Maximum Likelihood Model Selection for 1-Norm Soft Margin SVMs with Multiple Parameters

Tobias Glasmachers, Christian Igel, *Senior Member, IEEE*

Abstract—Adapting the hyperparameters of support vector machines (SVMs) is a challenging model selection problem, especially when flexible kernels are to be adapted and data are scarce. We present a coherent framework for regularized model selection of 1-norm soft margin SVMs for binary classification. It is proposed to use gradient-ascent on a likelihood function of the hyperparameters. The likelihood function is based on logistic regression for robustly estimating the class conditional probabilities and can be computed efficiently. Overfitting is an important issue in SVM model selection and can be addressed in our framework by incorporating suitable prior distributions over the hyperparameters. We show empirically that gradient-based optimization of the likelihood function is able to adapt multiple kernel parameters and leads to better models than four concurrent state-of-the-art methods.

Index Terms—support vector machines, model selection, regularization, maximum likelihood

1 Introduction

Support vector machines (SVMs, [1], [2]) are successful pattern recognition algorithms well embedded in statistical learning theory [3]. Still, in practice their application is not straightforward because they require the user to specify a kernel function, corresponding to a metric on the data, and a regularization parameter. It is well known that the performance of the resulting machine depends crucially on this choice, which poses the model selection problem for SVMs.

In simple cases, it is possible to adapt the hyperparameters manually by trial-and-error or based on coarse grid search using a cross-validation estimate of the generalization error. However, this standard proceeding does not scale to high-dimensional parameter spaces associated with classes of flexible candidate kernels. High flexibility in the choice of the kernel function allows the metric in the feature space to be tailored to the pattern recognition problem at hand and, therefore, promises better performance. In practice, limiting the space of potential kernel functions and non-expert ad-hoc decisions during model selection can reduce the success or even the applicability of SVMs.

Therefore, various algorithms for automatic SVM model selection have been suggested. General techniques include canonical hold-out-set based methods such as cross-validation [4]. Furthermore, specific methods for SVMs have been proposed [3], [5]–[12].

In this paper we consider the model selection problem for the widely used standard 1-norm soft margin SVM formulation for binary classification (the attribute 1-norm refers to the penalized norm of slack variables, corresponding to the hinge loss, not to the norm of the weight vector as, e.g., in [13]). We are interested in methods capable of selecting both the kernel parameters and the complexity control parameter of the SVM quickly and robustly. The focus on the standard case of the most widely used 1-norm SVM (in contrast to the SVM variant with 2-norm slack penalty term) narrows down the choice of the model selection method, because some methods proposed in the literature are only applicable to 2-norm SVMs.

We argue that the log-likelihood estimate introduced by Platt [14] is a natural objective function for model selection. Platt proposed logistic regression for estimating the class conditional probabilities. This method is a heuristic, but has become popular because it is simple, computationally cheap, and gives surprisingly good results (despite principal limitations, see [15]–[17]). We propose the resulting log-likelihood function as an optimization criterion for model selection and show that efficient gradient-based methods are applicable to its maximization.

Overfitting is a severe problem, in particular when adapting highly flexible kernel functions based on rather small datasets. Arguably the most elaborate and flexible ways to address the problem of overfitting in SVM model selection are Bayesian approaches [18]–[21]. Especially when the training data is limited, it is desirable to incorporate prior knowledge into the selection process. In a Bayesian framework such knowledge can naturally be included in the form of prior distributions. It is straight-forward to embed the log-likelihood objective function into this probabilistic framework for SVM model selection.

It has to be stressed that the goal of our work is to derive a robust model selection strategy for 1-norm SVMs, which have proven to yield sparse classifiers that generalize well. They are frequently used across a broad range of application domains, and our goal is

T. Glasmachers is with the Dalle Molle Institute for Artificial Intelligence (IDSIA), 6928 Manno-Lugano, Switzerland, and C. Igel is with the Institut für Neuroinformatik at the Ruhr-Universität Bochum, 44780 Bochum, Germany. tobias@idsia.ch, christian.igel@neuroinformatik.rub.de

to provide robust automatic model selection algorithms for these application scenarios. Support vector machines are a frequentist approach to pattern recognition. We do not want to change our learning machine for the reasons just mentioned, but nevertheless want to profit from a Bayesian framework for model selection if expert knowledge in terms of a hyperparameter prior is available. We show how to achieve these goals by using a likelihood function as an objective function for SVM model selection.

The paper is organized as follows. We first describe 1-norm SVMs for binary classification to fix our notation. Then we review state-of-the-art techniques for automatic model selection in section 3. In section 4 we turn to the log-likelihood for gradient-based model selection, possibly augmented by a hyperparameter-prior. An extensive experimental evaluation is presented in section 5. In the supplementary material, we provide software for validating the results and for applying our approach to new model selection problems. The paper ends with a discussion and conclusions.

2 Support Vector Machines

Support vector machines are state-of-the-art in machine learning for pattern recognition, in particular for binary classification. We consider an input space X and the output space $Y = \{+1, -1\}$. Learning is driven by sample data $S = \{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$ with (x_i, y_i) for $1 \le i \le \ell$ drawn independently from some fixed unknown distribution p over $X \times Y$. The goal of binary classification is to infer from S a hypothesis $h : X \to Y$ minimizing the risk $R_p(h) = \int\limits_{X \times Y} L(y, h(x))\, \mathrm{d}p(x, y)$ corresponding to the generalization error. We consider the 0-1-loss given by $L(y, h(x)) = (-h(x)y + 1)/2$ such that the risk becomes the expected classification error.

