Multi-objective evolutionary feature selection for online sales forecasting

Article *in* Neurocomputing · December 2016

DOI: 10.1016/j.neucom.2016.12.045

5 authors, including:

Fernando Jiménez
University of Murcia
77 PUBLICATIONS **987** CITATIONS
SEE PROFILE

Gracia Sánchez
University of Murcia
49 PUBLICATIONS **494** CITATIONS
SEE PROFILE

Jose M. Garcia
University of Murcia
229 PUBLICATIONS **1,885** CITATIONS
SEE PROFILE

Guido Sciavicco
University of Murcia
115 PUBLICATIONS **1,130** CITATIONS
SEE PROFILE

Multi-Objective Evolutionary Feature Selection for Online Sales Forecasting

Jiménez, F.*, Sánchez, G., García, J.M.

Faculty of Computer Science, University of Murcia (Spain)

Sciavicco, G.

Department of Mathematics and Computer Science, University of Ferrara (Italy)

Miralles, L.

Universidad Panamericana, Campus México, Facultad de Ingeniería, Augusto Rodin 498, México, Ciudad de México 03920, México

Abstract

Sales forecasting uses historical sales figures, in association with products characteristics and peculiarities, to predict short-term or long-term future performance in a business, and it can be used to derive sound financial and business plans. By using publicly available data, we build an accurate regression model for online sales forecasting obtained via a novel feature selection methodology composed by the application of the multi-objective evolutionary algorithm ENORA (Evolutionary NOn-dominated Radial slots based Algorithm) as search strategy in a wrapper method driven by the well-known regression model learner Random Forest. Our proposal integrates feature selection for regression, model evaluation, and decision making, in order to choose the most satisfactory model according to an *a posteriori* process in a multi-objective context. We test and compare the performances of ENORA as multi-objective evolutionary search strategy against a standard multi-objective evolutionary search strategy such as NSGA-II (Non-dominated Sorted Genetic Algorithm), against a classical backward search strategy such as RFE (Recursive Feature Elimination), and against the original data set.

Keywords: Multi-objective evolutionary algorithms, feature selection, random forest, regression model, online sales forecasting

*Corresponding author.

Email addresses: fernan@um.es (Jiménez, F.), gracia@um.es (Sánchez, G.), jmgarcia@um.es (García, J.M.), scvgdu@unife.it (Sciavicco, G.), lmiralles@up.edu.mx (Miralles, L.)

1. Introduction

Sales forecasting plays an essential role in Business Intelligence, which can be defined as the set of methodologies and techniques used for acquiring and transforming raw data into structured information for analytic purposes. *Forecasting* is the process of making predictions about the future based on past and present data [1]. Analytical techniques for forecasting can broadly be grouped into *regression techniques* and *machine learning techniques*. Sales forecasting uses historical sales figures, in association with products characteristics and peculiarities to predict short-term or long-term future performance, and it can be used to derive sound financial and business plans. *Online* sales (generally referred to as *e-commerce*) are nowadays very common, and are gradually becoming the most important selling channel; according to [2], the increase in online sales activities during the period 2002-2010 accounts for 18 per cent of the total growth in labour productivity. The online sales forecast problems can be thought as a multivariate regression problem, where dependent variable is numeric. Online *advertising* [3] accounts for a very high fraction of all advertising (for example, according to the *Annual IAB Internet Advertising Revenue Report*, the increase over the period 2011 - 2013 has been of around 17 per cent each year), and it is naturally associated to online sales. The combination of online sales and advertising is able to generate a very high quantity of raw data that can be used for sales forecasting, taking into account not only sales figures and products characteristics but also online advertising campaigns.

Sales forecasting has been approached in several different ways in recent research. Examples include neural networks [4], logistics regression [5], and support vector regression [6], among others. In this paper, we consider a data set that includes sales figures, product characteristics, and online advertising data, and we want to apply a regression model (see, e.g., [7]) to predict future sales. The considered data set is taken from the Kaggle community (see *https://www.kaggle.com*). *Kaggle* is a platform for predictive modeling competitions that comprises experts from over 100 countries and 200 universities covering many quantitative fields and industries (science, statistics, econometrics, maths, physics), and it has recently made its data sets available for academic use. Our data comes from the *Online Product Sales* competition, which include 751 instances with 546 attributes per instance, and cover 12 consecutive months; their structure, with relatively few instances and relatively many attributes, is the ideal environment for a regression model learning combined with a feature selection step. *Feature selection* is an independent process, commonly used in combination with classification model learning, whose main objective is to reduce the number of attributes in order to increase the performance of a classifier. Feature (or *attribute*) selection techniques [8] range from *filter models* to *wrapper models* to *embedded models*, depending on their grade of interaction with the learning algorithm, and it is relatively common in applications to genomics, health sciences, economics, finance, among others [9, 10, 11], as well as in psychology and social sciences [12, 13]. Feature selection in combination with regression model learning is less common; this is probably due to the fact that a regression model includes a feature selection *built-in*, so to say.

