is estimated from a validation set.

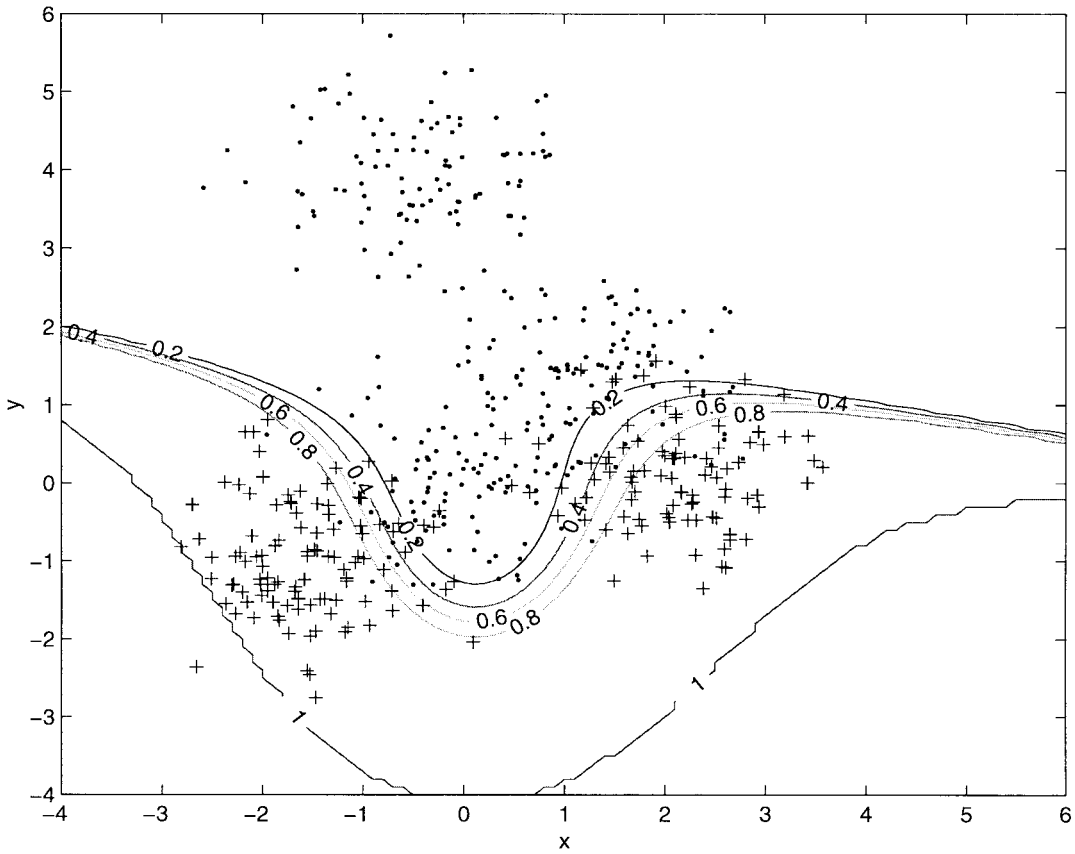


(a)

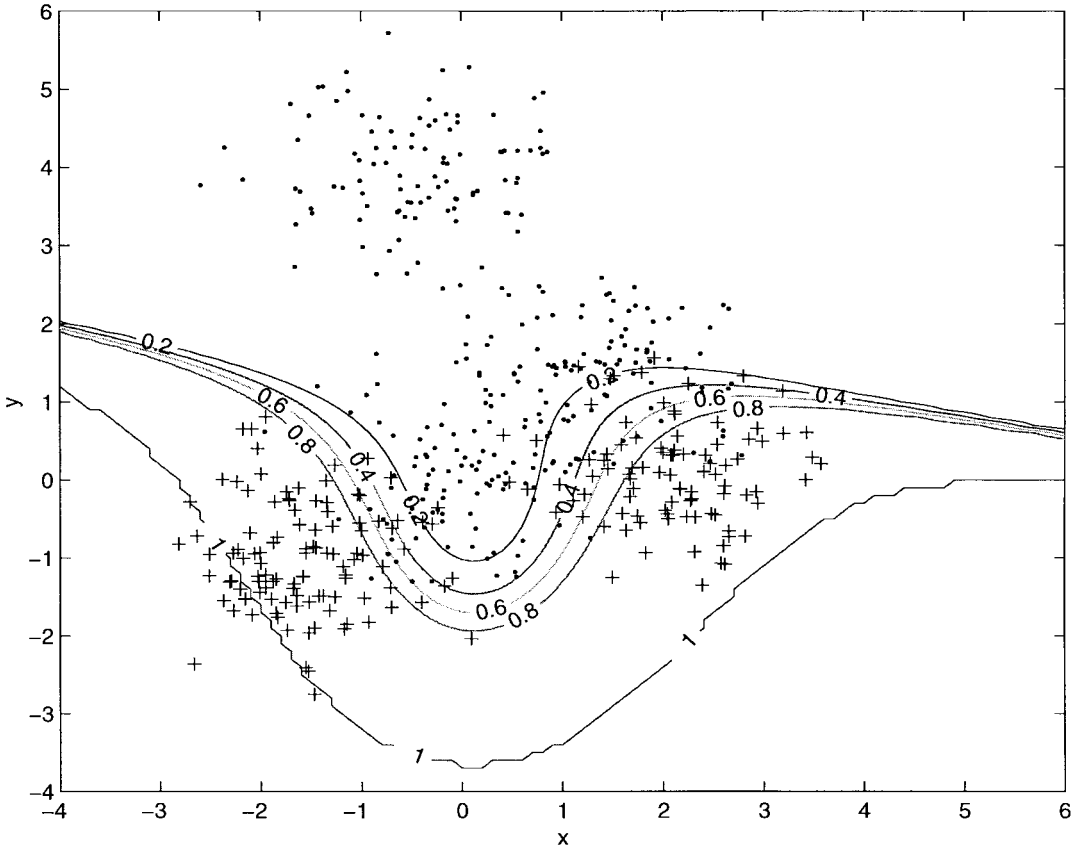


(b)