Support vector machines transfer the input data to a feature space and perform linear classification in that space. Given a positive semi-definite kernel function $k : X \times X \to \mathbb{R}$, we consider the feature space $\mathcal{H}_k = \mathrm{span}\{k(x, \cdot) \,|\, x \in X\}$ and the function class $\mathcal{H}_k^b = \{f = g + b \,|\, g \in \mathcal{H}_k, b \in \mathbb{R}\}$. We classify according to the sign of a decision function $f \in \mathcal{H}_k^b$. The decision boundary induced by f is a hyperplane in $\mathcal{H}_k$. The decision function generated by a 1-Norm Soft Margin SVM corresponds to the solution of

$$\underset{f \in \mathcal{H}_k^b}{\text{minimize}}\ \frac{1}{\ell} \sum_{i=1}^{\ell} L_{\text{hinge}}(y_i, f(x_i)) + \frac{\gamma_\ell}{2} \|f\|_k^2 \ ,$$

where $\gamma_\ell = (\ell C)^{-1}$ and the (semi-)norm $\| \cdot \|_k$ is inherited from $\mathcal{H}_k$ to $\mathcal{H}_k^b$. The loss function is given by $L_{\text{hinge}}(y, f(x)) = \max\{0, 1 - y f(x)\}$. The parameter $C > 0$ controls the trade-off between the optimization goals of reducing the empirical loss measured by L_{hinge} and the complexity of the hypothesis measured by $\|.\|_k^2$.

The SVM decison function f takes the form $f(x) = \sum_{i=1}^{\ell} \alpha_i k(x, x_i) + b$ with $\alpha \in \mathbb{R}^\ell$ and $b \in \mathbb{R}$. We solve the SVM optimization problem using the algorithm proposed in [22], which is implemented in the Shark machine learning library [23]. Examples x_i with corresponding coefficient $\alpha_i \neq 0$ are called support vectors. We talk about bounded support vectors if the coefficient is at its bound $y_i \cdot C$.

The regularization parameter C and the parameters θ of a family k_θ of candidate kernel functions are called hyperparameters. Their proper selection is the model selection problem for SVMs. To stress the dependency of the decision function f on the hyperparameters and the training data we denote the function resulting from training on S with regularization parameter C and kernel k by $f_{C,k,S} \in \mathcal{H}_k^b$.

3 SVM Model Selection

A number of techniques have been proposed for the automatic selection of hyperparameters [3], [6], [8]–[12]. These range from general-purpose techniques such as hold out set-based methods to specialized approaches that exploit properties of the SVM margin to bound the leave-one-out error. Most approaches propose an objective function, typically an error measure or bound, and an algorithm for its minimization.

Cross-validation (CV) is a standard technique, with the leave-one-out (LOO) error as a special case. Its sole parameter, the number of data folds, has undergone numberous investigations. Values of five or ten turn out to be a good compromise between low variance (few folds) and low bias (many folds) [4]. Hold out set-based error measures such as CV count the number of misclassified patterns in the validation sets and are therefore discrete-valued and non-differentiable. Simple grid search is the standard method for their minimization in low-dimensional parameter spaces. Alternatively, evolutionary direct search can be applied [24], [25].

However, in high-dimensional search spaces one would like to base search on direction information as provided by the gradient of the objective function. Therefore it has been proposed to smoothen the 0-1-loss with a sigmoid approximation [11].

Several other differentiable objective functions have been proposed for model selection of 2-norm SVMs (see, e.g., [8] and `http://olivier.chapelle.cc/ams/`), in which the hinge loss in the SVM optimization problem is replaced with the squared hinge loss. In this special case the regularization parameter can be treated like a kernel parameter [8]. The most important examples are the radius-margin bound [3], [9] and the span-bound [6], [8], which are both derived as upper bounds of the LOO error. We see some problems with these error measures. At least for small datasets, the frequent argument that the LOO error is nearly unbiased is not convincing due to its high variance. Second, actively minimizing error bounds violates their statistical prerequisites (i.e., the objective function values cannot be interpreted as bounds anymore). Third, the radius-margin quotient can not directly be used as a criterion for standard 1-norm soft-margin SVMs, because the radius does not depend on C. Adding an appropriate term depending on C does not give satisfactory results [26], [27]. While there is a variant of the span-bound for 1-norm SVMs, it provides only a loose upper bound by treating all bounded support vectors as errors.

4 Regularized SVM Model Selection

For the integration of SVM learning into a Bayesian framework it is common practice to start with a probabilistic interpretation of a term of the generic form $F + \lambda R$, where F depends on the empirical loss, R is a regularization term, and λ is a tradeoff parameter. These terms can occur either at the first level of inference [20], for example with $F = \frac{1}{\ell}\sum_{i=1}^{\ell} L_{\text{hinge}}(y_i, f(x_i))$, $R = \frac{1}{2}\|f\|_k^2$ and $\lambda = \gamma_\ell$, or at the second level of inference [21], namely model selection, for example with F being a model selection criterion, R the negative logarithm of a hyperparameter prior, and λ a normalization constant. In both cases, the Bayesian framework is rooted in the interpretation of $F + \lambda R$ as the negative logarithm of a *posterior* probability

$$\begin{aligned} F + \lambda R + \text{const} &= -\log(\text{likelihood} \cdot \text{prior}) \\ &= -\log(\text{posterior}) \ . \end{aligned}$$

Then minimizing $F + \lambda R$ corresponds to a maximum a posteriori (MAP) estimate of the parameters. These approaches combine the frequentist-motivated SVM with a Bayesian interpretation in order to take advantage of priors, for example for the selection of hyperparameters.

