Multi-Objective Model Type Selection

Alejandro Rosales-Pérez[a,*], Jesus A. Gonzalez[a], Carlos A. Coello Coello[b],
Hugo Jair Escalante[a], Carlos A. Reyes-Garcia[a]

[a] *Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Computer Science Department, Luis Enrique Erro No.1, Santa María Tonantzintla, Puebla, 72840, Mexico*
[b] *Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV-IPN), Computer Science Department, Evolutionary Computation Group (EVOCINV), Av. IPN No. 2508, San Pedro Zacatenco, Mexico City, 07360, Mexico*

Abstract

Classification is a mainstream within the machine learning community. As a result, a large number of learning algorithms have been proposed. The performance of many of these could highly depend on the chosen values of their hyper-parameters. This paper introduces a novel method for addressing the model selection problem for a given classification task. In our model selection formulation, both the learning algorithm and its hyper-parameters are considered. In our proposed approach, model selection is tackled as a multi-objective optimization problem. The empirical error, or training error, and the model complexity are defined as the objectives. We adopt a multi-objective evolutionary algorithm as the search engine, due to its high performance and its advantages for solving multi-objective problems. The model complexity is estimated experimentally, in a general fashion, as normally done for any learning algorithm, through the VC dimension. Strategies for choosing a single model or for constructing an ensemble of models from

---

[*] Corresponding author. Tel.: +52 (222) 2663100 x 3413
Email address: `arosales@inaoep.mx` (Alejandro Rosales-Pérez)

the resulting non-dominated set are also proposed. Experimental results on benchmark data sets indicate the effectiveness of the proposed approach. Furthermore, a comparative study shows that the obtained models are highly competitive, in terms of generalization performance, with other methods in the state of the art that focus on a single-learning algorithm, or a single-objective approach.

Keywords: Model type selection, VC dimension, Multi-objective optimization, Ensemble methods

1. Introduction

Classification is a common task in supervised learning. Its popularity is due to its use in a wide range of applications, such as medical diagnosis, text categorization, etc. In the machine learning community, several learning algorithms to fit a model have been proposed, including decisions trees, artificial neural networks, those based on statistical learning, etc. However, to date there is not a universal "best" model; this is referred to as the **No Free Lunch Theorem** [?]. Moreover, many of these learning algorithms have a set of adjustable parameters, called hyper-parameters, whose fine-tuning can affect their generalization ability. Taking that into consideration, one might ask the questions: what learning algorithm should be used for a specific problem? Also, given a learning algorithm, what hyper-parameters values should be chosen? These questions are related to the issue of model selection.

In the literature, there are several studies that address the model selection problem. Among these, some of them have approached it as an optimization problem, differing in the search technique adopted, including gradient-based methods [? ? ?], grid-search [?], or bio-inspired meta-heuristics such as evolutionary algorithms [? ? ? ? ?], artificial immune systems [?] or particle swarm optimizers [? ? ?], etc. Grid-search is the simplest one, but it could be time-consuming. Although gradient-based methods tend to be more (computationally) efficient, they are very susceptible to the initial search point and they can easily get trapped in a local optimum. Evolutionary algorithms have gained popularity because of their ease of use and their ability to overcome these shortcomings. Indeed, evolutionary algorithms can be less computationally expensive than grid-search, and are less susceptible to their initial search points than gradient-based methods. Furthermore, evolutionary algorithms do not require gradient information and can be easily parallelized.

Another major issue in model selection is the criterion used for this purpose. In this direction, we can differentiate the works that consider a single-objective criterion and those that consider multiple criteria. The single-objective criterion approaches are generally based on an estimation of the generalization error through the well-known k fold cross validation [? ? ? ?]. Attention has also been paid to considering multiple criteria. These works typically consider the model performance and some criterion for penalizing the model complexity [? ?]. Others have considered either to minimize the sensitivity and specificity [? ?], or different estimates of the model performance [? ? ?]. Alternatively, multiple criteria have also been approached by simplifying the objectives in a weighted linear combination of these [?] instead of simultaneously optimizing the objectives.

Despite these efforts, most of the existing studies consider a single model type (i.e., the learning algorithm is fixed *a priori* and the model selection task consists of choosing its hyper-parameters), which could not be the most suitable for a particular problem. To the best of the authors' knowledge, nowadays the works that address both the learning algorithm and the hyper-parameters selection are scarce (e.g. [**?** **?** **?**]), and most of them tackle the problem as a single-objective one. Notwithstanding, the disadvantages of using a single-objective approach for hyper-parameters optimization with respect to the generalization performance have been pointed out by several authors [**?** **?** **?**].

Inspired from previous ideas, we address both the problem of choosing a learning algorithm and its hyper-parameters during the model selection, which is faced as a multi-objective optimization problem. The error on training samples and the model complexity are considered as the objectives in our formulation. Unlike previous works in which the model complexity estimation depends on the learning algorithm (e.g., the number of support vectors in support vector machines), we propose to estimate it through the VC-dimension (for Vapnik-Chervonensky dimension) [**?**].

The main contribution of this paper is a general model selection framework, whose formulation makes it applicable to any learning algorithm. Additional contributions of the paper are as follows: (i) a multi-objective approach for tackling the model type selection problem (i.e., model type plus its hyper-parameters), (ii) the use of the VC-dimension in the model type selection formulation for estimating the model complexity to any model type, and (iii) since the outcome of the multi-objective optimization process is a set of solutions (models), that satisfy an optimal trade-off between the objectives from which a model should be chosen, the strategies proposed for constructing a final classification model from the non-dominated solutions set are an additional contribution. The performance of our proposed approach is assessed on several binary classification benchmark data sets widely used in the literature. The experimental results and comparisons show that our proposal is able to select highly effective classification models.

The remainder of this paper is organized as follows. In Section 2, we describe the VC-dimension theory and the way in which it can be estimated in an experimental fashion. Section 3 presents our proposal, describing in detail how the model selection problem is formulated as a multi-objective one. It also describes the proposal for constructing a final model from solutions in the resulting non-dominated front. Section 4 presents the experiments performed to test the validity of our proposal using benchmark data sets, and the results obtained from these. Finally, the main conclusions and future work direction paths are presented in Section 5.

2. VC Dimension Estimation

Vapnik and Chervonenkis defined the VC dimension [**?**] as a measure of the capacity of a learning algorithm. The VC Dimension is defined through the notion of “shattering”, which is described as follows: if we have a set of n samples that can be separated by a set of indicator functions F (functions that map a sample to its corresponding binary label) in all 2^n possible ways, we say that the set of samples is shattered by the set of functions F. The VC dimension can be formally defined as [**?**]:

> A set of functions F has a VC dimension h if there are h samples that can be shattered by the set of functions F, but there are not $h+1$ samples that can be shattered by the set of functions F.

Notwithstanding that the VC dimension can be seen as a measure of the model complexity [**?**], exact analytic estimates of this are only known for a few classes of functions (linear models), whereas for many others it is unknown. To overcome this, Vapnik *et al.* [**?**] proposed a method to experimentally estimate the effective VC dimension of a model. This approach is based on the best fitting between an analytic formula and measurements of the maximum deviation between the error rates on two independent data sets of varying sizes. Conceptually, this approach can be applied to any learning algorithm [**?**].

The maximum deviation, $\xi(n)$, of the error rates between two independent labeled data sets is defined as:

$$\xi(n) = \max_{\omega} \left(\mid \mathrm{err}\left(\mathbf{Z}_n^1\right) - \mathrm{err}\left(\mathbf{Z}_n^2\right) \mid \right) \tag{1}$$

where $\mathbf{Z}_n^1$ and $\mathbf{Z}_n^2$ are two independent labeled data sets of size n, $\mathrm{err}\left(\mathbf{Z}_n\right)$ is the error rate on the data set $\mathbf{Z}_n$, and ω is the set of parameters of a binary classifier.