Fig. 4. Various outputs from the SVM: (a) MAP output a_{MP}, (b) a_{MP} clipped at ± 1.

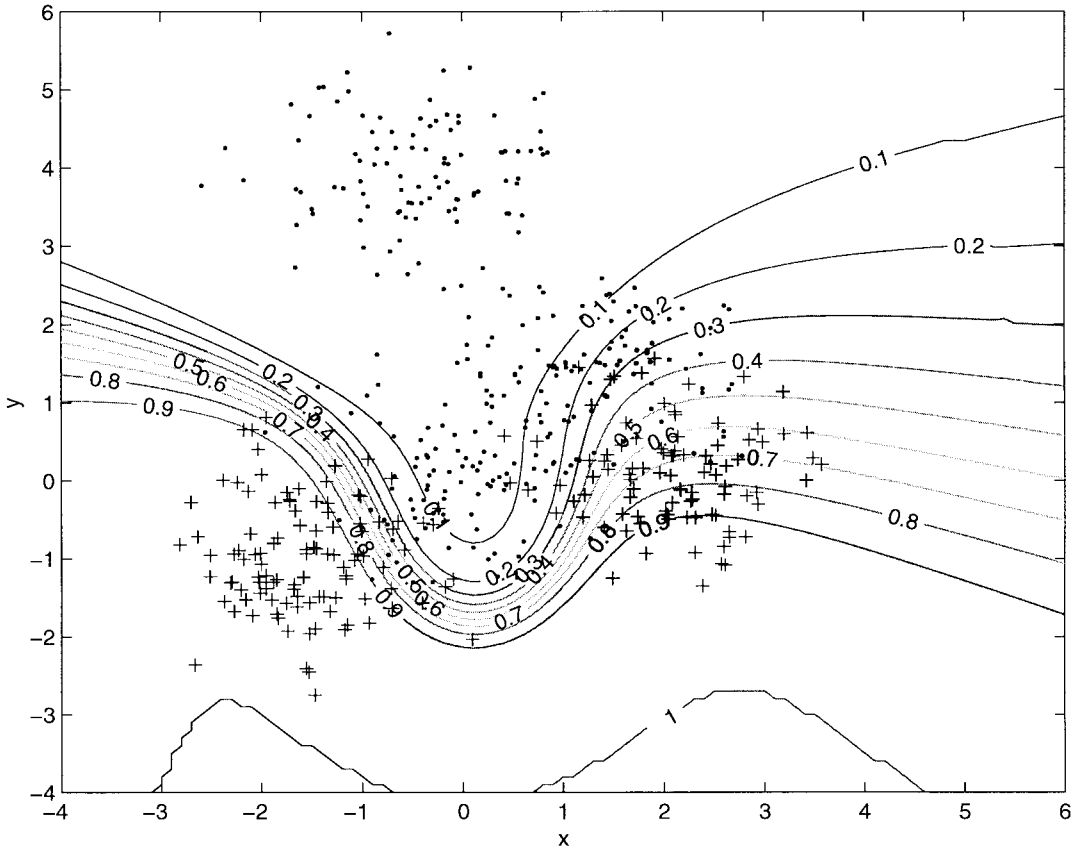


(c)



(d)

Fig. 4. (*Continued.*) Various outputs from the SVM: (c) unmoderated output, (d) regularized unmoderated output.



(e)

Fig. 4. (*Continued.*) Various outputs from the SVM: (e) moderated output.

TABLE I
VALUES OF G FROM VARIOUS SVM OUTPUTS

clipped MAP	unmoderated	regularized, unmoderated	moderated
∞	∞	3801	1417

as in [19]. Here, $o_i \in [0, 1]$ is the output for pattern i, and N_t is the size of the independent test set (equals 10 201 in this experiment). As mentioned in [19], $\exp(-G)$ can be interpreted as the data likelihood. The smaller the value of G, the more likely is the data generated by the model. Table I shows the results for the various outputs. For the clipped MAP solution, there are some misclassified test patterns (i.e., $y_i = -1$ but $a_{\mathrm{MP}} > 1$, or vice versa) and G goes infinite. Similarly is the case for the unmoderated estimate (16). The value of G obtained from regularized maximum likelihood fitting is also inferior to that based on the moderated output.

B. Image Segmentation Problem

The second data set is the image segmentation data from the UCI machine learning repository [37]. Each pattern has 19 continuous attributes and corresponds to a 3×3 region of an outdoor image. The problem is to classify the pattern into one of the seven classes: `brickface`, `cement`, `foliage`, `grass`, `path`, `sky`, and `window`. There are 210 patterns in the training set and 2100 patterns in the test set (each class has 300 test patterns). Polynomial kernel with $d = 1$ is used.