In this paper we propose a wrapper-based feature selection mechanism for sales prediction. We use the multi-objective evolutionary algorithm known as ENORA (Evolutionary NOn-dominated Radial slots based Algorithm) as selection strategy for a random search method [14, 15, 16], with the following two objectives: minimizing the number of selected features and minimizing the *root mean squared error* ($\mathcal{RMSE}$) of the model learned by Random Forest (RF), a well-known regression model learning algorithm known for its low tendency to overtraining (or over-fitting) and its high accuracy [17]. In order to choose among the non-dominated individuals (i.e., selections) generated by several runs, we use 10-folds cross-validation, and we choose the individual with the best $\mathcal{RMSE}$. The selection of attributes that emerges from this analysis is then used for a test phase, in which we compare the performances of several regression model learners under several different measures. Moreover, we compare the performance of ENORA as selection strategy with those of the multi-objective evolutionary algorithm known as NSGA-II (Non-dominated Sorted Genetic Algorithm) [18], which is considered a standard in the multi-objective evolutionary computation community, both in terms of *hypervolume* statistics of the last population, and in terms of the $\mathcal{RMSE}$ of the chosen individual. Finally, we compare the $\mathcal{RMSE}$ of both selections against that of the selection produced by the well-known wrapper method *Recursive Feature Elimination* (RFE) [19]. In terms of usability of the results, our strategy not only produces an accurate regression model, but also a selection of relevant features that gives information on which characteristics of the advertising campaigns and/or the sold products actually influence the sales.

The original data set encompasses sales figures of 12 consecutive months; we separated such data into 12 different data sets, and applied the above methodology and test to each of them separately. This is not only in compliance with the standard approaches to sales prediction (as it is understood that sales of most products are strongly influenced by the time of the year), but also allowed us to perform a meta-analysis that relates the relevant features and the month(s) in which they have been selected. We found that: *(i)* the regression model learned from the reduced data set (i.e., with the subset of features that has been selected) presents better $\mathcal{RMSE}$ values in cross-validation than the one from the original data set, in all 12 months; *(ii)* the selection produced by ENORA is better, in terms of $\mathcal{RMSE}$, than the selection produced by NSGA-II in 11 cases out of 12, and both of them are better than the selection produced by RFE in all 12 cases; *(iii)* the hypervolume statistics for ENORA are better than those for NGSA-II in all 12 months. Summarizing, the distinctive contributions of this work, compared to the existing literature, are:

- We propose the use of a multi-objective search strategy (ENORA), already used in real parameter optimization and fuzzy classification, as a strategy for feature selection.

- We describe a suitable configuration for the selection of features for regression (which, in particular, is applied to sales prediction), given by an open-source wrapper-based methodology and composed by our subset generation algorithm, by Random Forest as regression method (configured with only 10 trees to ensure a reasonable computation time; a greater number of trees does not guarantee relevant improvement in the accuracy and, in opposition, makes prohibitive the computation time), and by $\mathcal{RMSE}$ as evaluation measure.

- We propose a full methodology that integrates pre-processing, feature selection for regression, model evaluation (and statistical test), and decision making (generally not properly dealt with in the literature), in order to choose the most satisfactory model according to a *a posteriori* process (driven by an external process) in a multi-objective context.

- We propose suitable limits for the computation of the hypervolume for the problem of learning a regression model.

- We compare in detail our proposal with the most popular multi-objective search strategy and with another, non-evolutionary, mono-objective backward search strategy, both in terms of hypervolume (only for the multi-objective strategies) and in terms of accuracy of the result, giving sufficient insights to understand why ENORA and NSGA-II behave differently, especially in terms of diversity of the solutions.

- We provide an interpretation of the results in terms of sales prediction month-by-month, relating the months and the selected features, based on simple clustering.

The rest of the paper is structured as follows. Section 2 briefly reviews the evolutionary algorithms ENORA and NSGA-II, the literature on feature selection, multi-objective evolutionary feature selection, regression models, the Recursive Feature Elimination algorithm, and the related work. Section 3 describes how we adapted both ENORA and NSGA-II to the task of feature selection for regression. Section 4 describes the experiment and its results, and Section 5 contains the main conclusions of the paper.

2. Background

In this section we briefly review the multi-objective evolutionary algorithms known as ENORA and NSGA-II, and we give an account of feature selection, as well as multi-objective feature selection techniques in the recent literature. Moreover, we briefly discuss regression models and the Recursive Feature Elimination algorithm.

2.1. The evolutionary multi-objective algorithms ENORA and NSGA-II

Evolutionary (or *genetic*) computation makes use of a metaphor of natural evolution. According to this metaphor, a problem plays the role of an environment in which a population of individuals lives, each representing a possible solution to the problem. The degree of adaptation of each individual to its environment is expressed by an adequacy measure known as fitness function. Like evolution in nature, evolutionary algorithms have the potential to produce gradually improving solutions to the problem. The algorithms begin with an initial population of random solutions and, in each iteration, the best individuals are selected and combined using variation operators such as *crossing* and *mutation* to build the next generation. This process is repeated until some stop criterion is met. Some problems require *multi-objective optimization* (MO), in particular when there exists an intrinsic conflict between two or more problem goals;

feature selection, in which one has to maximize accuracy of a classifier and minimize the number of features, is an example of such problem. *Multi-objective evolutionary algorithms* [20, 21] have proved themselves to be very effective in searching for optimal solutions to multiple objective problems. A MO problem is formulated as a set of minimization/maximization problems of a tuple of n objective functions

$$f_1\left(\vec{x}\right), \ldots, f_n\left(\vec{x}\right),$$

where $\vec{x}$ is a vector of parameters belonging to a given domain. A set $\mathcal{F}$ of solutions for a MO problem is *not dominated* (or *Pareto optimal*) if and only if for each $\vec{x} \in \mathcal{F}$, there exists no $\vec{y} \in \mathcal{F}$ such that (i) there exists i $(1 \leq i \leq n)$ that $f_i\left(\vec{y}\right)$ improves $f_i\left(\vec{x}\right)$, and (ii) for every j, $(1 \leq j \leq n, j \neq i)$, $f_j\left(\vec{x}\right)$ does not improve $f_j\left(\vec{y}\right)$. In other words, a solution $\vec{x}$ *dominates* a solution $\vec{y}$ if and only if $\vec{x}$ is better than $\vec{y}$ in at least one objective, and it is not worse than $\vec{y}$ in the remaining objectives; $\vec{x}$ is *non-dominated* if and only if there is not other solution that dominates it. Multi-objective evolutionary algorithms are particularly suitable for multi-objective optimization, as they search for multiple optimal solutions in parallel and are capable of finding a set of optimal solutions in their final population in a single run. Once the set of optimal solutions is available, the most satisfactory can be chosen by applying a preference criterion. Thus, the aim of a multi-objective search algorithm is to discover a family of solutions that are a good approximation to the Pareto front. In the case of multi-objective feature selection, each solution at the front might represents a subset of features with an associated trade-off between, for example, accuracy and model complexity.