As it is stated in [**?**], $\xi(n)$ is bounded as follows:

$$\xi(n) \leq \Phi(n/h) \tag{2}$$

where

$$\Phi\left(\tau\right)=\begin{cases}1 & \text{if } \tau<0.5\\ a\dfrac{\log\left(2\tau\right)+1}{\tau-k}\left(1+\sqrt{1+\dfrac{b\left(\tau-k\right)}{\log\left(2\tau\right)+1}}\right) & \text{if } \tau\geq 0.5\end{cases} \tag{3}$$

where $\tau = n/h$, and the values of the parameters $a = 0.16$ and $b = 1.2$ were empirically determined. The value of $k = 0.14928$ is determined such that $\Phi\left(0.5\right) = 1$.

Since the bound in Equation (2) is tight, it can be assumed that

$$\xi\left(n\right)\approx\Phi\left(n/h\right) \tag{4}$$

The VC dimension h can be estimated from Equations (3) and (4). The maximum deviation $\xi\left(n\right)$ can be estimated by simultaneously minimizing the error rate on one labeled set and maximizing the error rate in the other one. This can be accomplished through the following procedure [**?** **?**]:

1. Generate a random labeled set $\mathbf{Z}_{2n}$ of size $2n$.
2. Split the set $\mathbf{Z}_{2n}$ into two sets of size n: $\mathbf{Z}_n^1$ and $\mathbf{Z}_n^2$.
3. Flip the labels of the set $\mathbf{Z}_n^1$, to form $\overline{\mathbf{Z}}_n^1$.
4. Merge the two sets: $\overline{\mathbf{Z}}=\overline{\mathbf{Z}}_n^1\cup\mathbf{Z}_n^2$, and train the binary classifier with the set $\overline{\mathbf{Z}}$.
5. Evaluate $\mathbf{Z}_n^1$ and $\mathbf{Z}_n^2$ with the trained classifier. Measure the difference of the error rates between the two sets: $\xi\left(n\right)=\mid \mathrm{err}\left(\mathbf{Z}_n^1\right)-\mathrm{err}\left(\mathbf{Z}_n^2\right)\mid$.

This procedure gives an estimate of $\xi\left(n\right)$ from which an estimate of h can be obtained. In order to reduce the variability in the estimation, this procedure is repeated for different data sets varying the samples sizes $n_1, \ldots, n_k$. Moreover, to reduce the variability due to the random samples, the procedure is repeated several times (m_j) for each sample set of size n_i. The average value for each experiment is taken for each n_i: $\overline{\xi}(n_1), \ldots, \overline{\xi}(n_k)$. The effective VC dimension can be estimated by finding the parameter h^* that best fits $\xi(n)$ with the theoretical formula $\Phi(n/h)$, as follows:

$$h^* = \underset{h}{\operatorname{argmin}} \sum_{i=1}^{k} \left[\overline{\xi}(n_i) - \Phi(n_i/h)\right]^2 \tag{5}$$

3. Multi-Objective Approach for Model Selection

The proposed approach formulates the model selection problem as a multi-objective optimization one, where the training error and the model complexity are considered as the objectives to be minimized. The general diagram of our proposal is shown in Figure 1.

The process starts by creating an initial population. In our problem, each individual in the initial population represents a potential model for a classification task. After that, we compute the components to be optimized: the training error and the model complexity, which is estimated in a general fashion via the VC-dimension, as it is explained in Section 2. Next, the models are evolved through by applying the evolutionary operators to create an offspring population, which represents new potential models for the given classification task. Thereafter, the models that satisfy the best trade-off between the two objectives to be optimized are stored in an external archive. This process is repeated until a stopping criterion is reached. At the end of the search, a final classification model is constructed, which is used for predicting the class labels of unknown samples.

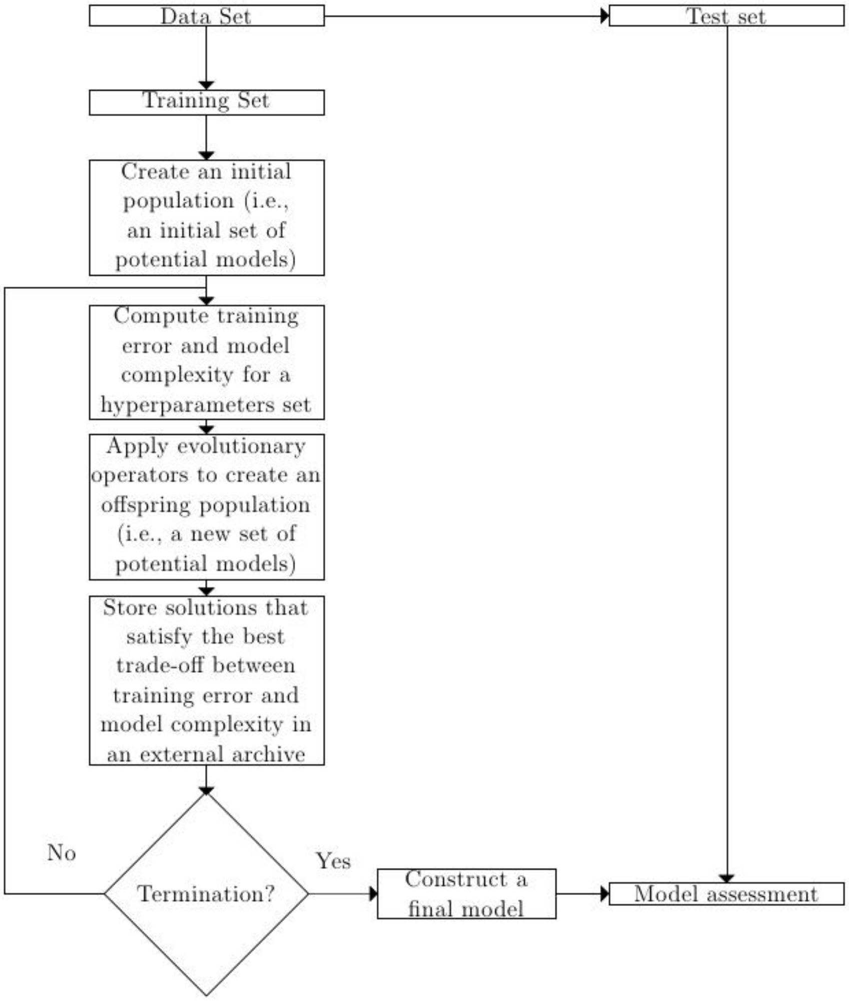

Data Set
Test set
Training Set
Create an initial population (i.e., an initial set of potential models)
Compute training error and model complexity for a hyperparameters set
Apply evolutionary operators to create an offspring population (i.e., a new set of potential models)
Store solutions that satisfy the best trade-off between training error and model complexity in an external archive
No
Termination?
Yes
Construct a final model
Model assessment

Figure 1: General diagram for the multi-objective model selection process.

In the proposed approach, we consider five different model types: support vector machines (SVMs), neural networks (NNs), random forest (RF), j48 and random trees (RTs). All of these are available in the WEKA [**?**] toolbox, and LibSVM [**?**] for the SVM. Table 1 shows the learning algorithms considered in our study. It also shows for each method the corresponding hyper-parameters. In the rest of the section we explain our proposal in detail.

Table 1: Description of the learning methods considered in our study.

Learn. Alg.	Hyper-parameters	Description
J48	Confidence: A confidence threshold for pruning. K: A minimum number of instances per leaf.	It constructs a pruned or unpruned C4.5 decision tree.
NNs	Neurons: Number of neurons in the hidden layer lr: Learning Rate for the backpropagation algorithm. Momentum: Momentum Rate for the backpropagation algorithm. Epochs: Number of epochs to train through. Seed: The value used to seed the random number generator.	It constructs a multi-layer perceptron using the backpropagation algorithm.
RF	Trees: Number of trees to build. K: Number of features to consider. Depth: The maximum depth of the trees. Seed: The value used to seed the random number generator.	It constructs a forest of random trees.
RT	K: Number of features to randomly investigate. Depth: The maximum depth of the tree. Seed: Seed used for the random number generator.	It constructs a tree that considers K randomly chosen attributes at each node. It does not perform a pruning step.
SVMs	Kernel: The kernel type to be used. d: The degree of a polynomial kernel. γ: Gamma value of an RBF kernel. B: A bias value in polynomial kernel. C: The complexity constant. Seed: Seed for the random number generator.	It constructs a support vector classifier.

