Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference

Bias and Variance Optimization for SVMs Model Selection

Alejandro Rosales-Pérez, Hugo Jair Escalante, Jesus A. Gonzalez and **Carlos A. Reyes-Garcia**
Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Santa Maria Tonantzintla, Puebla, Mexico
{arosales,hugojair,jagonzalez,kargaxxi}@inaoep.mx

Abstract

Support vector machines (SVMs) are among the most used methods for pattern recognition. Acceptable results have been obtained with such methods in many domains and applications. However, as most learning algorithms, SVMs have hyperparameters that influence the effectiveness of the generated model. Thus, choosing adequate values for such hyperparameters is critical in order to obtain satisfactory results for a given classification task, a problem known as model selection. This paper introduces a novel model selection approach for SVMs based on multi-objective optimization and on the bias and variance definition. We propose an evolutionary algorithm that aims to select the configuration of hyperparameters that optimizes a trade-off between estimates of bias and variance; two factors that are closely related to the model accuracy and complexity. The proposed technique is evaluated using a suite of benchmark data sets for classification. Experimental results show the validity of our approach. We found that the model selection criteria resulted very helpful for selecting highly effective classification models.

Introduction

A support vector machine (SVM) (Cortes and Vapnik 1995) is a supervised learning algorithm able to build a classification model from a labeled data set. The underlying idea of SVMs is to find the hyperplane that maximizes the separation of examples from two classes. SVM has become a quite popular method in classification and regression tasks, mainly due to its high performance and scalability. Nonetheless, a SVM has some adjustable parameters, usually called hyperparameters (Guyon et al. 2010), that can affect its performance. Thus, determining the adequate hyperparameter values is needed, this problem is usually known as *model selection*.

Model selection is the task of choosing the model that best describes a data set (Hastie, Tibshirani, and Friedman 2009). Therefore, the model selection seeks for hyperparameters values that maximize the generalization performance of the associated SVM. The generalization error of a classifier can be decomposed in two terms: squared bias and variance, components that are closely related to accuracy and model complexity. In general, bias describes the extent to which the systematic error of the learning algorithm contributes to the generalization error of the model, while variance describes the extent to which variations in the training data or a random behavior of the learning algorithm contributes to the error. So, minimizing both components is important in order to select a model that performs well on unseen data. However, these two components are in conflict, and minimizing one of them causes an increase in the other one. In this sense, the model selection problem can be seen as a multi-objective optimization problem.

Previous studies have tackled the SVM model selection task using evolutionary algorithms. In (Huaitie et al. 2010), the authors propose a combination of genetic algorithms and simulated annealing to choose the parameters for a RBF kernel function for an SVM. Multi-objective approaches have also been proposed. In (Chatelain et al. 2007; 2010; Ethridge, Ditzler, and Polikar 2010; Li, Liu, and Gong 2011), the authors propose to optimize the specificity and sensitivity as objectives, tackling the problem of model selection for unbalanced data sets (i.e. preventing to choose a model that performs well for one class but not for the other one). Other works have considered the accuracy and the number of support vectors as the objectives to optimize (Suttorp and Igel 2006; Ayd), under the assumption that the number of support vectors is associated to the model complexity. The previous works have taken into account the parameters selection for one kind of kernel function and they do not perform the kernel type selection. To the best of our knowledge, estimated values of bias and variance have not been previously used in a multi-objective approach for model selection. In this paper we face the problem of model selection for SVMs as a multi-objective optimization task. We propose a multi-objective evolutionary algorithm for model selection that simultaneously minimizes estimates of bias and variance, which are approximated from a finite data set. We used the NSGA-II (Deb et al. 2000) algorithm as search strategy because of its efficiency and because it could provide diverse solutions that satisfy a trade-off between these two components. We evaluated our approach using a suite of benchmark data sets for classification. Experimental results show that the proposed approach selects highly effective classification models, when it is compared to an SVM without performing hyperparameters selection, and a related method for model selection.

