A multi-model selection framework for unknown and/or evolutive misclassification cost problems

Clément Chatelain, Sébastien Adam, Yves Lecourtier,
Laurent Heutte [*], Thierry Paquet

University of Rouen, LITIS EA 4108, BP12, 76801 Saint Etienne du Rouvray, FRANCE

Abstract

In this paper, we tackle the problem of model selection when misclassification costs are unknown and/or may evolve. Unlike traditional approaches based on a scalar optimization, we propose a generic multi-model selection framework based on a multi-objective approach. The idea is to automatically train a pool of classifiers instead of one single classifier, each classifier in the pool optimizing a particular trade-off between the objectives. Within the context of two-class classification problems, we introduce the "ROC front concept" as an alternative to the ROC curve representation. This strategy is applied to the multi-model selection of SVM classifiers using an evolutionary multi-objective optimization algorithm. The comparison with a traditional scalar optimization technique based on an AUC criterion shows promising results on UCI datasets as well as on a real-world classification problem.

Key words: ROC front, multi-model selection, multi-objective optimization, ROC curve, handwritten digit/outlier discrimination.

1 Introduction

Tuning the hyper-parameters of a classifier is a critical step for building an efficient pattern recognition system as this crucial aspect of model selection strongly impacts the generalization performance. In the literature, many contributions in this field have focused on the computation of the model selection criterion, *i.e.* the value which is optimized with respect to the hyperparameters. These contributions have led to efficient scalar criteria and strategies used

[*] Corresponding author: Email address: Laurent.Heutte@univ-rouen.fr

to estimate the expected generalization error. One can cite Xi-Alpha bound of [24], the Generalized Approximate Cross-Validation of [33], the empirical error estimate of [3], the radius-margin bound of [9] or the maximal-discrepancy of [2]. Based on these criteria, hyperparameters are usually chosen using a grid search, coupled with a cross-validation procedure. In order to decrease the computational cost of grid search, some authors suggest to use gradient-based techniques (e.g. [4], [25]). In these works, the performance validation function is adapted in order to be differentiable with respect to the parameters to be optimized.

All the approaches mentioned above, though efficient, use a single criterion as the objective during the optimization process. Now, it is well known that a single criterion is not always a good performance indicator. Indeed, in many real-world pattern recognition problems (medical domain, road safety, biometry, etc...), the misclassification costs are (i) asymmetric as error consequences are class-dependant; (ii) difficult to estimate (for example when the classification process is embedded in a more complex system) or subject to change (for example in the field of fraud detection where the amount of fraud changes monthly). In such cases, a single criterion might be a poor performance indicator.

One solution to tackle this problem is to use as performance indicator the *Receiver Operating Characteristics* (ROC) curve proposed in [6]. Such a curve offers a synthetic representation of the trade-off between the *True Positive* rate (TP) and the *False Positive* rate (FP), also known as *sensitivity* vs. *specificity* trade-off. One way to take into account both FP and TP in the model selection process is to resume the ROC curve into a single criterion, such as the F-Measure (FM), the Break-Even Point (BEP) or the Area Under ROC Curve (AUC). However, we will show in the following that we can get more advantages in formulating the model selection problem as a true 2-D objective optimization task.

In this paper, our key idea is to turn the problem of the search for a global optimal classifier (*i.e.* the best set of hyperparameters) using a single criterion or a resume of the ROC curve, into the search for a pool of locally optimal classifiers (*i.e.* the pool of the best sets of hyperparameters) w.r.t. FP/TP rates. The best classifier among the pool can then be selected according to the needs of some practitioner. Consequently, the proposed framework can be viewed as a multiple model selection approach (rather than a model selection problem) and can naturally be expressed in a Multi-Objective Optimization (MOO) framework. Under particular conditions, we assume that such an approach leads to very interesting results since it enables a practitionner to (i) postpone the choice of the final classifier as late as possible and (ii) to change the classifier without a computationally expensive new learning stage when target conditions change.

Figure 1 depicts our overall multi-model selection process. The resulting output of such a process is a pool of classifiers, each one optimizing some FP/TP rate trade-off. The set of trade-off values constitutes an optimal front we call "ROC front" by analogy with MOO field.

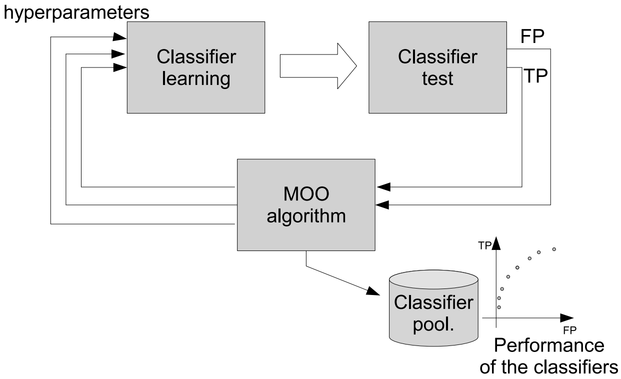


Fig. 1. Multi-model selection framework

The remainder of the paper is organized as follows. In section 2, we detail the rationale behind the ROC front concept and illustrate how our multi-model selection approach can outperform traditional approaches in a MOO framework. Section 3 gives an overview of Multi-Objective Optimization strategies and details the algorithm used in the proposed framework to compute the "ROC front". Section 4 presents a particular application of our approach to the problem of SVM hyperparameter selection and shows that our method enables to reach more interesting trade-offs than traditionnal model selection techniques on standard benchmarks (UCI datasets). In section 5, we discuss ways of selecting the best model from the pool of locally optimal models. Then, in order to assess the usefulness of our approach, we present in section 6 its application on a real world classification problem which consists in a digit/outlier discrimination task embedded in a numerical field extraction system for handwritten incoming mail documents. Finally, a conclusion and future works are drawn in section 7.

2 The "ROC front" concept

As stated in the introduction, a model selection problem may be seen from a multi-objective point of view, turning thus into a multi-model selection approach. In the literature, some multi-model selection approaches have been proposed. However, these approaches aim at designing a single classifier and thus cannot be considered as real multi-model selection approaches. Caruana for example proposed in [8] an approach for constructing ensembles of classifiers, but this method aims at combining these classifiers in order to optimize a scalar criterion (accuracy, cross entropy, mean precision, AUC). Bagging, Boosting or Error-Correcting-Output-Codes (ECOC) [17] are also classifier ensemble methods that can be viewed as producing single classifiers efficient with respect to a scalar performance metric. In [27], an Evolutionary Algorithm (EA) based approach is applied to find the best hyperparameters of a set of binary SVM classifiers combined to produce a multiclass classifier.

