2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Genetic Algorithms for Support Vector Machine Model Selection

Stefan Lessmann, Robert Stahlbock, Sven F. Crone

Abstract— **The support vector machine is a powerful classifier that has been successfully applied to a broad range of pattern recognition problems in various domains, e.g. corporate decision making, text and image recognition or medical diagnosis. Support vector machines belong to the group of semi-parametric classifiers. The selection of appropriate parameters, formally known as model selection, is crucial to obtain accurate classification results for a given task. Striving to automate model selection for support vector machines we apply a meta-strategy utilizing genetic algorithms to learn combined kernels in a data-driven manner and to determine all free kernel parameters. The model selection criterion is incorporated into a fitness function guiding the evolutionary process of classifier construction. We consider two types of criteria consisting of empirical estimators or theoretical bounds for the generalization error. We evaluate their effectiveness in an empirical study on four well known benchmark data sets to find that both are applicable fitness measures for constructing accurate classifiers and conducting model selection. However, model selection focuses on finding one best classifier while genetic algorithms are based on the idea of re-combining and mutating a large number of good candidate classifiers to realize further improvements. It is shown that the empirical estimator is the superior fitness criterion in this sense, leading to a greater number of promising models on average.**

I. Introduction

THE support vector machine (SVM) is a prominent classifier that has been introduced by Vapnik and co-workers in 1992 [1, 2]. In subsequent years the technique has received considerable attention in various application domains. Promising results have been obtained for e.g. medical diagnosis [3, 4], text and image recognition [5, 6] or the support of corporate decision making [7, 8].

SVMs are supervised learners that construct a model from available training data with known classification. In order to obtain accurate class predictions SVMs provide a number of free parameters that have to be tuned to reflect the requirements of the given task. We will use the term model to refer to a specific classifier, e.g. a SVM with specified kernel and kernel parameters.

The process of parameter fitting is known as model selection aiming at finding a model which will give minimum prediction error when being applied to classify unseen examples that originate from the same source as the training data. Since this true generalization performance is inaccessible we have to rely on appropriate estimators.

Within the scope of SVM model selection we can distinguish two major methodologies. The empirical approach to model selection involves estimating the generalization error by re-sampling techniques such as disjoint hold-out sets or cross-validation (CV) while theoretical approaches consist of constructing and minimizing algebraic bounds for the generalization error.

In this work, we propose a meta-strategy utilizing a genetic algorithm (GA) for model selection striving to determine all properties of the classifier in a solely data-driven manner. A particular classifier is assessed on the basis of its fitness that reflects arbitrary model selection criteria. Consequently, the fitness is the proxy for generalization error and is used to guide the evolutionary process of SVM model construction. We consider the CV performance as a popular empirical estimator for generalization error and the ratio of support vectors and data instances as a classical algebraic bound. Their effectiveness is contrasted in an empirical study using four well known benchmark data sets.

The remainder of the paper is organized as follows. Section II provides an introduction to SVMs while we review previous work on SVM model selection in Section III. Our GA based approach is presented in Section IV. The numerical results of an experimental study are described in Section V. Conclusions are given in Section VI.

II. Support Vector Machines

The SVM can be characterized as a supervised learning algorithm capable of solving linear and non-linear binary classification problems. Given a training set with m patterns $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, where $\boldsymbol{x}_i \in \mathrm{X} \subseteq \Re^n$ is an input vector and $y_i \in \{-1,+1\}$ its corresponding binary class label, the idea of support vector classification is to separate examples by means of a maximal margin hyperplane [9]. Therefore, the algorithm strives to maximize the distance between examples that are closest to the decision surface. The margin of separation is related to the so called Vapnik-Chervonenkis dimension (VCdim) which measures the complexity of a learning machine [10]. The VCdim is used in several bounds for the generalization error of a learner and it is known that margin maximization is beneficial for the generalization ability of the resulting classifier [11]. To construct the SVM classifier one has to minimize the norm of the weight vector $\boldsymbol{w}$ under the constraint that the training patterns of each class reside on opposite sides of the separating surface; see Fig. 1.

S. Lessmann, R. Stahlbock (corresponding author), Institute of Information Systems, University of Hamburg, Von-Melle-Park 5, D-20146 Hamburg, Germany (phone: 0049-40-42838-3063; fax: 0049-40-42838-5535; e-mail: [lessmann, stahlbock] @econ.uni-hamburg.de).

