Model Selection for Support Vector Machine Classification

Carl Gold
Computation and Neural Systems
California Institute of Technology, 139-74
Pasadena, CA 91125
Email carlg@caltech.edu

Peter Sollich
Department of Mathematics, King's College London
Strand, London WC2R 2LS, U.K.
Email peter.sollich@kcl.ac.uk

March 15, 2002

Abstract

We address the problem of model selection for Support Vector Machine (SVM) classification. For fixed functional form of the kernel, model selection amounts to tuning kernel parameters and the slack penalty coefficient C. We begin by reviewing a recently developed probabilistic framework for SVM classification. An extension to the case of SVMs with quadratic slack penalties is given and a simple approximation for the evidence is derived, which can be used as a criterion for model selection. We also derive the exact gradients of the evidence in terms of posterior averages and describe how they can be estimated numerically using Hybrid Monte Carlo techniques. Though computationally demanding, the resulting gradient ascent algorithm is a useful baseline tool for probabilistic SVM model selection, since it can locate maxima of the exact (unapproximated) evidence. We then perform extensive experiments on several benchmark data sets. The aim of these experiments is to compare the performance of probabilistic model selection criteria with alternatives based on estimates of the test error, namely the so-called “span estimate” and Wahba’s Generalized Approximate Cross-Validation (GACV) error. We find that all the “simple” model criteria (Laplace evidence approximations, and the Span and GACV error estimates) exhibit multiple local optima with respect to the hyperparameters. While some of these give performance that is competitive with results from other approaches in the literature, a significant fraction lead to rather higher test errors. The results for the evidence gradient ascent method show that also the exact evidence exhibits local optima, but these give test errors which are much less variable and also consistently lower than for the simpler model selection criteria.

Keywords: Support Vector Machines, model selection, probabilistic methods, Bayesian evidence

1 Introduction

Support Vector Machines (SVMs) have emerged in recent years as powerful techniques both for regression and classification. One of the central open questions is model selection: how does one tune the parameters of the SVM algorithm to achieve optimal generalization performance? We focus on the case of SVM classification, where these “hyperparameters” include any parameters appearing in the SVM kernel, as well as the penalty parameter C for violations of the margin constraint.

Our aim in this paper is twofold. First, we extend our work on probabilistic methods for SVMs to the case of quadratic slack penalties; we also develop a “baseline” algorithm which can be used to find in principle exact maxima of the evidence. Second, we perform numerical experiments on a selection benchmark data sets to compare the model selection criteria derived from the probabilistic view of SVMs with alternatives that directly try to optimize estimates of test error. Our focus in these experiments is less on computational efficiency, but rather on the relative merits of the methods in terms of the resulting generalization performance.

We begin in Sec. 2 with a brief review of SVM classification and of its probabilistic interpretation; the setup will be such that the extension of the probabilistic point of view to the quadratic penalty case requires only small changes compared to linear penalty SVMs. In Sec. 3 we review some criteria for model selection that have been proposed based on approximations to the test error. We also describe previous approximations to the evidence for the linear penalty SVM, and then give an analogue for quadratic penalty SVMs. Exact expressions for gradients of the evidence with respect to the hyperparameters are then derived in terms of averages over the posterior. Sec. 4 has a description of the methods we use in our numerical experiments on model selection, including the Hybrid Monte Carlo algorithm which we use to calculate evidence gradients numerically. The results of our experiments on benchmark data sets are discussed in Sec. 5; we conclude in Sec. 6 with a summary and an outlook towards future work.

2 SVM classification

In this section, we give a very brief review of SVM classification; for details the reader is referred to recent textbooks or review articles such as [1, 2]. We also sketch the probabilistic interpretation of SVMs, from which we later obtain Bayesian criteria for SVM model selection.

Suppose we are given a set D of n training examples (x_i, y_i) with binary outputs $y_i = \pm 1$ corresponding to the two classes. The basic SVM idea is to map the inputs x to vectors $\boldsymbol{\phi}(x)$ in some high-dimensional feature space; ideally, in this feature space, the problem should be linearly separable. Suppose first that this is true. Among all decision hyperplanes $\mathbf{w}\cdot\boldsymbol{\phi}(x)+b = 0$ which separate the training examples (i.e. which obey $y_i(\mathbf{w}\cdot\boldsymbol{\phi}(x_i)+b) > 0$ for all $x_i \in X$, X being the set of training inputs), the SVM solution is chosen as the one with the largest *margin*, i.e. the largest minimal distance from any of the training examples. Equivalently, one specifies the margin to be equal to 1 and minimizes the squared length of the weight vector $||\mathbf{w}||^2$ [2], subject to the constraint that $y_i(\mathbf{w} \cdot \boldsymbol{\phi}(x_i) + b) \geq 1$ for all i. The quantities $y_i(\mathbf{w}\cdot\boldsymbol{\phi}(x_i)+b)$ are again called margins, although for an unnormalized weight vector they no longer represent geometrical distances [2]. This leads to the following optimization problem: Find a weight vector $\mathbf{w}$ and an offset b such that $\frac{1}{2}||\mathbf{w}||^2$ is minimized, subject to the constraint that $y_i(\mathbf{w} \cdot \boldsymbol{\phi}(x_i) + b) \geq 1$ for all training examples.

If the problem is not linearly separable, or if one wants to avoid fitting noise in the training data, 'slack variables' $\xi_i \geq 0$ are introduced which measure how much the margin constraints are violated; one thus writes $y_i(\mathbf{w}\cdot$

$\boldsymbol{\phi}(x_i) + b) \geq 1 - \xi_i$. To control the amount of slack allowed, a penalty term $(C/p)\sum_i \xi_i^p$ is then added to the objective function $\frac{1}{2}||\mathbf{w}||^2$, with a penalty coefficient C. Common values for the exponent parameter are $p = 1$ and $p = 2$, giving linear and quadratic slack penalties respectively. Training examples with $y_i(\mathbf{w} \cdot \boldsymbol{\phi}(x_i) + b) \geq 1$ (and hence $\xi_i = 0$) incur no penalty; the others contribute $(C/p)[1 - y_i(\mathbf{w} \cdot \boldsymbol{\phi}(x_i) + b)]^p$ each. This gives the SVM optimization problem: Find $\mathbf{w}$ and b to minimize

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_i l_p(y_i[\mathbf{w} \cdot \boldsymbol{\phi}(x_i) + b]) \tag{1}$$

where $l_p(z)$ is the loss function

$$l_p(z) = \frac{1}{p}(1 - z)^p H(1 - z) \tag{2}$$

The Heaviside step function $H(1 - z)$ (defined as $H(a) = 1$ for $a \geq 0$ and $H(a) = 0$ otherwise) ensures that this is zero for $z > 1$. For $p = 1$, $l_p(z)$ is called (shifted) hinge loss or soft margin loss.

In the following we modify the basic SVM problem by adding the quadratic term $\frac{1}{2}b^2/B^2$ to (1), thus introducing a penalty for large offsets b. A discussion of why this is reasonable, certainly within a probabilistic view, can be found in [3]; at any rate the standard formulation can always be retrieved by making the constant B large. We can now define an augmented weight vector $\tilde{\mathbf{w}} = (b/B, \mathbf{w})$ and augmented feature space vectors $\tilde{\boldsymbol{\phi}}(x) = (B, \boldsymbol{\phi}(x))$ so that the modified SVM problem is to find a $\tilde{\mathbf{w}}$ which minimizes

$$\frac{1}{2}||\tilde{\mathbf{w}}||^2 + C\sum_i l_p(y_i\tilde{\mathbf{w}} \cdot \tilde{\boldsymbol{\phi}}(x_i)) \tag{3}$$

This statement of the problem is useful for the probabilistic interpretation of SVM classification, of which more shortly. For a practical solution, one uses Lagrange multipliers α_i conjugate to the constraints $y_i\tilde{\mathbf{w}} \cdot \tilde{\boldsymbol{\phi}}(x_i) \geq 1 - \xi_i$ and finds in the standard way (see e.g. [2]) that the optimal (augmented) weight vector is $\tilde{\mathbf{w}}^* = \sum_i y_i \alpha_i \boldsymbol{\phi}(x_i)$. For the linear penalty case $p = 1$, the α_i are found from

$$\max_{0 \leq \alpha_i \leq C} \left(\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij} \right) \tag{4}$$

Here $K_{ij} = K(x_i, x_j)$ are the elements of the Gram matrix $\mathbf{K}$, obtained by evaluating the *kernel* $K(x, x') = \tilde{\boldsymbol{\phi}}(x) \cdot \tilde{\boldsymbol{\phi}}(x') = \boldsymbol{\phi}(x) \cdot \boldsymbol{\phi}(x') + B^2$ for all pairs of training inputs. The corresponding optimal latent or discrimination function is $\theta^{*}(x) = \tilde{\mathbf{w}}^{*} \cdot \tilde{\boldsymbol{\phi}}(x) = \sum_i y_i \alpha_i K(x, x_i)$. Only the x_i with $\alpha_i > 0$ contribute to this sum; these are called support vectors (SVs). SVs fall into two groups: If $\alpha_i < C$, one has $y_i \theta_i^{*} \equiv y_i \theta^{*}(x_i) = 1$; we will call these the "marginal SVs" because their margins are exactly at the allowed limit where no slack penalty is yet incurred. For $\alpha_i = C$, on the other hand, $y_i \theta_i^{*} \leq 1$, and these "hard SVs" are the points at which the slack penalty is active. Non-SVs have large margins, $y_i \theta_i^{*} \geq 1$.

For the quadratic penalty case $p = 2$, the α_i are obtained as the solution of (see e.g. [2])

$$\max_{0 \leq \alpha_i} \left(\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij}^{C} \right) \tag{5}$$

where $K_{ij}^{C} = K_{ij} + C^{-1}\delta_{ij}$. Apart from the replacement of K by K^{C}, this maximization problem is the same as (4) for the linear penalty case in the limit $C \to \infty$ where no violations of the margin constraints are allowed. There is now only one kind of SV, identified by $\alpha_i > 0$. It follows by differentiating (5) that for a SV one has $y_i \sum_j \alpha_j y_j K_{ij}^{C} = 1$. The margin for a SV is thus $y_i \theta_i^{*} = y_i \sum_j \alpha_j y_j K_{ij} = 1 - \alpha_i / C$, so that all SVs incur a nonzero slack penalty. Non-SVs again have $y_i \theta_i^{*} \geq 1$.