The approach which is proposed in this paper is different since our aim is not to build a single classifier but a pool of classifiers, each one optimizing both FP and TP rates in the ROC space. In such a context, let us recall that a problem arising when ROC space is used to quantify classifier performance is their comparison in a 2-D objective space : a classifier may be better for one of the objectives (e.g. FP) and worse for the other one (e.g TP). Consequently, the strict order relation that can be used to compare classifiers when a single objective is only considered becomes unusable and classical mono-objective optimization strategies can not be applied.

Usually, in ROC space, this problem is tackled using a reduction of the FP and TP rates into a single criterion such as the Area Under ROC Curve (AUC) [30]. However, such performance indicators are a resume of the ROC curve taken as a whole and do not consider the curve from a local point of view. The didactic example proposed in figure 2 illustrates this statement. One can see on this figure two synthetic ROC curves. The curve plotted as solid line has a better AUC value, but the corresponding classifier is not better for any specific desired value of FP rate (resp. TP). Consequently, optimizing such a scalar criterion to find the best hyperparameters could lead to solutions that do not fit the practitioner needs in certain context. A better idea could be to optimize simultaneously FP and TP rates using a MOO framework and a dominance relation to compare classifier performance.

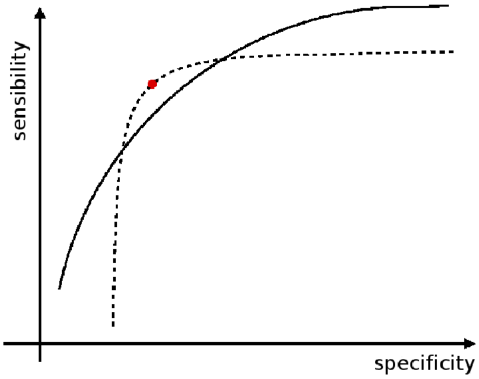


Fig. 2. Comparing ROC curves: the solid ROC curve provides a better AUC than the dashed ROC curve, but is not locally optimal for a given range of specificity (False Positive Rate).

Let us recall that the dominance concept has been proposed by Vilfredo Pareto in the 19th century. A decision vector $\overrightarrow{u}$ is said to dominate another decision vector $\overrightarrow{v}$ if $\overrightarrow{u}$ is not worse than $\overrightarrow{v}$ for any objective function and if $\overrightarrow{u}$ is better than $\overrightarrow{v}$ for at least one objective function. This is denoted $\overrightarrow{u} \prec \overrightarrow{v}$. More formally, in the case of the minimization of all the objectives, a vector $\overrightarrow{u} = (u_1, u_2, \ldots, u_k)$ dominates a vector $\overrightarrow{v} = (v_1, v_2, \ldots, v_k)$ if and only if:

$$\forall i \in \{1, \ldots, k\}, u_i \leq v_i \wedge \exists j \in \{1, \ldots, k\} : u_j < v_j$$

Using such a dominance concept, the objective of a Multi-Objective Optimization algorithm is to search for the Pareto Optimal Set (POS), defined as the set of all non dominated solutions of the problem. Such a set is formally defined as the set :

$$POS = \left\{ \overrightarrow{u} \in \vartheta / \neg \exists \overrightarrow{v} \in \vartheta, \overrightarrow{f(v)} \prec \overrightarrow{f(u)} \right\}$$

where ϑ denotes the feasible region (*i.e.* the parameter space regions where the constraints are satisfied) and $\overrightarrow{f}$ denotes the objective function vector. The corresponding values in the objective space constitute the so-called Pareto Front.

From our model selection point of view, the POS corresponds to the pool of non-dominated classifiers (the pool of the best sets of hyperparameters). In this pool, each classifier optimizes a particular FP/TP trade-off. The resulting set of FP/TP points constitutes an optimal front we call "ROC front". This concept is illustrated with a didactic example as shown in figure 3: let us assume that ROC curves have been obtained from three distinct hyperparameter sets. This could lead to the three synthetic curves plotted as dashed lines. One can see on this example that none of the classifiers dominates the others on the whole range of FP/TP rates. An interesting solution for a practitioner is the "ROC front" (the dotted solid curve), which is made of some non-dominated parts of each classifier ROC curves. The method proposed in this paper aims at finding this "ROC front" (and the corresponding POS), using an Evolutionary Multi-Objective Optimization (EMOO) Algorithm. This class of optimization algorithm has been chosen since Evolutionary Algorithms (EA) are known to be well-suited to search for multiple Pareto optimal solutions concurrently in a single run, through their implicit parallelism.

In the following section, a brief review of existing EMOO algorithms is proposed and the chosen algorithm is described.

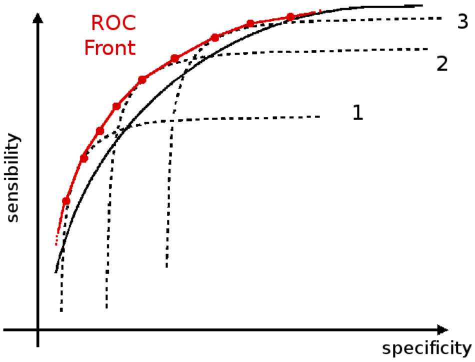


Fig. 3. Illustration of the ROC front concept : the ROC front depicts the FP/TP performance corresponding to the pool of non dominated operating points.

3 Evolutionary Multi-Objective Optimization

As stated earlier, our objective in this paper is to search for a pool of parametrized classifiers corresponding to the optimal set of FP/TP trade-offs. From a multi-objective optimization point of view, this set can naturally be seen as the Pareto Optimal Set and the set of corresponding FP/TP trade-offs is the ROC front. To tackle such a problem of searching a set of solutions describing the Pareto front, EA are known to be well-suited. This is why we do not consider in our review the approaches that optimize a single objective using the aggregation of different objectives into a single one (e.g. the use of the AUC) or the transformation of some objectives into constraints. For more details concerning these methods, see for example [16].

