# Multi-Objective Model Selection for Support Vector Machines

Christian Igel

Institute for Neurocomputing
Ruhr-University Bochum
44780 Bochum, Germany
christian.igel@neuroinformatik.rub.de

**Abstract.** In this article, model selection for support vector machines is viewed as a multi-objective optimization problem, where model complexity and training accuracy define two conflicting objectives. Different optimization criteria are evaluated: Split modified radius margin bounds, which allow for comparing existing model selection criteria, and the training error in conjunction with the number of support vectors for designing sparse solutions.

## 1 Introduction

Model selection for supervised learning systems requires finding a suitable trade-off between at least two objectives, especially between model complexity and accuracy on a set of noisy training examples ($\rightarrow$ bias vs. variance, capacity vs. empirical risk). Usually, this multi-objective problem is tackled by aggregating the objectives into a scalar function and applying standard methods to the resulting single-objective task. However, this approach can only lead to satisfactory solutions if the aggregation (e.g., a linear weighting of empirical error and regularization term) matches the problem. Thus, choosing an appropriate aggregation itself is an optimization task. A better way is to apply multi-objective optimization (MOO) to approximate the set of Pareto-optimal trade-offs and to choose a final solution afterwards, as discussed in the context of neural networks in [1–4]. A solution is Pareto-optimal if it cannot be improved in any objective without getting worse in at least one other objective [5–7].

In the following, we consider MOO of the kernel and the regularization parameter of support vector machines (SVMs). We show how to reveal the trade-off between different objectives to guide the model selection process, e.g., for the design of sparse SVMs. One advantage of SVMs is that theoretically well founded bounds on the expected generalization performance exist, which can serve as model selection criteria.[1] However, in practice heuristic modifications of these bounds—e.g., corresponding to different weightings of capacity and empirical

[1] When used for model selection in the described way, the term "bound" is slightly misleading.

risk—can lead to better results [8]. The MOO approach enables us to compare model selection criteria proposed in the literature after optimization.

As we consider only kernels from a parameterized family of functions, our model selection problem reduces to multidimensional real-valued optimization. We present a multi-objective evolution strategy (ES) with self-adaptation for real-valued MOO. The basics of MOO using non-dominated sorting [6, 9] and the ES are presented in the next section. Then SVMs and the model selection criteria we consider are briefly described in section 3.2 and the experiments in section 4.

## 2 Evolutionary multi-objective optimization

Consider an optimization problem with $M$ objectives $f_1, \ldots, f_M : X \to \mathbb{R}$ to be minimized. The elements of $X$ can be partially ordered using the concept of Pareto dominance. A solution $\boldsymbol{x} \in X$ dominates a solution $\boldsymbol{x}'$ and we write $\boldsymbol{x} \prec \boldsymbol{x}'$ iff $\exists m \in \{1, \ldots, M\} : f_m(\boldsymbol{x}) < f_m(\boldsymbol{x}')$ and $\nexists m \in \{1, \ldots, M\} : f_m(\boldsymbol{x}) > f_m(\boldsymbol{x}')$. The elements of the (Pareto) set $\{\boldsymbol{x} \mid \nexists \boldsymbol{x}' \in X : \boldsymbol{x}' \prec \boldsymbol{x}\}$ are called Pareto-optimal. Without any further information, no Pareto-optimal solution can be said to be superior to another. The goal of multi-objective optimization (MOO) is to find in a single trial a diverse set of Pareto-optimal solutions, which provide insights into the trade-offs between the objectives. When approaching a MOO problem by linearly aggregating all objectives into a scalar function, each weighting of the objectives yields only a limited subset of Pareto-optimal solutions. That is, various trials with different aggregations become necessary—but when the Pareto front (see below) is not convex, even this inefficient procedure does not help (cf. [6, 7]). Evolutionary multi-objective algorithms have become the method of choice for MOO [5, 6]. The most popular variant is the non-dominated sorting genetic algorithm NSGA-II, which shows fast convergence to the Pareto-optimal set and a good spread of solutions [6, 9]. On the other hand, evolution strategies (ES) are among the most elaborated and best analyzed evolutionary algorithms for real-valued optimization. Therefore, we propose a new method that combines ES with concepts from the NSGA-II.