ENORA (Evolutionary NOn-dominated Radial slots based Algorithm) is an elitist Pareto-based multi-objective evolutionary algorithm which was proposed for multi-objective constrained real parameter optimization in [22], and for fuzzy classification in survival prediction in [23]. Among others, it has been applied to feature selection for supervised [24] and unsupervised classification [25]. ENORA uses a $(\mu + \lambda)$ survival strategy, where μ is population size and λ is number of children created. It was originally developed in [26] under the name $(1 + 1) - ES$ as an *evolution strategy*, and it used selection, adapting mutation, and a population of size one. Recombination and populations with more than one individual were later introduced in [27]. The $(\mu + \lambda)$ technique allows the μ best children and parents to survive and it is, therefore, an *elitist* method. ENORA uses a $(\mu + \lambda)$ survival with $\mu = \lambda = N$, where N is the size of the population, binary tournament selection, and *self-adaptive* crossover and mutation for multi-objective evolutionary optimization.

For each of T generations, a pair of parents are selected by *binary tournament selection* from the population P. This selection algorithm returns the best from two random individuals according to a *rank-crowding-better function*, by means of which an individual I is considered better than an individual J if its rank is better (lower) than the rank of the individual J in the population P. The *rank* of an individual I in a population P, (denoted $rank\left(P, I\right)$), is the *non-domination level* of the individual I among the individuals J of the same slot. The *slot* function is calculated according to the equation (1) where $d = \left\lfloor \sqrt[n-1]{N} \right\rfloor$ and h_j^I is the objective function f_j^I normalized in $[0, 1]$:

$$slot\,(I) = \sum_{j=1}^{n-1} d^{j-1} \lfloor d \frac{\alpha_j^I}{\pi/2} \rfloor$$
$$\alpha_j^I = \begin{cases} \frac{\pi}{2} & \text{if } h_j^I = 0 \\ \arctan(\frac{h_{j+1}^I}{h_j^I}) & \text{if } h_j^I \neq 0 \end{cases} \tag{1}$$

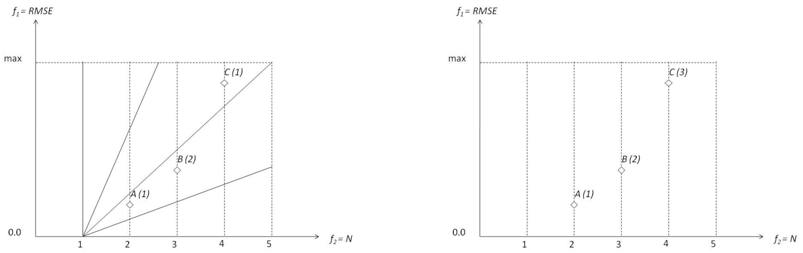
If two individuals I and J have the same rank, the best is the one with the greater crowding distance at its front. The *crowding distance* (CD) of an individual I in a population P is a measure of the search space around individual I which is not occupied by any other individual in the population P. This quantity serves as an estimation of the perimeter of the cuboid formed by using the nearest neighbours as the vertices, and it is calculated as follows:

$$CD\,(P, I) = \begin{cases} \infty & \text{if } f_j^I = f_j^{max} \text{ or } f_j^I = f_j^{min} \text{ for some } j \\ \\ \sum_{j=1}^{n} \frac{f_j^{sup_j^I} - f_j^{inf_j^I}}{f_j^{max} - f_j^{min}} & \text{otherwise} \end{cases} \tag{2}$$

where $f_j^{max} = \max_{I \in P} \{f_j^I\}$, $f_j^{min} = \min_{I \in P} \{f_j^I\}$, $f_j^{sup_j^I}$ is the value of the jth objective for the individual higher adjacent in the jth objective to the individual I, and $f_j^{inf_j^I}$ is the value of the jth objective for the individual lower adjacent in the jth objective to the individual I.

The selected pair of parents is crossed, mutated, evaluated and added to an initially empty auxiliary population Q. This process is repeated until Q contains a number N of individuals. An auxiliary population R is obtained via the union of populations P and Q. Next, the rank of all individuals in population R is calculated, and the N best individuals of R according to the rank-crowding-better function survive to the next generation.

NSGA-II (Non-dominated Sorted Genetic Algorithm) [21] is, as well, an elitist Pareto-based multi-objective evolutionary algorithm. It was designed to improve the previous NSGA [28] algorithm by incorporating an explicit diversity technique, and it is, perhaps, one of the most used Pareto-based multi-objective evolutionary algorithms described in the literature. NSGA-II uses, as ENORA, a $(\mu + \lambda)$ strategy with a binary tournament selection and a rank-crowding better function. The difference between NSGA-II and ENORA is how the calculation of the ranking of the individuals in the population is performed. In ENORA, the rank of an individual in a population is the non-domination level of the individual in its slot, whereas in NSGA-II the rank of an individual in a population is the non-domination level of the individual in the whole population. Although NSGA-II and ENORA algorithms are similar, they behave quite differently. The main difference is the following: when NSGA-II compares two individuals through binary tournament, an individual dominated by the other is never selected, while in ENORA an individual dominated by the other can be the winner of



ENORA

NSGA-II

Figure 1: Individuals rank assignment with ENORA vs NSGA-II.

the tournament. Fig. 1 graphically shows this behaviour. For example, if individuals B and C are selected for binary tournament with NSGA-II, as B dominates C, it certainly beats it. In contrast, individual C beats B in ENORA, because individual C has a better rank in its slot than individual B. So, ENORA encourages diversity because it allows the individuals in each slot to evolve towards the Pareto front although these individuals may not be not the best when they are compared, thus obtaining better hypervolume than NSGA-II during the course of the generations.