3.1. Multi-Objective Evolutionary Algorithms

A multi-objective optimization problem (MOOP) can be stated as follows:

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}), \ldots, f_l(\mathbf{x})\right]^T \\ &\text{subject to } \mathbf{x} \in \mathcal{X} \end{aligned} \tag{6}$$

where $\mathbf{x} = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$ is a decision variables vector, $f_i(\mathbf{x})$, $i = 1, \ldots, l$, are the objective functions, and $\mathcal{X}$ is the set of feasible solutions.

When the objectives in an MOOP are in conflict, there is not a single solution that would be the best for all of them. Pareto optimality provides a framework for dealing with such cases. We say that a solution $\mathbf{x}^1$ dominates a solution $\mathbf{x}^2$ (denoted by $\mathbf{x}^1 \preceq \mathbf{x}^2$) 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, i.e.,

$$\forall i : f_i\left(\mathbf{x}^1\right) \leq f_i\left(\mathbf{x}^2\right) \wedge \exists i : f_i\left(\mathbf{x}^1\right) < f_i\left(\mathbf{x}^2\right) \tag{7}$$

A solution $\mathbf{x}^*$ is Pareto optimal if there is not another solution $\mathbf{x}' \in \mathcal{X}$ such that $\mathbf{x}' \preceq \mathbf{x}^*$. The set of all Pareto optimal solutions is called Pareto optimal set, and the image of this set in objective function space is referred to as the Pareto Front.

Evolutionary algorithms are stochastic search techniques inspired in Darwin's evolutionary theory. These algorithms have been successfully used for solving MOOPs, mainly because they can obtain several elements of the Pareto optimal set in a single run, and because they are less susceptible than mathematical programming techniques to the shape and continuity of the Pareto front [**?** **?**].

Since the seminal work of Schaffer [**?**], a considerable number of multi-objective evolutionary algorithms (MOEAs) have been proposed, such as: Multi-Objective Genetic Algorithm (MOGA) [**?**], Niched Pareto Genetic Algorithm (NPGA) [**?**], Strength Pareto Evolutionary Algorithm (SPEA) [**?**], and its improved version SPEA2 [**?**], Pareto Archived Evolutionary Strategy (PAES) [**?**], Non-dominated Sorting Genetic Algorithm (NSGA) [**?**]

and NSGA-II [?], and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [?], among others. A comprehensive review of MOEAs can be found in [? ? ?].

In the context of multi-objective model selection, evaluating the objectives is computationally expensive, inasmuch as each candidate model has to be trained and tested, possibly several times. Furthermore, in the model selection problem, the optimal model is unknown *a priori*. The latter makes necessary that the generated solutions are diverse to each other in order to use *a posteriori* processing for constructing a final model. Taking this information into account, in this study we used MOEA/D, due to its high performance over different difficult problems [?]. Additionally, MOEA/D has a lower computational complexity than other MOEAs (such as the NSGA-II), and is able to provide well-distributed solutions along the Pareto front [?].

3.1.1. MOEA/D

MOEA/D [?] is one of the most recent MOEAs reported in the state of the art. It is based on the idea of decomposing an MOOP into a number of scalar objective optimization problems, also called subproblems, through a weighted aggregation of the objectives. MOEA/D minimizes all these subproblems iteratively in a single run. A neighborhood relation based on the distance of the aggregation weights vectors is defined among the subproblems. The optimal solutions to two neighboring subproblems should be very similar. Each subproblem has its best solution found so far in the population and is optimized in MOEA/D by using information from its neighbors.

A description of MOEA/D is presented in Algorithm 1. MOEA/D starts by creating an empty external population (EP) (step 1), which is used to store the non-dominated solutions found so far during the search. In MOEA/D, the T closest weight vectors in $\left\{\lambda^1, \ldots, \lambda^N\right\}$ to a weight vector λ^i constitute the neighborhood of λ^i. Thus, for each vector λ^i, it is computed the Euclidean distance between it and the others, and their T closest weight vectors are determined, where T defines the neighborhood size. The indexes of such T closest weight vectors are assigned to $B\left(i\right)$ (step 2). Next, the initial population of N individuals is randomly created (step 3). The individuals of the initial population are evaluated by using the fitness functions. For each objective, the lowest value attained by the individuals in the initial population is used to initialize a reference vector $\mathbf{z}$ (step 4).

The process to generate a new solution $\mathbf{y}$ comes here. To do this, the parents are randomly selected from the neighborhood, to which evolutionary operators (such as crossover and mutation) are applied to create $\mathbf{y}$ (step 7). In case $\mathbf{y}$ violates any constraint, the next step consists of applying some repair heuristic in order to make $\mathbf{y}$ a feasible solution $\mathbf{y}'$ (step 8). Next, reference vector $\mathbf{z}$ is updated in case an objective with a lower value is found (step 9). After that, the neighboring solutions are updated by considering all the neighbors of the i^{th} subproblem and replacing $\mathbf{x}^j$ by $\mathbf{y}'$ if $\mathbf{y}'$ performs better than $\mathbf{x}^j$ (step 10). The external population EP that was initialized in step 1 is updated by the new generated solution if and only if this solution is non-dominated with respect to those that are in EP. Moreover, if the new solution dominates any of those stored in EP, such solutions are removed from EP (step 11). Steps 7 to 11 are repeated while a stopping criterion is not reached. A detailed description of MOEA/D is beyond the scope of this paper, but interested readers are referred to [**?**] for more information about this approach.

Algorithm 1 MOEA/D [**?**]

```
Require: A stopping criterion,
         $N$: number of subproblems considered in MOEA/D,
         A uniform spread of $N$ weight vectors: $\lambda^1, \dots, \lambda^N$,
         $T$: the number of weight vectors in the neighborhood of each weight vector
Ensure: $EP$: an external population
Initialize $EP \rightarrow \emptyset$
Compute the Euclidean distance between any two weight vectors and then work out the $T$ closest weight vectors to each weight vector. For each $i = 1, \dots, N$, set $B(i) = \{i_1, \dots, i_T\}$, where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are the closest weight vectors to $\lambda^i$.
Generate an initial population $\mathbf{x}^1, \dots, \mathbf{x}^N$
Initialize $\mathbf{z} = [z_1, \dots, z_m]$ by setting $z_j = \min_{1 \leq i \leq N} f_j\left(\mathbf{x}^i\right)$
while stopping criterion is not satisfied do
    for $i = 1$ to $N$ do
        Randomly select two indexes $k, l$ from $B(i)$, and then generate a new solution $\mathbf{y}$ from $\mathbf{x}^k$ and $\mathbf{x}^l$ by using genetic operators.
        Apply a repair/improvement heuristic on $\mathbf{y}$ to produce $\mathbf{y}'$.
        Update $\mathbf{z}$, for each $j = 1, \dots, m$ if $z_j > f_j(\mathbf{y})$, then set $z_j = f_j(\mathbf{y})$
        Update of neighboring solutions: For each index $j \in B(i)$, if $g^{te}\left(\mathbf{y}'\lambda^j, \mathbf{z}\right) \leq g\left(\mathbf{x}^j\lambda^j, \mathbf{z}\right)$ , then set $\mathbf{x}^j = \mathbf{y}'$, $FV^j = F(\mathbf{y}')$
        Update of EP: Add $F(\mathbf{y}')$ to $EP$ if it is non-dominated with respect to the vectors stored in EP, and remove from $EP$ the vectors dominated by $F(\mathbf{y}')$.
    end for
end while
```

As evolutionary operators we used a differential evolution crossover-mechanism [**?**], and polynomial-based mutation [**?**]. In the differential evolution operator adopted ,each element $\bar{y_j}$ of a new solution $\bar{\mathbf{y}} = [\bar{y_1}, \dots, \bar{y_n}]$ is generated as follows:

$$\bar{y_j} = \begin{cases} x_j^i + F \times \left(x_j^k - x_j^l\right) & \text{with probability } CR, \\ x_j^i & \text{with probability } 1 - CR \end{cases} \tag{8}$$

where CR and F are two control parameters.