It is possible to choose priors on parameters in order to integrate these parameters out. This truly Bayesian proceeding effectively replaces parameters by priors, and the hope is that the selection of a prior is more intuitive and more robust than the direct selection of a parameter value. The drawback of this method is that the integration is usually computationally intractable and one must resort to approximations. It is tempting to choose (e.g., conjugate) priors such that the integration can be computed efficiently. However, we argue that in general the selection of arbitrary priors just to simplify computations is not reasonable.

Although the existing strategies often give good results, we want to remark that it is in most cases unnatural to interpret $\exp(-F)$ (suitably normalized, if possible) as a likelihood function of the SVM model parameters. Such approaches re-motivate quantities as being probabilities that have no probabilistic interpretation. We find it more natural to directly introduce a likelihood function, such as the one proposed by Platt [14], without re-interpreting some term of interest in an unnatural way.

4.1 Platt's Likelihood Estimate

Given a trained SVM, the class conditional probabilities can be estimated using logistic regression. Let us first consider a validation dataset $\tilde{S} = \big\{(\tilde{x}_1, \tilde{y}_1), \dots, (\tilde{x}_{\tilde{\ell}}, \tilde{y}_{\tilde{\ell}})\big\}$ independent of the training dataset S (later we will use a cross-validation procedure). Platt [14] has shown that the conditional probability $P(\tilde{y} = +1 \,|\, \tilde{f}_{C,k,S}(\tilde{x}))$ of positive label given the SVM prediction can be well estimated with a simple sigmoidal function $\sigma : \mathbb{R} \to [0, 1]$ squashing the decision function. The parameters of this function are determined directly by maximizing the log-likelihood

$$\begin{aligned} \mathcal{L}(\tilde{S}, \sigma, f_{C,k,S}) = &\sum_{\substack{(\tilde{x},\tilde{y})\in\tilde{S} \\ \tilde{y}=+1}} \log \sigma\big(f_{C,k,S}(\tilde{x})\big) \\ &+ \sum_{\substack{(\tilde{x},\tilde{y})\in\tilde{S} \\ \tilde{y}=-1}} \log\Big(1 - \sigma\big(f_{C,k,S}(\tilde{x})\big)\Big) \ , \end{aligned}$$

see also [28].

The sigmoid takes the form $\sigma_{(r,s)}(t) = 1/(1 + \exp(s \cdot t + r))$. In the original work of Platt [14] a uniform prior is used to regularize the selection of the sigmoid parameters r and s. For efficiency we drop the regularization. As proposed in [14] we use gradient-based optimization of a cross-validation based objective function to determine the sigmoid parameters and to compute the resulting log-likelihood. Now we define a probabilistic output SVM as a standard 1-norm SVM combined with a sigmoid $\sigma_{(r,s)}$.

The hyperparameters of a probabilistic output SVM are the kernel parameters, the complexity control parameter C, and the sigmoid parameters (r, s). For selecting these parameters based on a training dataset S, we propose to maximize the N-fold cross-validation log-likelihood $\bar{\mathcal{L}} = \sum_{n=1}^{N} \mathcal{L}(S_n, \sigma_{(r,s)}, f_{C,k,S\setminus S_n})$, where $S = S_1 \dot\cup \dots \dot\cup S_N$ is a partition of the dataset S into N disjoint subsets S_n of roughly equal size. Here, $f_{C,k,S\setminus S_n}$ is the SVM decision function obtained on the training data $S \setminus S_n$. Then $\mathcal{L}(S_n, \sigma_{(r,s)}, f_{C,k,S\setminus S_n})$ denotes the log-likelihood of the model $\sigma_{(r,s)} \circ f_{C,k,S\setminus S_n} : X \to [0, 1]$ given the hold-out set S_n.

The log-likelihood is differentiable w.r.t. the SVM hyperparameters whenever the SVM hypotheses $f_{C,k,S\setminus S_n}$ are. This is the case everywhere except for a low-dimensional zero set in which one of the sets of support vectors or bounded support vectors changes. In these points the dependency is still continuous, such that gradient-based optimization makes sense. A highly efficient way to compute the derivatives of $f_{C,k,S\setminus S_n}(x)$ w.r.t. C and k – often even faster than machine training itself – is provided by Keerthi et al. in [11]. Similar ideas have been presented in [8] for 2-norm SVMs.

If expert knowledge is available in form of a prior $\pi(C, \theta)$ over the hyperparameters then we add $\log(\pi(C, \theta))$ to the log-likelihood in order to obtain a maximum-*a-posteriori* (MAP) parameter estimate. The resulting posterior can be maximized based on gradients if the prior is differentiable, which is a weak assumption in practice.

4.2 Relation to Previous Work

Our likelihood computation is based on the work by Platt [14]. The only hint in the literature towards using the log-likelihood for kernel selection is found in [29], where the derivative of the probability $P(y = +1 \,|\, x)$ is computed in the context of SVM model selection. However, to the best of our knowledge, there has never been a thorough investigation of this arguably natural objective function, and in particular we are not aware of any experimental study. It has also never been noticed that this objective function can be combined with a prior distribution in order to encode priori knowledge into the model selection process.

Bartlett and Tewari [17] have shown that no consistent estimator of the form $\Pr(y = +1 \,|\, f(x))$ can be constructed based on the 1-norm SVM decision function f. However, these asymptotic arguments are not relevant for the case of small datasets. We never observed the asymptotic effects discussed in [17] in practice.