S. F. Crone, Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom (e-mail: s.crone@lancaster.ac.uk).

0-7803-9490-9/06/$20.00/©2006 IEEE

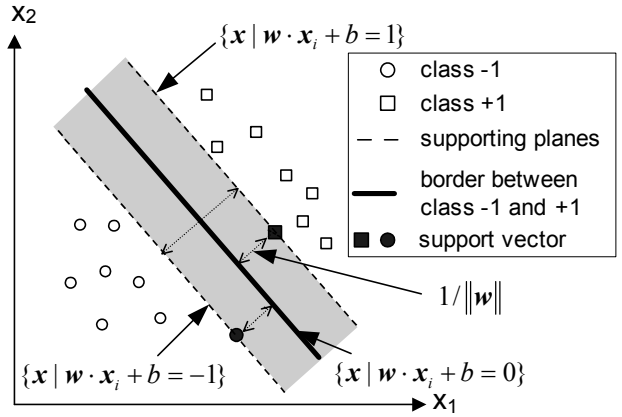

Fig. 1: Linear separation of two classes -1 and +1 in two-dimensional space with SVM classifier [12].

Since $y_i \in \{-1,+1\}$ we can formulate this constraint as

$$y_i((\boldsymbol{w} \cdot \boldsymbol{x}_i) + b) \geq 1, \ i = 1,...,m. \tag{1}$$

Examples which satisfy (1) with equality are called support vectors since they define the orientation of the resulting hyperplane.

To account for misclassifications, e.g. examples where constraint (1) is not met, the soft margin formulation of SVM introduces slack variables $\xi_i \in \Re$ [9]. Hence, to construct a maximal margin classifier one has to solve the convex quadratic programming problem (2):

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\|\boldsymbol{w}\| + C\sum_{i=1}^{m}\xi_i \\ &s.t.: \ y_i((\boldsymbol{w} \cdot \boldsymbol{x}_i) + b) \geq 1 - \xi_i, \ i = 1,...,m. \end{aligned} \tag{2}$$

C is a tuning parameter which allows the user to control the trade off between maximizing the margin (first term in the objective) and classifying the training set without error. The primal decision variables $\boldsymbol{w}$ and b define the separating hyperplane, so that the resulting classifier takes the form

$$y(\boldsymbol{x}) = sgn\left((\boldsymbol{w}^* \cdot \boldsymbol{x}) + b^*\right), \tag{3}$$

where $\boldsymbol{w}^*$ and b^* are determined by (2).

Instead of solving (2) directly, it is common practice to solve its dual (4):

$$\begin{aligned} \max_a = \quad & \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i,j=1}^{m} a_i a_j y_i y_j \boldsymbol{x}_i \boldsymbol{x}_j \\ s.t.: \quad & \sum_{i=1}^{m} a_i y_i = 0 \\ & 0 \leq a_i \leq C \ \ \forall i. \end{aligned} \tag{4}$$

In (4), a_i denotes the Lagrange variable for the i^{th} constraint of (1). Since the input vectors enter the dual only in form of dot products the algorithm can be generalized to non-linear classification by mapping the input data into a high-dimensional feature space via an a priori chosen non-linear mapping function Φ. Constructing a separating hyperplane in this feature space leads to a non-linear decision boundary in the input space. Expensive calculation of dot products $\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$ in a high-dimensional space can be avoided by introducing a kernel function K (5):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j). \tag{5}$$

We obtain the general SVM classifier (6) with decision function (7):

$$\begin{aligned} \max_a = \quad & \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i,j=1}^{m} a_i a_j y_i y_j \boldsymbol{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \\ s.t.: \quad & \sum_{i=1}^{m} a_i y_i = 0 \\ & 0 \leq a_i \leq C \ \ \forall i \end{aligned} \tag{6}$$

$$y(\boldsymbol{x}) = \text{sgn}\left(\sum_{i=1}^{m} \boldsymbol{a}_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right). \tag{7}$$

This kernel trick makes SVM flexible allowing the construction of special purpose kernels, e.g. for text classification [13].

III. Approaches for SVM Model Selection

Regarding the final SVM formulation (6), the free parameters of SVMs to be determined within model selection are given by the regularization parameter C and the kernel, together with additional parameters of the respective kernel function.