We take the usual approach of formulating this multiclass classification problem as a series of binary classification problems [38], [39]. Seven classifiers are constructed, with one for each class (e.g., the `brickface` classifier separates patterns belonging to the `brickface` class from those that do not). For a particular test pattern, the classifier with the highest output value is selected as winner and the corresponding class label assigned. Table II shows the misclassification errors on incorporating the classes one by one (in alphabetical order). The performance of the moderated output is always better than that of the MAP output. Moreover, performance can be improved further by rejecting some uncertain patterns. As the moderated output estimates the posterior class probability, meaningful rejection thresholds can be set on the winning classifier and the resultant improvement is shown in Table III. In contrast, selection of meaningful rejection thresholds in a traditional SVM is not easy, as its output has no clear relationship to the posterior class probability.

C. Text Categorization Problem

In this experiment, we use the Reuters-21 578 Distribution 1.0 test collection.[6] This is a set of 21 578 Reuters newswire stories from the year 1987 that have been manually indexed into 135 financial topic categories to support document rout-

[6]The Reuters-21 578 Distribution 1.0 test collection is available from http://www.research.att.com/~lewis. Arrangements for access were made by David Lewis.

TABLE II
RESULTS ON THE IMAGE SEGMENTATION PROBLEM (FOR COMPARISON, THE PERCENTAGE ERROR OF THE ONE NEAREST-NEIGHBOR CLASSIFIER IN THE DISCRIMINATION OF ALL SEVEN CLASSES IS 12.3%)

classes combined	# test patterns in the classes	# error (moderated)	% error (moderated)	# error (MAP)	% error (MAP)
brickface + cement	600	18	3.0	25	4.2
brickface + cement + foliage	900	40	4.4	48	5.3
brickface + cement + foliage + grass	1200	43	3.6	59	4.9
brickface + cement + foliage + grass + path	1500	60	4.0	71	4.7
brickface + cement + foliage + grass + path + sky	1800	62	3.4	72	4.0
brickface + cement + foliage + grass + path + sky + window	2100	180	8.6	205	9.8

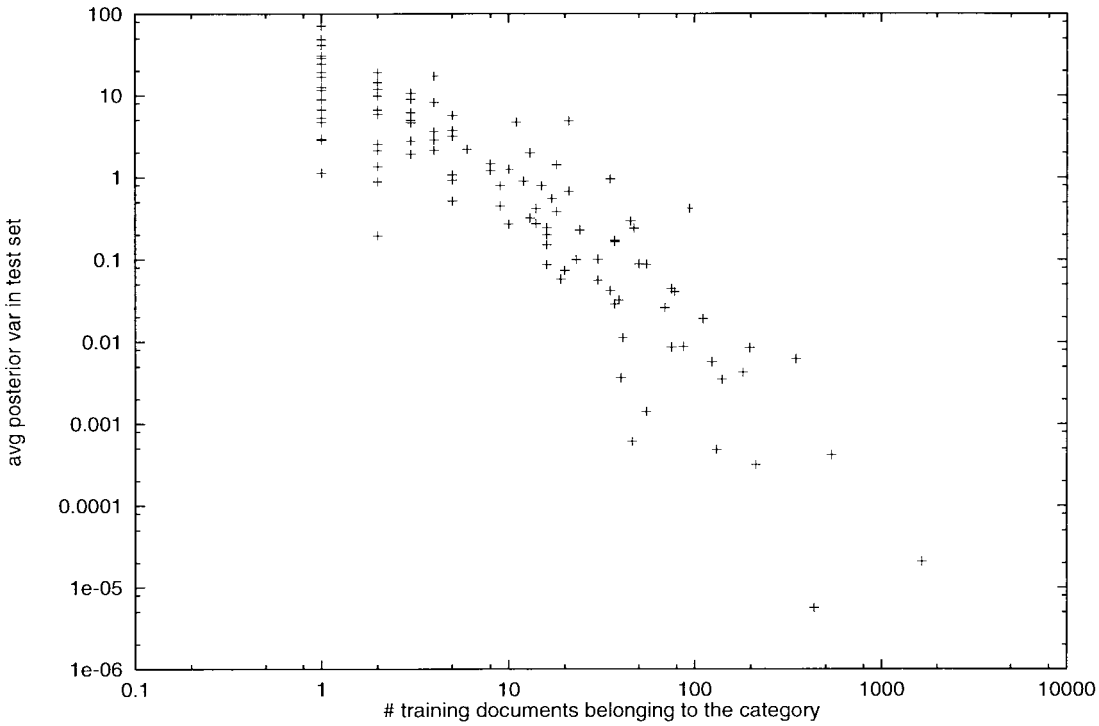


Fig. 5. Variation of the average posterior variance in the test set with the number of training documents belonging to the category (each point corresponds to one category).