3.1 Short review of existing approaches

Since the pioneering work of [31] in the mid eighties, a considerable amount of EMOO approaches have been proposed (MOGA from [21], NSGA from [32], NPGA from [23], SPEA from [37], NSGA II from [15], PESA from [12], SPEA2 [36]). In a study reported in [26] the performance of the three most popular algorithms (SPEA2, PESA and NSGA-II) are compared. These three approaches are elitist, i.e. they all use a history archive that records all the non-dominated solutions previously found in order to ensure the preservation of good solutions. This comparative study has been performed on different test problems using as quality measurement the two important criteria of an EMOO, *i.e.* the closeness to the Pareto front and the solution distribution in the objective space. Indeed, achieving a good spread and a good diversity of solutions on the obtained front is important to give the user as many choices as possible. The results obtained in [26] (which are corroborated in [36] and [7]) showed that none of the proposed algorithms ”dominate” the others in the Pareto sense. SPEA2 and NSGA-II perform equally well in convergence and diversity maintenance. Their convergence through the real Pareto Optimal Set is inferior to that of PESA but diversity among solutions is better maintained. The study also showed that NSGA-II is faster than SPEA2, because of the expensive clustering of solutions in SPEA2.

In the context of multi-model selection, computation of the objective values is often very time consuming since it involves learning and testing the classifier for each hyperparameter set. Moreover, a good diversity of solutions is necessary since there is no *a priori* information concerning the adequate operating point on the Pareto front. That is why we have chosen to use NSGA-II in the context of our study. We give in the next subsection a concise description of this algorithm. For more details, we refer to [15].

3.2 NSGA-II

NSGA II is a modified version of a previously proposed algorithm called NSGA [32]. It is a population-based, fast, elitist and parameter free approach that uses an explicit diversity preserving mechanism.

Algorithm 1. NSGA-II algorithm

```
$P_0 \leftarrow$ pop-init()
$Q_0 \leftarrow$ make-new-pop ($P_0$)
$t \leftarrow 0$
while $t < M$ do
   $R_t \leftarrow P_t \cup Q_t$
   $\mathcal{F} \leftarrow$ non-dominated-sort($R_t$)
   $P_{t+1} \leftarrow \emptyset$
   $i \leftarrow 0$
   while $|P_{t+1}| + |\mathcal{F}_i| \leq N$ do
      $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$
      crowding-distance-assignment($\mathcal{F}_i$)
      $i \leftarrow i + 1$
   end while
   Sort ($\mathcal{F}_i, \prec_n$)
   $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$
   $Q_{t+1} \leftarrow$ make-new-pop ($P_{t+1}$)
   $t \leftarrow t + 1$
end while
```

As one can see in Algorithm 1, the approach starts with the random creation of a parent population P_0 of N solutions (individuals). This population is used to create an offspring population Q_0. For this step, P_0 is first sorted using a non-domination criterion. This sorting assigns to each individual a domination rank. The non-dominated individuals have rank 1, they constitute the front

$\mathcal{F}_1$. Then, the others front $\mathcal{F}_i$ are defined recursively by ignoring the lower ranked solutions. This ranking is illustrated on the left of figure 4 in the case of a two-objective problem (f_1,f_2). Using the results of the sorting procedure, each individual is assigned a fitness equal to its non-domination level. Then, binary tournament selection, recombination and mutation operators (see [22] and [15]) are used to create a child population Q_0 with the same size as P_0.

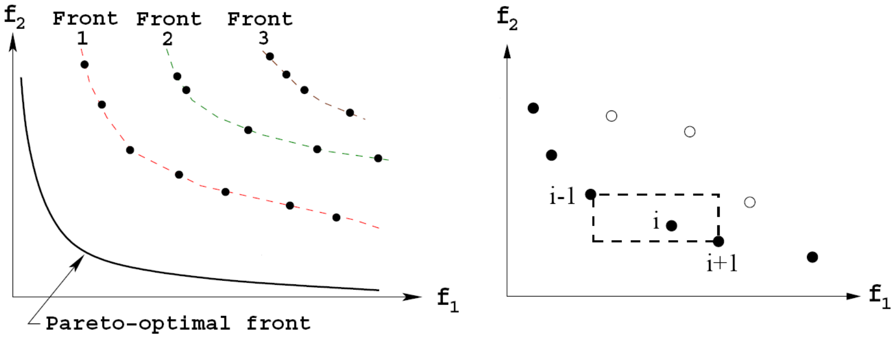


Fig. 4. Illustration of the $\mathcal{F}_i$ concept (left). Illustration of the crowding distance concept (right). The black points stand for the dominant vectors, whereas white ones are dominated.

After these first steps, the main loop is applied for M generations. In each loop of this algorithm, t denotes the current generation, $\mathcal{F}$ denotes the result of the non domination sorting procedure, i.e $\mathcal{F} = \{\mathcal{F}_i\}$ where $\mathcal{F}_i$ denotes the i^{th} front. P_t and Q_t denote the population and the offspring at generation t respectively and R_t is a temporary population.

As one can see, the main loop of the algorithm starts with a merging of the current P_t and Q_t to build R_t. This population of 2N solutions is sorted using the non domination sorting procedure in order to build the population P_{t+1}. In this step, a second sorting criterion is used to keep P_{t+1} to a constant size N during the integration of the successive $\mathcal{F}_i$. Its aim is to take into account the contribution of the solutions to the spread and the diversity of objective function values in the population. This sorting is based on a measure called *crowding_distance*. This measure which is precisely described in [15] is based on the average distance of the two points on both sides of this point along each of the objectives. This measure is illustrated on the right of figure 4. The larger the surface around the considered point, the better the solution from the diversity point of view. Using such values, the solutions in R_t that most contribute to the diversity are preferred in the construction of P_{t+1}. This step is illustrated in Algorithm 1 through the use of Sort($\mathcal{F}_i$,$\prec_n$), where $\prec_n$ denotes a partial order relation based on both domination and crowding distance. According to this relation, a solution i is better than a solution j if $i_{rank} < j_{rank}$ or if $(i_{rank} = j_{rank})$ and $(i_{distance} > j_{distance})$. One can note that $\prec_n$ is also used in the tournament operator.