A generic approach to model selection, applicable with any learning algorithm, involves cross-validating a parameterized classifier on a sub-sample of available data that has not been used for training. Repetitive evaluation of a model on k disjoint sub-samples while the union of the remaining k-1 sub-samples is used to form the training set gives the well known CV estimate of generalization performance. We obtain the leave-one-out estimate [14] as a special case of CV by setting $k = m$-1. While being computationally expensive the leave-one-out estimator is appealing since it uses the largest possible amount of training data for model building.

For SVMs, CV-based model selection is popular in conjunction with previously determined kernels. In particular, when considering only Gaussian kernels (Table 1) the number of free parameters reduces to two (regularization parameter C and kernel width). These are routinely determined by means of a grid-search varying the parameter settings with a fixed step-size through a wide range of values and assessing the performance of every combination [7, 15]. To reduce the potentially large number of parameter combinations, Keerthi and Lin proposed a heuristic that starts with a linear kernel to determine C and subsequently executes a line search to find promising candidates for the parameters of a Gaussian SVM [16].

Due to extensive re-sampling and re-training of the classifier, these empirical techniques, and the calculation of the leave-one-out estimate in particular, are expensive. A computationally more feasible alternative is to construct algebraic bounds for the generalization error, or the leave-one out estimate respectively, which are easier to calculate. Using this approach, model selection is accomplished by as-

sessing a classifier's capability to minimize these bounds.

For SVMs, the task of developing classifier specific bounds has received considerable attention in the literature; e.g. [1, 17-19], see [20] for comparisons. For example, (8) describes a simple bound T for the leave-one-out error, given by the ratio of support vectors ($\#SV$) to the number of training examples [11]:

$$T = \frac{\#SV}{m}. \tag{8}$$

This bound is inspired by the idea that removing a non-support vector from the training set does not change the optimal solution of (6) and leaves the resulting classifier unchanged [21].

By calculating the derivatives of such bounds with respect to the free parameter one can develop efficient search techniques for finding high quality parameterizations, e.g. [21-23]. However, these bounds usually depend on certain assumptions, e.g. they are valid only for a specific kernel or require a separation of the training set without error. Therefore, meta-heuristics as generic search procedures have been proposed as an alternative facilitating the use of arbitrary, non-differentiable model selection criteria [24, 25].

IV. Genetic Algorithms for SVM Model Selection

A. Genetic algorithms

GA are meta-heuristics that imitate the long-term optimization process of biological evolution for solving mathematical optimization problems. They are based upon Darwin's principle of the 'survival of the fittest'. Problem solutions are abstract 'individuals' in a population. Each solution is evaluated by a fitness function. The fitness value expresses survivability of a solution, i.e. the probability of being a member of the next population and generating 'children' with similar characteristics by handing down genetic information via evolutionary mechanisms like reproduction, variation and selection, respectively. Reproduction and variation is achieved by mutation of genes and crossover. The latter combines characteristics of two solutions for deriving two new solutions. The coding of the problem into a genetic representation, e.g. the sequence of the phenotype's parameters on a genotype, is crucial to the performance of GA. Moreover, the fitness function has great impact on performance. The reader is referred to [26, 27] for more detailed information regarding GA.

B. Data driven construction of SVM kernels

Meta-heuristics like GA have been used in conjunction with SVM in several ways, e.g. for feature selection [28], optimizing SVM's parameters (assuming a fixed kernel) [29], and kernel construction [24, 25].

We believe that the task of feature selection resides more in the realms of data pre-processing than within model selection and discard it from further analysis. While GA can be used to tune the parameters of a specific SVM with fixed kernel, a data driven kernel construction is obviously more flexible so that we follow this approach.

It has been shown that if *K1* and *K2* are kernels, we can derive a new valid kernel $\tilde{K}$ by $\tilde{K} = K1 + K2$ and $\tilde{K} = K1 \bullet K2$, respectively [9]. Consequently, we can use any number of base kernels and combine them to build a combined kernel. This idea has been proposed by [25, 30, 31] and we implement it by using the basic kernels of Table 1.