Polynomial-based mutation generates the new solution, $\mathbf{y} = [y_1, \ldots, y_n]$ as follows:

$$y_j = \begin{cases} \bar{y_j} + \Delta_j \times (U_b - L_b) & \text{with probability } pm \\ \bar{y_j} & \text{with probability } 1 - pm, \end{cases} \tag{9}$$

where pm is the probability of mutation, U_b and L_b are the upper and lower bounds, respectively, and Δ_j is a polynomial distribution for random numbers generation in the following way:

$$\Delta_j = \begin{cases} (2 \times \text{rand})^{\frac{1}{\eta+1}} - 1 & \text{if rand } < 0.5 \\ 1 - [2 \times (1 - \text{rand})]^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \tag{10}$$

where “rand” is a uniform random number in $[0, 1]$, and η is the distribution index for the mutation operator.

One of the key issues in MOEA/D is the method used for decomposing the MOOP into a number of scalar objective problems. A simple method in this regard is the weighted sum approach, but it has the disadvantage of not being able to generate concave portions of a Pareto front [**?**]. We used instead, the Tchebycheff approach [**?**], due to the fact that it is more robust to a concave shape of the Pareto front than the weighted sum approach. However, any other decomposition approach could be used in MOEA/D.

Using the Tchebycheff approach, an MOOP is decomposed into a N scalar optimization subproblem as follows:

$$\text{minimize } g\left(\mathbf{x} \mid \lambda, \mathbf{z}^{*}\right) = \max_{1 \leq i \leq m} \left\{\lambda_i \mid f_i\left(\mathbf{x}\right) - \mathbf{z}_i^{*} \mid\right\} \tag{11}$$

where $\lambda = [\lambda_1, \ldots, \lambda_m]$ is a weight vector, $\mathbf{z}^{*} = [z_1, \ldots, z_m]$ is a reference point, and m is the number of objectives in the problem.

In the literature, several repair heuristics have been proposed [**?**]. Nevertheless, we formulate the multi-objective model selection problem as an unconstrained one. For this reason, a repair heuristic is not used in our study; therefore, step 8 is not performed. The following sections explain the proposed approach for multi-objective model selection using MOEA/D.

3.2. Representation

Evolutionary Algorithms work with a population of solutions. In our proposed approach, each solution, also called individual, represents a potential model for the classification task. As previously stated, the task approached by our model selection proposal is to choose among a pool of learning algorithms and their corresponding hyper-parameters. To achieve this task, each model (the learning algorithm plus its hyper-parameters) should be encoded in a D-dimensional vector. In this study, each solution is encoded in a 7-dimensional vector as follows:

$$\mathbf{x}^{i} = \left[x_m^{i}, x_{hp_1}^{i}, \ldots, x_{hp_{D-1}}^{i}\right] \tag{12}$$

where x_m^{i} controls the learning algorithm, and $\left[x_{hp_1}^{i}, \ldots, x_{hp_{D-1}}^{i}\right]$ represents the hyper-parameters for the learning algorithm.

Since the hyper-parameters are numerical values, and in order to have them as accurate as possible, we used a real encoding for the individuals. By applying the evolutionary operators, such as crossover (Eq. (8)) and mutation, the individuals are evolved in an iterative process. One should note that there are some discrete variables, such as x^i_m, which represent a learning algorithm, or $x^i_{hp_x}$, which could represent a kernel type in the SVM case. For the evolutionary operators, this type of variable is internally treated as a real number, but we round it off to its nearest allowable discrete value.

From Table 1, we can observe that different learning algorithms require different hyper-parameters. For example, in J48 two hyper-parameters are considered, whilst in SVMs there are six hyper-parameters. Thus, x^i_m and the six hyper-parameters are the seven variables in our representation. The configuration given by an individual and the training set are used to fit a model.

The seven variables constitute the search space for our problem. An initial population is created using the Latin Hypercube sampling technique [**?**] with the aim of having a representative distribution of solutions in the search space. Once the initial population is created, it is used for producing an offspring population by applying the evolutionary operators until a stopping criterion is satisfied, and a set of non-dominated solutions is obtained.

3.3. Fitness Functions

In the proposed approach, the model selection problem is tackled as a multi-objective optimization problem, and an MOEA is used for solving it. Since the search is based on a population of solutions, it is required to have a way to measure how well a model performs in order to choose the best one.

The fitness function is in charge of this, and its definition is a crucial issue in model selection. One could try to estimate the effectiveness of the model based on the error on the training samples, also known as empirical error, and the optimization problem would try to minimize that error. Nonetheless, this would result in optimistic estimations of model performance, and could lead to highly complex models, causing the problem known as over-fitting. In other words, the model has a good performance on the training samples, but not on unseen samples (see [**? ? ?**] for more information about this problem). To overcome this handicap, the model complexity should also be controlled. Taking this into account, in this paper we propose not only to minimize the error on the training data, but also to minimize the model complexity.

The VC-dimension is a measure of the capacity of the model, which is related to its complexity, and it is used in the present study. The fitness functions defined for our problem are stated as follows:

$$\begin{aligned} err &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(y_i, y_i^*\right) \\ complexity &= \underset{h}{\operatorname{argmin}}\sum_{i=1}^{k}\left[\overline{\xi}\left(n_i\right) - \Phi\left(n_i/h\right)\right]^2 \end{aligned} \tag{13}$$

where N is the number of samples in the training set, y_i is the class label, y_i^* is the class predicted by the model, $\mathcal{L}\left(y_i, y_i^*\right)$ is a loss function, $\xi\left(n_i\right)$ is the experimental maximum deviation error rate of two observed independent labeled data sets, and $\Phi\left(n_i/h\right)$ is the expectation of the largest deviation error between two sets (see Section 2 for details about complexity estimation). We used the 0/1 loss function because it is well suited for classification tasks.

The 0/1 loss function is defined as:

$$\mathcal{L}\left(y_i, y_i^*\right) = \begin{cases} 1 & \text{if } y_i^* \neq y_i \\ 0 & \text{if } y_i^* = y_i \end{cases} \tag{14}$$

In consequence, the goal of performing this optimization is to simultaneously minimize the training error and model complexity. The outcome of this optimization step is a set of potential models that satisfies the best trade-off between the objectives, from which a model should be chosen. The next section explains how we approach this issue.

3.4. Constructing a Final Model

Once the evolutionary search is completed, a set of non-dominated solutions is obtained. Mathematically, all of them are equally acceptable solutions of the multi-objective optimization problem and, in our case, each of them represents a potential model for a given classification task. Therefore, it is desirable to select one model to be used to predict new samples from such set. In model selection for classification tasks, we have to choose the model with the highest possible generalization capability. Nevertheless, it is not clear what classification model from the non-dominated set is the "best" one. In this paper, we studied three strategies for constructing a final classification model, which are explained in the rest of this section.

3.4.1. Choosing a Single Model

As we previously stated, for our problem we seek the solution with the best possible generalization ability. In order to identify such solution, we studied the performance of the non-dominated solutions on unseen samples.

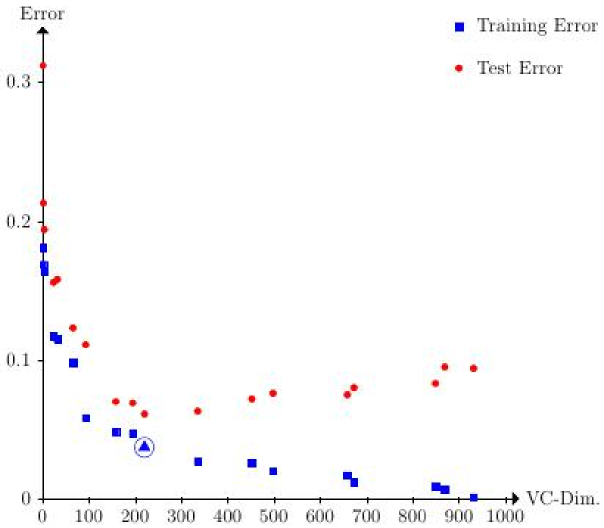


Figure 2: Behavior of non-dominated solutions on training samples and test samples

We noticed that the best solutions are those located at the knee of the curve, while solutions with low complexity and high complexity lead to models with a poor generalization performance. Both problems are well-known in machine learning as under-fitting and over-fitting, respectively. Figure 2 depicts an example of this behavior for a particular case. It also shows the trade-off between the training error and the model complexity, such that by increasing the model complexity, the training error is reduced.