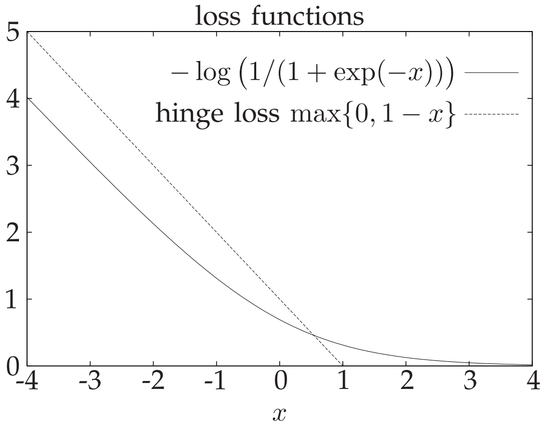

Fig. 1. Loss functions involved in the computation of the log-likelihood model selection criterion. The logarithm of the sigmoid resembles the hinge loss underlying SVM training.

Considering the technical aspects, the optimization of the likelihood in our approach fits into the general model selection framework presented in [11], where a sigmoid is used for smoothing the 0-1-loss. However, conceptually our approach is considerably different. The sigmoidal model is not used as a smooth approximation of some non-differentiable loss function, but in contrast takes the important role of a probabilistic model. The resulting model selection objective function $\bar{\mathcal{L}}$ is motivated by a probabilistic interpretation, and it is not an arbitrary approximation of the classification error. The asymptotic behavior of the corresponding loss resembles the hinge loss, which fits well into the 1-norm SVM framework, see Figure 1. The conceptual difference is best explained in the terms of soft and hard classification according to Wahba [30]: Coming from the hard classification framework considered in [11], we switch to a soft classification formulation in order to apply Bayesian methods. On the theoretical side this probabilistic interpretation is the key for the proper integration of a hyperparameter prior into the model selection process, and on the practical side our experiments show that maximization of the likelihood gives superior results.

Our method is inspired by and related to the approach proposed by Cawley and Talbot [21] for least squares SVMs and kernel logistic regression [31]. The conceptual similarity is the direct application of a hyperparameter prior to a model selection criterion, without involving the SVM training procedure. In [21], Cawley and Talbot use a simple squared loss (which naturally fits into the framework of least squares SVMs) to construct a model selection criterion, which is then *interpreted* as a negative log-likelihood. By contrast, we start with a meaningful likelihood term and propose it as a model selection criterion, to which Bayesian regularization can be applied naturally.

5 Experiments

The goal of our experiments is to assess the performance of the log-likelihood as an objective functions for SVM model selection. We experimentally compare it to four concurrent objective functions introduced in section 3 on a large collection of datasets. We use flexible Gaussian automatic relevance detection (ARD) kernels of the form $k(x, x') = \exp\Big(-\sum_{i=1}^{d} \gamma_i (x_i - x_i')^2\Big)$. All datasets in this study are relatively small, such that model selection for SVMs with Gaussian ARD kernel with a total of $(d + 1)$ parameters is challenging, where d is the dimension of the input space.

We consider two discrete-valued objective functions, namely 5-fold cross-validation and the leave-one-out error. Usually, simple grid search is conducted on non-differentiable objective functions. Because grid search is infeasible in high-dimensional search spaces, we employ the highly efficient elitist version [32] of the covariance matrix adaptation evolution strategy (CMA-ES) [33] for hyperparameter search. We refer to the corresponding model selection strategies as `cross-validation` and `leave-one-out`.

The span-bound (for efficiency, we use the approximate version of the span-bound [8]), the smoothed cross-validation error [11], and the negative log-likelihood form a second group of objective functions. They are differentiable w.r.t. the kernel and the regularization parameter.[1] We use the iRprop$^+$ algorithm [34] for their optimization. We refer to the corresponding model selection strategies as `span bound`, `smoothed CV`, and `likelihood`, respectively.

All experiments were carried out on the $(d{+}1)$-dimensional search space spanned by $(\log(C), \log(\gamma_1), \ldots, \log(\gamma_d))$. We start the optimization from the point $(0, -\log(2d), \ldots, -\log(2d))$ with an initial

[1] Strictly speaking, these functions are differentiable outside a zero set where the SVM hypothesis is not differentiable. Note that the span-bound is not continuous in the exceptional zero set, such that its gradient can be misleading (see Figure 2 in [8]).

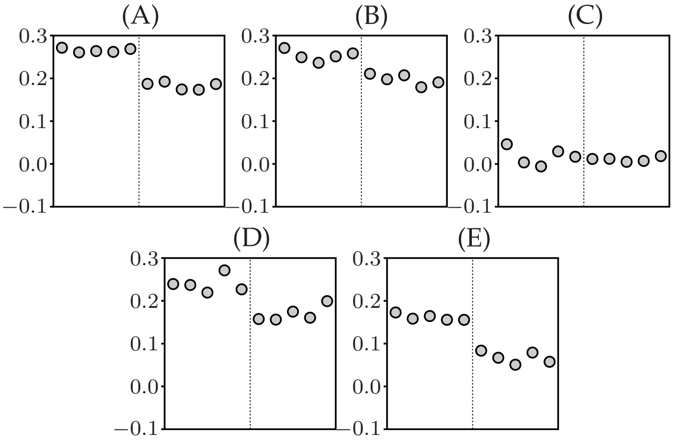

Fig. 2. Parameters γ_i of the ARD kernel (on log scale) found by (A) `cross-validation`, (B) `leave-one-out`, (C) `span-bound`, (D) `smoothed CV`, and (E) `likelihood`, averaged over 100 data partitions. Only the first five components are equally task relevant.

step size of $1/10$. This initial point corresponds to the educated guess of a radial kernel with width

$$\sigma = \sqrt{1/(2\gamma)} = \sqrt{1 \Bigg/ \left(2 \cdot \sum_{i=1}^{d} \gamma_i\right)} = \sqrt{d} \ ,$$

which is a reasonable configuration if all coordinates are normalized to unit variance. The optimization loop is stopped as soon as there is no progress made for 20 iterations for the discrete objective functions, or if the gradient step size falls below 10^{-3}, or in the worst case if the total number of iterations exceeds 200. These criteria ensure that the model selection problem is solved with reasonable accuracy, because the generalization performance depends only on the first few (one or two) digits of the hyperparameters.