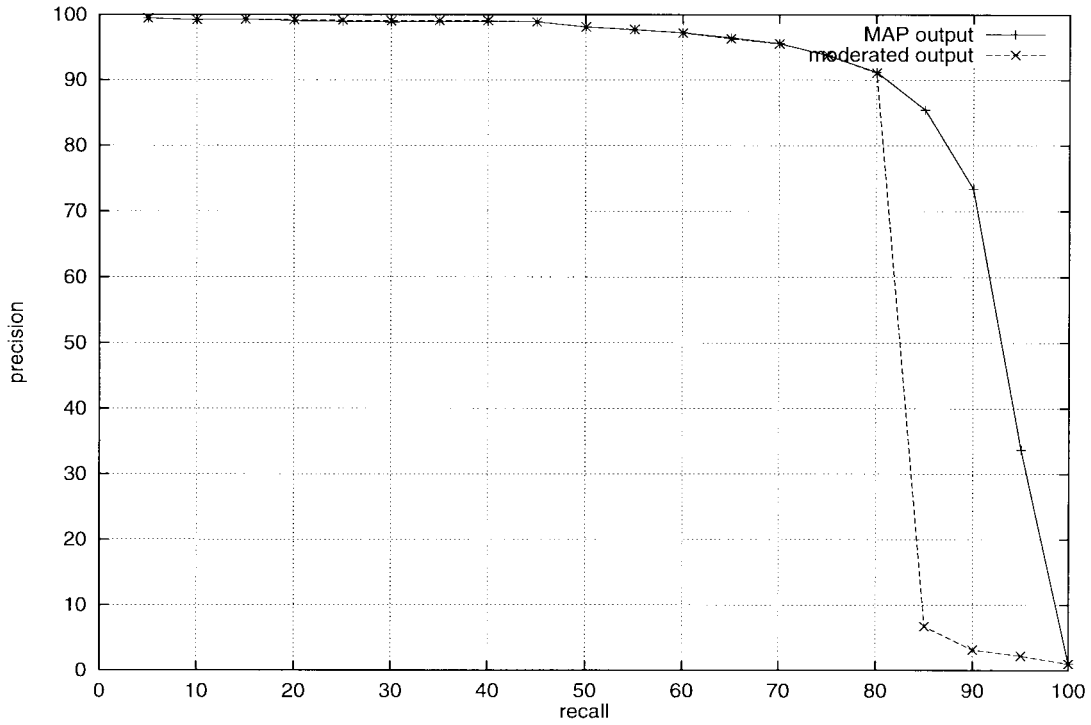
TABLE III
PERFORMANCE AFTER SETTING A REJECTION THRESHOLD ON THE MODERATED OUTPUT OF THE WINNING CLASSIFIER

rejection threshold	% test patterns rejected	%error
0.1	0.0	8.6
0.2	0.0	8.5
0.3	0.0	7.8
0.4	0.24	6.4
0.5	12.7	4.9
0.6	24.9	3.9
0.7	39.3	3.4
0.8	47.9	2.6
0.9	57.4	3.3

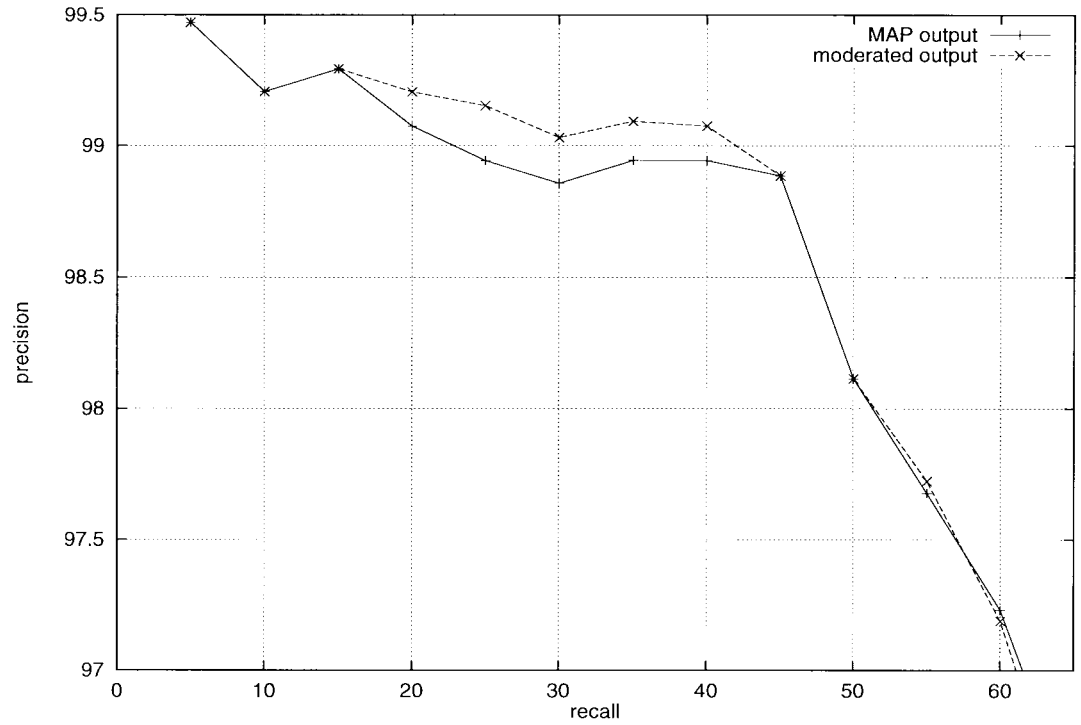
ing and retrieval by Reuters customers. Here, we use the ModApte (Modified Apte) split of the test collection, which contains 9603 training documents and 3299 test documents. Documents are coded using the traditional $tf \cdot idf$ (term frequency—inverse document frequency) representation [40]

$$\frac{f_i(t_j)}{\sqrt{\sum_{t_j \in D_i} f_i^2(t_j)}} \log\left(\frac{N}{N(t_j)}\right)$$

where $f_i(t_j)$ is the frequency of term t_j in document D_i, N is the total number of documents, and $N(t_j)$ the number of documents containing t_j. Polynomial kernel with $d = 1$ is used for all the categories.