2.2. Feature selection

Feature selection (FS) is defined in [29] as the process of eliminating features from the data base that are irrelevant to the task to be performed. Feature selection facilitates data understanding, reduces the measurement and storage requirements, reduces the computational process time, and reduces the size of a data set, so that model learning becomes an easier process. FS has become relatively common in applications to genomics, health sciences, economics, finance, among others [9, 10, 11], as well as in psychology and social sciences [12, 13].

Feature selection algorithms can be *supervised*, *unsupervised* and *semi-supervised*; this depends on whether the training set is, or not, labelled; moreover, feature selection models are also categorized into *filter, wrapper* and *embedded models*. The first ones apply statistical measures to assign a score to each feature; features are ranked by their score, and either selected to be kept or removed from the data set. Filter models do not interact with learning algorithms, and they can be *univariate* (when features are evaluated one by one) or *multivariate* (when they are evaluated in subsets). Wrapper methods define the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. Finally, the underlying idea of embedded models is learning which features best contribute to the accuracy of the model while the model is being created.

Feature selection methods consist of four steps, usually called *subset generation*, *subset evaluation*, *stopping criterion*, and *result validation.* Subset generation is a heuristic search in which candidate subsets are prepared for evaluation. Obviously, the search space for candidate subsets has cardinality, where N is the number of features. Examples of subset generation mechanisms include *greedy hill-climbing approach* [9], *sequential forward selection* [30], *sequential backward elimination* [31], *bi-directional selection* [29], *branch and bound* [32], *beam search* [33], *Las Vegas* algorithms [14], evolutionary algorithms [34, 35], and *particle swarm optimization* algorithms [36, 37]. During the phase of subset evaluation the goodness of a subset produced by a given subset generation procedure is measured. Examples of subset evaluation measures for multivariate filter methods are the *distance* [38], the *uncertainty* [39, 40]), the *dependence* [41], and the *consistency* [10], while wrapper methods mostly use the *accuracy* [42]. The stopping criterion establishes when the feature selection process must finish; it can be defined as a control procedure that ensures that no further addition or deletion of features does produce a better subset, or it can be as simple as a counter of iterations. Finally, in the phase of result validation the validity of the selected subset is tested. A recent overview, categorization, and comparison of existing feature selection methods is shown in [8]. A major drawback of such approaches is that they only consider a single criterion while searching for subset, and they do not try to minimize the number of chosen attributes; they can be then referred to as *single-objective* feature selection methods.

In this paper we propose a wrapper feature selection mechanism based on evolutionary subset generation. Wrapper schemata are more common in supervised classification rather than regression; this is probably due to the fact that regression model learning algorithms already perform feature selection and penalty [43]. However, these mechanisms do not suffice when the number of features is particularly high, and a separate feature selection process does improve the performances of the learned model, as we prove in our experiments.

2.3. Multi-objective evolutionary feature selection

The use of genetic algorithms for the selection of features in the design of automatic pattern classifiers was introduced in [44]. Since then, genetic algorithms have come to be considered as a powerful tool for feature selection [15], and have been proposed by numerous authors as a search strategy in filter, wrapper, and embedded models [45, 46, 47], as well as feature weighting algorithm and subset selection algorithms [48, 49]. A review of evolutionary techniques for feature selection can be found in [35], and a very recent survey of multi-objective algorithms for data mining in general can be found in [50, 51].

The first evolutionary approach involving multi-objective optimization for feature selection was proposed in [52] with three criteria: accuracy, number of features, and number of instances. In this approach, the three criteria are aggregated into a single one and, then, a single-objective algorithm is used. A formulation of feature selection as a multi-objective optimization problem has been presented in [53], and the multi-objective genetic algorithm proposed in [54] is applied to regression problems. A wrapper method to solve a two-objective optimization problem was proposed in [55]; in this case, the objectives were the accuracy of a fuzzy rule based classifier and an aggregated measure of the cardinality and granularity of the subset selection. A modified wrapper method that uses NSGA is proposed in [55] for minimizing the number of features and the error rate of a neural network-based classifier applied to handwritten digit recognition. The wrapper approach proposed in [56] takes into account the misclassification rate of the classifier, the difference in error rate among classes, and the size of the subset using a multi-objective evolutionary algorithm where a niche-based fitness punishing technique is proposed to preserve the diversity of the population. A wrapper approach is proposed in [57] which minimizes both the error rate and the size of the tree discovered by the C4.5 classification algorithm. Another wrapper method is proposed in [58] to maximize the cross-validation accuracy on the training set, maximize the classification accuracy on the testing set, and minimize the cardinality of feature subsets using support vector machines applied to protein fold recognition. An ensemble construction algorithm is proposed in [59] that combines an evolutionary multi-objective algorithm and a Bayesian *automatic relevance determination* methodology, using NSGA to minimize the error and the number of features.