TABLE 1:
BASIC KERNELS FOR CONSTRUCTION OF COMBINED KERNEL

Radial (K_{rad})	$K_{rad}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\alpha \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2\right)$
Polynomial (K_{poly})	$K_{poly}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\alpha(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + \beta\right)^d$
Sigmoidal (K_{sig})	$K_{sig}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\left(\alpha(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + \beta\right)$
Anova (K_{anova})	$K_{anova}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_j \exp\left(-\alpha\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\right)^2\right)^d$
Inverse multi-quadratic (K_{imq})	$K_{imq}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1/\sqrt{\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2 + \beta^2}$

Therewith, we obtain the combined kernel $\tilde{K}$ (9) with $\otimes_j \in \{+;\cdot\}\ \forall j = 1,...,4$:

$$\tilde{K} = K_{poly}^{\kappa_1} \otimes_1 K_{rad}^{\kappa_2} \otimes_2 K_{sig}^{\kappa_3} \otimes_3 K_{imq}^{\kappa_4} \otimes_4 K_{anova}^{\kappa_5}. \tag{9}$$

C. Genetic representation of SVM's combined kernel

In order to facilitate a data driven determination of the combined kernel (9) by means of GA we have to define a genotype encoding for the free parameters. This is accomplished by using five integer genes for the kernel exponents $(\kappa_1,...,\kappa_5)$, four binary genes for the kernel combination operators $(\otimes_1,...,\otimes_4)$, fifteen real valued genes for individual kernel parameters, e.g. (α, β, d) in Table 1, and one additional real valued gene for the regularization parameter. The overall genotype structure is shown in Fig. 2.

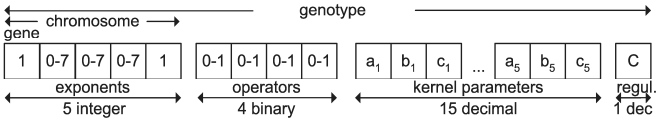

Fig. 2: Genotype encoding of SVM's combined kernel

We restrict the acceptable values for kernel exponent genes for computational reasons. In addition, these genes are superficial for polynomial and anova kernels that provide a kernel exponent as individual kernel parameter. Consequently, these genes have been set to one.

D. GA-based model selection

The GA-based development of SVMs is an iterative process starting with an initial population of randomly generated genotypes. Subsequently, SVMs are constructed by transferring the genotype's genetic code into a phenotype, i.e. a

SVM with a well defined combined kernel. After learning and (cross-)validation, each SVM is evaluated by the fitness function. Genetic operations use this quality information for building a new population of SVMs, which are trained and evaluated again. Thus, the whole learning process can be seen as subdivided into a microscopic cycle for learning of a SVM and a macroscopic evolutionary one; see Fig. 3.

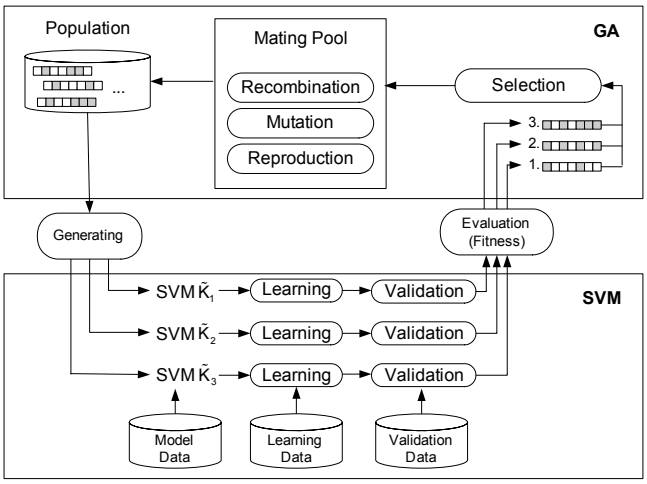

Fig. 3: Evolution of SVM by means of GA. Decoding of genotype into SVM is accomplished using the relationship between (9) and Fig. 2; here denoted as model data.

The fitness function is an important factor for evaluation and evolution of SVMs providing satisfactory and stable results in real-world applications. The fitness function guides the superordinated evolutionary learning process determining the probability that an individual can hand down genetic information to the subsequent population. Therefore, it should express the user's objective and should favour SVMs with satisfactory generalization ability in order to select useful classifiers systematically instead of accidentally. Consequently, the fitness function effectively conducts model selection and we can incorporate arbitrary model selection criteria as fitness measure.

Whereas the fitness function selects solutions for reproduction, the reproduction itself is conducted by means of mutation and crossover. The selection is implemented as tournament selection with a tournament size of two. Furthermore, an elitist mechanism is applied in order to ensure that the best SVM is member of the next generation.