Multi-objective optimization problem

A multi-objective optimization problem (MOOP), is the problem to find a solution that minimizes (or maximizes) two (or more) objectives that are usually in conflict (i.e. finding a solution that would give acceptable values for the objectives). According to Deb (Deb 2001), a MOOP can be stated as:

$$\begin{array}{ll} \text{minimize} & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_l(\mathbf{x})] \\ \text{subject to} & g_i(\mathbf{x}) \leq 0 \; i = 1, \ldots, p \\ & h_j(\mathbf{x}) = 0 \; j = 1, \ldots, q \end{array}$$

where $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbb{R}^n$ is a n-dimensional variable decision vector, l is the number of objectives, p the number of inequality constrains, and q is the number of equality constrains.

Most of the multi-objective optimization algorithms are based on the **dominance** concept to determine if a solution is better than another. We say that a solution $\mathbf{x}^{(1)}$ dominates a solution $\mathbf{x}^{(2)}$ ($\mathbf{x}^{(1)} \preceq \mathbf{x}^{(2)}$) if and only if $\mathbf{x}^{(1)}$ is better than $\mathbf{x}^{(2)}$ at least in one objective and it is not worse in the rest (Coello, Lamont, and Veldhuizen 2007; Deb 2001).

Generally, most of the multi-objectives problems do not have an unique solution, but a set of solutions. This set of solutions satisfies a trade-off between the different objectives being optimized. In order to establish this trade-off, the most accepted notion of optimum in MOOP is the so called Pareto optimal. Formally, the notion of **Pareto optimal** says that a solution $\mathbf{x}^* \in \mathbb{R}^n$ is a Pareto optimal if and only if $\nexists \mathbf{x} \in \mathbb{R}^n$, for which $\mathbf{x} \preceq \mathbf{x}^*$ (Coello, Lamont, and Veldhuizen 2007). This definition says that a solution $\mathbf{x}^*$ is a Pareto optimal if there does not exist another solution such that improving one objective causes any other objective to worsen. It is important to note that this definition does not produce a single solution, but a set of trade-off solutions between the different objectives. The set of trade-off solutions is known as **Pareto optimal set**. The vectors included in the Pareto optimal set are called **non-dominated** solutions. The plot of values of the objective functions which are non-dominated vectors in the Pareto optimal set is called **Pareto front**. Several techniques have been proposed for solving a MOOP, such as weighted sum, ϵ-constrains and evolutionary algorithms, the latter have shown an advantage over classical techniques (Coello, Lamont, and Veldhuizen 2007) and one of these is used in this work.

Bias-Variance Estimation

The ultimate goal of model selection in classification tasks is that of finding a model that obtains the highest possible generalization performance. That is, a model that guarantees to obtain a low error rate on samples that were not seen during the training process and that come from the same distribution than the training set. Generalization error of a classifier can be decomposed into the squared-bias and variance terms (Hastie, Tibshirani, and Friedman 2009):

$$\begin{aligned} err\left(f_D(x)\right) = &\left\{E_D\left[f_D(x)\right] - f(x)\right\}^2 + \\ &E_D\left[\left\{f_D(x) - E_D\left[f_D(x)\right]\right\}^2\right] \end{aligned} \tag{1}$$

where $f(x)$ is the target function (the desired output), $f_D(x)$ is the model trained with the data set D and $E_D[\cdot]$ is the expected value taken from all data sets D.

A number of studies have been addressed to extend the bias and variance decomposition into the classification field (Kong and Dietterich 1995; Kohavi and Wolpert 1996; Friedman 1997; Webb 2000).

Each of those definitions is able to provide information about the model's performance, giving insights of how much the bias and the variance contribute to the model error. In our study we adopted the Kohavi and Wolpert's definition (Kohavi and Wolpert 1996), because is close to the bias/variance decomposition formulated for regression tasks, and is one of the most used (Webb and Conilione 2005). The values for bias and variance can be estimated using sampling techniques, such as cross-validation, bootstrapping, etc.

In classification tasks, square bias is a measure of the contribution to the error of the central tendency (i.e. the class with the most votes across the multiple predictions) when a model is trained with different data sets. The variance is a measure of the deviations to the central tendency when a model is trained with different data sets (Webb 2000).