Using this algorithm, the population P_t necessarily converges through a set of points of the Pareto front of the problem since non-dominated solutions are preserved along generations. Furthermore, the use of the crowding-distance as a sorting criterion guarantees a good diversity in the population [15]. In the following section, NSGA-II is used in the proposed framework for SVM multi-model selection.

4 Application to SVM multi-model selection

As explained in the previous sections, the proposed framework aims at finding a pool of classifiers, optimizing simultaneously FP and TP rates. The approach can be used for any classifier that uses at least one hyperparameter. In this section, we have chosen to consider Support Vector Machines (SVM) since it is well known that the choice of SVM model parameters can dramatically affect the quality of their solution. Moreover, the problem of SVM model selection is known to be a difficult problem.

4.1 SVM classifiers and their hyperparameters for model selection

As stated in [28], classification problems with asymmetric and unknown misclassification costs can be tackled using SVM through the introduction of two distinct penalty parameters C_- and C_+. In such a case, given a set of m training samples x_i in $\Re^n$ belonging to class y_i :

$$(x_1, y_1) \ldots (x_m, y_m), x_i \in \Re^n, y_i \in \{-1, +1\}$$

the maximisation of the dual lagrangian with respect to the α_i becomes :

$$Max_\alpha \Big\{ \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \Big\}$$

$$\text{subject to the constraints: } \begin{cases} 0 \leq \alpha_i \leq C_+ \quad for \quad y_i = -1 \\ 0 \leq \alpha_i \leq C_- \quad for \quad y_i = +1 \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

where α_i denote the Lagrange multipliers and $K(.)$ denotes the kernel. In the case of a Gaussian (RBF) kernel, $K(.)$ is defined as :

$$K(x_i, x_j) = exp\left(-\gamma \times \|x_i - x_j\|^2\right)$$

Hence, in the case of asymmetric misclassification costs, three parameters have to be determined to perform an optimal learning of the SVM classifier:

- The kernel parameter of the SVM-rbf : γ.
- The penalty parameters introduced above : C_- and C_+.

In the following, the proposed framework is used in order to select the value of these three hyper-parameters.

4.2 Application of NSGA-II for SVM model selection

Two particular points have to be specified for the application of NSGA-II to SVM multi-model selection :

- the solution coding : as said before, three parameters are involved in the learning of SVM for classification problems with asymmetric misclassification costs : C_+, C_- and γ. These three parameters constitute the parameter space of our optimization problem. Consequently, each individual in NSGA-II has to encode these three real values. We have chosen to use a real encoding of these parameters in order to be as precise as possible.
- the evaluation procedure : each individual in the population corresponds to some given values of hyperparameters. In order to compute the performance associated to this individual, a classical SVM learning is performed using the encoded parameter values on a learning dataset. Then, this classifier is evaluated on a test dataset with the classical FP and TP rates as performance criteria.

One can see on figure 5 a synthetic scheme of our multi-model selection method.

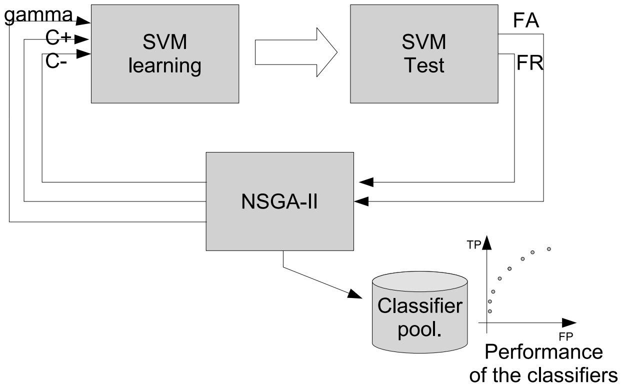


Fig. 5. SVM Multi-model selection framework

4.3 Experimental results on UCI datasets

In this subsection, the proposed multi-model selection approach based on the ROC front concept is evaluated and compared with other approaches on publicly available benchmark datasets [1]. First, the experimental protocol of our tests is described. Then, the results are shown and compared with some reference works, and finally several comments on these results are proposed.

Our approach has been applied on several 2-class benchmark datasets publicly available in the UCI Machine Learning repository on which state-of-the-art results have been published. The number of samples and the number of attributes for each problem are reported in table 1.

problem	# samples	# attributes
australian	690	14
wdbc	569	30
breast cancer	699	10
ionosphere	351	34
heart	270	13
pima	768	8

Table 1
Number of samples and number of attributes of the considered 2-class UCI problems.

As we propose a real multi-objective approach, the result of our experiments is a pool of classifiers describing the ROC Front. Thus, the evaluation of our approach is not easy since as mentioned in the introduction, comparing some results in a multi-dimensional space is a difficult task. Note that there exist some dedicated measures such as the Set Coverage Metric proposed in [35]. However, to the best of our knowledge, the other referred methods in the literature always consider a single classifier as a solution for a classification problem, which makes it difficult to compare our results with those found in the literature. To the best of our knowledge, the only way to compare our proposed approach to existing algorithms should be to apply the following experimental protocol :

Begin

- For each possible FP value (resp. TP value)
  - · Find the nearest classifier and the corresponding set of hyper parameters on the ROC front and determine the corresponding TP value (resp. FP value) on a validation dataset with a null threshold.
  - · For all existing approaches, find the corresponding operating point on the ROC curve, determine the corresponding threshold value and then

```
        evaluate the TP value (resp. FP value) on a validation dataset with the
        obtained threshold.
    · End for
    Use a statistical test of significance (e.g. McNemar test of significance) on all
    FP values (resp. TP values) to assess the superiority of the best approach.
    End
```

Unfortunately, in the literature, papers mention at best the AUC but do not provide the whole ROC curve. Such an AUC is an average over all FP values (resp. TP values) and thus loose the local information the practitioner needs in a real-world application. This trade-off between global and local evaluation is well-known in the EMOO community and is still an open issue to compare multi-objective optimization algorithms. Based on this statement, we have therefore chosen to average all the local performance of the ROC front to produce a way to compare our approach to existing ones based on AUC. For that, an Area Under the ROC Front (AUF) is calculated and compared with the Area Under the ROC Curve (AUC) of the other approaches. We do know that this comparison is not theoretically correct since the best results of a pool of classifiers are compared with a curve obtained by varying the threshold of a single classifier. However, the aim of this comparison is not to show that our approach gives better performance but only to highlight the fact that more interesting trade-offs may be locally reached through the ROC front approach. This comparison may also be justified by the fact that finally, in both cases, only one classifier with a unique threshold will be retained for a given problem. We discuss in section 5 how to select the best model among the pool of classifiers and offer a solution to this problem.