Let us start with an artificial toy problem to demonstrate the effects of the different model selection strategies. Samples $(x, y) \in \mathbb{R}^{10} \times \{\pm 1\}$ are drawn as follows: First we fix a label y with equal probability, then we set $x_i = y/2 + z_i$ for $i \in \{1, \dots, 5\}$ and $x_i = z_i$ for $i \in \{6, \dots, 10\}$, where the $z_i \sim \mathcal{N}(0, 1)$ are standard normally distributed and independent. Thus, all coordinates are noisy, and only the first five coordinates carry task relevant information. We drew 10000 examples which were split into 100 random partitions of 500 training and 9500 test examples. The resulting performance of all five methods is reported in Table 1, and the corresponding kernel parameters are depicted in Figure 2.

Our artificial toy dataset is clearly tailored towards the ARD kernel, but by no means towards the `likelihood` method. Nevertheless, the `likelihood` strategy high significantly outperforms its competitors. Figure 2 reveals that this method does the best job in adapting the kernel parameters, maybe on par with `cross-validation`, which selects slightly larger values on average. Interestingly, the gradient of the `span bound` is not predicting the global trend well, such that this method is unable to capture the relative importance of the different components in this task.

We applied the same experimental procedure to 27 benchmark datasets. 13 of these datasets have been introduced in [35] as a benchmark collection. They are pre-partitioned into 100 different training and test datasets (only 20 for image and splice). We obtained 14 more datasets from the UCI machine learning repository [36] and processed them as similarly as possible. All nominal features have been encoded symmetrically,[2] and all features have been normalized to zero mean and unit variance. In case of multi-class classification, minority classes have been merged to form binary classification problems. For each partitioning of a data set, an independent model selection was performed. The pre-processed datasets, including all partitioning information, are included in the supplementary material. The results of the five model selection methods are summarized in Tables 1 and 2.

Note that the results achieved on the different partitions are not strictly independent, because the same data are used to form training and test sets for the different partitions. We use the paired Wilcoxon rank sum test to compare the results achieved by different methods *as if the performances on the different partitions were statistically independent*. Thus, the significance levels reported most probably need adjustment, but the test results can nevertheless be used to judge the differences in performance. To stay on the safe side we used relatively strict significance levels of 0.01 and 0.001.

The results show that on 15 of the problems the `likelihood` strategy performs best. It is outperformed by `cross-validation`, `leave-one-out`, and/or `smoothed-CV` in 5 cases. For 8 datasets there is no clear winner.

The performance of the `span bound` is poor due to its very local gradient information that does not reflect the global trend. In control experiments we directly minimized the span-bound with CMA-ES. Because the span upper bounds the LOO error, it is not surprising that the resulting performance is usually close to `leave-one-out`, but slightly worse. In total, the `smoothed-CV` strategy performed surprisingly poorly compared to `likelihood` (significanty worse on 15 bechmarks, better on 2).

6 Discussion

In our experience, many model selection criteria proposed in the literature are not very robust. They work on selected test problems, but often fail on others. We therefore decided *a priori* to evaluate our algorithms on the complete benchmark suite considered in [35], and

[2] We represent two possible nominal feature values with a single binary variable. In case of more than two nominal values per feature we resort to one binary variable per value. Data points with missing values have been removed.