We empirically found that in most cases, the solution with a good generalization performance is the one nearest to a reference point z^*, which is defined as:

$$z_j^* = \min_{1 \leq i \leq L} f_j\left(x^i\right) \text{ for } j = \{1, 2\} \tag{15}$$

where L is the cardinality of the non-dominated set.

As it is shown in Figure 2, the objectives are measured in different scales.

In order to avoid that one objective has a higher impact than the other one in the distance computation, both objectives are firstly normalized in the range 0 to 1. Subsequently, the Euclidean distance is computed on the normalized objective vector. In the end, the closest solution is chosen[1], and is used to predict future samples of the problem. One should note that since the objectives are normalized, the reference vector $\mathbf{z}^*$ corresponds to the $(0, 0)$ point. Figure 2 shows with a triangle the solution that would be chosen with this strategy.

3.4.2. Ensemble of the Whole Non-Dominated Front

Ensembles of classifiers are based on the idea of combining the predicted outputs from different individual classification models. They have been successfully used for improving the performance of individual models [**? ?**]. One should remember that the output of the MOEA is a set of non-dominated solutions. Based on this, one might ask why not to construct an ensemble with the potential models (solutions) in the non-dominated front instead of choosing a single model.

Now the problem is to determine which models should be used in the ensemble. In the absence of knowledge about the preferences, all non-dominated solutions are equally good. With these ideas in mind, we used all of them for constructing an ensemble. One should recall that the non-dominated front could contain models with a very low complexity or a very high complexity, which could lead to over-fitted and under-fitted models, respectively. In a

---

[1] One should note that this is a suggestion. If the model, however, does not satisfy the performance requirements, any other solution can be chosen from the non-dominated set without performing a new search.

majority vote scheme all models are equally important, even those whose generalization performance could not be good enough. For that reason, we argue in favor of a weighted linear combination approach over a majority vote one[2]. This is because models with a less satisfactory performance are kept in the ensemble, but with a lower weight. Therefore, the final prediction given by the classification model would consist of a weighted linear combination of the individual predictions, as follows:

$$y^* = \sum_{i=1}^{L} \omega_i y_i^* \tag{16}$$

where L is the number of single classification models, and is equal to the cardinality of the non-dominated set, y_i^* is the prediction given by the i^{th} single model, and ω_i is the weight associated to that model. The weight vector $\omega = [\omega_1, \ldots, \omega_L]$ is a normalized vector, whose values depend on the distances between the reference point (defined by Equation (15)) and the potential solution. The normalized weight vector is computed as follows:

$$\omega_j = \frac{\frac{1}{d_j}}{\sum_{i=1}^{L} \frac{1}{d_i}} \tag{17}$$

where d_j is the Euclidean distance between the j^{th} solution and the reference point $\mathbf{z}^*$.

Alternatively, one could think in exploring other well-known ensemble techniques, such as bagging or boosting, or even trying to optimize the ensemble performance. Nevertheless, the main focus in this stage of our study

---

[2] Please note that the majority vote ensemble is a special case of the weighted linear combination approach.

is not the ensemble itself, but the optimization of the hyper-parameters for the classifiers. Our aim is to find the hyper-parameters that satisfy the best trade-off between the objectives, as well as to find ways to construct a classification model from the resulting non-dominated front, which could then be used for the prediction of unknown samples. These issues could be explored as part of our future work.

3.4.3. Ensemble of Some Solutions in the Non-Dominated Front

It is well-known from machine learning that for constructing an ensemble, two conditions have to be satisfied: the individual models should be accurate (i.e., the performance should be better than a random guessing), and they have to be diverse (i.e., single models should incur in different errors on new samples) among them [**?**]. This issue is explored in the third strategy for the final model construction. Therefore, for constructing an ensemble in this third strategy, we need to choose a subset of potential models in the non-dominated front, such that these models are accurate and diverse among them.

We would like to remark that the models were optimized during the optimization step, and the ones that satisfy the best trade-off are obtained as a result of this. Thus, we can assume that the models in the resulting non-dominated set are accurate (i.e., their performance is better than a random one). By making this, the problem is reduced to choosing a subset of these models that are as diverse as possible among them, which are used in the ensemble. In order to determine such subset, a forward aggregation approach is used. In the forward aggregation approach, we start by adding the solution closest to the reference point, z^* (as it was defined in Equation (15)). After that, a second model that maximizes the diversity is added, followed by a third model and so on. This process is repeated while the diversity among the models is not deteriorated.

Under the adopted approach, a diversity measure is required. There are a number of diversity measures reported in the literature, and a review of these can be found in [**?**]. In this study, we used one based on entropy, but any other can also be used. This measure is defined as follows:

$$E = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{L - \lceil \frac{L}{2} \rceil} \min \left\{ l\left(\mathbf{s}_i\right), L - l\left(\mathbf{s}_i\right) \right\} \tag{18}$$

where N is the number of samples, L is the number of individual models, and $l\left(\mathbf{s}_i\right)$ is the number of models that correctly predict the sample $\mathbf{s}_i$. This measure ranges between 0 and 1, where 0 indicates no difference and 1 is the highest possible diversity.

Finally, the prediction given by the ensemble is based on the weighted linear combination of the predictions of the individual models, as it is shown in Equation (16).

3.5. Final Remarks

One should note that under the proposed approach the expert's knowledge is not exploited. This could be a key issue in order to improve the performance of the models. In the agnostic learning vs. prior knowledge challenge [**?**] it was shown that, even when prior knowledge outperforms agnostic learning for most of the problems, there were some problems for which agnostic learning performs better than prior knowledge. In consequence, it is difficult to know when it is going to be better to use this kind of knowledge. Based on the results of this challenge, the organizers concluded that the agnostic learning approach is very powerful. Furthermore, there are cases in which this knowledge could not be available. For these reasons, we bet in favor of not using expert's knowledge.

Notwithstanding, if prior domain-knowledge is available, this could be integrated in several manners in the proposed approach. For example, based on the characteristics of the problem at hand, an expert could suggest that a particular learning algorithm would be more suitable than the others. This suggestion could be used for fixing *a priori* the learning algorithm. Thus, the search would be performed under its hyper-parameters set, reducing the search space. The expert's knowledge could also be used for choosing a single solution from the non-dominated front. Another way in which prior knowledge could be used would be during the ensemble construction, through the assignment of weights to each classifier. For our experiments, we assumed that expert's knowledge is not available. Next section describes the experiments and results obtained by our proposal.

4. Experiments and Results

In this section, we describe the experiments performed as well as the results obtained by our proposal using a benchmark test suite. We present a comparative study between the three proposed strategies for constructing a final classification model from the resulting non-dominated front. We also present statistical tests to validate our results when compared to other approaches reported in the specialized literature.

4.1. Experimental Settings

In order to evaluate the feasibility of our proposal in the model selection problem, we used the IDA benchmark[3] data sets introduced by [?]. This benchmark is well-suited for this purpose and it has been widely used in several related studies (e.g. [? ? ? ? ? ?]). Table 2 describes the suite of thirteen benchmark data sets, which are diverse in the number of samples and features. These data sets correspond to binary classification problems[4], and have been previously pre-processed in [?], in which the samples with missing values have been removed and all features have been standardized, i.e., all features have mean zero and a standard deviation of one.