The crossover operator is implemented as uniform crossover, i.e. all genes between two random points within a chromosome are interchanged between two genotypes representing parents for the resulting two new genotypes. Crossover is potentially applied to chromosomes for kernel aggregation and kernel exponent, whereas mutation can be applied to all chromosomes. The actual application of a genetic operation depends on user-defined rates. A high rate for crossing over and low rate for mutation are recommended. We set the crossover rate to 0.7 and the mutation rate for one gene to 0.3; see e.g. [26, 32]. Mutation is implemented as a stepwise increment or decrement with specific step size resulting in a new value within minimum and maximum limits. Binary genes are mutated by flipping 0 to 1 and vice versa.

V. Empirical Evaluation of GA-based Model Selection for SVM

A. Overview

We evaluate four data sets from the Statlog project and the UCI machine learning library. The data sets Australian credit (ac) and German credit (gc) exemplify a case of corporate credit scoring, e.g. classifying if an applicant is a good/bad credit risk. As examples for medical diagnosis we consider the data sets heart-disease (hrt), and Wisconsin breast cancer (wbc) each of which require a classification if a patient suffers from a certain disease or not. All sets are cases of binary classification so that examples either belong to a class +1 or a class -1 respectively. A brief description of each data set's characteristic is given in Table 2. For detailed information the reader is referred to [33-35].

Table 2:
Data set characteristics*

	#cases	#features	#class -1	#class +1
ac	690	14	307	383
gc	1000	20	700	300
hrt	270	13	150	120
wbc	683	10	239	444

* We use the pre-processed versions of the data sets available via the LIBSVM homepage [35].

The data sets have been partitioned into 2/3 training set for model building and 1/3 test set for out-of-sample evaluation. For each data set, the GA is used to construct a population of 50 individual SVMs. The evolutionary process of classifier assessment and fitness based recombination is run for 50 generations resulting in an overall number of 2,500 learned and evaluated SVMs per data set.

To consider empirical model selection procedures and algebraic bounds in a mutual framework we evaluated two different fitness criteria. In GA-1 fitness is measured by means of 10-fold CV balanced classification accuracy (*bca*) (10) whereas the bound (8) is used in GA-2. The *bca* is calculated as:

$$bca = \frac{1}{2}\left(\frac{\pi^-}{m^-} + \frac{\pi^+}{m^+}\right), \tag{10}$$

where m^- denotes the number of class -1 records in the data set and π^- the number of class -1 records that have been classified correctly with similar meanings for π^+ and m^+.

Results for GA-1 and GA-2 are contrasted with standard SVMs with linear, radial and polynomial kernel. Model selection for the standard SVMs is accomplished by means of extensive grid search, see Table 3.

TABLE 3:
PARAMETER RANGE FOR GRID SEARCH WITH STANDARD SVM *

	log(C)	d	log(α)	log(β)
Linear kernel	{-2,-1,…,3}	-	-	-
Radial kernel	{-2,-1,…,3}	-	{-2,-1,…,3}	-
Polynomial kernel	{-2,-1,…,2}	{2,3,4,5}	{-1,0,1}	{0,1,2}

*All parameters except the kernel exponent d for the polynomial kernel are varied on log scale. A minus sign indicates that the respective parameter is not present for the particular kernel.

B. Experimental Results

Following the idea of GA-based SVM model selection one chooses the individual with maximum overall fitness for future use on unseen data. To simulate this scenario, we assessed the performance by means of *bca* of the respective SVMs, e.g. the fittest member in the population, on the hold-out test set. To consider dynamical aspect of the GA, like the evolution of fitness and test performance, we report results on an aggregated generation level in Table 4 for GA-1 and Table 5 for GA-2 respectively.