We now turn to the probabilistic interpretation of SVM classification (see Refs. [3, 4, 5] and the works quoted below). The aim of such an interpretation is to allow the application of Bayesian methods to SVMs, without modifying the basic SVM algorithm which already has a large user community. (An alternative philosophy would be to consider similar inference algorithms which share some of the benefits of SVMs but are constructed directly from probabilistic models; Tipping's Relevance Vector Machine [6] is a successful example of this.) One regards (3) as defining a negative log-posterior probability for the parameters $\tilde{\mathbf{w}}$ of the SVM, given a training set D. The conventional SVM classifier is then interpreted as the maximum a posteriori (MAP) solution of the corresponding probabilistic inference problem. The first term in (3) gives the prior $Q(\tilde{\mathbf{w}}) \propto \exp(-\frac{1}{2}||\tilde{\mathbf{w}}||^2)$. This is a Gaussian prior on $\tilde{\mathbf{w}}$; the components of $\tilde{\mathbf{w}}$ are uncorrelated with each other and have unit variance. Because only the 'latent function' values $\theta(x) = \tilde{\mathbf{w}} \cdot \tilde{\boldsymbol{\phi}}(x)$—rather than $\tilde{\mathbf{w}}$ itself—appear in the second, data dependent term of (3), it makes sense to express the prior directly as a distribution over these. The $\theta(x)$ have a joint Gaussian distribution because the components of $\tilde{\mathbf{w}}$ do, with covariances given by

$$\langle\theta(x)\theta(x')\rangle = \left\langle(\tilde{\boldsymbol{\phi}}(x)\cdot\tilde{\mathbf{w}})(\tilde{\mathbf{w}}\cdot\tilde{\boldsymbol{\phi}}(x'))\right\rangle = K(x,x')$$

The SVM prior is therefore simply a *Gaussian process* (GP) over the functions θ, with zero mean and with the kernel $K(x,x')$ as covariance function. This link between SVMs and GPs has been pointed out by a number of authors, *e.g.* [7, 8, 9]. It can be understood from the common link to reproducing kernel Hilbert spaces [10], and can be extended from SVMs to more general kernel methods [11]. For connections to regularization operators see also [12]. A nice introduction to inference with Gaussian processes can be found in Ref. [13].

The second term in (3) similarly becomes a (negative) log-likelihood if we define the probability of obtaining output y for a given x (and θ) as

$$Q(y=\pm1|x,\theta) = \kappa(C)\exp[-Cl_p(y\theta(x))] \tag{6}$$

The constant factor $\kappa(C)$ is determined from $\kappa^{-1}(C) = \max_z\{e^{-Cl_p(z)} + e^{-Cl_p(-z)}\}$ to ensure that $\sum_{y=\pm1} Q(y|x,\theta) \leq 1$. In the linear penalty case this gives $\kappa(C) = 1/[1+\exp(-2C)]$; for the quadratic penalty SVM, the maximum in the definition of $\kappa^{-1}(C)$ is assumed at a value of θ obeying $\theta = \tanh(C\theta)$ and so $\kappa(C)$ can easily be found numerically. The likelihood for the complete data set (more precisely, for the training outputs $Y = (y_1 \ldots y_n)$ given the training inputs X) is then

$$Q(Y|X,\theta) = \prod_i Q(y_i|x_i,\theta)$$

With these definitions eq. (3) is, up to unimportant constants, equal to the log-posterior[1]

$$\ln Q(\theta|X,Y) = -\frac{1}{2}\sum_{x,x'}\theta(x)K^{-1}(x,x')\,\theta(x') - C\sum_i l_p(y_i\theta(x_i)) + \text{const} \tag{7}$$

By construction, the maximum of $\theta^*(x)$ gives the conventional SVM classifier, and this is easily verified explicitly [3].

[1] In (7) the unrestricted sum over x runs over all possible inputs, and $K^{-1}(x,x')$ are the elements of the inverse of $K(x,x')$, viewed as a matrix. We assume here that the input domain is discrete. This avoids mathematical subtleties with the definition of determinants and inverses of operators (rather than matrices), while maintaining a scenario that is sufficiently general for all practical purposes.

The probabilistic model defined above is not normalized, since $\sum_{y=\pm 1} Q(y|x,\theta) < 1$ for generic values of $\theta(x)$. The implications of this have been previously discussed in detail [3]. The normalization of the model is important for the theoretical justification of tuning hyperparameters via maximization of the data likelihood or "evidence". Nevertheless, experiments in [3] showed that promising results for hyperparameter optimization could be obtained also with the unnormalized version of the model. We will therefore, in common with other work on probabilistic interpretations of SVMs [7, 8, 9, 14, 15], disregard the normalization issue from now on. We will also focus on SVM classifiers constructed from radial basis function (RBF) kernels

$$K(x,x') = k_0 \exp\left[-\sum_a \frac{(x^a - x'^a)^2}{2l_a^2}\right] + k_{\text{off}} \tag{8}$$

where the x^a are the different input components, k_0 is the kernel amplitude and k_{off} the kernel offset; k_{off} corresponds to the term B^2 discussed above that arises by incorporating the offset b into the kernel. Each input dimension has associated with it a length scale l_a. Since in the probabilistic interpretation $K(x,x')$ is the prior covariance function of the latent function $\theta(x)$, each l_a determines the distance in the x^a-direction over which $\theta(x)$ is approximately constant; large l_a correspond to an input component of little relevance (see e.g. [16]).

3 Model selection criteria

3.1 Error bounds and approximations

Model selection aims to tune the hyperparameters of SVM classification (the penalty parameter C and any kernel parameters) in order to achieve the lowest test error ϵ, *i.e.* the lowest probability of misclassification of unseen test examples. The test error is not observable directly, and so one is lead to use bounds or approximations as model selection criteria. The simplest such bounds [17, 18] which have been applied as model selection criteria [19, 20] are expressed in terms of the quantity

$$\frac{R^2}{n}\sum_i \alpha_i \tag{9}$$

Here R is the radius of the smallest ball in feature space containing all training examples, while $\sum_i \alpha_i$ can be shown to equal the inverse square of the distance between the separating hyperplane and the closest training points. For RBF kernels, R is bounded by a constant since every input point has the same squared distance $\tilde{\boldsymbol{\phi}}(x) \cdot \tilde{\boldsymbol{\phi}}(x) = K(x,x)$ from the origin.

More recent work has shown that better bounds and approximations can be obtained for the leave-out-out error $\epsilon_{\rm loo}$. If $\theta^i(x)$ is the latent function obtained by training the SVM classifier on the data set with example (x_i, y_i) left out, then $\epsilon_{\rm loo}$ is the probability of misclassification of the left-out example if this procedure is applied to each data point in turn,

$$\epsilon_{\rm loo} = \frac{1}{n}\sum_i H(-y_i\theta_i^i) \tag{10}$$

where we have abbreviated $\theta_i^i \equiv \theta^i(x_i)$. Averaged over data sets this is an unbiased estimate of the average test error that is obtained from training sets of $n-1$ examples. This says nothing about the variance of this estimate; nevertheless, one may hope that $\epsilon_{\rm loo}$ is a reasonable proxy for the test error that one wishes to optimize. (This is in contrast to the training error, *i.e.* the fraction of all n training examples misclassified when training on the complete data set, which is general a strongly biased estimate of test error.) For large data sets, $\epsilon_{\rm loo}$ is time-consuming to compute and one is driven to look for cheaper bounds or approximations. Since removing non-SVs from the data set does not change the SVM classifier, a trivial bound on $\epsilon_{\rm loo}$ is the sum of the training error and the fraction of support vectors, both obtained when training on all n examples. To get better bounds, one writes

$$\epsilon_{\rm loo} = \frac{1}{n}\sum_i H\left(y_i[\theta_i^* - \theta_i^i] - y_i\theta_i^*\right)$$

which shows that an upper bound on $y_i[\theta_i^* - \theta_i^i]$ will give an upper bound on $\epsilon_{\rm loo}$. Jaakkola and Haussler proved a bound of this form, $y_i[\theta_i^* - \theta_i^i] \leq \alpha_i K_{ii}$; as before, the α_i are those obtained from training on the full data set. More sophisticated bounds were given by Chappelle and Vapnik [21, 20, 22] in terms of what they called the "span". We focus on the case of quadratic penalty SVMs, where the span estimates are simplest to state. In the simplified version of Ref. [22], and adapting to our formulation with the offset b incorporated into the kernel, the span S_i for a support vector can be defined as

$$S_i^2 = \min_{\lambda} \sum_{j,k} \lambda_j \lambda_k K_{jk}^C \tag{11}$$

where the minimum is over all $\lambda = (\lambda_1 \ldots \lambda_n)$ with $\lambda_i = -1$, and $\lambda_j = 0$ whenever $\alpha_j = 0$. With this definition, one can calculate $y_i[\theta_i^* - \theta_i^i]$ exactly under the assumption that dropping the point x_i from the training set leaves the "SV set" unchanged, in the sense that no new SVs arise in the new classifier and that all old SVs remain. One thus finds $y_i[\theta_i^* - \theta_i^i] = \alpha_i(S_i^2 - 1/C)$. S_i^2 can also be worked out explicitly as $S_i^2 = 1/[(\mathbf{K}_{\mathrm{SV}} + \mathbf{I}/C)^{-1}]_{ii}$, where $\mathbf{K}_{\mathrm{SV}}$ is the Gram matrix $\mathbf{K}$ restricted to the SVs and $\mathbf{I}$ is the unit matrix. (The same result was obtained by Opper and Winther [9] using a slightly different approach.) Using finally that $y_i\theta_i^* = 1 - \alpha_i/C$ for quadratic penalty SVMs, one thus has

$$\epsilon_{\mathrm{loo}} \approx \epsilon_{\mathrm{span}} = \frac{1}{n}\sum_i H\left(\alpha_i S_i^2 - 1\right), \qquad S_i^2 = 1/[(\mathbf{K}_{\mathrm{SV}} + \mathbf{I}/C)^{-1}]_{ii} \tag{12}$$

This is only an approximation because the assumption of an unchanged SV set will not hold for every SV removed from the training set.

The span estimate (12) of leave-one-out error has the undesirable property of being discontinuous as hyperparameters are varied, making numerical optimization difficult. The discontinuity arises from the discontinuity in the Heaviside step function H, and from the fact that the size of the matrix $\mathbf{K}_{\mathrm{SV}}$ changes as training examples enter or leave the set of SVs. To get around this [22], one can approximate $H(z)$ by a sigmoidal function $1/[1 + \exp(-c_1 x + c_2)]$ and smooth the span by adding a penalty that forces any nonzero λ_j to go to zero when $\alpha_j \to 0$. This gives the modified span definition

$$S_i^2 = \min_\lambda \sum_{j,k} \lambda_j \lambda_k K_{jk}^C + \eta \sum_{j \neq i} \frac{\lambda_j^2}{\alpha_j}$$

with the minimum taken over the same λ as in (11). Explicitly, one finds

$$S_i^2 = \frac{1}{[(\mathbf{K}_{\mathrm{SV}} + \mathbf{I}/C + \eta \mathbf{A}_{\mathrm{SV}}^{-1})^{-1}]_{ii}} - \frac{\eta}{\alpha_i}$$

where $\mathbf{A}_{\mathrm{SV}}$ is the diagonal matrix containing the nonzero α_i. This is easily seen to be continuous even when the set of SVs changes as hyperparameters are varied. For $\eta \to 0$ one recovers the original span definition (11); for $\eta \to \infty$, on the other hand, $S_i^2 \to K_{ii}^C = K_{ii} + 1/C$ and one recovers the Jaakkola and Haussler bound. Overall, the smoothed span estimate for ϵ_{loo} contains three smoothing parameters c_1, c_2 and η.

For linear penalty SVMs, Wahba [10] considered a modified version of ϵ_{loo}, obtained by replacing the Heaviside step function $H(-z)$ in (10) by the hinge loss $l_1(z) = (1-z)H(1-z)$; since $l_1(z) \geq H(-z)$, this actually gives an upper bound on ϵ_{loo}. Wahba's GACV (generalized approximate cross-validation) estimate for this modified ϵ_{loo} is

$$\epsilon_{\text{gacv}} = \frac{1}{n}\sum_i \left[l_1(y_i\theta_i^*) + \alpha_i K_{ii} f(y_i\theta_i^*)\right] \tag{13}$$

where

$$f(z) = \begin{cases} 2 & x < -1 \\ 1 & -1 \leq x \leq 1 \\ 0 & x > 1 \end{cases}$$

The first term in the sum in (13) would just give the naive estimate of the (modified) ϵ_{loo} from the performance on the training set; the second term effectively corrects for the bias in this estimate. Because of the nature of the function f, ϵ_{gacv} can exhibit discontinuities as hyperparameters are varied and this has to be taken into account when minimizing it numerically.