Multi-objective evolutionary algorithm

Evolutionary algorithms are heuristic search techniques inspired in Darwin's evolutionary theory. These kind of algorithms are based on the idea of the survival of the fittest individual where stronger individuals have a higher chance of reproduction. Generally, an evolutionary algorithm has five basic components: an encoding scheme, in a form of chromosomes or individuals, that represents the potential solutions to the problem, a form to create potential initial solutions, a fitness function to measure how close a chromosome is to the desired solution, selection operations and operators for selection and reproduction.

Evolutionary algorithms have been used for solving multi-objective problems. The main advantage of using this kind of algorithms is that they obtain several points in the Pareto front in a single run. Different evolutionary algorithms have been proposed for multi-objective optimization, including: Distance-based Pareto Genetic Algorithm (DPGA) (Osyczka and Kundu 1995), Niched-Pareto Genetic Algorithm (NPGA) (Horn, Nafpliotis, and Goldberg 1994), Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele 1999), Pareto Archived Evolution Strategy (PAES) (Knowles and Corne 2000), Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al. 2000), etc. A comprehensive review of evolutionary techniques for solving multi-objective problems can be found in (Coello, Lamont, and Veldhuizen 2007; Deb 2001).

In this work, we used the NSGA-II[1], which is an elitist genetic algorithm, that uses a crowding distance to preserve the diversity in the solutions. A general description of NSGA-II is presented in Algorithm 1. As most genetic algorithms, NSGA-II creates an offspring population, O_t, from a parent population, P_t. Nonetheless, this algorithm combines both

[1] An implementation of this algorithm in Matlab is available in http://delta.cs.cinvestav.mx/˜ccoello/EMOO/NSGA-II-Matlab.zip

populations, $T_t = O_t + P_t$, and the individuals are sorted based on non-dominance. Note that the size of T_t is $2N$, but the size of the new population, P_{t+1}, should be N. P_{t+1} is formed from the non-dominated fronts. This process begins adding the first non-dominated front to P_{t+1}, followed by the second front, and so on until the population has at least N individuals. The fronts that were not added are deleted. If the P_{t+1} population size is greater than N a niche strategy is used for choosing the individual of the last added front to be part of P_{t+1}.

Algorithm 1 NSGA-II (Deb et al. 2000)

```
Require: $N$ (number of individuals),
  $f$ (fitness functions),
  $g$ (number of generations in the evolutionary process)
  Initialize population $P_t$
  Evaluate objective functions
  Assign rank based on Pareto dominance
  for $t = 1 \rightarrow g$ do
    Generate child population $Q_t$
      Binary tournament selection
      Evolutionary operations
    for each parent and child in population do
      Assign rank based on Pareto dominance
      Generate set of non-dominate vectors
      Add solutions to next generation starting from the first
      front until individuals found determine crowding distance
      between points on each front
    end for
    Apply elitism over the lower front and those outside a crowd-
    ing distance
    Create next generation
  end for
```

SVM Model Selection

The generalization performance of an SVM is highly influenced by the choice of its hyperparameters. Therefore, in order to obtain acceptable performance in a given classification task, hyperparameters must be chosen appropriately. We propose a multi-objective evolutionary algorithm for selecting the kernel function together with its hyperparameters for SVM classifiers by minimizing the bias and variance. In general, it is said that a low bias is associated with a low error in the training set, but the model could be *overfitted*; in contrast, a low variance is associated with a low model complexity, and the model could be *underfitted*. With this premise, we believe that if both components are minimized, models with a good generalization ability can be obtained. Thus, we faced the model selection task as a multi-objective optimization problem using as objectives estimates of bias and variance, trying to select the model with the best trade-off between both components.

The proposed approach is as follows: given a labeled data set, we divide it into two different sets called training set and validation set. The training set is used to fit the parameters of the model during the hyperparameters space exploration. The NSGA-II algorithm is used as a search strategy for the exploration task. Once the search process is completed, a set of trade-off solutions is obtained, this set is called the Pareto optimal set. Each solution in the Pareto optimal set satisfies to some extent a trade-off between our objectives, that is bias and variance. The next step is to choose a final model to be used for classification. One could argue to use all solutions in an ensemble. However, since solutions with high variance (which are highly probable of being overffited) and solutions with high bias (which are highly probable of being underfitted) are contained in that Pareto optimal set (see Figure 1), they could affect the ensemble performance. Therefore, just one solution is chosen from the Pareto optimal set. We use a validation set to test each model in the Pareto optimal set. We select the solution with the lowest error rate in the validation set. Then, the model is trained using both, a training set and a validation set.