The typical experimental protocol used with this benchmark was introduced by [?], and is sometimes called the median protocol. The median protocol consists on performing the model selection on the first five partitions. After that, the median values of the hyper-parameters resulting from those partitions are taken, which are used to estimate the error rate for each partition. However, this protocol can introduce an optimistic bias into the performance estimation [?]. In order to overcome this bias in the performance evaluation, the model selection process is performed independently for each partition of each data set; this protocol is known as the internal protocol. The use of the internal protocol leads to a total of 1140 model selection experiments.

The parameters configuration used in our experiments is the following.

---

[3] Available at `http://www.raetschlab.org/Members/raetsch/benchmark`

[4] Without loss of generality, the experiments are performed on binary classification problems. Multi-class classification problems can be approached with multiple binary classifiers.

Table 2: Details of the data sets used in our experiments. The table shows the number of features for each data set and the number of instances for training and testing for each replication of each data set.

ID	Data set	Feat.	Training Samples	Testing Samples	Replications
1	Banana	2	400	4900	100
2	Breast Cancer	9	200	77	100
3	Diabetes	8	468	300	100
4	Flare Solar	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

For the differential evolution crossover, we fixed the value of $F = 0.5$, $CR = 0.7$. With respect to the mutation operator, the mutation probability pm was fixed to 0.1 and index distribution to 20. These parameters were experimentally tuned by evaluating the performance under each configuration of $pm = \{0.1, 0.2, 0.3\}$, $CR = \{0.5, 0.6, 0.7, 0.8, 0.9\}$, and $F = \{0.3, 0.4, 0.5, 0.6, 0.7\}$ on the first five partitions of splice data set, one of the largest both in number of training samples and features. The stopping criterion is defined as performing 1,000 fitness functions evaluations. To achieve this, the population size is set to 20, and the number of generations to 50. Moreover, the VC-dimension for each model is estimated experimentally. Thus, it is required to train and to test a number of times each model. In our experiments, we fixed this number to 10. Next, we present the results reached by our proposal, comparing the proposed strategies for a final model construction and with other evolutionary and non-evolutionary approaches for model selection.

4.2. Experimental Results and Discussion

In this section, we present the results obtained by our proposal (MOMTS, Multi-Objective Model Type Selection) so as to demonstrate its feasibility for the model selection problem. Table 3 shows the average error rates and standard deviations on the test sets attained for the three proposed strategies for constructing a final model, i.e., choosing a single model (MOMTS-S1), ensemble of the whole non-dominated front (MOMTS-S2), and the ensemble of some solutions in the non-dominated front (MOMTS-S3). As a baseline, we report the results obtained by using random forest (RF) with its default hyper-parameters, which is a standard learning algorithm based on an ensemble of decision tress.

We compare our results with those reported by Cawley and Talbot [**?**], who used Bayesian regularization at the second level of inference, adding a regularization term in the model selection criterion. Furthermore, in order to make a fair comparison, we also performed experiments considering approaches that consider different learning algorithms and their hyper-parameters during the model selection process. For that sake, we used PSMS [**?**] and SUMO [**?**], which are two evolutionary approaches that were proposed for model selection. PSMS is a single-objective approach based on a particle swarm optimizer that minimizes the error rate estimated through k fold cross validation. SUMO adopts a genetic algorithm as a search engine and the fitness function can be defined as minimizing some measure obtained via some evaluation strategy; in our case the measure was fixed to be the error rate and the evaluation strategy to be the k fold cross validation. In both cases, the number of particles/individuals was set to 20, and the number of iterations/generations to 50, resulting in 1,000 fitness function evaluations. This is the same number of fitness function evaluations set for our proposed approach. The reference results used the same 100 partitions (20 in case of the image and splice data sets) for training and testing, and also used the same experimental protocol (i.e., the internal protocol), making the results directly comparable.

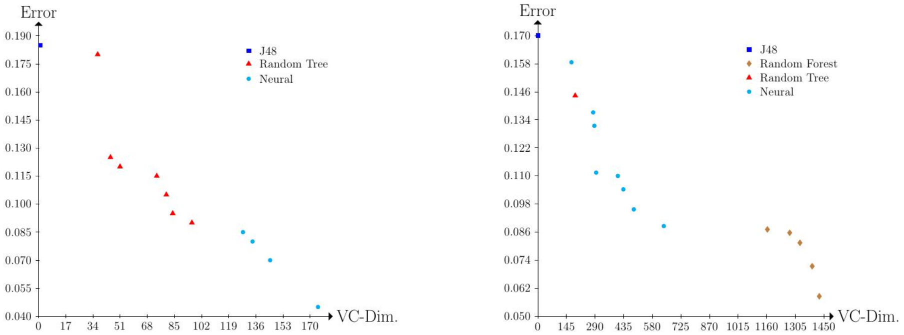
Figure 3 shows the non-dominated fronts generated by our proposal for some data sets in a particular trial. It is expected that these non-dominated fronts are an approximation to the true Pareto front. We can observe that different solutions are distributed along the non-dominated front. We can also note that the non-dominated front is formed by solutions that represent different learning algorithms. Each one of these solutions corresponds to models with different levels of complexity. Although, in some cases, a learning algorithm is represented by more than one solution, these correspond to different configurations of its hyper-parameters, which could lead to diverse models. Thus, in the resulting non-dominated front there are models, which are learned by different learning algorithms with a different hyper-parameters configuration[5].

---

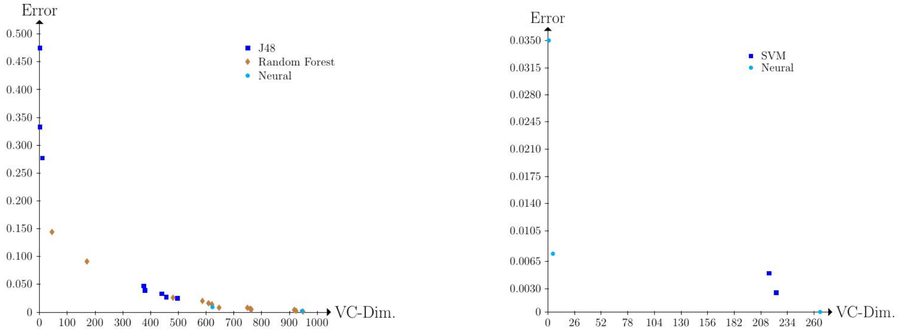
[5] The full list of the models generated by our proposed method for each partition of each data set is available at `http://ccc.inaoep.mx/~arosalesp/Resources/Models.zip`

Table 3: Results obtained by the proposed approach, and those obtained by random forest (RF), LS-SVM using Bayesian regularization, PSMS, and SUMO. The reported results are the average and standard error on test sets over the 100 or 20 replications of each data set. The best result for each data set is shown in **boldface**.