3.2 Approximations to the evidence

From a probabilistic point of view, it is natural to tune hyperparameters to maximize the likelihood of the data $Q(Y|X)$. By definition, $Q(Y|X) = \int d\theta\, Q(Y|X,\theta)Q(\theta)$ where the integration is over the values $\theta(x)$ of the latent function θ at all different input points x. The likelihood $Q(Y|X,\theta)$ only depends on the values $\theta_i \equiv \theta(x_i)$ of θ at the training inputs; all other $\theta(x)$ can be integrated out trivially, so that

$$Q(Y|X) = \int d\boldsymbol{\theta}\, Q(Y|X,\boldsymbol{\theta})Q(\boldsymbol{\theta})$$

where now the integral is over the n-dimensional vector $\boldsymbol{\theta} = (\theta_1 \ldots \theta_n)$. Because $Q(\theta)$ is a zero mean Gaussian process, the marginal $Q(\boldsymbol{\theta})$ is a zero mean Gaussian distribution with covariance matrix $\mathbf{K}$. The evidence is therefore

$$Q(Y|X) = |2\pi\mathbf{K}|^{-1/2}\kappa^n(C)\int d\boldsymbol{\theta}\,\exp\left[-\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\theta} - \sum_i Cl_p(y_i\theta_i)\right] \tag{14}$$

This n-dimensional integral is in general impossible to carry out exactly. But it can be approximated by expanding the exponent around its maximum

$\boldsymbol{\theta}^*$, the solution of the SVM optimization problem. For the linear penalty case, this requires some care: because of the kink in the hinge loss $l_1(z)$, the θ_i corresponding to marginal SVs have to be treated separately. The result for the normalized log-evidence, suitably smoothed to avoid spurious singularities, is [3]

$$E(Y|X) \equiv \frac{1}{n} \ln Q(Y|X) = \frac{1}{n} \ln Q^*(Y|X) - \frac{1}{2n} \ln \det(\mathbf{I} + \mathbf{L}_m \mathbf{K}_m) \qquad (15)$$

where $\mathbf{K}_m$ is the sub-matrix of the Gram matrix corresponding to the marginal SVs, $\mathbf{L}_m$ is the diagonal matrix with entries $2\pi[\alpha_i(C - \alpha_i)/C]^2$ and

$$\frac{1}{n} \ln Q^*(Y|X) = -\frac{1}{2n} \sum_i \alpha_i y_i \theta_i^* - \frac{C}{n} \sum_i l_p(y_i \theta_i^*) + \ln \kappa(C) \qquad (16)$$

The approximation (15) is computationally efficient because it only involves calculating the determinant of a single matrix of size equal to the number of marginal SVs. A related approximation was proposed by Kwok [15]. He suggested to smooth the kink in the hinge loss by using a sigmoidal approximation for the Heaviside function, giving $l_1(z) \approx s(z) = (1 - z)/[1 + \exp[-c(1 - z)]$ with a smoothing parameter c. This has the disadvantage that the SVM solution θ^* is no longer a maximum of the smoothed posterior. The analogous result to (15) also involves, instead of $\mathbf{K}_m$ and $\mathbf{L}_m$, the whole Gram matrix and a diagonal matrix with entries $Cs''(y_i\theta_i^*)$, respectively. One thus needs either to evaluate a large determinant of size n, or—somewhat arbitrarily—to truncate small values of $s''(y_i\theta_i^*)$ to zero to reduce the size of the problem. We therefore do not consider this approach further.

An approximation similar to (15) can easily be derived for the quadratic penalty case. The loss function $l_2(z)$ now has a continuous first derivative and so all θ_i can be treated on the same footing. The derivation of the Laplace approximation is thus standard, and involves the Hessian of the log-likelihood at the maximum $\boldsymbol{\theta}^*$; the resulting approximation to the (normalized log-) evidence is

$$E(Y|X) = \frac{1}{n} \ln Q^*(Y|X) - \frac{1}{2n} \ln \det(\mathbf{I} + \mathbf{M}_{\mathrm{SV}} \mathbf{K}_{\mathrm{SV}}) \qquad (17)$$

where $\mathbf{M}_{\mathrm{SV}}$ is a diagonal matrix containing the second derivatives $Cl_2''(y_i\theta_i^*)$ of the loss function evaluated for all the SVs. The calculation of $E(Y|X)$ according to (17) requires only the determinant of a matrix whose size is the number of SVs, again expected to be manageably small. The matrix $\mathbf{M}_{\rm SV}$ as defined above is just a multiple of the unit matrix, since $l_2''(z) = H(1-z)$ and $z = y_i\theta_i^* < 1$ for SVs. However, the step discontinuity in $l_2''(z)$ at $z = 1$ has the undesirable consequence that the Laplace approximation to the evidence will jump discontinuously when one or several of the α_i reach zero as hyperparameters are varied. We therefore smooth the result by replacing $l_2''(z) = H(1-z)$ in the definition of $\mathbf{M}_{\rm SV}$ by the approximation $l_2''(z) \approx \exp[-a/(1-z)]$ for $z < 1$ (and 0 for $z \geq 1$). This is smooth at $z = 1$ and also has continuous derivatives of all orders at this point. The value of a determines the range of values of $z = y_i\theta_i$ around 1 for which the smoothing is significant, with $a \to 0$ recovering the Heaviside step function.

3.3 Evidence gradients

Beyond the relatively simple approximations to the evidence derived above, it is difficult to obtain accurate numerical estimates of the evidence. This is a well-known general problem: while averages over probability distributions are straightforward to obtain, normalization constants for such distributions – such as the evidence, which is the normalization factor for the posterior – require much greater numerical effort (see *e.g.* [23]). To avoid this problem, one can estimate the gradients of the evidence with respect to the hyperparameters and use these in a gradient ascent algorithm, without ever calculating the value of the evidence itself. As we show in this section, these gradients can be expressed as averages over the posterior distribution, which one can then estimate by sampling as explained in Sec. 4.2.

Starting from eq. (14) we can find the derivative of the normalized log-evidence $E(Y|X) = n^{-1}\ln Q(Y|X)$ w.r.t. the penalty (or noise) parameter C:

$$\begin{aligned} \frac{\partial}{\partial C}E(Y|X) &= \frac{\partial \ln\kappa(C)}{\partial C} - \frac{\int d\boldsymbol{\theta}\, Q(\boldsymbol{\theta}) \sum_i l_p(y_i\theta_i) \exp\left[-C\sum_i l_p(y_i\theta_i)\right]}{\int d\boldsymbol{\theta}\, Q(\boldsymbol{\theta}) \exp\left[-C\sum_i l_p(y_i\theta_i)\right]} \\ &= \frac{\partial \ln\kappa(C)}{\partial C} - \left\langle \frac{1}{n}\sum_i l_p(y_i\theta_i) \right\rangle \end{aligned} \tag{18}$$

where the average is, as expected, over the posterior $Q(\boldsymbol{\theta}|D) \propto Q(Y|X,\boldsymbol{\theta})Q(\boldsymbol{\theta})$. Similarly, the derivative of the log-evidence w.r.t. any parameter λ appearing in the kernel is

$$\begin{aligned}\frac{\partial}{\partial\lambda}E(Y|X) &= -\frac{1}{2n}\frac{\partial}{\partial\lambda}\ln|2\pi\mathbf{K}| - \frac{\int d\boldsymbol{\theta}\,\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\frac{\partial}{\partial\lambda}\mathbf{K}^{-1}\boldsymbol{\theta}\exp\left[-\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\theta} - \sum_i Cl_p(y_i\theta_i)\right]}{\int d\boldsymbol{\theta}\,\exp\left[-\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\theta} - \sum_i Cl_p(y_i\theta_i)\right]} \\ &= -\frac{1}{2n}\mathrm{tr}\left(\frac{\partial\mathbf{K}}{\partial\lambda}\mathbf{K}^{-1}\right) + \frac{1}{2n}\left\langle\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\lambda}\mathbf{K}^{-1}\boldsymbol{\theta}\right\rangle \\ &= -\frac{1}{2n}\mathrm{tr}\left(\frac{\partial\mathbf{K}}{\partial\lambda}\mathbf{K}^{-1}\left\langle\mathbf{I} - \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\right\rangle\right) \qquad (19)\end{aligned}$$

Numerical evaluation of this expression as it stands would be unwise, since the difference $\left\langle\mathbf{I} - \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\right\rangle$ can be much smaller than the two contributions individually; in fact, for $n = 0$ we know that it is exactly zero. It is better to rewrite (19), using the fact that the elements of the matrix $\mathbf{I} - \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}$ can be obtained as

$$\delta_{ij} - \theta_i(\mathbf{K}^{-1}\boldsymbol{\theta})_j = \exp\left[\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\theta}\right]\frac{\partial}{\partial\theta_j}\theta_i\exp\left[-\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\theta}\right]$$

The posterior average can thus be worked out using integration by parts, giving

$$\left\langle\delta_{ij} - \theta_i(\mathbf{K}^{-1}\boldsymbol{\theta})_j\right\rangle = \left\langle Cl'_p(y_j\theta_j)y_j\theta_i\right\rangle$$

If we define the matrix $\mathbf{Y}$ as the diagonal matrix with entries y_i so that $(\mathbf{Y}\boldsymbol{\theta})_i = y_i\theta_i$, and denote by $l'_p(\mathbf{Y}\boldsymbol{\theta})$ the vector with entries $l'_p(y_i\theta_i)$, then this can be written in the compact form

$$\left\langle\mathbf{I} - \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\right\rangle = C\left\langle\boldsymbol{\theta}[l'_p(\mathbf{Y}\boldsymbol{\theta})]^{\mathrm{T}}\mathbf{Y})\right\rangle$$

Combining this with equation (19), one has finally

$$\frac{\partial}{\partial\lambda}E(Y|X) = -\frac{C}{2n}\left\langle[l'_p(\mathbf{Y}\boldsymbol{\theta})]^{\mathrm{T}}\mathbf{Y}\frac{\partial\mathbf{K}}{\partial\lambda}\mathbf{K}^{-1}\boldsymbol{\theta}\right\rangle \qquad (20)$$

This expression appears to require the inverse $\mathbf{K}^{-1}$ of the Gram matrix, which for large n would be computationally expensive to evaluate; however, as described in Sec. 4.2 the sampling from the posterior using Hybrid Monte Carlo can be arranged so that samples of both $\boldsymbol{\theta}$ and $\mathbf{K}^{-1}\boldsymbol{\theta}$ are obtained without requiring explicit matrix inversions.

4 Numerical methods

In Sec. 5 below, we report results from SVM model selection experiments using the criteria described above. Specifically, for linear penalty SVMs we compare maximizing the Laplace approximation to the evidence $E(Y|X)$, eq. (15), with minimizing Wahba's ϵ_{gacv}, eq. (13); for quadratic penalty SVMs we again use the relevant approximation to the evidence, eq. (17), and contrast with minimization of the span estimate ϵ_{span} of the leave-one-out error, eq. (12). These four model selection criteria are "simple" in the sense that they can be evaluated explicitly at moderate computational cost. In order to be able to compare the different criteria directly, and because one of them (ϵ_{gacv}) has possible discontinuities as a function of the hyperparameters, we use a simple greedy random walk algorithm for optimization that is described in Sec. 4.1. For the other three criteria, more efficient gradient-based optimization algorithms can be designed [22] but since our focus here is not on computational efficiency we do not consider these.