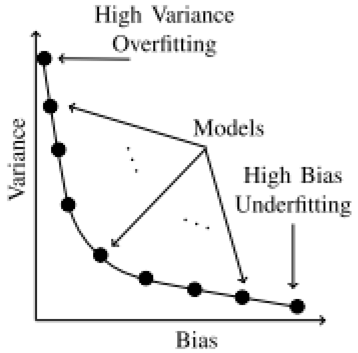


Figure 1: Several points in the Pareto front represent a trade-off between the bias and variance of the model. Models with high bias could be underfitted while models with high variance could be overfitted.

0-3	0-∞	0-∞	0-∞
kernel	n	γ	C_0

Figure 2: Codification schema to represent a SVM model

A multi-objective evolutionary algorithm requires a way to codify solutions, a way to evaluate the individuals fitness, and evolutionary operators. The rest of this section provides a detailed description of the components of the proposed evolutionary algorithm.

Representation

Evolutionary algorithms require a codification to represent the potential solutions for the optimization problem. This codification is usually called chromosome or individual. For the purpose of our study, the individual codifies the SVM hyperparameters with a 4D numerical vector, see Figure 2.

The kernel parameter can take values between 0 and 3. This is an integer value and represents the type of kernel function, according to the ID presented in Table 1. n, γ and $C0$ are the hyperparameters for the kernels, according to the mathematical expression shown in Table 1. Note that the n, γ and $C0$ hyperparameters can take very large values (infinity, in theory), for computational reasons these values are initially limited, although during the evolutionary process they could be increased.

We used Matlab to implement the proposed method. We used the SVM implementation from the LIBSVM (Chang and Lin 2011) package.

Fitness Function

Bias and variance are the objectives to be minimized. Since only a finite sample of data is available for the model selection process, it is only possible to obtain approximations to the bias and variance of a model. For estimating the bias and variance, we used the Kohavi and Wolpert's definition for the classification task, because it is one of the closest to the bias and variance decomposition for regression task, and it is one of the most used (Webb and Conilione 2005). Under this definition, bias and variance are computed as follows:

$$bias^2 = \frac{1}{2}\sum_{y\in Y}\left[P_{Y,X}\left(Y_F = y \mid X = x\right) - P\left(Y_H = y\right)\right]^2$$

$$var = \frac{1}{2}\left(1 - \sum_{y\in Y} P_D\left(Y_H = y\right)^2\right)$$

where Y is the set of output classes, Y_F is the fixed function that maps each sample x to a class y, and Y_H is a hypothesis estimating Y_F.

In order to estimate the bias and variance values, $n \times k$-fold cross validation is used. We fixed the values of n to ten, and k to three, as it was employed by Webb (Webb 2000). In each three-fold cross validation, the data set is randomly divided into three disjoint subsets. A subset is used for testing once, and the rest for training, and this process is repeated three times. So, each sample is classified one time, and the three-fold cross validation process is repeated ten times. Therefore, each sample is classified ten times, and these classifications are used to compute the probabilities used in the above expressions to approximate bias and variance.

Computational Issues

Our approach can be considered a wrapper method. Wrapper methods explore the hyperparameters space and evaluate several models in order to select the best one. We used $n \times k$ fold cross validation to estimate the bias and variance values for our fitness function. Thus, the model has to be trained and tested several times in order to determine the fitness of the model. This causes that the fitness function evaluation to be computationally expensive.

Table 1: Different kernels types used with SVM, where u and v are training vectors and n, γ and C_0 are the kernel parameters.

ID	Name	kernel
0	Linear	$u' \cdot v$
1	Polynomial	$(\gamma * u' \cdot v + C0)^n$
2	RBF	$e^{-\gamma\cdot\lvert u-v\rvert^2}$
3	Sigmoid	$\tanh\left(\gamma \cdot u' \cdot v + C0\right)$

Table 2: Data sets used in our experiments. Each data set has 100 partitions for training and testing, except Splice and Image data set with 20 partitions.