The result of our approach is compared with several works based on the optimization of a scalar criterion for various classifiers : [5] (Decision lists and rules sets), [13] (Rankboost), [19] (Decision trees), [30] (SVMs) and [34] (five models : naive Bayes, logistic, decision tree, kstar, and voting feature interval). We refer to these papers for more explanation of the criterion and the model used.

Concerning the application of our multi-objective strategy, a cross validation procedure has been performed with 5 folds for each dataset. The results are presented in table 2, where the first column is the best AUC found until now among the precited works based on the optimization of a scalar criterion, and the second one is the AUF of our approach.

As expected, one can see that for every dataset the ROC front yielded by the pool of classifiers leads to a higher area than the area under the ROC curve of the other single classifiers. As said before, it is important to emphasise that the AUF cannot theoretically be compared with AUC since the various operating points of the ROC front cannot be reached by a single classifier. However, this

problem	AUC literature	ref.	AUF
australian	90.25 ± 0.6	[34]	96.22 ± 1.7
wdbc	94.7 ± 4.6	[19]	99.59 ± 0.4
breast cancer	99.13	[5]	99.78 ± 0.2
ionosphere	98.7 ± 3.3	[30]	99.00 ± 1.4
heart	92.60 ± 0.7	[34]	94.74 ± 1.9
pima	84.80 ± 6.5	[13]	87.42 ± 1.2

Table 2
Comparison of the Area Under the ROC Curve (AUC) in the literature with the Area Under the ROC Front (AUF).

comparison with methods which that directly optimize AUC clearly shows that our approach enables to reach very interesting local operating points which cannot be reached at the same time by the AUC-based classifiers. Hence, we claim that if the good model can be selected among the pool of classifiers, our approach can lead to better results than AUC-based methods. Despite these interesting results, the model selection problem still remains partly open since the choice of the retained classifier among the set of locally optimal classifiers has to be performed. This crucial final model selection step is discussed in the following section.

5 How to select the best model ?

The problem of choosing an operating point in the ROC space is not specific to the proposed approach. For example, when training a single classifier with an AUC criterion, the practitioner still has to choose the appropriate threshold value, i.e. the operating point in the ROC space.

Theoretically, the best operating point must be determined according to Bayes theory by minimizing the following decision function, known as the expected cost and defined as :

$$\text{expected cost(FP,TP)} = p(p).(1 - TP).c(N,p) + p(n).FP.c(Y,n)$$

where $p(p)$ and $p(n)$ are respectively the prior probabilities of (p)ositive samples and (n)egative samples (class distribution), $c(N,p)$ is the cost of a false negative error and $c(Y,n)$ is the cost of a false positive error.

Obviously, target conditions ($p(p)$, $p(n)$, $c(N,p)$, $c(Y,n)$) are rarely all known at runtime. Consequently, two runtime conditions may be distinguished to select the best model on the ROC front, depending on whether the misclassifications costs and the class distributions are known with an acceptable precision or not.

- If the target conditions are known, then *iso-performance lines* proposed in [18] can be used to select the best model. It is based on the projection of the Bayes decision function onto the ROC space. An *iso-performance line* is defined as the set of points providing the same expected cost. The slope of an *iso performance line* is given by :

$$\text{slope} = \frac{p(n).c(Y,n)}{p(p).c(N,p)}$$

Using this *iso-performance line* on the ROC space, the optimal operating point is easy to find. Starting from the upper left corner, move the *iso-performance line* towards the lower right corner; the optimal operating point is the first intersection between the line and the ROC front. This method is illustrated on figure 6, where we can notice that the best classifier can be selected.

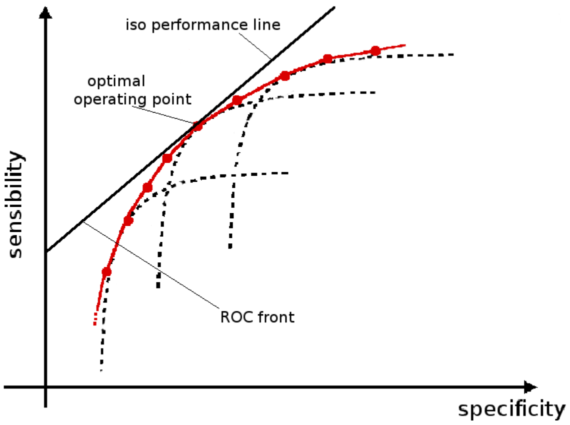


Fig. 6. When the target conditions of a given problem are known, representing the *iso-performance line* allows to select the appropriate operating point.

- If the target conditions are unknown at runtime, the expected cost cannot be evaluated. Consequently, the slope of the appropriate *iso-performance line* can not be determined. Then, the only way for choosing the best classifier is to perform a testing stage in context, i.e. testing each classifier of the ROC front, and choosing the one that best fits the application constraints. We present in section 6 a real world problem with this kind of scenario.

One can note that, in the second case, browsing all possible *iso-performance lines* could be used in order to "filter" the ROC-front by removing concavities. Indeed, classifiers lying on the concavities of the ROC front can not be theoretically optimal since any performance on a line segment connecting two ROC points can be achieved by randomly choosing between them [20]. This is illustrated on figure 7. Such an idea has been proposed in [29] to generate the ROC Convex Hull of a set of classifiers. Consequently, one can consider that our proposed method enables to find the optimal ROC-CH.

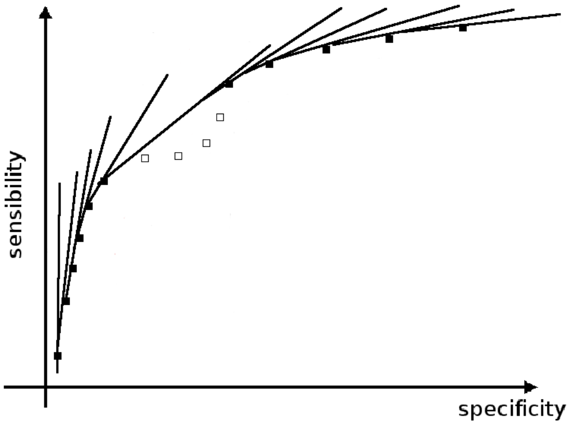


Fig. 7. Browsing all possible *iso-performance lines* on a concave ROC front allows to filter the non-filled squares the performance of which can be outperformed.