### 2.1 Non-dominated sorting

We give a concise description of the non-dominated sorting approach used in NSGA-II. For more details and efficient algorithms realizing this sorting we refer to [9]. First of all, the elements in a finite set $A \subseteq X$ of candidate solutions are ranked according to their level of non-dominance. Let the non-dominated solutions in $A$ be denoted by $\mathrm{ndom}(A) = \{a \in A \mid \nexists a' \in A : a' \prec a\}$. The Pareto front of $A$ is then given by $\{(f_1(a), \ldots, f_M(a)) \mid a \in \mathrm{ndom}(A)\}$. The elements in $\mathrm{ndom}(A)$ have rank 1. The other ranks are defined recursively by considering the set without the solutions with lower ranks. Formally, let $\mathrm{dom}_n(A) = \mathrm{dom}_{n-1}(A) \setminus \mathrm{ndom}_n(A)$ and $\mathrm{ndom}_n(A) = \mathrm{ndom}(\mathrm{dom}_{n-1}(A))$ for $n \in \{1, \ldots\}$ with $\mathrm{dom}_0 = A$. For $a \in A$ we define the level of non-dominance $\mathrm{r}(a, A)$ to be $i$ iff $a \in \mathrm{ndom}_i(A)$.

As a second sorting criterion, non-dominated solutions $A'$ are ranked according to how much they contribute to the spread (or diversity) of objective function values in $A'$. This can be measured by the crowding-distance. For $M$ objectives, the crowding-distance of $a \in A'$ is given by $\mathrm{c}(a, A') = \sum_{m=1}^{M} \mathrm{c}_m(a, A')/(f_m^{\max} - f_m^{\min})$, where $f_m^{\max}$ and $f_m^{\min}$ are (estimates of) the minimum and maximum value of the $m$th objective and

$$\mathrm{c}_m(a, A') = \begin{cases} \infty\,, \text{ if } f_m(a) = \min\{f_m(a') \,|\, a' \in A'\} \text{ or } f_m(a) = \max\{f_m(a') \,|\, a' \in A'\} \\ \min\{f_m(a'') - f_m(a') \,| \\ \qquad a', a'' \in A' : f_m(a') < f_m(a) < f_m(a'')\}\,, \text{ otherwise.} \end{cases}$$

Based on the level of non-dominance and the crowding distance we define the relation

$$a \prec_A a' \Leftrightarrow \mathrm{r}(a, A) < \mathrm{r}(a', A) \text{ or } \left[(\mathrm{r}(a, A) = \mathrm{r}(a', A)) \wedge (\mathrm{c}(a, \mathrm{ndom}_{\mathrm{r}(a', A)}(A)) > \mathrm{c}(a', \mathrm{ndom}_{\mathrm{r}(a', A)}(A)))\right]\,,$$

for $a, a' \in A$. That is, $a$ is better than $a'$ when compared using $\prec_A$ if either $a$ has a better (lower) level of non-dominance or $a$ and $a'$ are on the same level but $a$ is in a "lesser crowded region of the objective space" and therefore induces more diversity.

### 2.2 Multi-objective evolution strategy with mutative self-adaptation

Evolution strategies (ES, cf. [10, 11]) are one of the main branches of evolutionary algorithms (EAs), i.e., a class of iterative, direct, randomized optimization methods mimicking principles of neo-Darwinian evolution theory. In EAs, a (multi-) set of $\mu$ individuals representing candidate solutions, the so called parent population, is maintained. In each iteration (generation) $t$, $\lambda$ new individuals (the offspring) are generated based on the parent population. A selection procedure preferring individuals representing better solutions to the problem at hand determines the parent population of the next iteration.

In our ES, each individual $a_i^{(t)} \in \mathbb{R}^{2n}$ is divided into two parts, $a_i^{(t)} = (\boldsymbol{x}_i^{(t)}, \boldsymbol{\sigma}_i^{(t)})$, the object variables $\boldsymbol{x}_i^{(t)} \in \mathbb{R}^n = X$ representing the corresponding candidate solution and the strategy parameters $\boldsymbol{\sigma}_i^{(t)} \in \mathbb{R}^n$. For simplicity, we do not distinguish between $f_m(a_i^{(t)})$ and $f_m(\boldsymbol{x}_i^{(t)})$. The strategy parameters are needed for self-adaptation, a key concept in EAs [12, 13] that allows for an online adaptation of the search strategy leading to improved search performance in terms of both accuracy and efficiency (similar to adaptive step-sizes / learning rates in gradient-based steepest-descent algorithms). The initial parent population consists of $\mu$ randomly created individuals. In each iteration $t$, new individuals $a_i^{(t+1)}$, $i = 1, \ldots, \lambda$ are generated. For each offspring $a_i^{(t+1)}$, two individuals, say $a_u^{(t)} = (\boldsymbol{x}_u^{(t)}, \boldsymbol{\sigma}_u^{(t)})$ and $a_v^{(t)} = (\boldsymbol{x}_v^{(t)}, \boldsymbol{\sigma}_v^{(t)})$, are chosen from the

current parent population uniformly at random. The new strategy parameters $\boldsymbol{\sigma}_i^{(t+1)} = (\sigma_{i,1}^{(t+1)}, \ldots, \sigma_{i,n}^{(t+1)})$ of offspring $i$ are given by

$$\sigma_{i,k}^{(t+1)} = \underbrace{\frac{1}{2}\left[\sigma_{u,k}^{(t)} + \sigma_{v,k}^{(t)}\right]}_{\text{intermediate recombination}} \underbrace{\cdot \exp\left(\tau' \cdot \zeta_i^{(t)} + \tau \cdot \zeta_{i,k}^{(t)}\right)}_{\text{log-normal mutation}} .$$