Data Set	RF	LS-SVM-BR [?]	PSMS [?]	SUMO [?]	MOMTS-S1	MOMTS-S2	MOMTS-S3
Banana	13.14 ± 0.069	10.59 ± 0.050	11.08 ± 0.083	10.88 ± 0.074	14.34 ± 0.105	**10.48 ± 0.046**	12.91 ± 0.160
Breast Cancer	27.94 ± 0.412	27.08 ± 0.494	33.01 ± 0.658	26.27 ± 0.448	29.89 ± 0.736	**25.61 ± 0.593**	27.82 ± 0.676
Diabetes	25.83 ± 0.212	23.14 ± 0.166	27.06 ± 0.259	23.49 ± 0.177	28.34 ± 0.318	**23.08 ± 0.174**	25.66 ± 0.214
Flare Solar	36.44 ± 0.173	**34.07 ± 0.171**	34.81 ± 0.173	38.47 ± 0.573	34.90 ± 0.224	34.59 ± 0.189	34.52 ± 0.214
German	25.16 ± 0.240	**23.59 ± 0.216**	30.10 ± 0.720	23.83 ± 0.213	28.30 ± 0.274	23.67 ± 0.224	25.89 ± 0.218
Heart	20.26 ± 0.382	**16.19 ± 0.348**	20.69 ± 0.634	17.67 ± 0.355	23.14 ± 0.542	16.48 ± 0.241	18.75 ± 0.351
Image	**2.09 ± 0.095**	2.90 ± 0.154	2.90 ± 0.112	2.45 ± 0.126	3.79 ± 0.226	2.24 ± 0.123	3.03 ± 0.246
Ringnorm	9.57 ± 0.095	**1.61 ± 0.015**	7.98 ± 0.660	1.72 ± 0.071	2.66 ± 0.079	2.49 ± 0.074	3.02 ± 0.164
Splice	8.50 ± 0.210	10.91 ± 0.154	14.63 ± 0.324	10.94 ± 0.146	7.43 ± 0.373	**4.84 ± 0.156**	6.71 ± 0.269
Thyroid	5.89 ± 0.273	4.63 ± 0.218	4.32 ± 0.235	4.85 ± 0.224	6.48 ± 0.350	**4.00 ± 0.194**	6.11 ± 0.347
Titanic	22.51 ± 0.124	22.59 ± 0.120	24.18 ± 0.193	34.99 ± 0.523	26.53 ± 0.127	**22.08 ± 0.085**	22.22 ± 0.100
Twonorm	9.02 ± 0.074	2.84 ± 0.021	3.09 ± 0.127	**2.55 ± 0.022**	5.21 ± 0.555	3.73 ± 0.179	5.70 ± 0.679
Waveform	13.75 ± 0.071	**9.78 ± 0.044**	12.80 ± 0.325	**9.78 ± 0.060**	11.34 ± 0.180	9.93 ± 0.043	10.95 ± 0.256



(a) Breast-Cancer (b) german



(c) splice (d) twonorm

Figure 3: Non-dominated fronts generated from a particular trial of the proposed method. The solutions in the non-dominated front represent different learning algorithms with different hyper-parameter configurations.

4.2.1. Comparison of Strategies for Constructing the Final Model

The results of the proposed strategies for constructing a final model are shown in the last three columns of Table 3. The results of the ensemble approaches outperformed those obtained when a single model is chosen in most cases. The single model was better than the ensemble of some solutions in the non-dominated front in 2 out of 13 data sets (ringnorm and twonorm data sets). This seems reasonable insomuch as it is well-known that using an ensemble of models helps to improve the predictions. Between the two ensembles approaches, the one based on the whole non-dominated front showed better results in 12 out of 13 data sets.

An ANOVA statistical test with a 95% of confidence is applied so as to determine if the difference between the proposed strategies is significant, and Tukey's test is used as a post-hoc test. The results obtained by this test are shown in Table 4. In this table, we can note that the analysis of variance showed a statistical significance difference for most of the data sets, except for the flare solar one, to which the post-hoc test is not applied. According to the pairwise comparisons, we can also note that the ensemble of some solutions of the front (MOMTS-S3) performs significantly better than the single model approach in 6 out of 13 data sets (banana, diabetes, german, heart, image, and titanic). On the other hand, the ensemble of the whole front (MOMTS-S2) showed to be significantly better than the single model approach in 10 out of 13 benchmark data sets (banana, breast cancer, diabetes, german, heart, image, splice, thyroid, titanic, and waveform), and also significantly outperformed the ensemble of some solutions approach in 10 out of 13 data sets (banana, diabetes, german, heart, image, ringnorm, splice, thyroid, twonorm, and waveform). It seems clear that the ensemble of the whole front approach is the best of the three approaches. However, for assessing the statistical difference among them over the different data sets, Demšar [**?**] recommends the Friedman's test for comparing multiple classifiers over multiple data sets. This test is performed with a 95% of confidence, and the Nemenyi test as the post hoc test. According to these tests, the ensemble of the whole front approach is statistically superior to the others.

Table 4: F-statistic obtained from the ANOVA test and q-values from the Tukey HSD test for performing all possible pairwise comparisons among the proposed strategies for a final model construction. The critical values at the 0.05 level for ANOVA test are 3.16 ($F\,(2,57)$) for the image and splice data sets and 3.03 ($F\,(2,297)$) for the rest. The critical values at 0.05 level for the Tukey HSD test are 3.41 for the image and splice data sets (57 degrees of freedom) and 3.34 for the rest of the data sets (297 degrees of freedom). Cases that exceed the critical value are considered as a difference that is statistically significant at the fixed level and are marked with an asterisk (*).

	ANOVA	q Tukey HSD		
Data Set	F	MOMTS-S1 vs. MOMTS-S2	MOMTS-S1 vs. MOMTS-S3	MOMTS-S2 vs. MOMTS-S3
Banana	**294.899***	**33.967***	**12.584***	**21.384***
Breast Cancer	**10.178***	**6.380***	3.085	3.294
Diabetes	**117.12***	**21.643***	**11.027***	**10.616***
Flare Solar	0.932	−−	−−	−−
German	**93.107***	**19.293***	**10.042***	**9.251***
Heart	**72.394***	**16.734***	**11.032***	**5.705***
Image	**14.221***	**7.542***	**3.698***	**3.844***
Ringnorm	**5.690***	1.499	3.173	**4.672***
Splice	**22.736***	**9.238***	2.568	**6.670***
Thyroid	**19.140***	**8.110***	1.210	**6.900***
Titanic	**575.616***	**42.203***	**40.876***	1.328
Twonorm	**3.939***	2.864	0.948	**3.812***
Waveform	**15.937***	**7.731***	2.138	**5.593***

4.2.2. Comparison with Other Model Selection Approaches

Table 3 also shows the performance of random forest (RF), LS-SVM with Bayesian Regularization (LS-SVM-BR), which uses a radial basis function kernel reported by [**?**], as well as the results obtained with the application of PSMS and SUMO in the benchmark data sets. Due to the fact that the best results of our proposal were reached with the ensemble of the whole non-dominated front (MOMTS-S2), this approach is used for the comparison.

First, we compare with random forest (RF), which is used as a baseline to evaluate the benefits of performing model selection. From the reported results in Table 3, we can note that our proposal outperformed RF in 12 out of 13 data sets, being the image data set the only one in which RF performed better than our proposal.

Comparing the ensemble of the whole non-dominated approach (MOMTS-S2) with LS-SVM-BR, we can note that our proposal obtained better results in 7 out of 13 data sets (banana, breast cancer, diabetes, image, splice, thyroid, and titanic), but it was outperformed in the rest of the data sets. In addition, it is worth noting that an improvement of more than a 6% was reached in the splice data set.

With respect to PSMS, a single-objective approach that considers different learning algorithms and hyper-parameters selection, we note that our approach performed better than PSMS in 12 out of 13 benchmark data sets.

Comparing MOMTS-S2 with SUMO, another evolutionary approach for model selection, we note that MOMTS-S2 got better generalization performance on 10 out of 13 data sets.

Regarding statistical assessment, we applied the ANOVA test with a 95%

of confidence to compare the performance of the model selection approaches: LS-SVM-BR, PSMS, SUMO, and MOMTS-S2. Inasmuch as our goal is to compare the performance of the proposed approach with the reference results, the Dunnett's test is used as the post-hoc test. These statistical tests were conducted independently for each data set. The results of these are shown in Table 5, from which we can note that, for all cases, the analysis of variance revealed that there is a statistically significance difference at the 0.05 level, i.e., $p < 0.05$. Thus, the post-hoc test is applied.

According to the results shown in Table 5, statistical tests indicated that the proposed approach significantly outperformed LS-SVM-BR in 2 data sets (image and splice), and it was significantly outperformed in one data set (twonorm). Regarding SUMO, our method performed significantly better in 5 out of 13 data sets (banana, flare solar, splice, thyroid, and titanic), and it was significantly outperformed in the twonorm data set. On the other hand, our approach significantly outperformed PSMS in 10 of the benchmarks data sets (banana, breast cancer, diabetes, german, heart, image, ringnorm, splice, titanic, and waveform data sets), but it was significantly worse than PSMS in the twonorm data set.