	25% quantile	50% quantile	75% quantile	CV error	LOO error	span-bound	smoothed-CV	likelihood	25% quantile	50% quantile	75% quantile	CV error	LOO error	span-bound	smoothed-CV	likelihood
	artificial toy problem								**balance-scale**							
cross-validation	0.147	0.150	0.154	–		<<		>>	0.031	0.041	0.049	–		<<		>>
leave-one-out	0.145	0.149	0.154		–	<<	<	>>	0.029	0.038	0.044		–	<<		
span-bound	0.146	0.157	0.173	>>	>>	–	>>	>>	0.107	0.177	0.266	>>	>>	–	>>	>>
smoothed-CV	0.147	0.152	0.157		>	<<	–	>>	0.031	0.040	0.052			<<	–	>>
likelihood	0.140	0.143	0.147	<<	<<	<<	<<	–	0.024	0.034	0.044	<<		<<	<<	–
	banana								**cylinder bands**							
cross-validation	0.104	0.107	0.111	–		<<	<<	>>	0.253	0.280	0.317	–		<<		>>
leave-one-out	0.105	0.108	0.112		–	<<	<<	>>	0.256	0.280	0.329		–	<<	>	>>
span-bound	0.115	0.157	0.390	>>	>>	–	>>	>>	0.314	0.366	0.405	>>	>>	–	>>	>>
smoothed-CV	0.109	0.115	0.125	>>	>>	<<	–	>>	0.241	0.268	0.305		<	<<	–	
likelihood	0.101	0.106	0.109	<<	<<	<<	<<	–	0.220	0.256	0.293	<<	<<	<<		–
	breast cancer								**car evaluation**							
cross-validation	0.234	0.273	0.299	–		<<			0.071	0.087	0.099	–		<<		>>
leave-one-out	0.234	0.260	0.299		–	<<		<	0.074	0.086	0.095		–	<<		>>
span-bound	0.273	0.312	0.351	>>	>>	–	>>	>>	0.235	0.258	0.289	>>	>>	–	>>	>>
smoothed-CV	0.244	0.273	0.299			<<	–		0.056	0.073	0.104			<<	–	>>
likelihood	0.247	0.273	0.312		>	<<		–	0.041	0.049	0.056	<<	<<	<<	<<	–
	connect-4								**diabetis**							
cross-validation	0.185	0.188	0.196	–		<<	<<	>>	0.230	0.240	0.257	–		<<		
leave-one-out	0.184	0.187	0.195		–	<<	<<	>>	0.230	0.242	0.257		–	<<		>
span-bound	0.272	0.272	0.272	>>	>>	–	>>	>>	0.247	0.265	0.287	>>	>>	–	>>	>>
smoothed-CV	0.199	0.215	0.269	>>	>>	<<	–	>>	0.229	0.243	0.250			<<	–	
likelihood	0.169	0.171	0.178	<<	<<	<<	<<	–	0.226	0.237	0.253		<	<<		–
	diagnosis								**flare solar**							
cross-validation	0.000	0.000	0.000	–		<<			0.323	0.333	0.343	–		<<	<	<<
leave-one-out	0.000	0.000	0.000		–	<<			0.324	0.335	0.348		–	<<		<<
span-bound	0.000	0.067	0.200	>>	>>	–	>>	>>	0.324	0.345	0.438	>>	>>	–		
smoothed-CV	0.000	0.000	0.000			<<	–		0.325	0.335	0.355	>			–	<<
likelihood	0.000	0.000	0.000			<<		–	0.337	0.354	0.370	>>	>>		>>	–
	german								**heart**							
cross-validation	0.232	0.247	0.260	–		<<			0.150	0.180	0.210	–		<<		
leave-one-out	0.236	0.248	0.263		–	<<			0.160	0.180	0.200		–	<<	<	
span-bound	0.312	0.340	0.360	>>	>>	–	>>	>>	0.210	0.245	0.280	>>	>>	–	>>	>>
smoothed-CV	0.230	0.250	0.264			<<	–		0.160	0.200	0.210		>	<<	–	
likelihood	0.230	0.245	0.264			<<		–	0.160	0.190	0.210			<<		–
	image								**ionosphere**							
cross-validation	0.025	0.028	0.029	–		<<	<	>	0.059	0.065	0.075	–		<<	<<	<<
leave-one-out	0.026	0.030	0.036		–	<<		>>	0.060	0.065	0.075		–	<<	<<	<<
span-bound	0.101	0.150	0.247	>>	>>	–	>>	>>	0.239	0.350	0.366	>>	>>	–	>>	>>
smoothed-CV	0.031	0.035	0.046	>		<<	–	>>	0.065	0.083	0.101	>>	>>	<<	–	
likelihood	0.020	0.023	0.025	<	<<	<<	<<	–	0.070	0.085	0.096	>>	>>	<<		–
	king rook vs. king								**king rook vs. king pawn**							
cross-validation	0.164	0.168	0.174	–		<<		>>	0.049	0.060	0.066	–		<<	>	>>
leave-one-out	0.162	0.170	0.176		–	<<		>>	0.050	0.058	0.065		–	<<	>	>>
span-bound	0.177	0.182	0.192	>>	>>	–	>>	>>	0.240	0.287	0.361	>>	>>	–	>>	>>
smoothed-CV	0.167	0.174	0.178			<<	–	>>	0.040	0.051	0.062	<	<	<<	–	>>
likelihood	0.143	0.145	0.154	<<	<<	<<	<<	–	0.024	0.032	0.038	<<	<<	<<	<<	–
	magic gamma telescope								**mammographic mass**							
cross-validation	0.155	0.160	0.166	–		<<		>>	0.174	0.183	0.204	–		<<		>>
leave-one-out	0.155	0.159	0.165		–	<<		>>	0.173	0.181	0.201		–	<<	<	
span-bound	0.355	0.355	0.356	>>	>>	–	>>	>>	0.178	0.198	0.230	>>	>>	–	>>	>>
smoothed-CV	0.154	0.163	0.170			<<	–	>>	0.179	0.186	0.203		>	<<	–	>>
likelihood	0.148	0.151	0.154	<<	<<	<<	<<	–	0.170	0.179	0.192	<<		<<	<<	–
	ringnorm								**sonar (mines vs. rocks)**							
cross-validation	0.020	0.022	0.026	–		<<	<<	<<	0.176	0.213	0.241	–		<<	<<	>>
leave-one-out	0.020	0.022	0.024		–	<<	<<	<<	0.185	0.213	0.241		–	<<	<<	>>
span-bound	0.060	0.097	0.197	>>	>>	–	>>	>>	0.313	0.407	0.472	>>	>>	–	>>	>>
smoothed-CV	0.032	0.036	0.040	>>	>>	<<	–	>>	0.213	0.245	0.278	>>	>>	<<	–	>>
likelihood	0.027	0.029	0.033	>>	>>	<<	<<	–	0.167	0.194	0.213	<<	<<	<<	<<	–

TABLE 1
Absolute and relative performance of the different model selection strategies on the first 20 datasets. The 25%, 50%, and 75% quantiles of the test errors over the fixed partitions are reported. The symbols $<$, and $<<$ are used to indicate that the method in this row performs significantly better than the method in this column with significance levels 0.01 and 0.001 (paired Wilcoxon rank sum text), respectively, while the symbols $>$, and $>>$ indicate that the method in the corresponding row performs significantly worse.