In [60] a multi-objective evolutionary optimization and support vector machines are combined. NSGA-II is used to minimize the false positive rate, the false negative rate, and the number of support vectors to reduce the computational complexity. In [61] two wrapper methods with three and two objectives, respectively, applied to cancer diagnosis are compared. The three-objective version optimizes the sensitivity, the specificity and the number of genes, while the two-objective one optimizes the accuracy and the number of genes. NSGA-II is used as search strategy, and a support vector machine is used for the classification task. A filter local search embedded multi-objective *memetic* algorithm is presented in [62], which is a synergy of an evolutionary algorithm (NSGA-II) and a filter method for the identification relevant features in a multi-class problem. The filter approach proposed in [63] includes measures of consistency, dependency, distance and information, and it is based, again, on NSGA-II. A NSGA-II wrapper approach is proposed in [64] for named entities recognition. A modification of the dominance relation is introduced in [65] to treat an arbitrarily large number of objectives, and used in a combination of NSGA-II, logistic regression, and naïve Bayes with Laplace correction as classification algorithms. In [66], multi-objective feature selection is applied to a certain diagnosis problem in medicine. In [67], a multi-objective algorithm that minimizes error identification rate, undetected identification rate, and number of selected features is proposed for an application in engineering. In [68] a multi-objective Bayesian artificial immune system is applied to feature selection in classification problems, aimed at minimizing both the classification error and cardinality of the subset of features. In [69] a wrapper method is proposed to optimize the error rate of data-mining algorithm and the size of the model built by a learning algorithm by using NSGA and NSGA-II. A multi-objective estimation of distribution algorithm is proposed in [70] for the selection of a feature subset, based on joint modeling of objectives and variables. The authors use six different performance measures for the classifiers based on the classification accuracy, given by a confusion matrix and class-value probabilities, and adopt a wrapper approach to evaluate feature subsets using naïve Bayes and tree-augmented naïve Bayes classifiers.

In [71] the authors propose a multi-objective optimization algorithm to maximize the ROC (*receiver operating characteristic*) convex hull [72]. The proposal is compared with NSGA-II and other approaches in the experiments, and the conclusion is drawn that their approach gives better results. Very recently, in [73], a parallel multi-objective optimization approach was proposed to cope with high-dimensional feature selection problems. Several parallel multi-objective evolutionary alternatives are proposed and experimentally evaluated. Finally, in [74] a multi-objective unsupervised feature selection algorithm is proposed to incorporate the correlation coefficient and the cardinality of the feature subset, which not only evaluates the redundancy of selected features but also provides several objective values for each particular size of feature subset, and in [75] a new multi-objective evolutionary ensemble optimizer, coupled with neural network models, is proposed.

To conclude, we recall that in [24] ENORA and NSGA-II have been used to predict the outcome of a session in a contact center environment, and in [25] ENORA and NSGA-II have been compared as search strategies in feature selection for unsupervised classification.

2.4. Random Forest

Random Forest (RF) is a well-known regression model learning algorithm known for its low tendency to overtraining and its high accuracy [17]. The idea underlying the RF algorithm was first proposed in [76], where the authors argued how a generalization of the decision tree could gain accuracy and gradually lower the risk of overtraining. The proposal was later extended and improved in [77]. RF is based on the classical decision tree algorithm, which is quick and interpretable, but often not very accurate. The main drawback of a decision tree is its natural tendency to overtraining as the tree grows: this is mainly caused by the fact that decision tree have low bias, but very high variance. Random forests are a way of averaging multiple depths decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability. Random Forest is a package available open source in Weka (see *http://www.cs.waikato.ac.nz/ml/weka/*).

2.5. Recursive Feature Elimination

The Recursive Feature Elimination algorithm (also known as RFE) [19] is a wrapper method (so, as in all wrapper method we can choose the evaluator), but it encompasses a *backward* search strategy (in opposition to a random search strategy). RFE returns a ranking of the features of a classification problem by training a classifier (or a regression model learner) and it uses backwards selection, re-sampling and external validation for feature selection. Recursive Feature Elimination is available open source in Caret R (see *http://cran.r-project.org/web/packages/caret/caret.pdf*).

2.6. Discussion of Related Work

As shown above, most approaches that use multi-objective evolutionary algorithms for feature selection have been proposed in recent years. Although both filter and wrapper methods have been proposed, most authors tend to prefer the latter to the former. The optimization model most commonly used involves maximizing the accuracy of a classifier while minimizing the number of features, although many other models have been proposed for specific contexts. NSGA-II has been without doubt the search strategy most widely used, either directly or as reference algorithm in comparison with others. Finally, most existing work in feature selection is focused in classification rather than in regression. In this paper we focus on the problem of regression in supervised learning in online sales forecasting. In order to simultaneously attain a regression model which is highly accurate and simple we are interested in subset selection wrapper methods with a multi-objective evolutionary search strategies. To this end, we propose to use ENORA as search strategy; since NSGA-II is the reference in the literature, in this work we compare our results with those given by NSGA-II, from both points of view of the statistical results and the accuracy of the model; we also discuss the evolution of the two search methods and their characteristics. We use the Random Forest learning method for regression because of its low tendency to overtraining and its high accuracy, and it can be used to rank the importance of variables. Additionally, Random Forest runs efficiently in big data bases. The results of our proposal are also compared with another popular wrapper method for feature selection such as Recursive Feature Elimination.

3. Adapting ENORA and NSGA-II to Feature Selection for Regression

In this paper we adapt ENORA and NSGA-II to serve as multi-objective evolutionary search strategy for feature selection. Although such a search strategy can be used for both filter and wrapper feature selection methods, we propose to use it as wrapper method with RF for regression. In this section, the main components of both ENORA and NSGA-II, adapted for feature selection are described.