For linear penalty SVMs we also studied evidence optimization using numerical estimates of the evidence gradients (18,20). The Monte Carlo method used to perform the necessary averages over the posterior is outlined in Sec. 4.2, while Sec. 4.3 describes the details of the gradient ascent algorithm. Note that our use of evidence gradients provides a baseline for model selection methods based on approximations to the evidence since it locates, up to small statistical errors from the Monte Carlo sampling of posterior averages, a local maximum of the exact evidence.

In all experiments using approximations to the test error (ϵ_{span} and ϵ_{gacv}) as model selection criteria, the hyperparameters being optimized were the parameters of the RBF kernel (8), *i.e.* the amplitudes k_0, k_{off} and the logarithms of the length scales l_a (one per input dimension). The penalty parameter C can be fixed to *e.g.* $C = 1$ since these criteria depend only on the properties of the SVM solution (*i.e.* the maximum of the posterior, rather than the whole posterior distribution). This SVM solution only depends on the product $CK(x, x')$ rather than C and the kernel individually, as one easily sees from (4,5) or equivalently from (7). In contrast, the evidence (14) takes into account both the position of the posterior maximum and the shape of the posterior distribution around this maximum; the latter does depend on C. We therefore include C as a hyperparameter to be optimized in evidence maximization. The SVM predictor of the final selected model will of course again be dependent only on the product $CK(x, x')$; but the value of

C itself would be important *e.g.* for the determination of predictive class probabilities. This issue, which we do not pursue here, is discussed in detail in Ref. [3].

4.1 Optimization of "simple" model selection criteria

We optimized the simple model selection criteria using a simple greedy random walk, or "zero temperature Monte Carlo" search. This is a simple adaptation of the common Metropolis Algorithm (see e.g. [24]) used to sample from a probability distribution; in the zero temperature limit the algorithm reduces to repeatedly adding a small step (which we take to be Gaussian) to each parameter, recalculating the quantity being optimized, and moving to the new point if and only if the new point yields a better value (higher for the evidence, and lower for error estimates). The randomness in the algorithm may appear disadvantageous in terms of computational efficiency; but for our purposes, it is actually helpful since it allows us to assess whether the model selection criteria in question have a number of local optima or a single (global) optimum. It also made further randomization over the initial hyperparameter values unnecessary, and so the experiments with the simple model selection criteria were all started with a fixed set of initial values for the SVM hyperparameters. A few preliminary trials were used to choose initial values with an appropriate order of magnitude, and all results reported were initialized with the hyperparameters $C = 1$, $l_a = 1$ for all length scales, $k_0 = 1$ and $k_{\mathrm{off}} = 0.1$.

The span error estimate and the Laplace evidence for quadratic loss each have additional smoothing parameters that had to be selected (c_1, c_2 and η in the span estimate, a for the Laplace evidence; see Secs. 3.1 and 3.2 respectively). Appropriate values for these parameters were found by a simple (log) line search in the parameter values. These tests were not done extensively, but the results for SVM model selection did not seem to depend strongly on the values of these parameters as long as they were of a reasonable order of magnitude. For all tests presented here the values used were $c_1 = 5$, $c_2 = 0$, $\eta = 1$ for the span estimate, and $a = 0.1$ for the Laplace evidence with quadratic loss.

In the greedy random walk algorithm, the step size used for each hyperparameter was adapted separately by measuring the acceptance rate for proposed changes in the parameter and scaling the step size up or down to keep the acceptance rate close to 50%. Thus a decreasing step size can be taken as one measure of how well the process has converged to an optimum. The search is terminated when either the step size has become very small, or the change to the criterion being optimized becomes very small. It was also found during experimentation that a useful addition to the basic algorithm was to enforce minimum and maximum values of the hyperparameters. Without such bounds the algorithm would occasionally get "stuck" in a plateau region of the model selection criterion where one or more hyperparameters were either very large or very small. Note that for the kernel hyperparameters steps in the random walk were taken in the natural logarithm of the hyperparameter values, as these scale parameters were expected to show a significant range of variation. Steps for C were taken in a linear scale, reflecting the smaller range of variation.

4.2 Estimating evidence gradients

We used Hybrid Monte Carlo (HMC, see *e.g.* [23]) to estimate the posterior averages required in the expressions (18) and (20) for the exact evidence gradients. The HMC algorithm is a standard technique from statistical physics that works by simulating a stochastic dynamics with a Hamiltonian "energy" defined by the target distribution plus a "momentum", or kinetic energy term. Denoting the momentum variables $\mathbf{p}$, the Hamiltonian we choose for our case is

$$\mathcal{H}(\boldsymbol{\theta}, \mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{K}\mathbf{p} + \frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\theta} + V(\boldsymbol{\theta}), \qquad V(\boldsymbol{\theta}) = C\sum_i l_p(y_i\theta_i) \tag{21}$$

and the corresponding "Boltzmann" distribution $P(\boldsymbol{\theta}, \mathbf{p}) \propto \exp[-\mathcal{H}(\boldsymbol{\theta}, \mathbf{p})] \propto \exp(\frac{1}{2}\mathbf{p}^{\mathrm{T}}\mathbf{K}\mathbf{p})Q(\boldsymbol{\theta}|D)$ factorizes over $\boldsymbol{\theta}$ and $\mathbf{p}$, so that samples from $Q(\boldsymbol{\theta}|D)$ can be obtained by sampling from $P(\boldsymbol{\theta}, \mathbf{p})$ and discarding the momenta $\mathbf{p}$. The $\mathbf{p}$ are nevertheless important for the algorithm, since they help to ensure a representative sampling of the posterior. An update step in the HMC algorithm consists of two parts. First, one updates a randomly chosen momentum variable p_i by Gibbs sampling according to the Gaussian distribution $\exp(-\frac{1}{2}\mathbf{p}^{\mathrm{T}}\mathbf{K}\mathbf{p})$; this will in general change the value of the Hamiltonian. Second, one changes both $\boldsymbol{\theta}$ and $\mathbf{p}$ by moving along a Hamiltonian trajectory for some specified "time" τ; the trajectory is determined by solving an appropriately discretized version of the differential equations

$$\frac{d\theta_i}{d\tau} \;=\; \frac{\partial\mathcal{H}}{\partial p_i} = (\mathbf{K}\mathbf{p})_i \tag{22}$$

$$\frac{dp_i}{d\tau} = -\frac{\partial \mathcal{H}}{\partial \theta_i} = -(\mathbf{K}^{-1}\boldsymbol{\theta})_i - \frac{\partial V(\boldsymbol{\theta})}{\partial \theta_i} \tag{23}$$

For an exact solution of these equations, $\mathcal{H}$ would remain constant; due to the discretization, small changes in $\mathcal{H}$ are possible and one accepts the update of $\boldsymbol{\theta}$ and $\mathbf{p}$ from the beginning to the end of the trajectory with the usual Metropolis acceptance rule. Iterating these steps the algorithm will, after some initial equilibration period, produce samples from $P(\boldsymbol{\theta}, \mathbf{p})$.

The occurrence of $\mathbf{K}^{-1}$ in (23) is inconvenient. We circumvent this by introducing $\tilde{\boldsymbol{\theta}} = \mathbf{K}^{-1}\boldsymbol{\theta}$; $\boldsymbol{\theta}$ is initialized to the SVM solution $\boldsymbol{\theta}^*$, since then the corresponding $\tilde{\boldsymbol{\theta}}$ is obtained trivially as $\tilde{\theta}_i = y_i\alpha_i$ without requiring matrix inversions. The Hamiltonian equations (22,23) simplify to

$$\begin{aligned} \frac{d\tilde{\theta}_i}{d\tau} &= p_i \\ \frac{dp_i}{d\tau} &= -\tilde{\theta}_i - \frac{\partial V(\boldsymbol{\theta})}{\partial \theta_i} \end{aligned}$$

and the simple form of the first equation is in fact what motivated our choice of the momentum-dependent part of H, eq. (21). The correspondence between $\tilde{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ is maintained by updating $\boldsymbol{\theta} = \mathbf{K}\tilde{\boldsymbol{\theta}}$ whenever $\tilde{\boldsymbol{\theta}}$ is changed. As a by-product, we automatically obtain samples of $\mathbf{K}^{-1}\boldsymbol{\theta}$ as required for (20).

Averages over the posterior distribution are taken by sampling after each trajectory step, repeating the procedure over some large number of steps. In practice usually the first half of the steps are discarded to allow for equilibration. We chose a total of 40,000 samples, giving 20,000 "production samples" with which to calculate the averages needed for the calculation of the gradients, eqs. (18) and (20).

4.3 Gradient ascent algorithm

The numerical values for the gradient of the evidence, estimated as explained above, were used in a simple gradient ascent algorithm to move the hyperparameters to a local maximum of the evidence. More powerful optimization techniques are not feasible in our case because neither the evidence itself, nor the Hessian of the evidence are available. The conjugate gradient method, for example, incorporates information about the Hessian and also usually employs a line search using the values of the function to be optimized. Approximations to Newton's method such as the Levenberg-Marquardt algorithm also use second derivative information. Fortunately, using the first derivatives of the evidence with respect to the hyperparameters leads to convergence to some maximum in a reasonable amount of time: typically between 40 and 80 steps of gradient ascent are required before the gradients have shrunk to small values.

For the experiments described the “learning rate” multiplier for the derivative of each parameter is adapted separately throughout the optimization. This is necessary as the gradients vary over several orders of magnitude during a typical simulation. In our case the adaptation of the“learning rate” of the optimization must be based on the change in the gradients only rather than on the change in the evidence itself. We expect gradients to increase only at the start of a simulation, but thereafter they should decrease as the parameters approach a maximum in the evidence. If the gradients do not decline quickly then the learning rate is increased, if the gradients increase sharply then the ascent step is discarded and the learning rate is decreased. For vector parameters (like the length scales in an RBF kernel) the change in gradient diretion can also be used for learning rate adaptation: sudden and large changes in the gradient suggest that the optimization may have passed a maximum and the step should be redone with a smaller learning rate. As in the experiments with zero temperature Monte Carlo search, gradient ascent steps for the kernel hyperparameters were actually taken in the logarithms of these parameters.

As noted above, the HMC simulation calculates averages over the posterior with only a relatively small amount of noise. Consequently, for a given set of starting hyperparameters an optimization based on gradient ascent in the evidence is practically deterministic. So in order to investigate the properties of local maxima in the evidence repeated trials were performed with the SVM hyperparameters initialized to random values. A few preliminary trials were used to choose reasonable orders of magnitude, and unless specified otherwise all results reported begin with uniform random initalization in the ranges $C \in [0.4, 0.8]$, $\ln l_a \in [-1, 2]$ for all length scales, $\ln k_0 \in [-1, 1]$ and $\ln k_{\rm off} \in [-2, -1]$.

4.4 Computational effort

We conclude this section with a brief discussion of the computational demands of the various model selection methods; though we stress once more that our focus was not on computational efficiency, so that faster algorithms can almost certainly be designed for all of the model selection criteria that we consider.