	25% quantile	50% quantile	75% quantile	CV error	LOO error	span-bound	smoothed-CV	likelihood	25% quantile	50% quantile	75% quantile	CV error	LOO error	span-bound	smoothed-CV	likelihood
	spambase								**splice**							
cross-validation	0.094	0.098	0.107	–		<<	<<	>>	0.099	0.106	0.121	–		<<	>>	>>
leave-one-out	0.093	0.098	0.104		–	<<	<<	>>	0.099	0.106	0.115		–	<<	>>	>>
span-bound	0.132	0.141	0.156	>>	>>	–	>>	>>	0.248	0.306	0.338	>>	>>	–	>>	>>
smoothed-CV	0.102	0.111	0.118	>>	>>	<<	–	>>	0.055	0.058	0.068	<<	<<	<<	–	
likelihood	0.087	0.090	0.097	<<	<<	<<	<<	–	0.054	0.055	0.063	<<	<<	<<		–
	thyroid								**tic-tac-toe**							
cross-validation	0.040	0.053	0.067	–				<<	0.013	0.014	0.022	–	<<	<<	>>	>>
leave-one-out	0.037	0.053	0.067		–			<<	0.016	0.027	0.039	>>	–	<<	>>	>>
span-bound	0.040	0.053	0.080			–	>>		0.332	0.342	0.353	>>	>>	–	>>	>>
smoothed-CV	0.027	0.040	0.067			<<	–	<<	0.010	0.014	0.018	<<	<<	<<	–	>>
likelihood	0.040	0.060	0.080	>>	>>		>>	–	0.007	0.011	0.014	<<	<<	<<	<<	–
	titanic								**blood transfusion**							
cross-validation	0.224	0.226	0.230	–					0.226	0.235	0.243	–		<<		>
leave-one-out	0.224	0.228	0.231		–				0.225	0.233	0.242		–	<<		
span-bound	0.224	0.227	0.230			–			0.230	0.240	0.273	>>	>>	–	>>	>>
smoothed-CV	0.223	0.227	0.230				–		0.225	0.233	0.240			<<	–	
likelihood	0.223	0.227	0.230					–	0.223	0.233	0.240	<		<<		–
	twonorm								**waveform**							
cross-validation	0.027	0.031	0.037	–		<<	<<	<<	0.103	0.109	0.117	–		<<	<<	
leave-one-out	0.026	0.027	0.036		–	<<	<<	<<	0.103	0.109	0.115		–	<<	<<	
span-bound	0.116	0.189	0.248	>>	>>	–	>>	>>	0.157	0.178	0.200	>>	>>	–	>>	>>
smoothed-CV	0.034	0.036	0.041	>>	>>	<<	–		0.109	0.113	0.119	>>	>>	<<	–	>>
likelihood	0.034	0.037	0.040	>>	>>	<<		–	0.106	0.109	0.114			<<	<<	–

TABLE 2

Absolute and relative performance of the different model selection strategies on the remaining 8 datasets. For details refer to the caption of Table 1.

later added lots of datasets from the UCI repository to broaden the experimental basis. On these problems, gradient-based maximization of the approximated likelihood function has proven to be an efficient and highly robust method for adapting multiple kernel parameters, clearly outperforming other available methods.

Adding a prior and turning the maximum likelihood into a maximum a posteriori approach can further improve the performance of the model selection process and particularly its robustness. Of course, any performance gain depends on the quality of the prior, which amounts to the quality of available expert knowledge. Therefore, it is difficult to assess its potential experimentally on a large benchmark suite.

The good results achieved by the likelihood strategy are not self-evident, because the maximization of the log-likelihood objective function does not directly minimize the 0-1-loss used for testing. This is in contrast to the cross-validation error, its smoothed variant, the leave-one-out error, and the span bound (which upper bounds the LOO error). In the following, we discuss some hypotheses that may explain our empirical results.

The discrete-valued nature of CV errors leads to objective functions with many plateaus. Thus, cross-validation and leave-one-out do not provide sufficient direction information for search in high-dimensional parameter spaces. These objective functions are even difficult to optimize for the CMA-ES, which in principle can cope well with plateaus. However, optimizing cross-validation and leave-one-out with the CMA-ES can give good results whenever the starting point of the optimization is already sufficiently close to the optimum (e.g., whenever a radial kernel would be suitable for the problem at hand, which is often the case). Thus, both the differentiable structure and the ability to provide a trend by taking gradual real values are clear advantages of the log-likelihood over hold out set-based error measures in high-dimensional search spaces.

It was already found in [8] that gradient-based optimization of the span-bound is endangered to be misleading. This is because the gradient represents highly local information that does not extend beyond the many hyper-surfaces that split the search space into components where the span bound is continuous and differentiable. Fixes that require setting additional nursing parameters have been tried, but these solutions are not fully convincing.

The comparison of likelihood to the technically very similar smoothed-CV strategy is especially interesting. The significantly better success of the likelihood strategy is hard to understand, because smoothed CV is much closer to optimizing the 0-1-loss. Figure 1 provides a possible explanation of this phenomenon: Maximizing the log-likelihood corresponds to a loss function closely related to SVM training.

7 Summary and Conclusions

We propose a simple and coherent framework for support vector machine model selection. It is designed for 1-norm soft margin SVMs. Its core is maximization of a differentiable estimate of the likelihood function of the model parameters. The computations are based on an established approximation of class conditional probabilities. The likelihood can be combined with meaningful priors for robust maximum a posteriori inference of hyperparameters. In contrast to methods that approximate the classification error with a sigmoidal function just in order to obtain a smooth objective function, our approach has a natural probabilistic interpretation. The experimental results clearly indicate the benefits of the new model selection procedure for multiple hyperparameters and small datasets, where robust model selection techniques are of utmost importance, and the tools provided in the supplementary material allow our approach to be applied to new datasets.