Here, the $\zeta_{i,k}^{(t)} \sim \mathcal{N}(0,1)$ are realizations of a normally distributed random variable with zero mean and unit variance that is sampled anew for each component $k$ for each individual $i$, whereas the $\zeta_i^{(t)} \sim \mathcal{N}(0,1)$ are sampled once per individual and are identical for each component. The mutation strengths are set to $\tau = 1/\sqrt{2\sqrt{n}}$ and $\tau' = 1/\sqrt{2n}$ [12, 11]. Thereafter the objective parameters are altered using the new strategy parameters:

$$x_{i,k}^{(t+1)} = \underbrace{x_{c_{i,k}^{(t)},k}^{(t)}}_{\text{discrete recombination}} \underbrace{+ \sigma_{i,k}^{(t+1)} z_{i,k}^{(t)}}_{\text{Gaussian mutation}} ,$$

where $z_{i,k}^{(t)} \sim \mathcal{N}(0,1)$. The $c_{i,k}^{(t)}$ are realizations of a random variable taking the values $u$ and $v$ with equal probability. After generating the offspring, $(\mu, \lambda)$-selection is used, i.e., the $\mu$ best individuals of the offspring form the new parent population.

So far, we have described a canonical ES with mutative self-adaptation, for more details please see [11] and references therein. Now we turn this ES into a multi-objective algorithm by using the non-dominated sorting operator $\prec_{\{a_1^{(t+1)},\ldots,a_\lambda^{(t+1)}\}}$ for the ranking in the $(\mu, \lambda)$-selection in iteration $t$. In addition, we keep an external archive $\mathcal{A}^{(t+1)} = \text{ndom}(\mathcal{A}^{(t)} \cup \{a_1^{(t+1)}, \ldots, a_\lambda^{(t+1)}\})$ of all non-dominated solutions discovered so far starting from the initial population. This extends the ideas in [14] and yields the first self-adaptive ES using non-dominated sorting, which we call NSES.

The NSES uses a non-elitist selection scheme (i.e., the best solutions found so far are not kept in the parent population), because self-adaption does not work well together with elitism. This is in contrast to NSGA-II [9] and the self-adaptive SPANN [2]. Of course, the NSES is elitist when looking at the concurrent archive $\mathcal{A}^{(t)}$.

## 3 Models selection for SVMs

Support vector machines (SVMs, e.g., [15–17]) are learning machines based on two key elements: a general purpose linear learning algorithm and a problem specific kernel that computes the inner product of input data points in a feature space.

### 3.1 Support vector machines

We consider $L_1$-norm soft margin SVMs for the discrimination of two classes. Let $(\boldsymbol{x}_i, y_i), 1 \leq i \leq \ell$, be the training examples, where $y_i \in \{-1, 1\}$ is the

label associated with input pattern $\boldsymbol{x}_i \in X$. The main idea of SVMs is to map the input vectors to a feature space $F$ and to classify the transformed data by a linear function. The transformation $\phi : X \to F$ is implicitly done by a kernel $K : X \times X \to \mathbb{R}$, which computes an inner product in the feature space, i.e., $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$. The linear function for classification in the feature space is chosen according to a generalization error bound considering a margin and the margin slack vector, i.e., the amounts by which individual training patterns fail to meet that margin (cf. [15–17]). This leads to the SVM decision function

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{\ell} y_i \alpha_i^* K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \quad ,$$

where the coefficients $\alpha_i^*$ are the solution of the following quadratic optimization problem:

$$\begin{aligned} \text{maximize} \quad & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i, \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C, \quad i = 1, \ldots, \ell \ . \end{aligned}$$

The optimal value for $b$ can then be computed based on the solution $\boldsymbol{\alpha}^*$. The vectors $\boldsymbol{x}_i$ with $\alpha_i^* > 0$ are called support vectors. The number of support vectors is denoted by $\#\,\mathrm{SV}$. The regularization parameter $C$ controls the trade-off between maximizing the margin

$$\gamma = \left(\sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)^{-1/2}$$

and minimizing the $L_1$-norm of the final margin slack vector $\boldsymbol{\xi}^*$ of the training data, where

$$\xi_i^* = \max\left(0, 1 - y_i \left(\sum_{j=1}^{\ell} y_j \alpha_j^* K(\boldsymbol{x}_j, \boldsymbol{x}_j) + b\right)\right) \quad .$$