(a)



(b)

Fig. 6. Recall-precision curve for the text categorization problem: (a) range of recall: 0–100% and (b) zoomed-in range of recall: 0–65%.

Instead of using the misclassification error, performance in text categorization tasks is usually measured by *recall* and *precision* [41]. Recall is the percentage of total documents for a given category that are correctly classified. Precision is the percentage of predicted documents for a given category that are correctly classified. The point at which recall equals precision is the *break-even* point, and is often used as a single summarizing measure for comparison of results. Moreover, because the text categorization problem is also analyzed as a set of dichotomous classification problems as in Section V-B, we use both *micro-averaging* and *macro-averaging* [41] to evaluate overall performance across the entire set of categories.

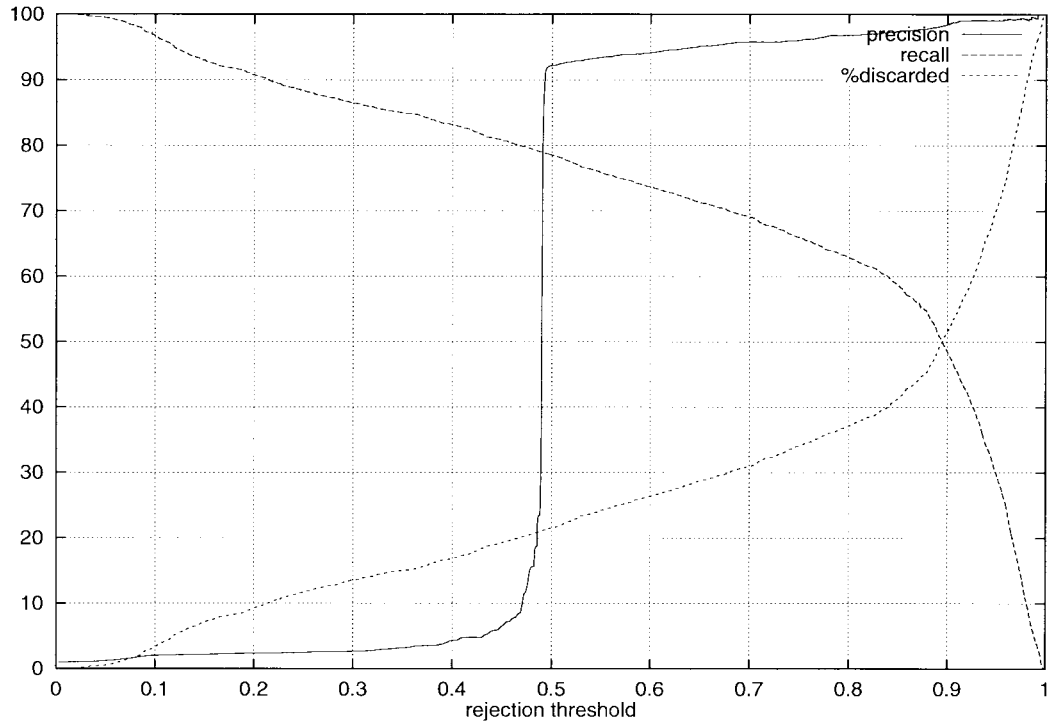


Fig. 7. Variations of recall and precision with the rejection threshold on the moderated output.