3.1. Initial Population, Representation of Solutions and Evaluation

We use a fixed-length representation, where each individual consists of a bit set. Each bit represents an attribute in the data set (1 for selected, and 0 for non-selected attributes); the length of the individuals is equal to the number N of attributes in the initial data set. Additionally, to carry out self-adaptive crossing and mutation, each individual has two discrete parameters $d_I \in \{0,\ldots,\delta\}$ and $e_I \in \{0,\ldots,\epsilon\}$ associated to crossing and mutation, where $\delta \geq 0$ is the number of crossing operators and $\varepsilon \geq 0$ is the number of mutation operators. Therefore, an individual I is represented as:

$$I = \left\{ b_1^I, \ldots, b_N^I, d_I, e_I \right\},$$

where $b_i^i \in \{0,1\}$ for $i = 1,\ldots,N$, $d_I \in \{0,\ldots,\delta\}$, and $e_I \in \{0,\ldots,\epsilon\}$.

An individual I is evaluated with two fitness functions, $f_1(I)$ and $f_2(I)$, corresponding to the two (minimization) objectives of the multi-objective optimization model:

$$\begin{cases} f_1(I) = \mathcal{RMSE}(I) \\ f_2(I) = \mathcal{C}(I) \end{cases}$$

where $\mathcal{RMSE}(I)$ is the root mean squared error computed over the reduced data set, and $\mathcal{C}(I)$ is the cardinality of the subset represented by the individual I, i.e, the number of bits equal to 1 in the individual I. As a regression model learner, there are

Algorithm 1 Initialize population

```
Require: $\delta > 0$ {Number of crossing operators}
Require: $\epsilon > 0$ {Number of mutation operators}
Require: $N$ {Number of input attributes in data set}
Require: $popsize \geq 0$ {Number of individuals in the population}
$P \leftarrow$ Empty Population
for $I = 1$ to $popsize$ do
    $I \leftarrow$ new Individual
    $Q \leftarrow \{1, \ldots, N\}$
    $q \leftarrow$ Int Random from $Q$
    $r \leftarrow N - q$
    for $i = 1$ to $q$ do
        $j \leftarrow$ Random from $Q$
        $b^I_j \leftarrow 1$
        $Q \leftarrow Q - \{j\}$
    end for
    for $i = 1$ to $r$ do
        $j \leftarrow$ Random from $Q$
        $b^I_j \leftarrow 0$
        $Q \leftarrow Q - \{j\}$
    end for{Random Discrete for self-adaptive variation}
    $d_I \leftarrow$ Int Random from $\{0, \delta\}$
    $e_I \leftarrow$ Int Random from $\{0, \epsilon\}$
    Add $I$ to population $P$
end for
return $P$
```

several possible measures of the quality of a model learned by Random Forest. The most relevant one, often referred to as *root mean squared error* ($\mathcal{RMSE}$) is typically defined as:

$$\mathcal{RMSE}(D) = \sqrt[2]{\frac{\Sigma_{i=1}^{|D|}(\bar{d}_i - d_i)}{|D|}}$$

where D is the data set composed by $|D|$ instances, each with a class value d_i and a *prediction* $\bar{d}_i$. The $\mathcal{RMSE}$ is computed on the data set projected on the attributes selected in I.

The initial population is randomly generated as described in Algorithm 1. For each individual I in the population, a number $q \in \{1, \ldots, N\}$ is first randomly generated. Next, q random bits in the individual I are fixed to 1 and the remaining $N - q$ bits are fixed to 0. Finally, d_I and e_I values for self-adaptive variation are randomly generated from $\{0, \delta\}$ and $\{0, \epsilon\}$ respectively.

3.2. Variation operators

We use one crossover operator (*uniform crossover*) and one mutation operator (*one flip mutation*), although any other variation operators can be considered; therefore, for us, $\delta = \epsilon = 1$. The selection of the operators is made by means of an adaptive technique that uses the parameters d_I and e_I (in our case, both in the set $\{0, 1\}$) to indicate which crossover (Algorithm 3) and which mutation (Algorithm 4) is carried out on the

Algorithm 2 Variation

```
Require: $Parent1$, $Parent2$ {Individuals to vary}
$Child1 \leftarrow Parent1$
$Child2 \leftarrow Parent2$
Self-adaptive crossover $Child1$, $Child2$
Self-adaptive mutation $Child1$
Self-adaptive mutation $Child2$
return $Child1, Child2$
```

Algorithm 3 Adaptive crossover

```
Require: $I$, $J$ {Individuals to cross}
Require: $p_v$ ($0 < p_v < 1$) {Probability of operator change}
Require: $\delta > 0$ {Number of different crossover operators ($\delta = 1$ in our case)}
if A random Bernoulli variable of probability $p_v$ takes the value $1$ then
    $d_I \leftarrow$ Int Random from $\{0, \delta\}$
end if
$d_J \leftarrow d_I$
Carry out the type of crossover specified by $d_I$: {0: No cross} {1: Uniform crossover}
```

Algorithm 4 Adaptive mutation

```
Require: $I$ {Individual to mutate}
Require: $p_v$ ($0 < p_v < 1$) {Probability of operator change}
Require: $\epsilon > 0$ {Number of different mutation operators ($\epsilon = 1$ in our case)}
if A random Bernoulli variable of probability $p_v$ takes the value $1$ then
    $e_I \leftarrow$ Int Random from $\{0, \epsilon\}$
end if
Carry out the type of mutation specified by $e_I$: {0: No mutation} {1: One flip mutation}
```

individual I. The Algorithm 2 is used to generate two children from two parents by self-adaptive crossing and mutation. We first fix a probability of variation $p_v = 0.1$, and, then, for each individual in each generation, we ask to a Bernoulli random variable with parameter p_v whether or not the values d_I and e_I have to be changed. In the particular case of crossover, fixed the first (resp., second) selected individual I (resp., J), the decision on whether the crossover takes place or not depends on the value of d_I. In summary, our proposal works as follows. If an individual comes from a given crossover or a given mutation, that specific crossover and mutation is preserved to their offspring unless the Bernoulli random variable returns *true*; for this reason, p_v must be a small value to ensure a controlled evolution. Although the probability of the crossover and mutation is not explicitly represented, it can be computed as the ratio the individuals for which crossover and mutation values are set to one.