ID	Data set	Feat.	Training Samples	Testing Samples	Replications
1	Banana	2	400	4900	100
2	BC	9	200	77	100
3	Diabetes	8	468	300	100
4	FS	9	666	400	100
5	German	20	700	300	100
6	Heart	13	170	100	100
7	Image	20	1300	1010	20
8	Ringnorm	20	400	7000	100
9	Splice	60	1000	2175	20
10	Thyroid	5	140	75	100
11	Titanic	3	150	2051	100
12	Twonorm	20	400	7000	100
13	Waveform	21	400	4600	100

Let l be the number of times that a specific model is trained and tested in the evaluation step, N is the number of evaluated models per generation, and g is the number of generations in the evolutionary algorithm, the number of trained models is given by $l \times N \times g$. Despite the computational cost, the model selection algorithm has the advantage of determining an adequate configuration for the SVM classifier without requiring a set of experiments manually performed for this purpose. The computational cost is the main drawback of the adopted approach, but we are actually working on ways to make it efficient.

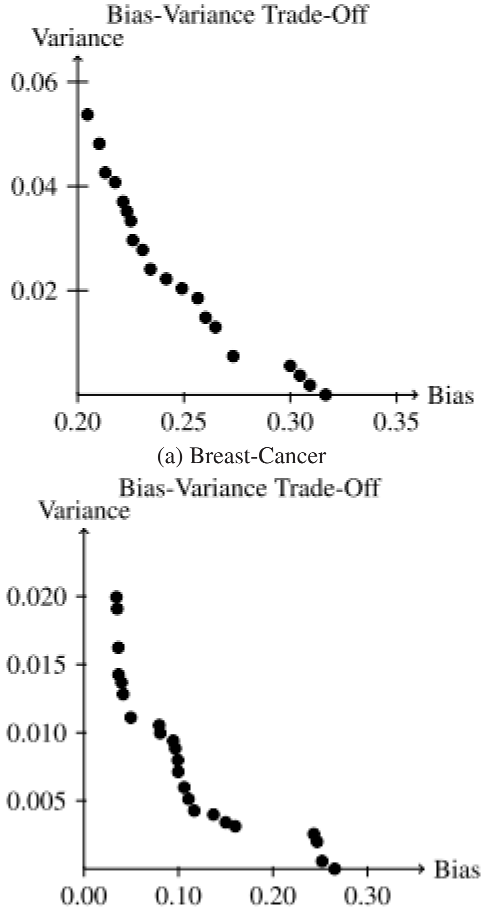
Experiments and results

We performed several experiments using a suite of benchmark data sets[2] described in Table 2. The data sets are diverse in terms of the number of features and samples and they have been used in several works (Rätsch, Onoda, and Müller 2001; Escalante, Montes, and Sucar 2009; Zhou and Xu 2009). For each data set, we randomly selected 10 partitions. For each trial, a population size equal to 25 and a number of generations equal to 50 are fixed.

Figure 3 shows the Pareto fronts obtained for some data sets in a particular trial. These plots show the trade-off that exists between the bias and variance. Each point plotted in the Pareto front represents the optimal solutions that were found by the NSGA-II algorithm. These solutions satisfy a trade-off between model bias and variance and they allow us to know the expected generalization error over new samples. Each of these solutions is evaluated using the validation data set to select the final model.

Table 3 shows the results obtained with our proposal, called MOSVMMS. We report the error rates and the standard deviation of ten replications obtained in the test set for each data set. These results are compared with those of the standard SVM (i.e. a SVM with the default hyperparam-

[2] These data sets are available in http://theoval.cmp.uea.ac.uk/matlab/benchmarks/



(a) Breast-Cancer

(b) Image

Figure 3: Obtained Pareto fronts from a particular trial of the proposed method.

eters), and of with PSMS (Escalante, Montes, and Sucar 2009), which is a full model selection method that has obtained an acceptable performance over data sets from different domains. PSMS uses particle swarm optimization (PSO) for selecting a combination of feature selection method, pre-processing method, learning algorithm and the associated hyperparameters. In order to make a fair comparison of our results, we fixed the learning algorithm to SVM and feature selection and pre-processing methods were not selected, we used the same ten partitions for both approaches.