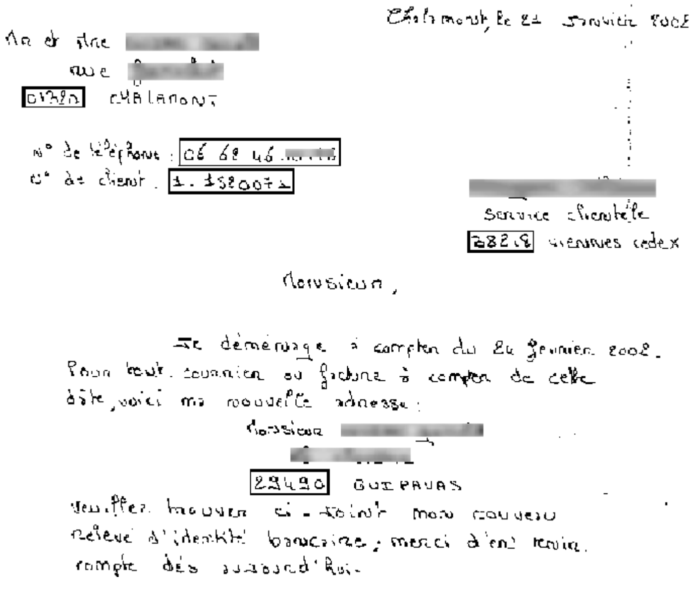
6 Application to a real-world pattern recognition problem

In this section, an interesting example of real-world problem for which our approach suits better than an AUC-based method is presented.

6.1 Digit/outlier discrimination

The work described in this paper has been motivated by the design of a more complex system that aims at extracting numerical fields (phone number, zip code, customer code, etc.) from incoming handwritten mail document images [10, 11] (see fig. 8). The main difficulty of such a task comes from the fact that handwritten digits may touch each other in the image while some textual parts sometimes are made of separated or touching characters. Figure 9 gives some examples of segmented components to deal with. In such a variable context, segmentation, detection and recognition of a digit and rejection of textual components must be performed simultaneously.

In this paper, the proposed approach is applied to a particular stage of the numerical field extraction system. More precisely, the SVM to be optimized is used as a fast two-class classifier prior to the digit recognizer itself, aiming at filtering the "obvious outliers" (see figure 9.a) from all the other shapes (see figure 9.b and 9.c) in order to avoid a costly digit recognition stage when it is not necessary. The choice of the SVM classifier has been motivated by



Chalamont, le 21 janvier 2002

M. et Mme [redacted]
rue [redacted]
01320 CHALAMONT

n° de téléphone : 06 68 46 [redacted]
n° de client : 1.1580071

[redacted]
Service clientèle
28218 Vienne cedex

Monsieur,

Je déménage à compter du 26 février 2002.
Pour tout courrier ou facture à compter de cette date, voici ma nouvelle adresse :

Monsieur [redacted]
[redacted]
29490 GUIPAVAS

Veuillez trouver ci-joint mon nouveau relevé d'identité bancaire ; merci d'en tenir compte dès aujourd'hui.

Fig. 8. Example of an incoming mail document. Numerical fields to extract are highligthed

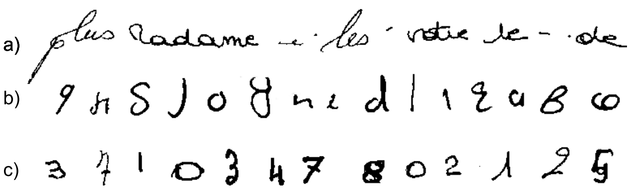


Fig. 9. Examples of digits and outliers. The first line (a) contains shapes which can be considered as "obvious" outliers. The last line (c) contains digits that should be accepted as they are, whereas the middle line (b) contains "ambiguous outliers" (*i.e.* shaped as digits) that should be rejected by the proposed approach.

its efficiency in a two-class context. Its objective is to reject as many outliers as possible, while accepting as many digits as possible. Further stages of the system deal with digit recognition and ambiguous outlier rejection. This context is a good example of a classification task with asymmetric and unknown misclassification costs since the influence of a FP or a FN on the whole system results is unknown at runtime. In the next subsection, the performance of the proposed system are assessed.

6.2 Experimental results and discussion

In this section, the experimental results obtained using the proposed approach are analysed. These results are compared with those obtained using a state-of-the-art algorithm [30], where a SVM classifier is trained with respect to an AUC criterion. Both NSGA-II and AUC-based approaches have been applied on a learning database of 7129 patterns (1/3 digit, 2/3 outliers), tested and evaluated on a test and a validation database of resp. 7149 and 5000 patterns with the same proportions of digits and outliers. In the case of NSGA-II, the range values for SVM hyperparameters are given in table 3. Concerning the NSGA-II parameters, we have used some classical values, proposed in [15]. Among them, one can note that the size of the population has been set to 40 in order to have enough points on the Pareto front. The resulting curves are presented in figure 10.

Hyperparameter	γ	C_-	C_+
Range	$0-1$	$0-5000$	$0-5000$

Table 3
Range values for SVM hyperparameters

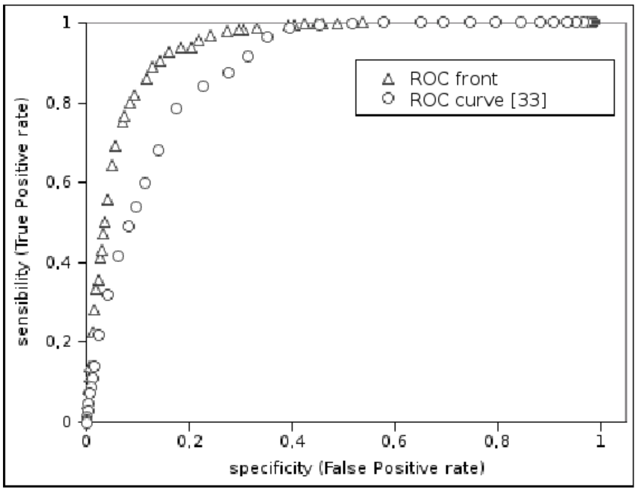


Fig. 10. FP/TP curves obtained using the two approaches : a set of SVM classifiers obtained with NSGA-II (ROC Front), and a single SVM classifier trained with AUC criterion (ROC Curve).