TABLE 4:
RESULTS FOR GA-1 BASED MODEL SELECTION *

GA-1		GA		Standard SVM		Deviation
	Gen.	Best fitness	*bca* on test	Best fitness	*bca* on test	between GA and standard SVM
ac	10	0.8878	***0.8376***	0.8761	0.7993	4.79%
	25	0.8903	*0.8376*			4.79%
	50	0.8903	*0.8376*			4.79%
gc	10	0.6719	0.5752	0.6794	0.6629	-13.23%
	25	0.6853	***0.6784***			2.34%
	50	0.6903	0.5611			-15.36%
wbc	10	0.9753	***0.9743***	0.9750	0.9659	0.87%
	25	0.9758	*0.9743*			0.87%
	50	0.9767	*0.9743*			0.87%
hrt	10	0.8592	0.7785	0.8611	**0.7997**	-2.65%
	25	0.8647	0.7810			-2.34%
	50	0.8770	0.7744			-3.16%

* Results are provided on an aggregated generation level. That is, the fittest individual within the first 10, 25, and 50 generations is selected and evaluated on the test set simulating a scenario where the GA is stopped after the respective number of iterations. We use bold letters to denote the classifier that performs best on test data (with lower number of iterations, if performances are equal). In addition, italic letters indicate that SVMs with a combined kernel outperform standard SVM.

Results for standard SVM are given for comparison purpose. These have been computed using the grid search approach of Table 3 and selecting the model within maximum overall performance. Here, performance is defined in the sense of 10-fold CV *bca* on training data (Table 4) and bound (8) (Table 5) mimicking the behaviour of GA-1 and GA-2.

Using the algebraic bound (8) as fitness criterion, the GA-based SVM outperforms standard SVM on all considered data sets whereas it fails to find a superior model on the heart data set when using the empirical estimator. Similarly, the deviation between test performance of GA-based SVMs and standard SVMs appears more favorable for GA-2. However, differences between GA-1 and GA-2 in absolute performance values on test data are minor so that we conclude that both are appropriate fitness criteria for GA.

TABLE 5:
RESULTS FOR GA-2 BASED MODEL SELECTION *

GA-2		GA		Standard SVM		Deviation
	Gen.	Best fitness	*bca* on test	Best fitness	*bca* on test	between GA and standard SVM
ac	10	0.7957	***0.8034***	0.7782	0.7292	10.18%
	25	0.8065	*0.7401*			1.49%
	50	0.8152	*0.7344*			0.71%
gc	10	0.6712	***0.6236***	0.6441	0.5853	6.54%
	25	0.6787	0.6192			5.79%
	50	0.6922	0.5480			-6.37%
wbc	10	0.9186	***0.9694***	0.9269	0.9646	0.50%
	25	0.9457	0.9528			-1.22%
	50	0.9520	0.9444			-2.09%
hrt	10	0.7937	***0.7810***	0.6825	0.7400	5.54%
	25	0.7937	*0.7810*			5.54%
	50	0.7937	*0.7810*			5.54%

* see Table 4.

Noteworthy, for both GA-1 and GA-2 we observe a trend to overfit the data when running for a large number of generations. Due to our elitist selection the fitness increases monotonically from generation to generation. Though, selecting a model after 50 generations is always equal or inferior, in the sense of final performance achieved on test data, to selecting a model in an earlier stage of the evolutionary process. While these differences are negligible for the medical data sets the performance drop-off is serious for ac (6.9% for GA-2) and gc (11.7% GA-1). To clarify on this issue we analyse the relationship between fitness and performance on hold-out data in more detail using generalization diagrams as shown exemplary for GA-1 on ac in Fig. 4.

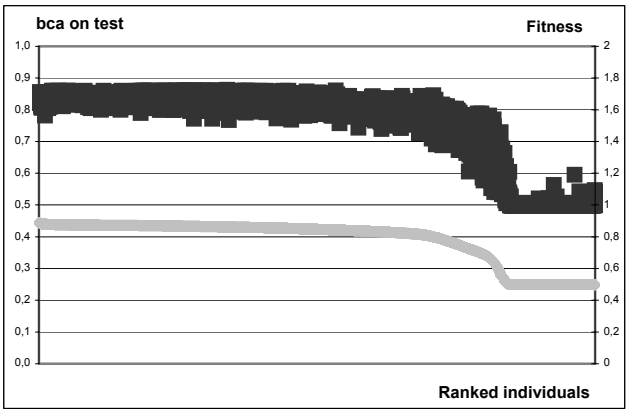

Fig. 4: Generalization diagram for GA-1 on ac showing all individual SVMs over all generations ranked by their fitness (grey squares) with according *bca* on test set (black squares). Note that fitness and test performance are scaled differently on individual axis to improve readability.