Acknowledgments

CI acknowledges support from the German Federal Ministry of Education and Research within the National Network Computational Neuroscience under grant number 01GQ0951 (Bernstein Fokus "Learning behavioral models: From human experiment to technical assistance").

References

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory (COLT 1992)*. ACM, 1992, pp. 144–152.

[2] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

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

[4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

[5] T. Jaakkola, M. Diekhaus, and D. Haussler, "Using the fisher kernel method to detect remote protein homologies," *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pp. 149–158, 1999.

[6] V. Vapnik and O. Chapelle, "Bounds on Error Expectation for Support Vector Machines," *Neural Computation*, vol. 12, pp. 2013–2036, 2000.

[7] N. Cristianini, A. Elisseeff, J. Shawe-Taylor, and J. Kandola, "On Kernel-Target Alignment," in *Neural Information Processing Systems*. MIT Press, 2001, pp. 367–373.

[8] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131–159, 2002.

[9] S. S. Keerthi, "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1225–1229, 2002.

[10] T. Glasmachers and C. Igel, "Gradient-based adaptation of general Gaussian kernels," *Neural Computation*, vol. 17, no. 10, pp. 2099–2105, 2005.

[11] S. S. Keerthi, V. Sindhwani, and O. Chapelle, "An Efficient Method for Gradient-Based Adaptation of Hyperparameters in SVM Models," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, 2007.

[12] C. Igel, T. Glasmachers, B. Mersch, N. Pfeifer, and P. Meinicke, "Gradient-based Optimization of Kernel-Target Alignment for Sequence Kernels Applied to Bacterial Gene Start Detection," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 216–226, 2007.

[13] P. S. Bradley and O. L. Mangasarian, "Feature Selection via Concave Minimization and Support Vector Machines," in *International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 1998, pp. 82–90.

[14] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in Large Margin Classifiers*, pp. 61–74, 1999.

[15] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[16] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *The Annals of Statistics*, vol. 32, no. 1, pp. 56–85, 2004.

[17] P. L. Bartlett and A. Tewari, "Sparseness vs Estimating Conditional Probabilities: Some Asymptotic Results," *Journal of Machine Learning Research*, vol. 8, pp. 775–790, 2007.

[18] M. Opper and O. Winther, "Gaussian Process Classification and SVM: Mean Field Results," in *Large Margin Classifiers*, P. Bartlett, B. Schölkopf, D. Schuurmans, and A. Smola, Eds. MIT Press, 1999.

[19] M. Seeger, "Bayesian Model Selection for Support Vector Machines, Gaussian Processes and Other Kernel Classifiers," in *Neural Information Processing Systems 12*. MIT Press, 2000, pp. 603–609.

[20] C. Gold and P. Sollich, "Model selection for support vector machine classification," *Neurocomputing*, vol. 55, no. 1-2, pp. 221–249, 2003.

[21] G. C. Cawley and N. L. C. Talbot, "Preventing Over-Fitting during Model Selection via Bayesian Regularisation of the Hyper-Parameters," *Journal of Machine Learning Research*, vol. 8, pp. 841–861, 2007.

[22] T. Glasmachers and C. Igel, "Maximum-Gain Working Set Selection for Support Vector Machines," *Journal of Machine Learning Research*, vol. 7, pp. 1437–1466, 2006.

[23] C. Igel, T. Glasmachers, and V. Heidrich-Meisner, "Shark," *Journal of Machine Learning Research*, vol. 9, pp. 993–996, 2008.

[24] F. Friedrichs and C. Igel, "Evolutionary Tuning of Multiple SVM Parameters," *Neurocomputing*, vol. 64, no. C, pp. 107–117, 2005.

[25] T. Glasmachers and C. Igel, "Uncertainty Handling in Model Selection for Support Vector Machines," in *Parallel Problem Solving from Nature (PPSN X)*, G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, Eds. Springer, 2008, pp. 185–194.

[26] K. M. Chung, W. C. Kao, C. L. Sun, L. L. Wang, and C.-J. Lin, "Radius margin bounds for support vector machines with the RBF kernel," *Neural Computation*, vol. 15, no. 11, pp. 2643–2681, 2003.

[27] K. Duan, S. S. Keerthi, and A. N. Poo, "Evaluation of simple performance measures for tuning SVM hyperparameters," *Neurocomputing*, vol. 51, no. 1, pp. 41–60, 2003.

[28] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, pp. 267–276, 2007.

[29] O. Chapelle, "Support Vector Machines: Induction Principle, Adaptive Tuning and Prior Knowledge," Ph.D. dissertation, Laboratoire d'Informatique de Paris 6, 2002.

[30] G. Wahba, "Soft and hard classification by reproducing kernel Hilbert space methods," *Proceedings of the National Academy of Sciences*, vol. 99, no. 26, pp. 16 524–16 530, 2002.

[31] G. C. Cawley and N. L. C. Talbot, "Efficient approximate leave-one-out cross-validation for kernel logistic regression," *Machine Learning*, vol. 71, no. 2, pp. 243–264, 2008.

[32] T. Suttorp, N. Hansen, and C. Igel, "Efficient covariance matrix update for variable metric evolution strategies," *Machine Learning*, vol. 75, no. 2, pp. 167–197, 2009.

[33] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[34] C. Igel and M. Hüsken, "Empirical Evaluation of the Improved Rprop Learning Algorithm," *Neurocomputing*, vol. 50, pp. 105–123, 2003.

[35] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[36] A. Asuncion and D. J. Newman, "UCI machine learning repository," 2007. [Online]. Available: www.ics.uci.edu/∼mlearn/MLRepository.html