Several comments can be made from the obtained results. First, one can remark that each point of the ROC curve obtained for a single classifier trained with AUC criterion is dominated by at least one of the point of the ROC front. Such a result stems from the fact that using an EMOO approach, FP and TP rates are minimized simultaneously through the variation of the three involved SVM hyperparameters whereas in the case of an AUC approach, a single parametrized classifier is trained to optimize every possible FP/TP trade-offs. Figure 11 is another illustration of the interest of the ROC front concept. It shows the ROC curves computed from four classifiers which have been selected using the proposed framework. This figure clearly shows that the ROC front corresponds to a set of classifiers which are specialized on some specific ranges of FP/TP trade-offs.

A second remark concerns the possibility when using an EMOO to apply some constraints on the objective values (as in the parameter space). Such a possibility is very useful in the context of our application since it enables to

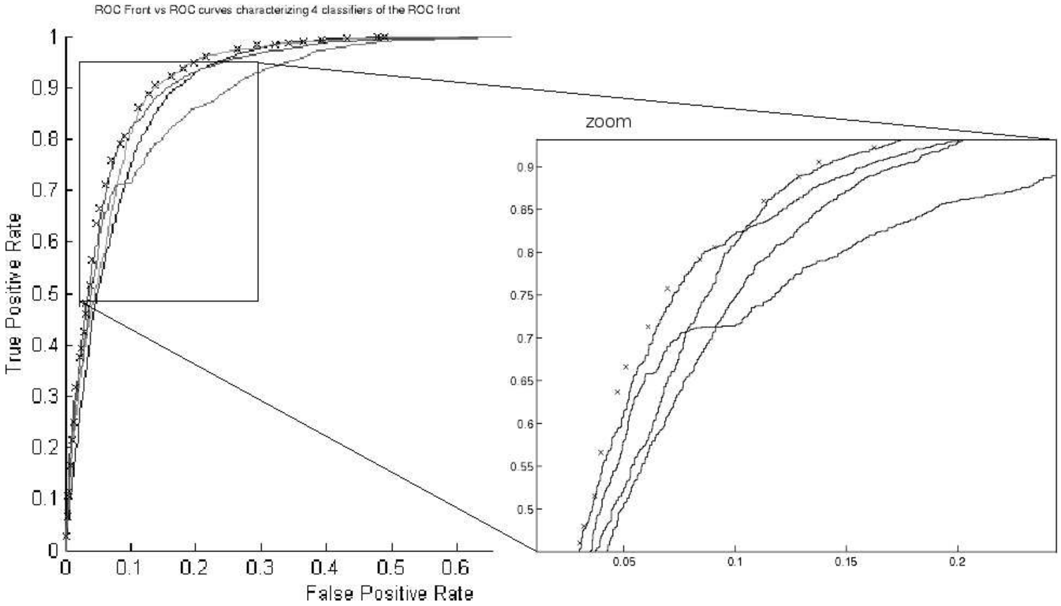


Fig. 11. Illustration of the ROC front concept on a classification dataset. The solid lines are the ROC curves computed from 4 of the 40 classifiers selected using the proposed framework. The performance of the classifiers of the ROC front appear as 'x'

focus on a small part of the ROC front. Indeed, we are particularly interested by a small part of the ROC front since we want the rejection of a digit be as rare as possible to prevent errors in the whole recognition process, this would imply a null False Negative rate (*i.e.* a 100% TP rate). But on the other hand, figure 10 shows that a 100% TP rate leads to a FP higher than 50%. Such a result involves a very time consuming recognition stage, that can not be accepted regarding our processing time constraints during the decision stage. Thus, we have applied an lower bound of 97% to the TP rate in order to obtain an acceptable trade-off between the recognition quality of the system and the computational constraints. Figure 12 shows the results obtained with this additional constraint. One can see that such a setting enables to obtain more diversity among the FP/TP trade-offs in the chosen TP range.

6.3 How to select the best model ?

Once the ROC front has been built for our application, the final best model among the classifiers has to be selected. As discussed in section 5, two scenarii may occur at runtime, whether the expected cost can be computed or not. In our digit/outlier discrimination problem, this expected cost cannot be computed since the classification task is embedded in the whole numerical field extraction application and is evaluated by recall/precision measures. Hence, a test stage in context has to be performed by successively embedding each classifier of the front in the whole system. Table 4 presents the results ob-

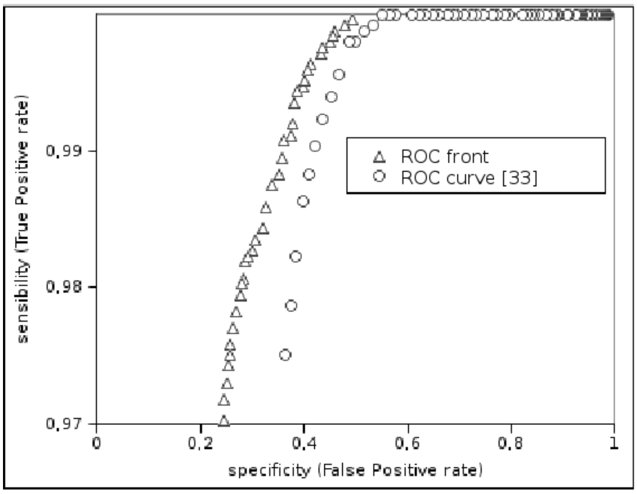


Fig. 12. ROC curve obtained for a True Positive rate between 97 and 100%

tained by the whole numerical field extraction system for several digit/outlier classifiers of the ROC front, i.e. for several FP/TP trade-offs.

Classifier TP rate in %	98.8	99.04	99.26	99.48	99.76	99.96	100
recall	0.370	0.410	0.440	0.458	0.462	0.481	0.488
precision	0.110	0.130	0.150	0.176	0.246	0.223	0.152
F1-Measure	0.170	0.197	0.224	0.254	0.321	0.305	0.232

Table 4
Recall/Precision values of the whole numerical field extraction system for several digit/outlier classifiers, represented here by their TP rate.