The computationally cheapest of the simple model selection criteria is ϵ_{gacv}, which can be evaluated in time $\mathcal{O}(n)$ from the properties of the trained SVM classifier. The span estimate ϵ_{span} requires the inversion of a matrix of size equal to the number of SVs; assuming that the number of SVs is some finite fraction of n for large n this gives a cost of $\mathcal{O}(n^3)$ for large n. The Laplace approximations to the evidence, for both linear and quadratic penalty SVMs, are dominated by the evaluation of determinants whose size is also the number of SVs (or, for linear penalty SVMs, the number of marginal SVs), giving again a scaling of approximately $\mathcal{O}(n^3)$.

Table 2 lists the running times for a single optimization step with each of the different methods. The evidence approximations and the test error approximations showed more or less similar running times, although the span estimate took somewhat longer on average. By far the greatest run time was needed for the gradient ascent on the evidence, due to the HMC sampling involved; a typical optimization run on a single processor HP V-Class took anywhere from 6 hours to 6 days. In comparison, most optimizations based on the simple model criteria were under an hour. The run time of the HMC algorithm should scale relatively benignly as $O(n^2)$ in the size of the training set, but our experiments show that the prefactor is large. The n^2 scaling comes mainly from the conversion from $\tilde{\boldsymbol{\theta}}$ to $\boldsymbol{\theta}$ via $\boldsymbol{\theta} = \mathbf{K}\tilde{\boldsymbol{\theta}}$ which is necessary during the solution of the Hamiltonian equations. (The length of the Hamiltonian trajectory, *i.e.* the time τ, does not need to be increased with n; the same is true for the number of samples required to obtain the posterior averages to a given accuracy.)

Note that the theoretical dependence of running time on training set size was not strictly followed in reality. One reason for this is that the average time per step presented includes time spent on discarded steps in the zero temperature Monte Carlo search algorithms. That is, the speed of the simple optimization techniques used here depends on the complexity of the search space.

As stated above, we were interested in the evidence gradient ascent algorithm mainly as a baseline for SVM model selection based on probabilistic criteria. Computational efficiency could however be increased in a number of ways; the Nyström method [25], for example, could significantly reduce the dimensionality (currently n) of the space over which the posterior needs to be sampled using HMC.

Data set	Inputs	Training set size	Test set size
Crabs	5	80	120
Pima	7	200	332
WDBC	30	300	269
Twonorm	20	300	7100
Ringnorm	20	300	7100

Table 1: Number of input dimensions, and sizes of training and test sets for the data sets used in our experiments.

5 Numerical results

5.1 Data sets

The model selection methods under consideration were applied to five two-class classification problems that are common in the machine learning literature. Three of these are from real world problems: the Pima Indian Diabetes data set, the Crabs data set and the Wisconsin Diagnostic Breast Cancer (WDBC) data set. The remaining two data sets, Twonorm and Ringnorm, are synthetic. The dimensionality of the inputs x and the size of the training and test sets for each data set are given in Table 1. All benchmark data sets are available through the UCI Machine Learning Repository (http://www1.ics.uci.edu/∼mlearn/MLRepository.html) and/or the DELVE archive (http://www.cs.toronto.edu/∼delve/). More detailed descriptions are also available on the web. Inputs were standardized so that across each complete data set all input components had zero mean and unit variance. For each data set the training and test sets were held constant for all experiments. The first n points in the data set were used for training and the remaining points were used for testing. The one exception is the Crabs data set, where the 6th attribute (color) was not used for classification and the remaining points were sampled to ensure an even distribution of the unused color attribute in the training and test sets. The number of training points, given in Table 1, was the same as that used in previous research.

Data Set	SVM	LE1	LE2	GACV	Span	Evid grad
Crabs	0.81	3	5	4	6	137
Pima	1.23	10	9	11	21	1805
WDBC	2.1	19	26	39	64	9352
Twonorm	2.5	35	26	27	82	7779
Ringnorm	3.7	58	71	68	216	10665

Table 2: Average CPU time (seconds) per optimization step. Times are given for: training of the SVM classifier (SVM); evaluation of the Laplace approximation to evidence for $p = 1$ and $p = 2$ (LE1, LE2); evaluation of ϵ_{gacv} and ϵ_{span} (GACV, Span); and evaluation of the evidence gradients (Evid grad)

5.2 Model selection using simple criteria

We discuss first the results obtained by optimizing the four simple model selection criteria: the Laplace evidence (LE), eq. (15), and the GACV (13) for linear penalty SVMs, and the Laplace evidence (17) and span error estimate (12) for quadratic penalty SVMs. The experiments with gradient ascent on the evidence, for linear penalty SVMs, are described separately in Sec. 5.3.

A typical example of selecting parameters for a linear penalty SVM by optimizing the Laplace approximation of the evidence is shown in Fig. 1. The data set for the experiment shown is the Twonorm data set. This example is chosen because it shows several typical features that appear in similar forms in all of the optimizations. To what extent optimizations for the other data sets match this example will be noted where appropriate.

The parameters all move more or less stochastically to stable final values as the evidence is optimized and it is clear that maximizing the Laplace Evidence correlates to reducing the error on a test set. Both the Laplace evidence and the GACV are shown alongside the test error, although only the Laplace evidence is used for optimization. Maximizing the evidence generally reduces the GACV, although this correspondence is not strict. Similar behaviour is observed for optimization with the Laplace evidence for the quadratic penalty SVM, and for optimization of the GACV and the span estimate. For quadratic penalty SVMs, the Laplace evidence and the span estimate also have the same qualitative correlation as the Laplace evidence

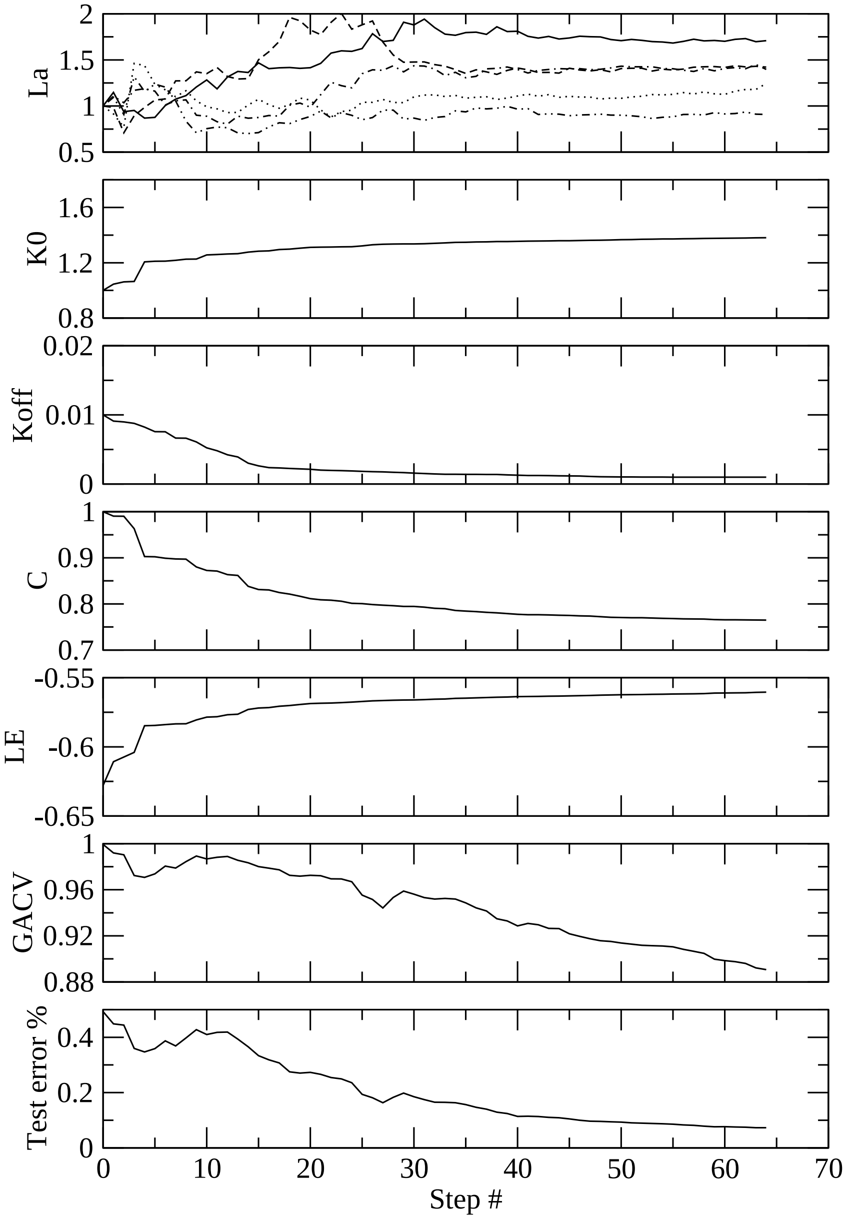

Figure 1: Hyperparameter tuning for the Twonorm data set by optimizing the Laplace approximation to the evidence for linear penalty SVMs. The top four graphs show the evolution of the hyperparameters; 5 out of the 20 length scale parameters are shown. Below, the Laplace evidence (LE) is shown; the GACV error estimate and the test error are also displayed, to demonstrate the correlation with the Laplace evidence.

and the GACV for the linear penalty case.

An important issue in all of these methods is the existence of many local optima in the model selection criteria. Starting from the same initialization, the hyperparameters converged to significantly different values in repeated trials. We verified explicitly, *e.g.* by evaluating the chosen model selection criterion along a line in hyperparameter space connecting different end points of two optimization runs, that the different local optima found were genuine and not artefacts due to incomplete convergence of the optimization algorithms. The search criteria always deteriorated in between the points found by the search, confirming that the latter were in fact local optima.

To analyse the characteristics of the local optima, 25 repeated trials were performed on all data sets for the simple optimization criteria. Comparison of the final SVM hyperparameter values at the local optima showed that they were highly variable. For all methods and all data sets the variance of the final parameter values was always of the same order of magnitude as the average value of the final parameters. Tuning of the length scales is often interpreted as "relevance determination" for the different dimensions of the data because a large length scale indicates that the classification does not vary significantly with changes in that parameter. The results here however indicate that the relevance of each dimension probably depends in a complicated way on the relevance assigned to the other dimensions, and that different assignments of the length scales can yield similar results.

In addition to variance in the final SVM hyperparameter values, the test error also showed significant trial to trial variation. For all methods, many of the trials result in a final test error that is close to the best achieved by any method, but for some methods a large portion of the trials end in a test error that is significantly worse. The average and standard deviation of the test errors achieved with the different methods are shown in Table 3. Table 4 shows the best test errors achieved on any trial for each method and data set. For reference, Table 5 shows the test errors achieved on the same data sets by comparable methods in previous research. Unfortunately, these previous studies do not always include error bars for test error results so it is hard to compare the results for the averages and standard deviations of the error.