The diagram reveals that GA-1 provides excellent model selection capabilities for this particular data set. Individuals with high fitness exhibit similarly high test set performance

so that fitness based model selection will produce reliable classifiers with good generalization performance. Conducting this analysis over all data set revealed that GA-1 exceeds GA-2 in terms of correlation between fitness and generalization performance on average.

At the right side of Fig. 4 we observe a clear fitness drop-off. The test performance reaches a constant level of 0.5. This is explained by the fact, that the respective classifiers become naïve, predicting only one class for all instances. We refrained from incorporating prior knowledge into the GA, e.g. what kernel types/parameters to avoid for a given data set, range of the regularization parameter, etc., striving for a generic model selection mechanism. Equipping the algorithm with maximum flexibility allowed the construction of accurate and generalizable classifiers but at the cost that a certain amount of the derived models become futile. While extensive grid search usually leads to a number of naïve predictors as well, we analyze the ratio of naïve SVMs to overall SVMs for the GA and grid search in Fig. 5 to find that the number of ineffective models is in fact larger for the GA-based approach and GA-2 in particular.

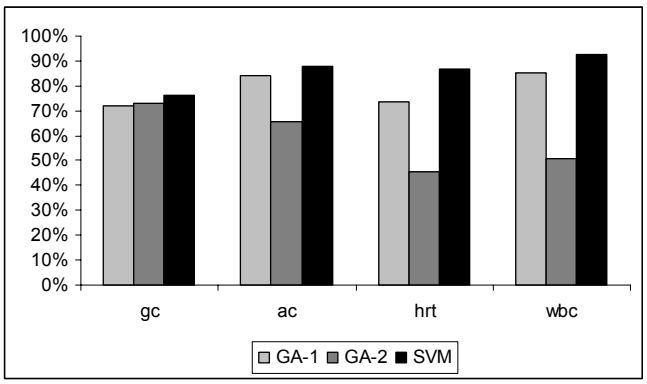

Fig. 5: Ratio of non-naïve models for GA-1, GA-2 and standard SVM.

This analysis explains our previous finding regarding the superiority of GA-1 in terms of generalization ability. While GA-1 and GA-2 are both promising for the task of selecting one best model out of a large candidate list, GA-1 is superior for steering the process of SVM kernel construction leading to a larger number of suitable classifiers on average.

VI. Conclusions

In this paper, we developed a GA-based approach to automate the task of model selection for the SVM classifier. This involved the construction of a combined kernel and the tuning of all resulting parameters. Requiring an appropriate fitness criterion for the GA we evaluated the well known CV performance on training data as an empirical model selection criterion. On the other hand, the minimization of algebraic bounds is well established within the SVM community facilitating model selection without re-sampling and re-training. Comparing these two model selection measures in the context of GA-based SVM parameterization we found that both are appropriate to choose a classifier that will generalize well to unknown data. However, model selection aims at finding only one classifier and from a GA perspective the empirical estimate of generalization performance is the better choice to guide the evolutionary process of SVM construction. Using the support vector bound (8) as fitness criterion delivered a larger number of futile classifiers decreasing reliability on average. To overcome this shortcoming, partly present in GA-1 as well, we will develop GAs that incorporate prior knowledge regarding SVM kernels and parameters, e.g. tuning heuristics like [16], in further research. However, such approaches will come at the cost of sacrificing generality and dissociate from the appealing vision of automatic model selection.

References

[1] N. E. Ayat, M. Cheriet, and C. Y. Suen, "Automatic model selection for the optimization of SVM kernels," *Pattern Recognition*, vol. 38, pp. 1733-1745, 2005.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proc. of the 5th Annual Workshop on Computational Learning Theory*, Pittsburgh, Pennsylvania, USA, 1992.

[3] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines," *Machine Learning*, vol. 46, pp. 389-422, 2002.

[4] J. Zhang and Y. Liu, "Cervical Cancer Detection Using SVM Based Feature Screening," *Proc. of the 7th Medical Image Computing and Computer-Assisted Intervention*, Saint-Malo, France, 2004.

[5] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. of the 10th European Conf. on Machine Learning*, Chemnitz, Germany, 1998.

[6] G. Guo, S. Z. Li, and K. L. Chan, "Support vector machines for face recognition," *Image and Vision Computing*, vol. 19, pp. 631-638, 2001.

[7] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," *Journal of the Operational Research Society*, vol. 54, pp. 627-635, 2003.