Self-adaptive crossover and mutation help to realize both goals of maintaining diversity in the population and sustaining the convergence capacity of the evolutionary algorithm, and by using self-adaptive operators it is not necessary to set *a priori* the probability of application of the different operators; among others, Srinivas and Patnaik [78] propose a similar approach, in which the probabilities of crossover and mutation are varied depending on the fitness value of the solutions.

4. Experiments and results

Here we describe the proposed methodology, and we report the results of applying it. We discuss them in terms of hypervolume (to compare ENORA and NSGA-II), and in terms of quality of the selected subsets of features. Our methodology can be summarized as follows: for each month, separately, it includes pre-processing of the data, feature selection, a comparison of optimizer performances (based on hypervolume metrics), regression model construction, and test, as it is depicted in Fig. 2.

4.1. The "Online Sales" Data Set

The considered data set is taken from the Kaggle community. *Kaggle* is a platform for predictive modeling competitions that comprises experts from over 100 countries and 200 universities covering many quantitative fields and industries (science, statistics, econometrics, mathematics, physics), and it has recently made its data sets available for academic use. Our data comes from the *Online Product Sales* competition, which include 751 instances with 546 attributes per instance, and cover 12 consecutive months.

Data are provided in block for the 12 months; the first 12 columns (*Outcome_M1* through *Outcome_M12*) contain the monthly online sales for the first 12 months after product launch. The 546 remaining columns are the input attributes, and contain information on the date in which the major advertising campaign began and the product was launched (*Date_1*) and the date in which the product was announced and a pre-release advertising campaign began (*Date_2*), information on the products' features, and on the advertising campaign. Both quantitative variables (*Quan_x*) and categorical (*Cat_x*) variables are present. As one may expect, there are some missing data. Moreover, no common-sense considerations can be made as data have been codified so that their source cannot be traced; the dates are given in terms of number of days that have passed after some unspecified "zero".

As a first step, we transformed our data set into 12 copies, each with precisely one outcome, corresponding to the online sales for a single month.

4.2. Data pre-processing

The initial data bases are pre-processed as follows. First, all the missing values for nominal and numerical attributes are replaced with, respectively, the modes and means from the training data; to this end, the procedure *ReplaceMissingValues* from the package *weka.filters.unsupervised.attribute* is used to this end. Second, the features showing a too small variation are eliminated; we have used the procedure *nearZeroVar* from Caret R for this task. As a result, each monthly data bases has been reduced to 163 features.

4.3. Feature selection and decision making

In this work we used three different feature selection methods: ENORA, NSGA-II and RFE. Both ENORA and NSGA-II are probabilistic methods for multi-objective optimization, and so they require multiple runs with different seeds. The following steps are performed with both ENORA and NSGA-II. First, we perform feature selection 30 times with a RF-based wrapper method and multi-objective evolutionary search

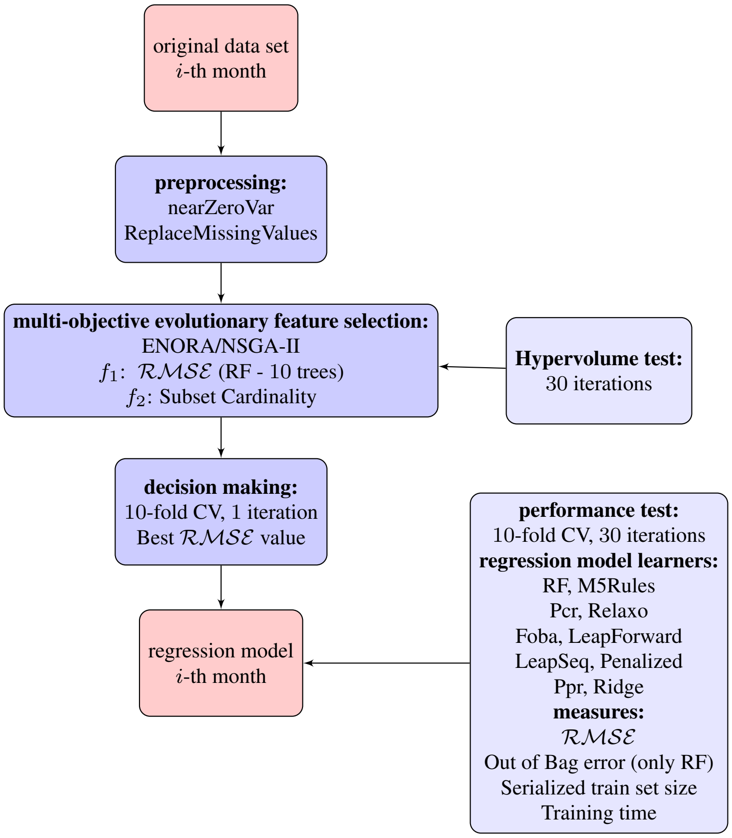


Figure 2: Proposed methodology.

strategy, following the optimization model explained in Section 3. Both ENORA and NSGA-II are implemented in Java using the *Weka* package. For each run, we used the following evaluator for both ENORA and NSGA-II:

weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.trees.RandomForest -F 5 -T 0.01 -R 1 -E DEFAULT – -I 10 -K 0 -S 1 -num-slots 1