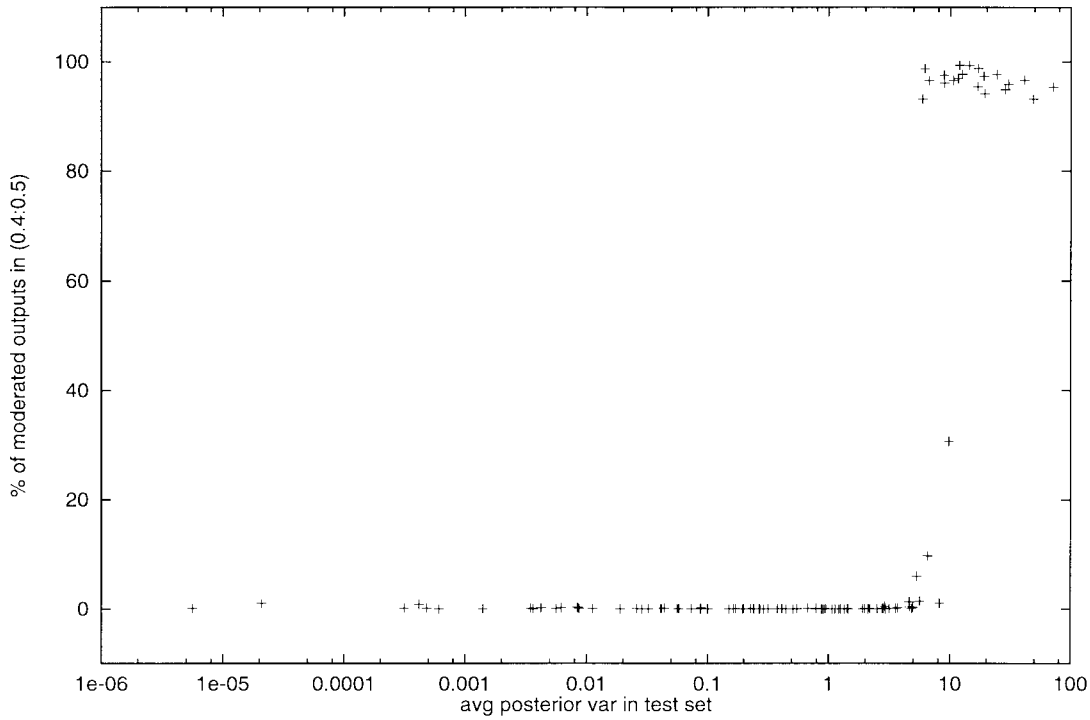
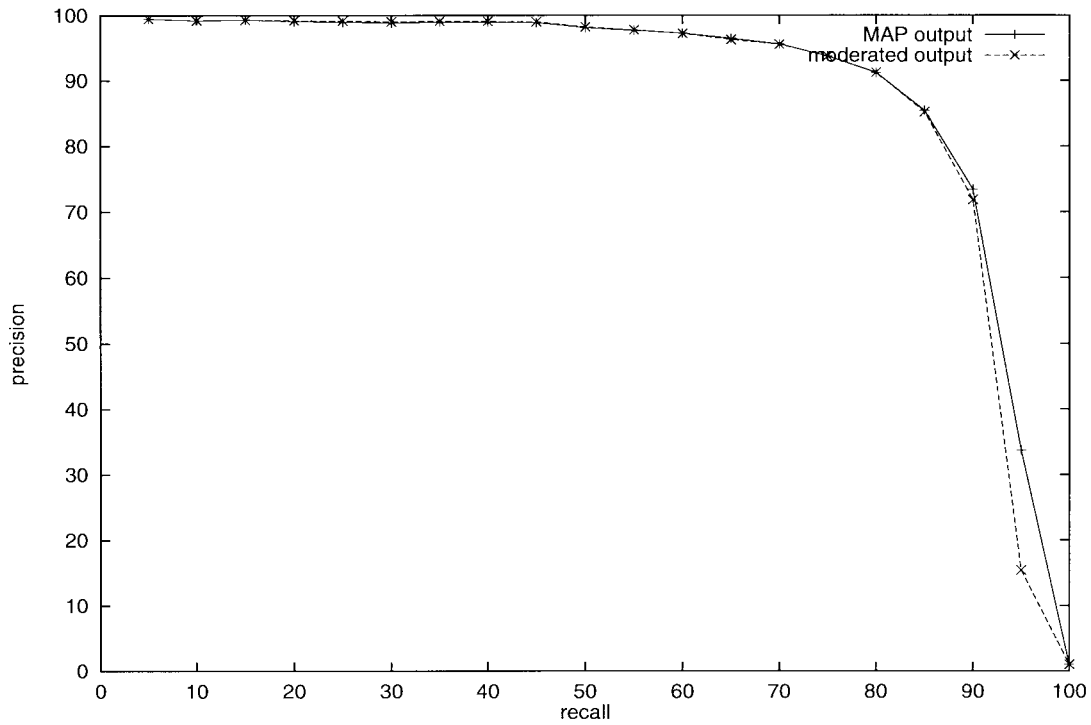


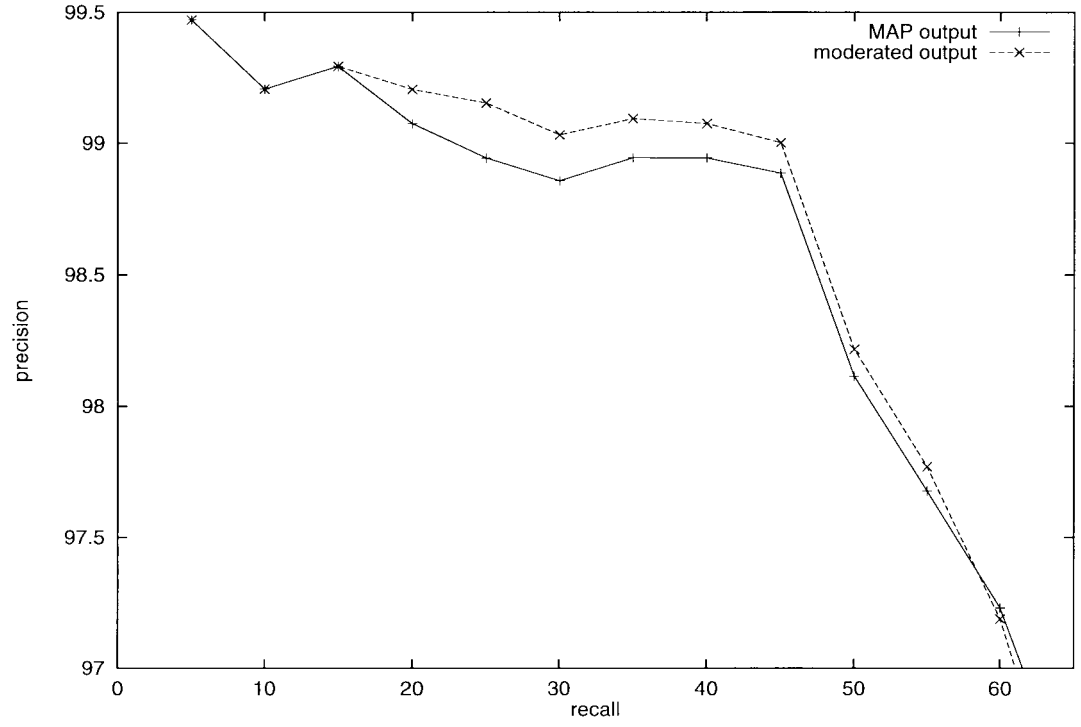
Fig. 8. Plot showing the percentage of test patterns with moderated output in the range (0.4, 0.5), versus the average posterior variance in the test set (each point corresponds to one category).