[8] S. Viaene, B. Baesens, T. Van Gestel, J. A. K. Suykens, D. Van den Poel, J. Vanthienen, B. De Moor, and G. Dedene, "Knowledge discovery in a direct marketing case using least squares support vector machines," *International Journal of Intelligent Systems*, vol. 16, pp. 1023-1036, 2001.

[9] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press, 2000.

[10] V. N. Vapnik, *Estimation of Dependences Based on Empirical Data*. New York: Springer, 1982.

[11] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[12] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

[13] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text Classification using String Kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419-444, 2002.

[14] A. Lunts and V. Brailovskiy, "Evaluation of attributes obtained in statistical decision rules," *Engineering Cybernetics*, vol. 3, pp. 98-109, 1967.

[15] T. van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. de Moor, and J. Vandewalle, "Benchmarking Least Squares Support Vector Machine Classifiers," *Machine Learning*, vol. 54, pp. 5-32, 2004.

[16] S. S. Keerthi and C.-J. Lin, "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel," *Neural Computation*, vol. 15, pp. 1667-1689, 2003.

[17] T. Joachims, "Estimating the Generalization Performance of an SVM Efficiently," *Proc. of the 17th Intern. Conf. on Machine Learning*, Stanford, CA, USA, 2000.
[18] O. Chapelle and V. Vapnik, "Model selection for support vector machines," *Proc. of the 13th Annual Conference on Neural Information Processing Systems*, Denver, CO, USA, 2000.
[19] K.-M. Chung, W.-C. Kao, L.-L. Wang, and C.-J. Lin, "Radius Margin Bounds for Support Vector Machines with RBF kernel," *Neural Computation*, vol. 15, pp. 2643-2681, 2003.
[20] K. Duan, S. S. Keerthi, and A. N. Poo, "Evaluation of simple performance measures for tuning SVM hyperparameters," *Neurocomputing*, vol. 51, pp. 41-59, 2003.
[21] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing Multiple Parameters for Support Vector Machines," *Machine Learning*, vol. 46, pp. 131-159, 2002.
[22] S. S. Keerthi, "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1225-1229, 2002.
[23] S. Boughorbel, J. P. Tarel, and N. Boujemaa, "The LCCP for Optimizing Kernel Parameters for SVM," *Prof. of the 15th Intern. Conf. on Artificial Neural Networks*, Warsaw, Poland, 2005.
[24] F. Friedrichs and C. Igel, "Evolutionary Tuning of multiple SVM parameters," *Neurocomputing*, vol. 64, pp. 107-117, 2005.
[25] H.-N. Nguyen, S.-Y. Ohn, and W.-J. Choi, "Combined Kernel Function for Support Vector Machine and Learning Method Based on Evolutionary Algorithm," *Proc. of the 11th Intern. Conf. on Neural Information Processing*, Calcutta, India, 2004.
[26] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading: Addison-Wesley, 1989.
[27] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, 6 ed. Cambridge: MIT Press, 2001.
[28] L. Li, W. Jiang, X. Li, K. L. Moser, Z. Guo, L. Du, Q. Wang, E. J. Topol, Q. Wang, and S. Ra, "A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset," *Genomics*, vol. 85, pp. 16-23, 2005.
[29] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," *Mechanical Systems and Signal Processing*, vol. 18, pp. 625-644, 2004.
[30] S.-Y. Ohn, H.-N. Nguyen, and S.-D. Chi, "Evolutionary Parameter Estimation Algorithm for Combined Kernel Function in Support Vector Machine," *Proc. of the Advanced Workshop on Content Computing*, ZhenJiang, JiangSu, China, 2004.
[31] S.-Y. Ohn, H.-N. Nguyen, D. S. Kim, and J. S. Park, "Determining optimal decision model for support vector machine by genetic algorithm," *Proc. of the 1st Intern. Symposium on Computational and Information Science*, Shanghai, China, 2004.
[32] S. Bhattacharyya, "Direct Marketing Response Models using Genetic Algorithms," *Fourth International Conference on Knowledge Discovery and Data Mining*, New York, 1998.
[33] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, *Machine learning, neural and statistical classification*. New York: Horwood, 1994.
[34] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "UCI Repository of machine learning databases," Department of Information and Computer Science, University of California, Irvine, CA, 1998.
[35] C.-C. Chang and C.-J. Lin, "LIBSVM - A Library for Support Vector Machines," software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.