From Table 3, we can observe that the multi-objective SVM model selection (MOSVMMS) obtained lower error rates for most of the data sets, and it was worse for four data sets only. Demšar (Demšar 2006) recommends the Wilcoxon Signed Rank test to compare two classifiers over different data sets. We applied this statistical test with 95% of confidence to compare our obtained results with those obtained using just a SVM and those using PSMS. The statistical test showed that MOSVMMS outperforms significantly to other approaches in the *banana, image, ringnorm, splice, thyroid, twonorm* and *waveform* data sets, when it is compared to the SVM without performing hyperparameters selection. With respect to PSMS, a related method in the state of the art, there was a statistical significance difference for the *heart* and *twonorm* data sets. For the *breast-cancer, flare-solar, german* and *titanic* data sets, in which our proposal was outperformed, the statistical test did not show that the differences were statistically significant. When we evaluated them over all data sets, according to the Wilcoxon Signed Rank test, the proposed method performs significantly to better than SVM without hyperparameters selection and PSMS.

Table 3: Comparison of the multi-objective SVM model selection (MOSVMMS), with SVM (with default parameters) and PSMS. The reported results are the error rates averages over ten trials for each data set, and the best result is shown in bold.

ID	SVM	PSMS	MOSVMMS
1	46.11 ± 3.64	10.81 ± 0.64	**10.69 ± 0.56**
2	**29.87 ± 3.77**	31.95 ± 3.93	30.39 ± 7.30
3	23.17 ± 1.69	27.73 ± 1.95	**23.10 ± 1.94**
4	**32.73 ± 1.63**	32.80 ± 1.50	32.98 ± 1.86
5	**23.60 ± 2.22**	25.80 ± 3.98	24.30 ± 2.67
6	17.90 ± 2.85	24.90 ± 10.73	**17.40 ± 2.80**
7	15.37 ± 1.01	3.90 ± 0.83	**3.42 ± 0.62**
8	24.75 ± 0.51	2.37 ± 2.20	**1.63 ± 0.11**
9	16.37 ± 0.85	12.78 ± 1.92	**11.70 ± 0.90**
10	11.60 ± 3.61	4.80 ± 2.82	**4.27 ± 2.72**
11	**22.51 ± 0.16**	22.81 ± 1.10	24.37 ± 5.86
12	3.57 ± 0.59	7.82 ± 14.88	**2.64 ± 0.28**
13	13.45 ± 0.63	12.08 ± 1.23	**10.53 ± 0.87**
Ave.	21.61 ± 1.78	16.66 ± 3.67	**15.19 ± 2.19**

Conclusions and Future Work

We presented a novel evolutionary multi-objective optimization approach for model selection of SVMs. Bias and variance estimates of a model are related to its accuracy and complexity. We propose using estimates of both terms as the objectives to be minimized in order to obtain models with an acceptable generalization performance. An advantage of the proposed method is that it can be applied to data sets from different domains as shown in our reported experiments. Since the bias and variance estimates are based on a cross-validation approach, our proposal does not depend of the model, thus it can be easily extended to other models than SVM and to the full model selection formulation.

Even though the computational work load, due to the intensive search to explore the hyperparameters space, we consider that it can be acceptable if we take into account that the final user does not have to deal with the selection of the values for the parameters of the SVM classifier.

Our experimental results showed an advantage of our proposal, when it was compared to a SVM without performing hyperparameters selection, and with a related method from the state of the art. Statistical significance tests showed that the difference was significant, giving evidence of the advantage of our proposal with respect to the other approaches considered for comparison.

Our current work is focused to study alternative methods to estimate the bias and variance of the models. We are also studying to extend our proposed method for different learning algorithms, that is, the method will be able to choose among different learning algorithms and their associated hyperparameters. As a future work, we want to study the effect of the population size and the number of generations in the quality of the selected model, as well as alternatives to reduce the computational cost of the model selection algorithm. We also want to study strategies to choose accurate and diverse solutions from the Pareto optimal set, perhaps considering an ensemble. Finally, we want to compare our obtained results with PSMS using all its components and with other multi-objective model selection approaches and to test our proposed method with high dimensional data sets.

Acknowledges

The first author was supported by CONCACyT under scholarship No. 329013.