To illustrate the variability in the final error resulting from each optimization method histograms of the errors achieved in all trials are shown for the Twonorm, Pima and WDBC data sets in Figs. 2, 3, and 4 respectively. These plots show the difficulty of picking a "best" method from among the simple model selection criteria. For the Twonorm data set all of the methods pro-

	LE1	LE2	GACV	Span	Evid grad	ES
Crabs	10.7 ± 2.1	10.5 ± 1.3	13.0 ± 1.8	6.0 ± 1.8	9.2 ± 1.5	5.5 ± 1.3
Pima	30.3 ± 2.0	33.5 ± 2.2	23.2 ± 2.7	21.0 ± 1.1	20.8 ± 1.5	19.7 ± 1.5
WDBC	5.8 ± 2.5	5.8 ± 3.6	9.6 ± 2.8	7.8 ± 4.2	4.0 ± 1.2	2.4 ± 1.0
Twonorm	13.5 ± 12.6	12.6 ± 5.0	5.2 ± 1.9	4.6 ± 0.9	4.0 ± 0.2	3.7 ± 0.4
Ringnorm	4.7 ± 1.6	2.5 ± 5.3	3.3 ± 1.3	3.5 ± 1.3	3.2 ± 0.6	3.2 ± 0.6

Table 3: Test error ϵ for all data sets (in %), written in the form "mean ± standard deviation". Statistics for the simple model selction criteria are taken over 25 trials. For gradient ascent in the evidence averages are over 25 trials for the Crabs data set, and over 10 trials for all other data sets. Abbreviations for the model selection criteria are as in Table 2, except for the last column (ES = gradient ascent with "early stopping"; see Sec. 5.3.2).

duce test errors that are around the best for any method in previous research, but with the evidence approximation for linear penalty SVMs around a third of the trials end in errors that are signficantly greater. For the Pima data set, all of the simple methods are inferior to the best methods in previous research, and both of the evidence approximations perform worse than the error estimates; while on the WDBC data set the evidence approximations are superior to the error estimates.

Comparing Tables 3, 4, and 5, it is clear that the high variability in the results achieved by optimizing the four "simple" model selection criteria is undesirable; while the best trials for each method and data set are approximately the same as the best results reported in previous research, the average performance over trials is rather disappointing. One possible productive use of the high variability of classifiers produced by convergence to local optima of the model selection criteria could be to combine the resulting classifiers in some ensemble or voting scheme. Such approaches normally benefit precisely from high variability among the classifiers being combined, so this could be an interesting subject for future research.

5.3 Model selection using evidence gradients

Figs. 5 and 6 show a typical run of evidence gradient ascent on the Twonorm data set. Fig. 5 shows the tuning of a subset of the RBF kernel length scales

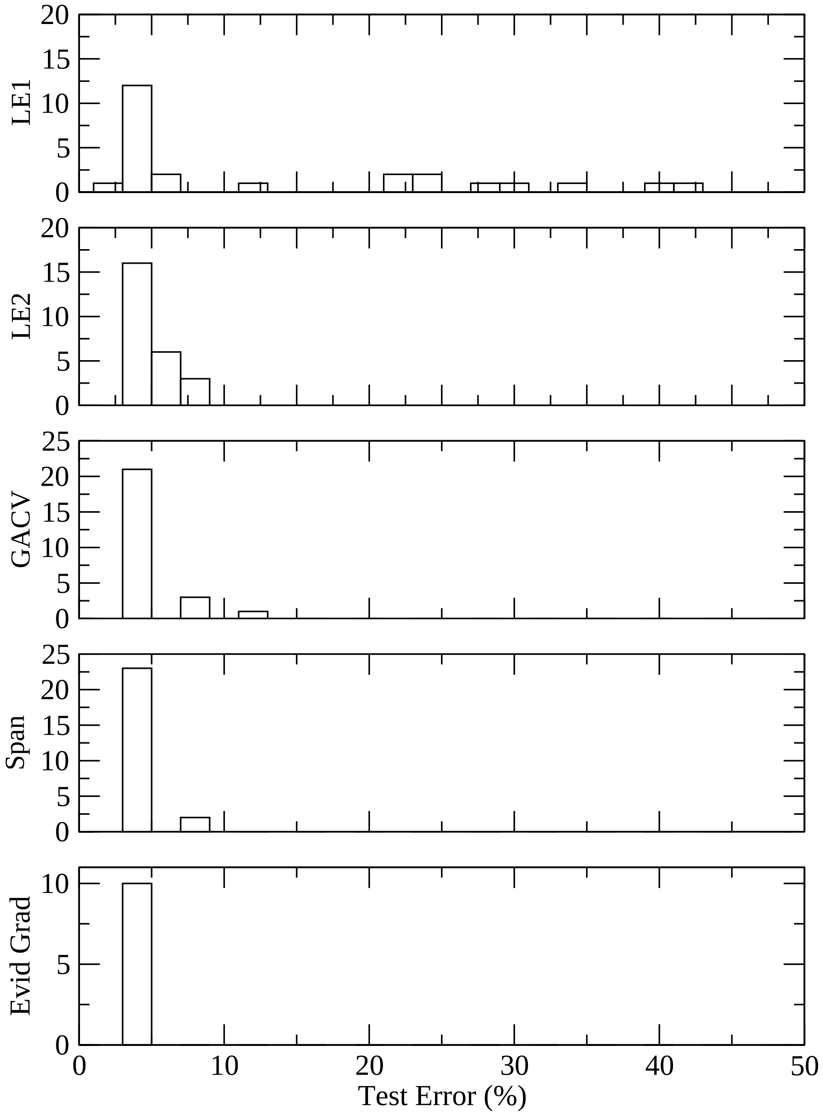

Figure 2: Histogram of test errors (in %) achieved on the Twonorm data set. Shown are the results of 25 trials with the simple model selection criteria, and 10 trials of evidence gradient ascent. The bin size for the histogram is 2%.

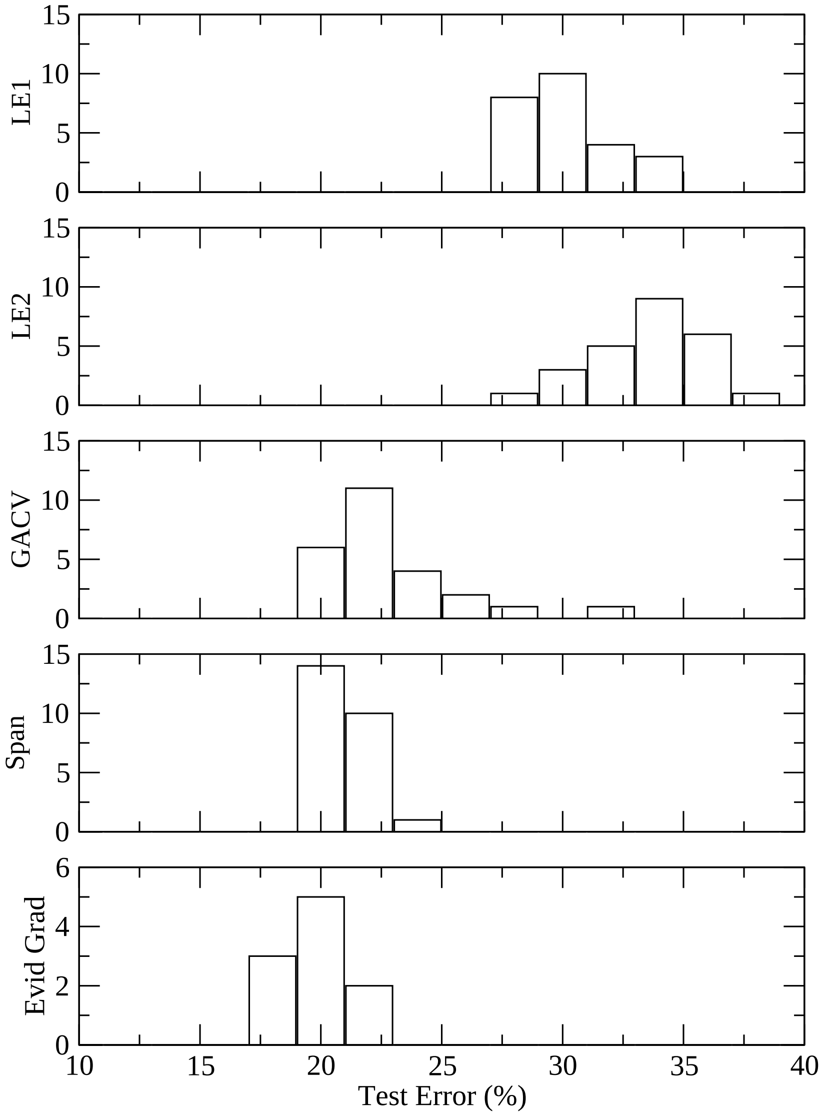

Figure 3: Histogram of test errors (in %) achieved on the Pima data set. Shown are the results of 25 trials with the simple model selection criteria, and 10 trials of evidence gradient ascent. The bin size for the histogram is 2%.

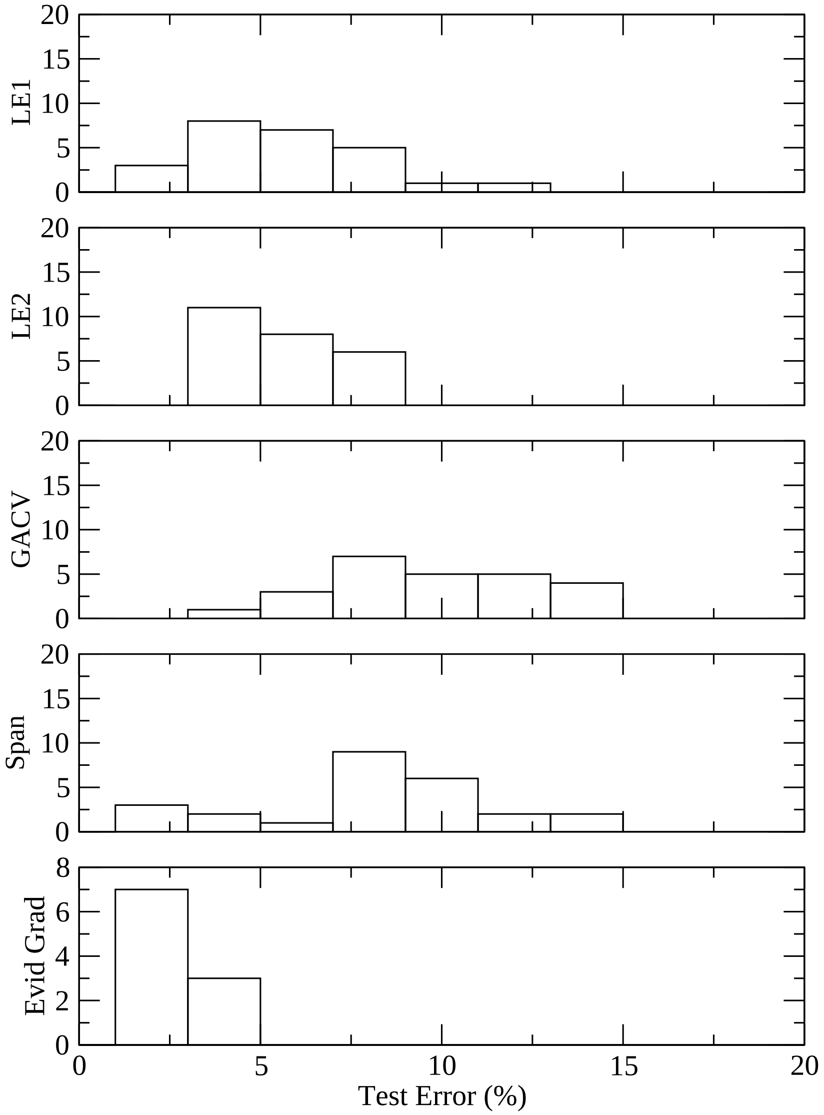

Figure 4: Histogram of test errors (in %) achieved on the WDBC data set. Shown are the results of 25 trials with the simple model selection criteria, and 10 trials of evidence gradient ascent. The bin size for the histogram is 2%.