### 3.2 Model selection criteria for $L_1$-SVMs

Choosing the right kernel for a SVM is important for its performance. When a parameterized family of kernel functions is considered, kernel adaptation reduces to finding an appropriate parameter vector. These parameters together with the regularization parameter are called hyperparameters of the SVM. In practice,

the hyperparameters are usually determined by grid search. Because of the computational complexity, grid search is only suitable for the adjustment of very few parameters. Further, the choice of the discretization of the search space may be crucial. Perhaps the most elaborate techniques for choosing hyperparameters are gradient-based approaches [18, 8, 19, 20]. When applicable, these methods are highly efficient. However, they have some drawbacks and limitations: The kernel function has to be differentiable. The score function for assessing the performance of the hyperparameters (or at least an accurate approximation of this function) also has to be differentiable with respect to all hyperparameters, which excludes reasonable measures such as the number of support vectors. In some approaches, the computation of the gradient is only exact in the hard-margin case (i.e., for separable data / $L_2$-SVMs) when the model is consistent with the training data. Further, as the objective functions are indeed multi-modal, the performance of gradient-based heuristics depends on the initialization—the algorithms are prone to getting stuck in local optima. In [21, 22], single-objective evolution strategies were proposed for adapting SVM hyperparameters, which partly overcome these problems; in [23] a single-objective genetic algorithm was used for SVM feature selection (see also [24–26]) and adaptation of the (discretized) regularization parameter. Like gradient-based techniques, these methods are not suitable for MOO. Therefore we apply the NSES to directly address the multi-objective nature of model selection.

We optimize Gaussian kernels $k_{\boldsymbol{A}}(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-(\boldsymbol{x}-\boldsymbol{z})^T \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{z})\right)$, $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^m$. We look at two parameterizations of the symmetric, positive definite matrix $\boldsymbol{A}$. In the standard scenario, we adapt $k_{\gamma \boldsymbol{I}}$, where $\boldsymbol{I}$ is the unit matrix and $\gamma > 0$ is the only adjustable parameter. In addition, we optimize $m$ independent scaling factors weighting the input components and consider $k_{\boldsymbol{D}}$, where $\boldsymbol{D}$ is a diagonal matrix with arbitrary positive entries.

We want to design classifiers that generalize well. The selection criteria we consider are therefore (partly) based on bounds on the number of errors in the leave-one-out procedure, which gives an estimate of the expected generalization performance. However, most bounds were derived for the hard-margin case (i.e., for separable data / $L_2$-SVMs) and $L_1$-SVMs cannot be reduced to this scenario. Thus, we combine heuristics and results for the hard-margin case to selection criteria for $L_1$-SVMs.

### 3.3 Modified radius margin bounds

Let $R$ denote the radius of the smallest ball in feature space containing all $\ell$ training examples given by

$$R = \sqrt{\sum_{i=1}^{\ell} \beta_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{\ell} \beta_i^* \beta_j^* K(\boldsymbol{x}_i, \boldsymbol{x}_j)} \ ,$$

where $\boldsymbol{\beta}^*$ is the solution vector of the quadratic optimization problem

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{maximize}} \quad & \sum_{i=1}^{\ell} \beta_i K(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{\ell} \beta_i \beta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \beta_i = 1 \\ & \beta_i \geq 0 \quad , \quad i = 1, \ldots, \ell \ , \end{aligned}$$

see [27]. Following a suggestion by Olivier Chapelle, the modified radius margin bound

$$T_{\mathrm{DM}} = (2R)^2 \sum_{i=1}^{\ell} \alpha_i^* + \sum_{i=1}^{\ell} \xi_i^* \ ,$$

was considered for model selection of $L_1$-SVMs in [28]. In practice, this expression did not lead to satisfactory results [8, 28]. Therefore, in [8] it was suggested to use

$$T_{\mathrm{RM}} = R^2 \sum_{i=1}^{\ell} \alpha_i^* + \sum_{i=1}^{\ell} \xi_i^* \ ,$$

based on heuristic considerations and it was shown empirically that $T_{\mathrm{RM}}$ leads to better models than $T_{\mathrm{DM}}$.[2] Both criteria are not differentiable [28]. They can be viewed as two different aggregations of the following two objectives

$$f_1 = R^2 \sum_{i=1}^{\ell} \alpha_i^* \text{ and } f_2 = \sum_{i=1}^{\ell} \xi_i^* \tag{1}$$

penalizing model complexity and training errors, respectively. For example, a highly complex SVM classifier that very accurately fits the training data has high $f_1$ and small $f_2$.