In micro-averaging, the performance tables for each of the categories are added and the overall recall and precision are computed. In macro-averaging, performance measures are computed separately for each category, and then the mean is reported.

With macro-averaging, both the MAP and the moderated output yield the same break-even point of 72.0%. This is because, for most of the categories, the posterior variances s^2 in (9) for the test patterns are too small to make a difference (Fig. 5). From the figure, one can also observe the clear



(a)



(b)

Fig. 9. Recall-precision curve with the modified scheme: (a) range of recall: 0–100% and (b) zoomed-in range of recall: 0–65%.

inverse relationship between the number of training documents belonging to the category and the average posterior variance for the corresponding classifier.

The difference becomes more apparent in micro-averaging, when outputs from different dichotomous classifiers are ranked together. Moderated output yields slightly better precision at medium recall levels[7] [Fig. 6(b)], but is worse at high recall levels [Fig. 6(a)]. The sudden deterioration can be understood by observing the effect of using a rejection threshold (Fig. 7).

[7] Note that although the improvement in precision is small, this may still be useful when the text categorizer is used in information extraction applications [24].

There is a large drop in precision near a threshold of 0.5. This is because for those classifiers with large average posterior variance, almost all the test patterns yield moderated outputs close to 0.5 (Fig. 8), indicating maximum uncertainty in the predictions. To have a high recall in micro-averaging, one has to label all these uncertain patterns as belonging to the corresponding category, thus seriously hampering the overall precision.

To alleviate this problem, one alternative is to first label patterns that the classifiers have high confidence, and come back to the uncertain patterns only when a very high recall level is required. Fig. 9 shows the resulting performance with this modified scheme. The use of moderated output still yields improved precision at medium recall levels, while worsens only at very high recall levels. However, in general, we recommend the use of the rejection threshold instead, which offers direct control over the acceptance of uncertain patterns. For example, with a rejection threshold of 0.5, we can obtain reasonably good precision and recall values of 92.2 and 78.5%, respectively.

Finally, notice that our results also concur with others [8], [22] that the SVM is very suitable for text categorization. The micro-averaged break-even point obtained here is 85%, which is among the best known results in this collection.[8] Favorable results using the SVM have also been reported on the older Reuters-22 173 collection [9].

VI. Conclusion

In this paper, we extend the use of moderated output to SVM based on the evidence framework. Because of the Bayesian nature of this approach, the moderated output derived serves naturally as an approximation to the posterior class probability. Consequently, meaningful rejection thresholds can be assigned and outputs from several networks (as in using dichotomous networks for multiclass problems) can be directly compared. Moreover, unlike the use of unmoderated outputs in approximating the posterior class probabilities (as in [22]), the moderated output is more in line with the Bayesian idea that the posterior weight distribution should be taken into account on prediction, and this alleviates the usual tendency of assigning overly high confidence to its predictions in regions of low training data density. Experimentally, performance improvements on both artificial and real-world data sets are observed, especially in cases when the posterior variance is not too small and the moderated output is used in tandem with the rejection threshold.

In this paper, we have focused on classification problems. Extension to regression problems for the calculation of error bars is also straight-forward. Potentially, the ability to transform SVM output to posterior class probability estimate can yield a lot more benefits [20], such as compensating for different prior probabilities and using an ensemble of networks. These will be investigated in the future. Finally, as this work is based on the evidence framework, variability in the hyperparameter [i.e., the regularization parameter λ in (5)] is ignored. This issue will be studied and the application of other Bayesian techniques, like Markov Chain Monte Carlo methods [44] and the Gaussian process [45], will also be considered.

[8] On this Reuters-21 578 collection, Dumais *et al.* obtained 87% [22] and Joachims obtained 84.2% [8] by using SVM. The best result so far is 87.8% [42], obtained using a system with 100 decision trees. The best previously reported result is 85% [43]. However, notice that each of these schemes typically uses a slightly different text representation and preprocessing. For example, here we used stemming, the $tf \cdot idf$ representation and without feature selection. Dumais *et al.* [22], on the other hand, did not perform stemming and used a simpler binary representation (which encodes whether a particular word appears in the document or not), but required feature selection based on the mutual information.