	LE1	LE2	GACV	Span	Evid grad	ES
Crabs	5.9	9.2	10.9	3.4	5.0	3.4
Pima	27.8	28.4	20.2	19.0	19.3	18.4
WDBC	1.9	3.4	4.1	1.9	1.5	1.2
Ringnorm	2.1	2.2	1.9	2.0	2.5	2.5
Twonorm	2.9	3.1	3.4	3.5	3.4	3.0

Table 4: Best single trial test error ϵ (in %). Abbreviations for the model selection criteria are as in Table 3.

and Fig. 6 shows the tuning of kernel amplitude k_0, the kernel offset k_{off} and the penalty parameter C. (Although statistics for the performance of the evidence gradient method were determined by initialization to random parameter values, for the specific sample shown we started all length scales with identical parameters.) Both the gradients of the evidence with respect to each parameter and the parameter values themselves are shown. The gradients typically start at small values, rise to a peak and then decline. Most parameter ultimately arrive at a constant value with small gradients, indicating that the evidence is at a local maximum with respect to that parameter. The optimization is terminated when the gradients have reached a small fraction of their peak magnitude. During this process the error on the test set decreases significantly.

As with the simple model selection criteria analysed in the previous section (Laplace approximations to the evidence and error approximations), repeated trials of gradient ascent in the evidence showed the existence of many local maxima in the evidence at widely varying parameter values. Due to the long run time required for the HMC sampling used to calculate the evidence gradients only 10 trials were performed for each benchmark data set, with the exception of the Crabs data set where 25 trials were performed. (See Section 4.4 for a discussion of the running time of the algorithm on the various data sets.) As explained in Sec. 4.3, the gradient ascent algorithm is essentially deterministic once the initial hyperparameter values have been fixed. Consequently, repeated trials were started from random initial values for the hyperparameters in order to investigate the existence and variability of local maxima in the evidence.

Tables 3 and 4 above list the resulting test errors obtained with gradient ascent optimization of the evidence, along with results obtained from the

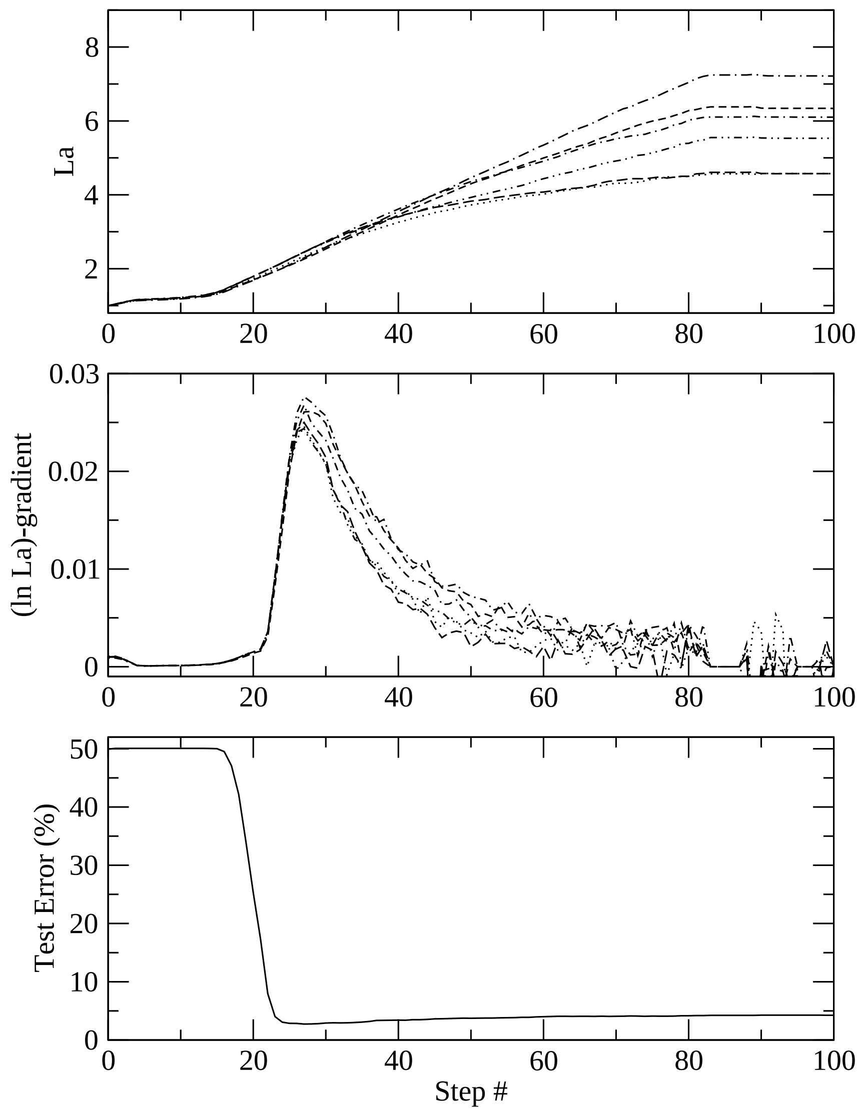

Figure 5: Tuning the length scales l_a on the Twonorm data set, using gradient ascent on the evidence. 6 out of 20 length scale parameters are shown, along with the corresponding gradients; the bottom plot shows the evolution of the test error.

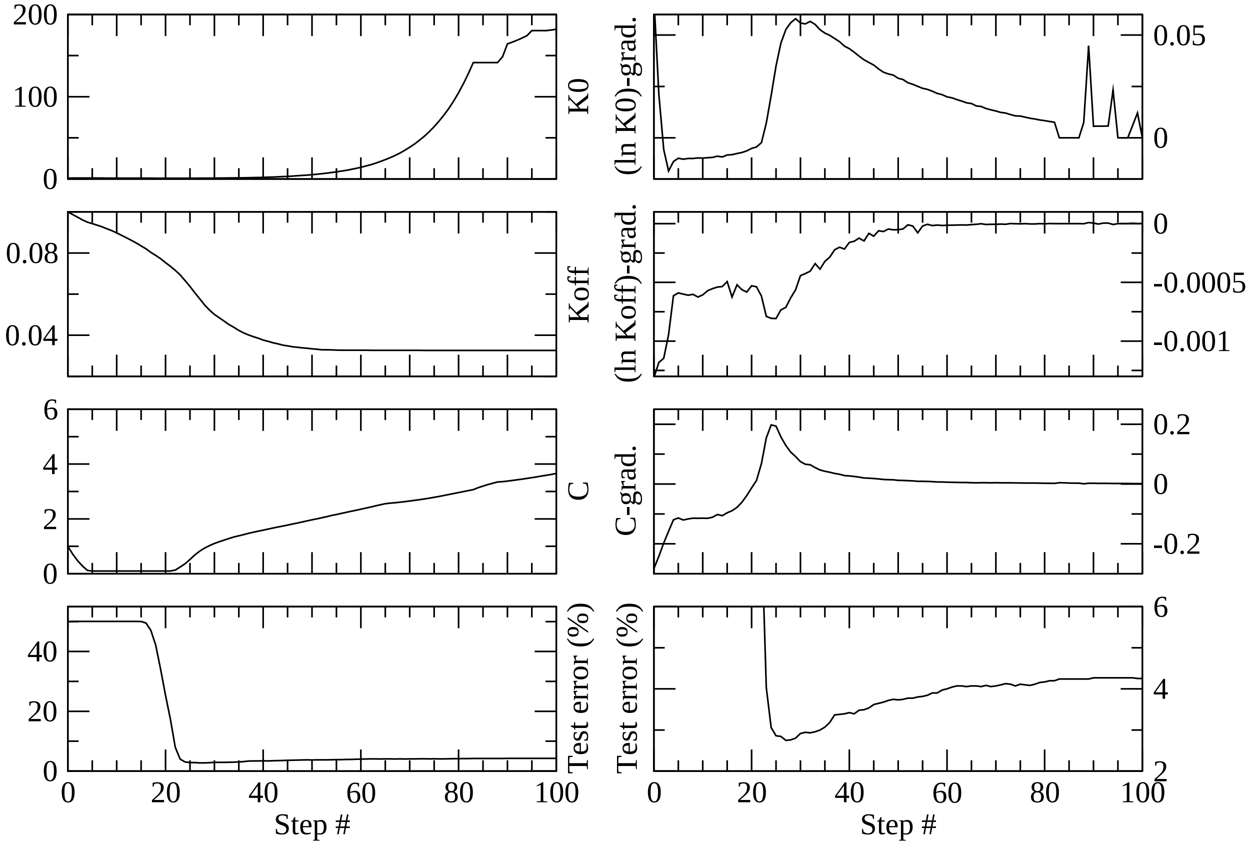

Figure 6: Tuning k_0, k_{off} and C on the Twonorm data set, using gradient ascent on the evidence. The gradients for each parameter are shown alongside the actual parameter values. The two bottom panels show the evolution of the test error, with the right one being a zoom on the range of small error values; see discussion in Sec. 5.3.2.

Data set	GP Var	SVM Var	SVM CV	GP Lap	GP MF
Crabs	2.5	3.3	3.3	3.3	1.7
Pima	19.9	20.5	20.2	20.2	19.0
WDBC	3.7	3.7	3.3	3.3	2.6
Twonorm	3.2	3.7	2.3	4.0	—
Ringnorm	1.7	1.9	2.3	3.0	—

Table 5: Test errors ϵ (in %) found on the benchmark data sets in previous work. The methods used were as follows. GP Var: Gaussian process classifier (see *e.g.* [26, 27]), with hyperparameters determined by maximizing a variational approximation to the evidence [7]. SVM Var: SVM with hyperparameters selected by the same variational method [7]. SVM CV: SVM, with all length scales $l_a = l$ set equal and l and k_0 determined by ten-fold cross validation [7]; the offset was unrestricted, *i.e.* effectively $k_{\rm off} \to \infty$. GP Lap: Gaussian process classifier, hyperparameters determined by maximizing a Laplace approximation to the evidence [7]; GP MF: Gaussian process classifier trained by a mean field method [9].

simpler methods discussed earlier. Comparing with the results found in previous studies (Table 5), one sees that gradient ascent on the evidence for SVMs with radial basis function kernels achieves approximately the same test error as the best methods that have been previously applied. For some data sets the best performance obtained by evidence gradient ascent is superior to the performance previously reported. An interesting point of comparison is with SVM model selection by optimization of a variational approximation of the evidence, as described in [7]. (This comparison is somewhat tentative because of the small number of trials for our evidence gradient ascent method, combined with the lack of information about trial-to-trial variation in [7].) Still, it is worth noting that although the method desribed here uses gradients of the evidence without further approximation, and also tunes the C parameter (which is effectively fixed to unity in the approach of [7]), it does not seem to achieve systematically superior performance than the variational approximation.

Figs. 2, 3 and 4 above contain the histograms of the test error produced by SVM model selection by evidence gradient ascent, and the comparison with the simple model selection criteria, for the Twonorm, Pima and WDBC data sets. Although strong conclusions cannot be drawn due to the small number of trials, gradient ascent in the evidence seems to produce significantly better performance on all of the data sets than any of the other methods. In all tests the distribution of resulting errors is both closer to the best results found in previous studies, and less variable. The most likely explanation for the superior performance of the gradient ascent method is the fact that it actually maximizes an exact, unapproximated model selection criterion (the evidence), while the simple model selection criteria (Laplace evidence and error estimates) are all to some extent approximate. The poorly performing local optima of these simple criteria may then arise from errors introduced by the approximations.