### 3.4 Number of SVs and training error

There are good reasons to prefer SVMs with few support vectors: In the hard-margin case, the number of SVs is an upper bound on the expected number of errors made by the leave-one-out procedure (e.g., see [18, 17]). Further, the space and time complexity of the SVM classifier scales with the number of SVs. A natural measure for the performance of a classifier on a training set is the

[2] Also for $L_2$-SVMs it was shown empirically that theoretically better founded weightings of such objectives (e.g., corresponding to tighter bounds) need not correspond to better model selection criteria [8].

percentage of misclassified patterns of the training data $\mathrm{CE}(D_{\mathrm{train}})$. Hence, we consider

$$f_1' = \#\,\mathrm{SV} + \theta\left(R^2 \sum_{i=1}^{\ell} \alpha_i^*\right) \text{ and } f_2' = \ell \cdot \mathrm{CE}(D_{\mathrm{train}}) + \theta\left(\sum_{i=1}^{\ell} \xi_i^*\right) \quad , \qquad (2)$$

where $\theta(x) = x/(1+x)$. For example, it is easy to achieve zero classification error when all training points become support vectors, but this solution is not likely to generalize well.

The optional $\theta(\dots)$ terms are used for smoothing the objective functions. In recent experiments, it turns out that they can be omitted without deterioration of performance.

## 4 Experiments

For the evaluation of our hyperparameter optimization method we used the common medical benchmark datasets *breast-cancer*, *diabetes*, *heart*, and *thyroid* with input dimensions $m$ equal to 9, 8, 13, and 5, and $\ell$ equal to 200, 468, 170, and 140. The data originally from the UCI Benchmark Repository [29] are preprocessed and partitioned as in [30], where we consider the first of the splits into training and test set $D_{\mathrm{train}}$ and $D_{\mathrm{test}}$. We applied the NSES to adapt $C$ and the parameters of $k_{\gamma \boldsymbol{I}}$ or $k_{\boldsymbol{D}}$; the quality of a candidate solution was determined after SV learning. We set $\mu = 15$ and $\lambda = 75$. The strategy parameters $\boldsymbol{\sigma}$ were initialized to 1 when adapting $k_{\gamma \boldsymbol{I}}$ and because of the increased number $(m+1)$ of objective parameters to 0.1 when adapting $k_{\boldsymbol{D}}$. The objective parameters of the initial individuals were chosen uniformly at random from $[0.1, 100]$ and $[0.01, 10]$ for $C$ and the kernel parameters, respectively. The NSES parameters were not tuned, i.e, the efficiency of the ES could surely be improved. Figures 1, 2, and 3 depict the solutions in the final archives.[3]

### 4.1 Modified radius margin bounds

Figure 1 shows the results of optimizing $k_{\gamma \boldsymbol{I}}$ using the objectives (1); for each $f_1$ value of a solution in the final archive the corresponding $f_2$, $T_{\mathrm{RM}}$, $T_{\mathrm{DM}}$, and $100 \cdot \mathrm{CE}(D_{\mathrm{test}})$ are given. For *diabetes*, *heart*, and *thyroid*, the solutions lie on typical convex Pareto fronts; in the *breast-cancer* example the convex front looks piecewise linear. Assuming convergence to the Pareto-optimal set, the results of a single MOO trial are sufficient to determine the outcome of single-objective optimization of any (positive) linear weighting of the objectives. Thus, we can directly determine and compare the solutions that minimizing $T_{\mathrm{RM}}$ and $T_{\mathrm{DM}}$ would suggest. Our experiments substantiate the findings in [8] that the heuristic bound $T_{\mathrm{RM}}$ is better suited for model selection than $T_{\mathrm{DM}}$: When looking at

[3] Because it is difficult to present sets of sets, we discuss outcomes from typical (randomly chosen) trials. Additional results showing the robustness of our approach can be downloaded from http://www.neuroinformatik.rub.de/PEOPLE/igel/moo .