Overall, our ensemble approach was able to get lower error rates than the other model selection methods in 7 out of 13 data sets, while the Bayesian regularization approach did the same in 5 out of 13 data sets, and SUMO in 2 out of 13 data sets. There is not a clear advantage of LS-SVM-BR and MOMTS-S2 when multiple data sets are considered. In order to statistically assess the four model selection approaches over the suite of 13 benchmark data sets, the Friedman test with a 95% of confidence was used. As a post-

Table 5: F-statistic obtained from the ANOVA test and t_d-values from Dunnett's test when MOMTS-S2 is compared with LS-SVM-BR, PSMS, and SUMO. The critical values at the 0.05 level for the ANOVA test are 2.72 ($F\,(3,76)$) for the image and splice data sets and 2.63 ($F\,(3,396)$) for the rest. For Dunnett's test, the critical values at the 0.05 level are 2.40 for the image and splice data sets (76 degrees of freedom) and 2.36 for the rest of the data sets (396 degrees of freedom). Cases that exceed the critical value are considered as a difference that is statistically significant at the fixed level and are marked with an asterisk (*).

	ANOVA (F-value)	MOMTS-S2 versus (t_d Dunnett)		
Data Set	F	LS-SVM-BR	PSMS	SUMO
Banana	**17.594***	1.191	**6.513***	**4.346***
Breast Cancer	**37.582***	1.875	**9.438***	0.841
Diabetes	**94.396***	0.216	**14.239***	1.466
Flare Solar	**38.338***	1.130	0.478	**8.435***
German	**62.129***	0.138	**11.188***	0.280
Heart	**23.938***	0.488	**7.080***	2.001
Image	**6.543***	**3.605***	**3.605***	1.156
Ringnorm	**83.108***	1.862	**11.621***	1.629
Splice	**377.328***	**20.560***	**33.152***	**20.660***
Thyroid	**2.881***	2.042	1.036	**2.752***
Titanic	**445.726***	1.250	**5.150***	**31.664***
Twonorm	**20.588***	**5.683***	**4.086***	**7.534***
Waveform	**78.231***	0.631	**12.074***	0.631

hoc test, we used the Bonferroni-Dunn test, to compare the performance of our proposal (MOMTS-S2) with the references. According to these tests, MOMTS-S2 is statistically better than PSMS, but there is not a statistical significance difference between MOMTS-S2 and LS-SVM-BR and MOMTS-S2 and SUMO.

Another aspect to take into consideration is the computational cost of the methods. In this regard, we compare the execution time required by our proposal against PSMS and SUMO. The average execution time of our proposal (MOTMS) was 54.29 minutes, whilst PSMS and SUMO required, respectively, 30.36 and 31.90 minutes on average. As one could note, our proposal is more time-consuming than the others. This is due to the fact that under the proposed approach two objectives have to be evaluated, while in PSMS and SUMO only a single objective is evaluated. In our case, estimating the model complexity through the VC-dimension implies to train and to test a model a number of times (10 times, according to the parameter that we used). Measuring the training error also implies to train and test such model. Hence, evaluating both objectives involves training and testing the model. This could represent a disadvantage with respect to the others, in terms of computational cost. Notwithstanding, we can argue that the task of model selection can be performed off-line. Moreover, since the models are in the non-dominated set, several strategies for constructing a final classification model can be performed without significantly increasing the computational cost. In addition to this, our proposal (MOTMS) has the advantage of getting highly competitive models, outperforming SUMO and PSMS in most of the data sets.

4.2.3. Discussion

From the experimental results shown in Table 3, we can note how over-fitting can be present in model selection. Among the three strategies for constructing a final model, those based on ensembles proved being beneficial, reducing the over-fitting effect. In spite of this, we cannot say that ensemble approaches completely solve the problem. We can also note that in most cases, the use of the solutions in the whole non-dominated front in an ensemble achieved a better generalization performance than when a subset of these are considered for the ensemble. This is a surprising result, since it was expected that by taking into account the diversity as a criterion for the ensemble construction, a better performance would be attained than when not doing so. Observing the diversity between both approaches, we noted that the whole non-dominated approach has better diversity, whereas the ensemble of a subset of solutions approach gets trapped in a local optimal solution.

A comparison with random forest (RF) showed the benefits of performing model selection against not doing so. This is specially remarkable in the ringnorm, splice, and twonorm data sets, in which an improvement above a 4% is reached. Even though a simple RF outperformed our proposal in the image data set, a pairwise comparison did not show a statistical significant difference between both. Therefore, it is worth performing the computational effort in order to construct a reliable classification model.

The ensemble of the whole front of the proposed approach (MOMTS-S2) significantly outperformed LS-SVM-BR on three benchmark data sets, but it was significantly worse in one data set. The greatest improvement was obtained in the splice data set, reducing the error rate in 6.07%. The greatest degradation was on the twonorm data set, with a difference of 0.89%. In spite of this, the overall performance of both approaches was similar. Neither the reference nor the proposed approach were significantly better than each other. It is interesting to note that MOMTS-S2 does not outperform LS-SVM-BR, which is a model selection method of the state of the art. This is due to the fact that MOMTS-S2 deals with different model types and their corresponding hyper-parameters. Nevertheless, we can argue that in LS-SVM-BR there are only two parameters to be optimized, while in

MOMTS-S2, seven parameters are taken into consideration, which considerably increase the search space and makes it harder to reach the “optimal” solutions with a lower number of iterations. Moreover, we gain generality without significantly over-fitting the models.

The experimental evaluations showed that MOMTS-S2 significantly outperformed PSMS. Although there was not a statistical significant difference between MOMTS-S2 and SUMO, when different data sets were considered, MOMTS-S2 significantly outperformed SUMO on several data sets. This gives evidence about the suitability of using a multi-objective approach in contrast to a single-objective approach, in spite of the computational cost of doing so. The experimental results showed that only minimizing the error rate estimated through k-fold cross validation could lead to choose a model with a small degree of over-fitting. In spite of this, the k-fold cross validation approach has the advantage of being free from the model assumptions, which makes it applicable to any learning algorithm and feasible to the full model selection formulation[6]. On the other hand, the use of the VC-dimension for controlling the model complexity, and avoiding over-fitting, as much as possible, also shows its potential for being applicable to different model types. Therefore, we believe that this approach can also be applicable to the full model selection formulation.

---

[6] The full model selection formulation consists of the task of finding the best combination of pre-processing, feature selection, and learning algorithms together with their parameters [**? ?**].

5. Conclusions and Future Work

In this paper, we have proposed a multi-objective approach for dealing with the problem of model selection. Our model selection approach takes into account both the learning algorithm and the hyper-parameters during the search process. We defined the training error, or empirical error, and the model complexity, which is estimated through the VC dimension, as the objectives to be optimized. The adopted formulation showed the following advantages: (i) the experimental way for measuring the VC dimension allows us to consider different learning algorithms in a general framework, and makes the method applicable to the full model selection problem; (ii) our proposal had a competitive performance over different benchmark data sets, making it applicable to problems from diverse domains; and (iii) the multiple non-dominated solutions obtained through the multi-objective formulation makes it easy to extend it to ensembles of models.

The experimental results showed that constructing an ensemble of models performs better than choosing a single model. Furthermore, the ensemble approach showed to be effective for reducing the effect of over-fitting. The advantages of the multi-objective approach over a single-objective formulation such as PSMS were also supported by the experiments. The experimental results also show that highly competitive classification models were generated by our proposal, without significantly degrading the performance in most cases. Hence, we can conclude that our proposed approach can be an useful framework for model selection in real world problems.

In the proposed approach, the VC dimension is experimentally estimated, making it computationally expensive. Alternatives such as approximating this value through surrogate-assisted evolutionary computation, or computing it using parallel computing would be interesting paths of future research. Other future research directions also include the extension to the full model selection problem, i.e., considering feature selection and data pre-processing into the model selection process. Studying more effective ways for constructing an ensemble (possibly) by using a second level of optimization would be another interesting direction for future research.

Acknowledgments

The authors would like to show gratitude to the anonymous reviewers, whose valuable comments helped to improve this paper. The first author is grateful for the support from CONACyT scholarship no. 329013.