References

A multi-objective artificial immune algorithm for parameter optimization in support vector machine. *Appl. Soft. Comput.*

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: A library for support vector machines. *ACM Trans Intell Syst Technol* 2(3):27:1–27:27.

Chatelain, C.; Adam, S.; Lecourtier, Y.; Heutte, L.; and Paquet, T. 2007. Multi-objective optimization for svm model selection. In *Ninth ICDAR*, volume 1, 427 –431.

Chatelain, C.; Adam, S.; Lecourtier, Y.; Heutte, L.; and Paquet, T. 2010. A multi-model selection framework for unknown and/or evolutive misclassification cost problems. *Pattern Recogn.* 43(3):815 – 823.

Coello, C. A. C.; Lamont, G. B.; and Veldhuizen, D. A. V. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer US.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Mach Learn* 20:273–297. 10.1007/BF00994018.

Deb, K.; Agrawal, S.; Pratap, A.; and Meyarivan, T. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In Schoenauer, M.; Deb, K.; Rudolph, G.; Yao, X.; Lutton, E.; Merelo, J.; and Schwefel, H.-P., eds., *PPSN*, volume 1917 of *LNCS*. Springer Berlin / Heidelberg. 849–858.

Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*. Wiley.

Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30.

Escalante, H. J.; Montes, M.; and Sucar, L. E. 2009. Particle swarm model selection. *J Mach Learn Res* 10:405–440.

Ethridge, J.; Ditzler, G.; and Polikar, R. 2010. Optimal v-svm parameter estimation using multi objective evolutionary algorithms. In *IEEE-CEC 2010*, 1 –8.

Friedman, J. H. 1997. On bias, variance, 0/1loss, and the curse-of-dimensionality. *Data Min Knowl Disc* 1:55–77.

Guyon, I.; Saffari, A.; Dror, G.; and Cawley, G. C. 2010. Model selection: Beyond the bayesian/frequentist divide. *J Mach Learn Res* 11:61–87.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York.

Horn, J.; Nafpliotis, N.; and Goldberg, D. 1994. A niched pareto genetic algorithm for multiobjective optimization. In *IEEE WCCI. P*, 82 –87 vol.1.

Huaitie, X.; Guoyu, F.; Zhiyong, S.; and Jianjun, C. 2010. Hybrid optimization method for parameter selection of support vector machine. In *IEEE-ICIS 2010*, volume 1, 613 –616.

Knowles, J., and Corne, D. 2000. Approximating the nondominated front using the pareto archived evolution strategy. *Evol Comp* 8(2):149–172.

Kohavi, R., and Wolpert, D. 1996. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the 13th ICML*, 275–283. Morgan-Kaufmann Publishers.

Kong, E., and Dietterich, T. 1995. Error-correcting output coding corrects bias and variance. In *Proceedings of the 12th ICML*, 313–321. Morgan-Kaufmann Publishers.

Li, W.; Liu, L.; and Gong, W. 2011. Multi-objective uniform design as a svm model selection tool for face recognition. *Expert Syst. Appl.* 38(6):6689 – 6695.

Osyczka, A., and Kundu, S. 1995. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Struct Multidiscip O* 10:94–99. 10.1007/BF01743536.

Rätsch, G.; Onoda, T.; and Müller, K.-R. 2001. Soft margins for adaboost. *Mach Learn* 42:287–320. 10.1023/A:1007618119488.

Suttorp, T., and Igel, C. 2006. Multi-objective optimization of support vector machines. In Jin, Y., ed., *Multi-Objective Machine Learning*, volume 16 of *Studies in Computational Intelligence*. Springer Berlin / Heidelberg. 199–220.

Webb, G., and Conilione, P. 2005. Estimating bias and variance from data. Technical report, .

Webb, G. I. 2000. Multiboosting: A technique for combining boosting and wagging. *Mach Learn* 40:159–196.

Zhou, X., and Xu, J. 2009. A svm model selection method based on hybrid genetic algorithm and empirical error minimization criterion. In Wang, H.; Shen, Y.; Huang, T.; and Zeng, Z., eds., *ISNN 2009*, volume 56 of *Advances in Intelligent and Soft Computing*. Springer Berlin / Heidelberg. 245–253.

Zitzler, E., and Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE T Evol Comput* 3(4):257 –271.