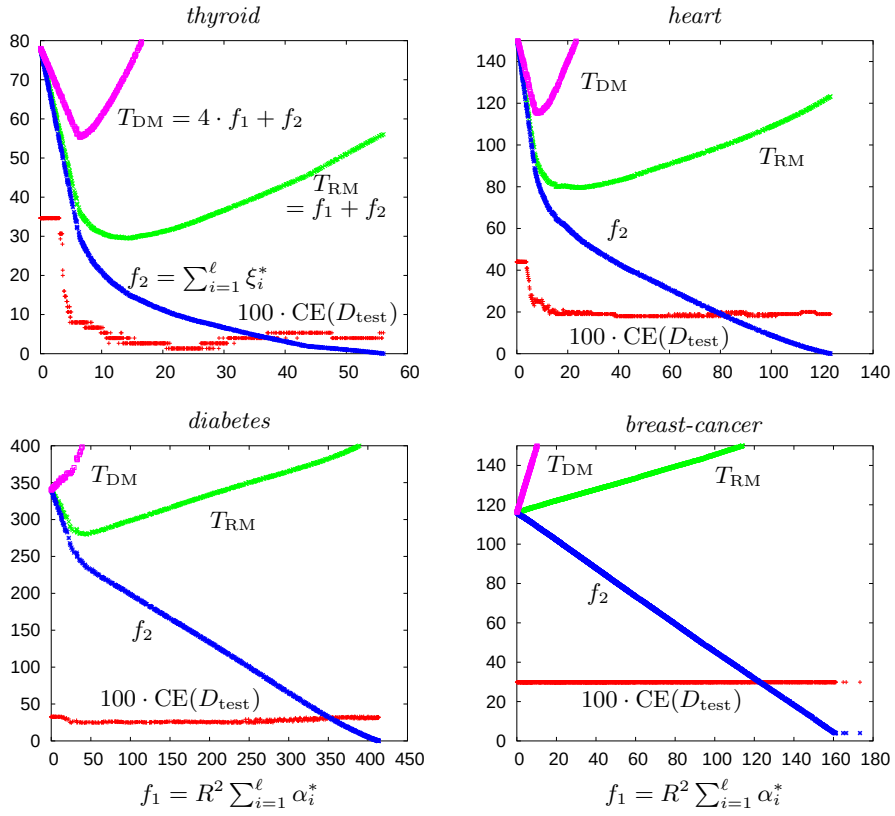


**Fig. 1.** Pareto fronts (i.e., $f_2$ vs. $f_1$) of $\mathcal{A}^{(100)}$—the outcome of the optimization after $100 \cdot \lambda$ evaluations—for $k_{\gamma \boldsymbol{I}}$ and the four benchmark problems. Additionally, for every solution in $\mathcal{A}^{(100)}$ the values of $T_{\mathrm{RM}}$, $T_{\mathrm{DM}}$, and $100 \cdot \mathrm{CE}(D_{\mathrm{test}})$ are plotted against the corresponding $f_1$ value. Projecting the minimum of $T_{\mathrm{RM}}$ (for $T_{\mathrm{DM}}$ proceed analogously) along the $y$-axis on the Pareto front gives the $(f_1, f_2)$ pair suggested by the model selection criterion $T_{\mathrm{RM}}$—this would also be the outcome of single-objective optimization using $T_{\mathrm{RM}}$. Projecting an $(f_1, f_2)$ pair along the $y$-axis on $100 \cdot \mathrm{CE}(D_{\mathrm{test}})$ yields the corresponding error on an external test set (which is assumed to be not available for model selection).

$\mathrm{CE}(D_{\mathrm{test}})$ and the minima of $T_{\mathrm{RM}}$ and $T_{\mathrm{DM}}$, we can conclude that $T_{\mathrm{DM}}$ puts too much emphasis on the "radius margin part" yielding worse classification results on the external test set (except for *breast-cancer* where there is no difference on $D_{\mathrm{test}}$). The *heart* and *thyroid* results suggest that even more weight should be given to the slack-variables (i.e., the performance on the training set) than in $T_{\mathrm{RM}}$.

In the MOO approach, degenerated solutions resulting from a not appropriate weighting of objectives (which we indeed observed—without the chance to

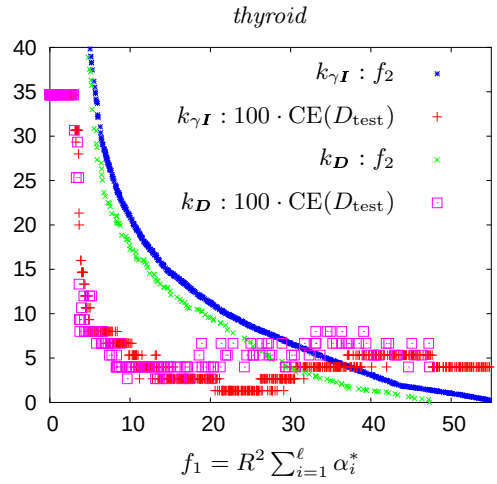


**Fig. 2.** Pareto fronts after optimizing $k_{\gamma \boldsymbol{I}}$ and $k_{\boldsymbol{D}}$ for objectives (1) and *thyroid* data after 100 iterations. For both kernel parameterizations, $f_2$ and $100 \cdot \mathrm{CE}(D_{\mathrm{test}})$ are plotted against $f_1$.