We comment briefly on the actual values of the hyperparameters found by gradient ascent on the evidence, in particular for the kernel amplitude k_0 and the offset k_{off}. For the Twonorm data set, the maximum in the evidence occurs at a relatively large value of k_0, with an average of $k_0 \approx 180$ across trials. (Large values of k_0 were also found with model selection with the approximate evidence, but were much less common when using the error estimates.) This may appear surprising. However, it should be born in mind that from the probabilistic view the prior variance of the latent function θ is $\langle \theta^2(x) \rangle = K(x, x) = k_0 + k_{\text{off}}$. The typical prior scale for $\theta(x)$ is therefore $\sqrt{k_0}$ (since k_{off} is small, see below), which equates to around 13 for $k_0 \approx 180$; this is not unreasonably large compared to the scale of 1 set by the SVM margin. Similar final values of k_0 were obtained for the Crabs and WDBC data sets, while for Pima and Ringnorm k_0 was rather smaller. Previous experiments with simple synthetic data sets [3] suggest that an evidence maximum at large k_0 correlates with small apparent levels of noise in the data set; we have not attempted to verify this correlation for our five benchmark data sets.

The offset hyperparameter k_{off} was typically tuned to very small values by evidence gradient ascent (*e.g.* around 0.03 for the Twonorm data set). This provides a posteriori justification for our approach of including the offset parameter b from the conventional SVM framework into the kernel.

5.3.1 Noise in evidence gradients

In the final portions of the optimization shown in Figs. 5 and 6 it can be observed that there is significant noise in the gradients as the evidence approaches a maximum. This arises from statistical fluctuations in the HMC sampling, which come to dominate when the true gradient values are small. Although the noise could be decreased by increasing the length of the HMC runs, that did not seem to be necessary for the cases considered here: because of the learning rate adaptation the learning rate is quite small by the time the evidence is close to its maximum and the noise in the gradients has little effect on the final results.

In the Twonorm example it can also be seen that when the parameters are nearly at a maximum in the evidence the gradients with respect to the kernel parameters are calculated as zero in some steps. This effect occurred typically for larger values of C. Regions of $\boldsymbol{\theta}$-space where the potential $V(\boldsymbol{\theta})$ in the Hamiltonian (21) is zero, *i.e.* where all $y_i\theta_i \geq 1$, are then much more probable then regions where $y_i\theta_i < 1$ for some i. It is then possible that the HMC sampling only returns samples from the region with $V(\boldsymbol{\theta}) = 0$, where $l'_p(y_i\theta_i) = 0$ for all i so that (20) gives an estimate of zero for all gradients with respect to kernel parameters. Experiments showed that scaling the trajectory length in the HMC runs proportionally to $\frac{1}{C}$ for large C could avoid this effect. The rationale is that the shorter trajectories makes the HMC sampling more likely to sample values of $\boldsymbol{\theta}$ which are just outside the boundary of the $V(\boldsymbol{\theta}) = 0$ region; these still have appreciable posterior probability but do give nonzero values for some of the $l'_p(y_i\theta_i)$. We did not explore this issue in detail, however.

5.3.2 "Overfitting" by evidence maximization

Close inspection of progress of the test error in Figs. 5 and 6 shows an interesting aspect of tuning SVM hyperparameters using the evidence. While the overall evolution of the test error shows a large decline as gradient ascent on the evidence progresses, a closer look at the region of small error values (see the lower right plot of Fig. 6) shows that the test error goes through a shallow minimum before a small rise to its final value. Not all data sets show such a clean example of this behaviour as the Twonorm data set, but all except Pima did exhibit the phenomenon to some degree.

One possible explanation for the observed test error minimum is the fact that we are not using the evidence of a properly normalized probabilty model (see Sec. 2). An alternative interpretation, which seems to us more likely, is that we are observing here a kind of overfitting. This takes place not on the level of the "network" parameters ($\mathbf{w}$ or $\theta(x)$) as in conventional overfitting – which is due to a lack of regularization – but on the level of the hyperparameters: If we imagine sampling a number of data sets of size n from a given true distribution, then the evidence as a function of the hyperparameters, and hence the position of its maximum, will depend on the particular data set. Only for large n would the evidence become independent of the data set (and related to the Kullback-Liebler divergence, or cross-entropy, between the true distribution over data sets and the one predicted by the inference model; see *e.g.* [3]). For finite n, maximization of the evidence for a specific data set is therefore not expected to lead to strict minimization of the error on an independent test set.

This interpretation leads naturally to the idea of using an early stopping mechanism when optimizing the evidence, where the gradient ascent is abandoned when performance on an independent validation set ceases to improve. Note that this is not the same as simply returning to hyperparameter tuning by cross-validation; in fact, a grid search using cross-validation error over the large number of hyperparameters in our examples (C, k_0, k_{off} and the length scales l_a associated with each of the d input dimensions) would be essentially impossible (see also [22]). To gauge the possible benefits of such an approach, we have included in Table 3 above both the final test error when the optimization is run until the gradients are small, and the minimal value of the test error during the gradient ascent. True early stopping with an independent validation set would be expected to yield a performance in between these two values; the results in Table 3 suggest that this could be useful for some data sets.

6 Conclusion

In this paper we have investigated the issue of model selection for SVM classifiers. We have restricted ourselves to model selection in the sense of tuning the parameters of an RBF kernel and the penalty parameter C, though the general approaches described could also be used for choosing between different functional forms of the kernel.

We reviewed briefly the probabilistic view of SVMs, and extended our previous work on Laplace approximations to the evidence to the case of SVMs with quadratic slack penalties. Exact expressions for the gradients of the evidence in terms of posterior averages were also derived, and we described how these averages can be estimated numerically using Hybrid Monte Carlo techniques and used in a model selection algorithm which performs gradient ascent on the exact (unapproximated) evidence.

In our numerical experiments on five benchmark data sets, we compared optimization of four "simple" model criteria with the evidence gradient descent. Two of the simple criteria were estimates of test error: the generalized approximate cross-validation error (GACV) for SVMs with linear slack penalties, and the span error estimate for SVMs with quadratic penalties. The two other criteria were derived from probabilistic concepts; these were the Laplace approximations to the evidence for the linear and quadratic penalty cases. Our main result is that all the simple model criteria exhibit multiple local optima with respect to the hyperparameters. While some of the resulting "locally optimal" SVM classifiers give test performance that is competitive with results from other approaches in the literature, a significant fraction lead to rather higher test errors. The results for the evidence gradient ascent method show that also the exact evidence exhibits local optima. But these give much less variable test errors, which are also typically lower test errors than for the simpler model selection criteria. In this sense, "you get what you pay for": the computationally rather more expensive evidence gradient ascent approach gives better and more consistent performance than the cheaper model selection criteria.

There are a number of directions for possible future work. First, our results strongly suggest that the hunt is still on for a model selection criteria for SVM classification which is both simple and gives consistent generalization performance. Alternatively, one could try to cope with the existence of local maxima in the simple model selection criteria by testing the selected models on a validation set and performing repeated optimizations until satisfactory performance is found. A more interesting approach might be to try to exploit the large variability in the locally optimal classifiers by using them in some scheme for combining classifiers. Finally, if evidence gradient ascent turned out in more comprehensive tests to be the model selection method of choice, it would be worth investigating possible speed-ups of the algorithm. We already hinted at the Nyström method [25] above, but one could also explore running the model selection only on randomly sampled subsets of data, and then possibly combining the resulting classifiers appropriately.

Acknowledgements

Access to the Hewlett-Packard V2500 was provided by the Caltech Center for Advanced Computing Research (http://www.cacr.caltech.edu) through the National Partnership for Advanced Computational Infrastructure–A Distributed Laboratory for Computational Science and Engineering, supported by the NSF cooperative agreement ACI-9619020. We also thank Andrew Buchan for assistance with the early stages of the numerical experiments for quadratic penalty SVMs.

References

[1] C J C Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[2] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

[3] P Sollich. Bayesian methods for Support Vector Machines: Evidence and predictive class probabilities. *Machine Learning*, 46:21–52, 2002.

[4] P Sollich. Probabilistic interpretation and Bayesian methods for Support Vector Machines. In *ICANN99 – Ninth International Conference on Artificial Neural Networks*, pages 91–96, London, 1999. The Institution of Electrical Engineers.

[5] P Sollich. Probabilistic methods for Support Vector Machines. In S A Solla, T K Leen, and K-R Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 349–355, Cambridge, MA, 2000. MIT Press.

[6] Michael E. Tipping. The relevance vector machine. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 652–658, Cambridge, MA, 2000. MIT Press.

[7] M Seeger. Bayesian model selection for Support Vector Machines, Gaussian processes and other kernel classifiers. In S A Solla, T K Leen, and K R Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 603–609, Cambridge, MA, 2000. MIT Press.

[8] M Opper and O Winther. Gaussian process classification and SVM: Mean field results and leave-one-out estimator. In A J Smola, P Bartlett, B Schölkopf, and D Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 43–65, Cambridge, MA, 2000. MIT Press.

[9] M Opper and O Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Comput.*, 12(11):2655–2684, 2000.

[10] G Wahba. Support Vector Machines, reproducing kernel Hilbert spaces and the randomized GACV. In B Schölkopf, C Burges, and A J Smola, editors, *Advances in Kernel Methods: Support Vector Machines*, pages 69–88. MIT Press, Cambridge, MA, 1998.

[11] T Jaakkola and D Haussler. Probabilistic kernel regression models. In David Heckerman and Joe Whittaker, editors, *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, San Francisco, CA, 1999. Morgan Kaufmann.

[12] A J Smola, B Schölkopf, and K R Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, 1998.

[13] C K I Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M I Jordan, editor, *Learning and Inference in Graphical Models*, pages 599–621. Kluwer Academic, 1998.

[14] J T Y Kwok. Moderating the outputs of Support Vector Machine classifiers. *IEEE Trans. Neural Netw.*, 10(5):1018–1031, 1999.

[15] J T Y Kwok. The evidence framework applied to Support Vector Machines. *IEEE Trans. Neural Netw.*, 11(5):1162–1173, 2000.

[16] R M Neal. *Bayesian learning for neural networks.* Springer, New York, 1996.

[17] V Vapnik. *The nature of statistical learning theory.* Springer, New York, 1995.

[18] V Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

[19] N Cristianini, C Campbell, and J Shawe-Taylor. Dynamically adapting kernels in Support Vector Machines. In M Kearns, S A Solla, and D Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 204–210, Cambridge, MA, 1999. MIT Press.

[20] O Chapelle and V N Vapnik. Model selection for Support Vector Machines. In S A Solla, T K Leen, and K-R Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 230–236, Cambridge, MA, 2000. MIT Press.

[21] V Vapnik and O Chapelle. Bounds on error expectation for Support Vector Machines. *Neural Comput.*, 12(9):2013–2036, 2000.

[22] O Chapelle, V Vapnik, O Bousquet, and S Mukherjee. Choosing multiple parameters for Support Vector Machines. *Mach. Learn.*, 46(1-3):131–159, 2002.

[23] R M Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[24] W Krauth. Introduction to Monte Carlo algorithms. In J Kertesz and I Kondor, editors, *Advances in Computer Simulation*. Springer, 1998.

[25] C K I Williams and M Seeger. Using the Nyström method to speed up kernel machines. In T K Leen, T G Dietterich, and V Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[26] D Barber and C K I Williams. Gaussian processes for Bayesian classification via hybrid Monte Carlo. In M C Mozer, M I Jordan, and T Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 340–346, Cambridge, MA, 1997. MIT Press.

[27] C K I Williams and D Barber. Bayesian classification with Gaussian processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(12):1342–1351, 1998.