As one can expected the True Positive Rate has to be very high to provide good recall and precision values since rejecting a digit may imply to miss a numerical field. We do not show the results for the classifiers the TP rate of which is lower than 98.8% since both recall and precision are lower than those presented in table 4. Finally, given the final application constraints, the system designer is able to choose the model that best fits the industrial needs. As an example, if one choose to maximise the F1-measure, the classifier providing TPR=99.76% will be selected. The results of this real-world application corroborate the idea that model selection must be considered as long as possible as a multi-objective optimization task in a pattern recognition system.

7 Conclusion

In this paper, we have presented a framework to tackle the problem of classifier model selection with unknown and/or evolutive misclassification costs. The approach is based on a multi-model selection strategy in which a pool of classifiers is trained in order to depict an optimal ROC front. Using such a front, it is possible to choose the FP/TP trade-off that best fits the application constraints. An application of this strategy with Evolutionary Multi-Objective Optimization for the training of a set of SVM classifiers has been proposed, with a validation on both UCI datasets and a real-world application on the discrimination of handwritten digits from outliers. Obtained results have shown that our approach enables to reach better local operating points that state-of-the-art approaches based on the area under ROC curve criterion. As a conclusion, one can say that an AUC-based approach suits pattern recognition problems where the operating point may vary, whereas our approach better suit problems where the operating point is supposed to be static.

The proposed approach is simple and generic and can thus be of great interest for the practitioner who has to optimize a classifier in the context of unknown and/or evolutive misclassification costs. It can be applied to other parametric classifiers (KNN, Neural network, etc.) with other optimization methods [14]. Moreover, it can be easily extended through the introduction of other parameters (kernel type) or objectives (number of support vectors, decision time).

In our future works, we plan to extend the approach to the multiclass problem. We also plan to apply a multi-objective optimization strategy to the whole numerical field extraction system, using recall and precision as criteria.

References

[1] D.J. Newman A. Asuncion. UCI machine learning repository, 2007.

[2] D. Anguita, S. Ridella, F Rivieccio, and R Zunino. Hyperparameter design criteria for support vector classifiers. *Neurocomputing*, 55(1-2):109–134, 2003.

[3] N.E. Ayat, M. Cheriet, and C.Y. Suen. Automatic model selection for the optimization of svm kernels. *Pattern Recognition*, 30:1733–1745, 2004.

[4] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural Computation*, 12:1889–1900, 2000.

[5] Henrik Boström. Maximizing the area under the roc curve using incremental reduced error pruning. In *Proceedings of ROCML*, 2005.

[6] A.P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.

[7] L.T. Bui, D. Essam, H.A. Abbass, and D. Green. Performance analyis of multiobjective evolutionary methods in noisy environnments. In *Proceedings of APS 2004*, pages 29–39, 2004.

[8] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *proceedings of ICML*, 2004.

[9] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing

multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

[10] C. Chatelain, L. Heutte, and T. Paquet. Segmentation-driven recognition applied to numerical field extraction from handwritten incoming mail documents. *Document Analysis System, LNCS 3872*, pages 564–575, 2006.

[11] C. Chatelain, L. Heutte, and T. Paquet. A two-stage outlier rejection strategy for numerical field extraction in handwritten documents. In *Proceddings of ICPR*, pages 224–227, 2006.

[12] D.W. Corne, J.D. Knowles, and M.J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel problem solving from nature*, pages 839–848, 2000.

[13] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *Advances in NIPS*. MIT Press, 2004.

[14] B.F. de Souza, A.C.P.L.F. de Carvalho, R. Calvo, and R. Porfirio Ishii. Multiclass svm model selection using particle swarm optimization. In *Proceedings of HIS*, pages 31–31, 2006.

[15] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist nondominated sorting genetic algorithm for multiobjective optimization : Nsga-ii. *IEEE Transactions on Evolutionary Computation*, pages 182–197, 2002.

[16] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[17] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[18] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. Technical report, HP Laboratories, 2004.

[19] C. Ferri, P. Flach, and J. Hernandez-Orallo. Learning decision trees using the area under the roc curve. In *Proceedings of ICML*, pages 139–146, 2002.

[20] P.A. Flach and S. Wu. Repairing concavities in roc curves. In *Proc. 2003 UK Workshop on Computational Intelligence*, pages 38–44. University of Bristol, August 2003.

[21] C.M. Fonseca and P.J. Flemming. Genetic algorithm for multiobjective optimization: formulation, discussion and generalization. In *Proceedings of ICGA*, pages 416–423, 1993.

[22] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[23] J. Horn, N. Nafpliotis, and Goldberg D.E. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of IEEE-WCCC*, pages 82–87, 1994.

[24] T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Scholkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[25] Sathiya Keerthi, Vikas Sindhwani, and Olivier Chapelle. An efficient

method for gradient-based adaptation of hyperparameters in svm models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 673–680. MIT Press, Cambridge, MA, 2007.

[26] V. Khare, X. Yao, and K. Deb. Performance scaling of multiobjective evolutionary algorithm. In *Technical report - SCS, University of Birmingham*, pages 1–70, 2002.

[27] G. Lebrun, O. Lezoray, C. Charrier, and H. Cardot. An ea multi-model selection for svm multiclass schemes. In *Proceedings of IWANN*, pages 257–264, 2007.

[28] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical report, 1997.

[29] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[30] A. Rakotomamonjy. Optimizing auc with support vector machine. *Proceedings of ECAI Workshop on ROC Curve and AI*, pages 469–478, 2004.

[31] J.D. Schaffer and J.J. Grefenstette. Multiobjective learning via genetic algorithms. In *Proceedings of IJCAI 1985*, pages 593–595, 1985.

[32] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[33] G. Wahba, X. Lin, F. Gao, D. Xiang, R. Klein, and B. Klein. The bias-variance tradeoff and the randomized gacv. In *Proceedings of NIPS*, pages 620–626, 1999.

[34] S. Wu. A scored auc metric for classifier evaluation and selection. *proceedings of ROCML*, 2005.

[35] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms : Empirical results. *IEEE Transactions on Evolutionary Computation*, 2(8):173–195, 1999.

[36] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2001.

[37] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparison case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.