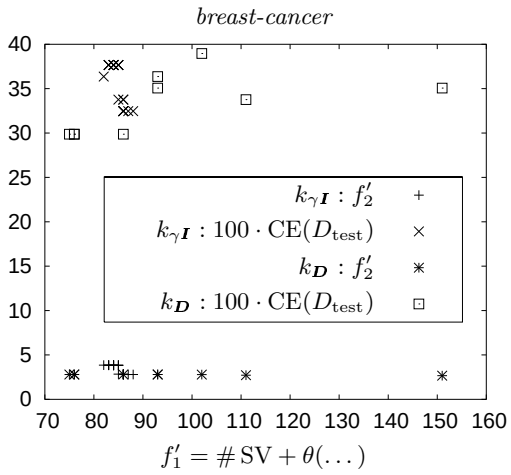


**Fig. 3.** Pareto fronts after optimizing $k_{\gamma \boldsymbol{I}}$ and $k_{\boldsymbol{D}}$ for objectives (2) and *breast-cancer* data after 200 generations. For both kernel parameterizations, $f_2'$ and $100 \cdot \mathrm{CE}(D_{\mathrm{test}})$ are plotted against $f_1'$.

change the trade-off afterwards—in single-objective optimization of SVMs) become obvious and can be excluded. For example, one would probably not pick the solution suggested by $T_{\mathrm{DM}}$ in the *diabetes* benchmark. A typical MOO heuristic is to choose a solution from the archive that corresponds to an "interesting"

part of the Pareto front. In case of a typical convex front, this would be the area of highest "curvature" (the "knee", see figure 1). In our benchmark problems, this leads to results on a par with $T_{\mathrm{RM}}$ and much better than $T_{\mathrm{DM}}$ (except for *breast-cancer*, where the test errors of all optimized trade-offs were the same). Therefore, this heuristic combined with $T_{\mathrm{RM}}$ (derived from the MOO results) is an alternative for model selection based on modified radius margin bounds.

Adapting the scaling of the kernel (i.e., optimizing $k_{\boldsymbol{D}}$) sometimes led to better objective values compared to $k_{\gamma \boldsymbol{I}}$, see figure 2 for an example, but not to better generalization performance.

### 4.2 Number of SVs and training error

Now we describe the results achieved when optimizing the objectives (2). We write solutions as triples $[\#\,\mathrm{SV}, \mathrm{CE}(D_{\mathrm{train}}), \mathrm{CE}(D_{\mathrm{test}})]$. In the *thyroid* example with two hyperparameters, the archive (the Pareto set) after 100 generations contained a single solution $[14, 0, 0.027]$. When additionally adjusting the scaling, the archive contained the single solution $[11, 0, 0.027]$. In case of the *heart* data, the archives did not change qualitatively after 200 generations. For $k_{\gamma \boldsymbol{I}}$ we got a Pareto set containing solutions from $[49, 0.17, 0.2]$ to $[57, 0, 0.17]$. With scaling, the NSES converged to an archive containing only consistent solutions with 49 SVs. However, the classification error on the test set was either 0.23 or 0.24. In the *diabetes* task, the archives did not change qualitatively after 100 generations, where we got solutions from $[208, 0, 0.297]$ to $[212, 0, 0.287]$ for $k_{\gamma \boldsymbol{I}}$ and an archive where all solutions corresponded to $[178, 0, 0.297]$ when adjusting $k_{\boldsymbol{D}}$. Results for the *breast-cancer* data are shown in figure 3. In all four scenarios, we achieved lower objective values when adapting the scaling. These results did not necessarily correspond to lower $\mathrm{CE}(D_{\mathrm{test}})$.

## 5 Conclusion

Model selection is a multi-objective optimization (MOO) problem. We presented an evolution strategy combining non-dominated sorting [9] and self-adaptation [12, 14] for efficient MOO. It was successfully applied to optimize multiple hyperparameters of SVMs with Gaussian kernels based on conflicting, not differentiable criteria. In most experiments, better objective values were achieved when adapting individual scaling factors for the input components. However, these solutions did not necessarily correspond to lower errors on external test sets. The final Pareto fronts visualize the trade-off between model complexity and learning accuracy for guiding the model selection process. When looking at split modified radius margin bounds, standard MOO heuristics based on the curvature of the Pareto front led to comparable models as using the modified bound proposed in [8]. The latter puts more emphasis on minimizing the slack vector compared to the bound considered in [28], a strategy that is strongly supported by our results. In practice, the experiments involving minimization of the number of support vectors are of particular interest, because here the complexity objective

is directly related to the speed of the classifier. Knowing the speed vs. accuracy trade-off is helpful when designing SVMs that have to obey real-time constraints.