Acknowledgment

The author would like to thank R. Vanderbei for the LOQO quadratic programming solver used in the experiments, and also 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] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods.* New York: Wiley, 1998.
[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
[4] B. Schölkopf, C. Burges, and A. Smola, *Advances in Kernel Methods: Support Vector Machines.* Cambridge, MA: MIT Press, 1998.
[5] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," NeuroCOLT2, Tech. Rep. NC2-TR-1998-030, 1998.
[6] V. Vapnik, *The Nature of Statistical Learning Theory.* Berlin, Germany: Springer-Verlag, 1995.
[7] ——, *Statistical Learning Theory.* New York: Wiley, 1998.
[8] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European Conf. Machine Learning*, 1998.
[9] J. T. Kwok, "Automated text categorization using support vector machine," in *Proc. Int. Conf. Neural Inform. Processing*, Kitakyushu, Japan, Oct. 1998, pp. 347–351.
[10] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proc. Comput. Vision Pattern Recognition*, Puerto Rico, June 1997.
[11] B. Schölkopf, "Support vector learning," Ph.D. dissertation, Technische Universität Berlin, Germany, 1997.
[12] C. J. C. Burges and B. Schölkopf, "Improving the accuracy and speed of support vector machines," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. San Mateo, CA: Morgan Kaufmann, 1996, pp. 375–381.
[13] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *NNSP*, 1997.
[14] B. Schölkopf, C. Burges, and V. Vapnik, "Incorporating invariances in support vector learning machines," in *Proc. Int. Conf. Artificial Neural Networks*, 1996.
[15] H. D. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. San Mateo, CA: Morgan Kaufmann, 1997, pp. 155–161.
[16] V. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. San Mateo, CA: Morgan Kaufmann, 1997, pp. 281–287.
[17] K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Proc. Int. Conf. Artificial Neural Networks*, 1997.
[18] J. Weston, A. Gammerman, M. Stitson, V. Vapnik, V. Vovk, and C. Watkins, "Density estimation using support vector machines," Royal Holloway Univ. London, U.K., Tech. Rep. CSD-TR-97-23, 1998.
[19] D. J. C. MacKay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, Sept. 1992.
[20] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Comput.*, vol. 3, pp. 461–483, 1991.

[21] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," Stanford Univ. Univ. Toronto, Tech. Rep., 1996.
[22] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proc. 7th Int. Conf. Inform. Knowledge Management,* 1998.
[23] J. Platt, 1998, private communication.
[24] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, "Learning to extract symbolic knowledge from the World Wide Web," in *Proc. 15th Nat. Conf. Artificial Intell.,* 1998.
[25] 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.
[26] C. K. I. Williams, "Prediction with Gaussian processes: From linear regression to linear prediction and beyond," in *Learning and Inference in Graphical Models,* M. I. Jordan, Ed. Boston, MA: Kluwer, 1998.
[27] J. T. Kwok, "The evidence framework applied to support vector machine," *IEEE Trans. Neural Networks,* submitted for publication.
[28] ——, "Integrating the evidence framework and the support vector machine," in *Proc. European Symp. Artificial Neural Networks,* Bruges, Belgium, Apr. 1999, pp. 177–182.
[29] D. J. C. MacKay, "Bayesian interpolation," *Neural Comput.,* vol. 4, no. 3, pp. 415–447, May 1992.
[30] ——, "Bayesian model comparison and backprop nets," in *Advances in Neural Information Processing Systems 4,* J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 839–846.
[31] ——, "A practical Bayesian framework for backpropagation networks," *Neural Comput.,* vol. 4, no. 3, pp. 448–472, May 1992.
[32] ——, "Probable networks and plausible predictions—A review of practical Bayesian methods for supervised neural networks," *Network: Comput. Neural Syst.,* vol. 6, no. 3, pp. 469–505, Aug. 1995.
[33] H. H. Thodberg, "A review of Bayesian neural networks with an application to near infrared spectroscopy," *IEEE Trans. Neural Networks,* vol. 7, pp. 56–72, 1996.
[34] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis,* 2nd ed. New York: Springer-Verlag, Springer Series in Statistics, 1985.
[35] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory.* New York: Wiley, 1994.
[36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C,* 2nd ed. New York: Cambridge Univ. Press, 1992.
[37] C. Blake, E. Keogh, and C. J. Merz, "UCI repository of machine learning databases," Univ. California, Irvine, Dept. Inform. Comput. Sci., 1998. Available http://www.ics.uci.edu/~mlearn/MLRepository.html
[38] V. Blanz, B. Schölkopf, H. Bulthoff, C. Burges, V. Vapnik, and T. Vetter, "Comparison of view-based object recognition algorithms using realistic 3-D models," in *Proc. Int. Conf. Artificial Neural Networks,* 1996.
[39] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," in *Proc. 1st Int. Conf. Knowledge Discovery Data Mining,* 1995.
[40] W. B. Frakes and R. Baeza-Yates, *Information Retrieval: Data Structures and Algorithms.* Englewood Cliffs, NJ: Prentice-Hall, 1992.
[41] D. D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task," in *SIGIR Forum,* Copenhagen, Denmark, June 1992, pp. 37–50.
[42] C. Apte, F. Damerau, and S. Weiss, "Text mining with decision rules and decision trees," in *Proc. Conf. Automat. Learning Discovery,* June 1998.
[43] Y. Yang, "An evaluation of statistical approaches to text categorization," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, CMU-CS 97-127, 1997.
[44] R. M. Neal, *Bayesian Learning for Neural Networks, Lecture Notes in Statistics.* Berlin, Germany: Springer, 1996.
[45] 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: Morgan Kaufmann, 1997, pp. 295–301.

James Tin-Yau Kwok (M'98) 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 with the Department of Computer Science at the Hong Kong Baptist University. His research interests include support vector machines, artificial neural networks, and information retrieval.