## References


1. Abbass, H.A.: An evolutionary artificial neural networks approach for breast cancer diagnosis. Artificial Intelligence in Medicine **25** (2002) 265–281
2. Abbass, H.A.: Speeding up backpropagation using multiobjective evolutionary algorithms. Neural Computation **15** (2003) 2705–2726
3. Jin, Y., Okabe, T., Sendhoff, B.: Neural network regularization and ensembling using multi-objective evolutionary algorithms. In: Congress on Evolutionary Computation (CEC'04), IEEE Press (2004) 1–8
4. Wiegand, S., Igel, C., Handmann, U.: Evolutionary multi-objective optimization of neural networks for face detection. International Journal of Computational Intelligence and Applications **4** (2004) 237–253 Special issue on Neurocomputing and Hybrid Methods for Evolving Intelligence.
5. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers (2002)
6. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley (2001)
7. Sawaragi, Y., Nakayama, H., Tanino, T.: Theory of Multiobjective Optimization. Volume 176 of Mathematics in Science and Engineering. Academic Press (1985)
8. Chung, K.M., Kao, W.C., Sun, C.L., Lin, C.J.: Radius margin bounds for support vector machines with RBF kernel. Neural Computation **15** (2003) 2643–2681
9. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6** (2002) 182–197
10. Beyer, H.G.: The Theory of Evolution Strategies. Springer-Verlag (2001)
11. Beyer, H.G., Schwefel, H.P.: Evolution strategies: A comprehensive introduction. Natural Computing **1** (2002) 3–52
12. Bäck, T.: An overview of parameter control methods by self-adaptation in evolutionary algorithms. Fundamenta Informaticae **35** (1998) 51–66
13. Igel, C., Toussaint, M.: Neutrality and self-adaptation. Natural Computing **2** (2003) 117–132
14. Laumanns, M., Rudolph, G., Schwefel, H.P.: Mutation control and convergence in evolutionary multi-objective optimization. In Matousek, R., Osmera, P., eds.: Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001), Brno, Czech Republic: University of Technology (2001) 24–29
15. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press (2000)
16. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
17. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)
18. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning **46** (2002) 131–159
19. Gold, C., Sollich, P.: Model selection for support vector machine classification. Neurocomputing **55** (2003) 221–249
20. Keerthi, S.S.: Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. IEEE Transactions on Neural Networks **13** (2002) 1225–1229

21. Friedrichs, F., Igel, C.: Evolutionary tuning of multiple SVM parameters. In Verleysen, M., ed.: 12th European Symposium on Artificial Neural Networks (ESANN 2004), Evere, Belgium: d-side publications (2004) 519–524
22. Runarsson, T.P., Sigurdsson, S.: Asynchronous parallel evolutionary model selection for support vector machines. Neural Information Processing – Letters and Reviews **3** (2004) 59–68
23. Fröhlich, H., Chapelle, O., Schölkopf, B.: Feature selection for support vector machines by means of genetic algorithms. In: 15th IEEE International Conference on Tools with AI (ICTAI 2003), IEEE Computer Society (2003) 142–148
24. Eads, D.R., Hill, D., Davis, S., Perkins, S.J., Ma, J., Porter, R.B., Theiler, J.P.: Genetic algorithms and support vector machines for time series classification. In Bosacchi, B., Fogel, D.B., Bezdek, J.C., eds.: Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V. Volume 4787 of Proceedings of the SPIE. (2002) 74–85
25. Jong, K., Marchiori, E., van der Vaart, A.: Analysis of proteomic pattern data for cancer detection. In Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G., eds.: Applications of Evolutionary Computing. Number 3005 in LNCS, Springer-Verlag (2004) 41–51
26. Miller, M.T., Jerebko, A.K., Malley, J.D., Summers, R.M.: Feature selection for computer-aided polyp detection using genetic algorithms. In Clough, A.V., Amini, A.A., eds.: Medical Imaging 2003: Physiology and Function: Methods, Systems, and Applications. Volume 5031 of Proceedings of the SPIE. (2003) 102–110
27. Schölkopf, B., Burges, C.J.C., Vapnik, V.: Extracting support data for a given task. In Fayyad, U.M., Uthurusamy, R., eds.: Proceedings of the First International Conference on Knowledge Discovery & Data Mining, AAAI Press (1995) 252–257
28. Duan, K., Keerthi, S.S., Poo, A.: Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing **51** (2003) 41–59
29. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
30. Rätsch, G., Onoda, T., Müller, K.R.: Soft margins for adaboost. Machine Learning **42** (2001) 287–32