Both algorithms were executed with population size equal to 1000 and for 100 generations, for a total of 100000 evaluations (and a different seed) in each run. ENORA was incorporated by the authors into Weka as an official package (called *MultiObjectiveEvolutionarySearch*). Then, we run a decision making phase: for each run, we performed 10-folds cross-validation on each non-dominated solution, and identified the solution with the best value for the cross-validation test. Finally, we constructed a reduced data base with the selected attributes. As we have explained, we compared the results with an execution of RFE (see Section 2), where, to ensure a correct comparison, RF has been chosen as evaluator, and the following control command was

Month	ENORA	NSGA-II	RFE
Jan	17	17	95
Feb	15	11	120
Mar	19	11	137
Apr	49	10	78
May	10	13	18
Jun	9	9	16
Jul	9	11	26
Aug	23	14	130
Sep	27	11	80
Oct	14	5	162
Nov	33	9	22
Dec	13	11	157

Table 1: Number of selected attributes with ENORA, NSGA-II and RFE.

used:

rfeControl(functions=rfFuncs, method="repeatedcv", repeats=30, number=10).

All experiments have been executed on a 8 processors machine Intel Xeon X7550 @ 2.00 GHz, RAM 1 TByte at 1067MHz and storage Lustre Distributed File System v2.5.2, with interconnection network Infiniband QDR (40Gbps), and a single-processor Intel(R) Core (TM) i5 2400 CPU at 3.10 GHz with 16GB RAM, running Windows 7 Pro Service Pack 1 64 bit. The run time for each execution for ENORA was, on average, of 9.08 hours, for NSGA-II of 5.23 hours, and for RFE of 0.84 hours.

The result of our decision making process in terms of cardinality of the chosen subset, is shown in Tab. 1, specified per month, and compared with the selection provided by RFE.

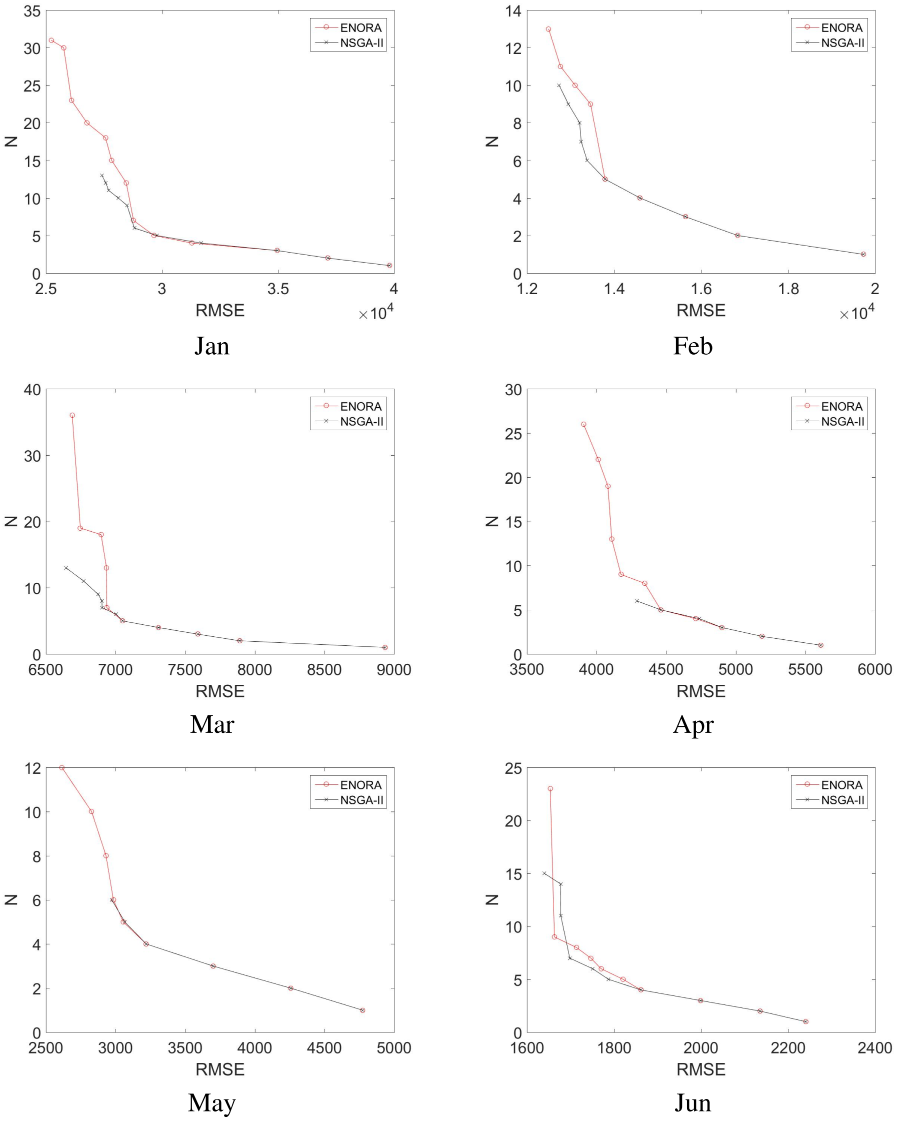
4.4. Hypervolume test

Here the performances of ENORA and NSGA-II are compared against each other, and the results of such a comparison are shown. The aim of this set of experiments was to identify the best performing optimization algorithm to the task of feature selection for regression. To compare the algorithms, we used the *hypervolume indicator*, which is defined, in general terms [21], as the volume of the search space dominated by a population P, expressed as:

$$HV(P) = \bigcup_{i=1}^{|Q|} v_i$$

where $Q \subseteq P$ is the set of non-dominated individuals of P, and v_i is the volume of the individual i. We use, for technical convenience, the equivalent measure *hypervolume ratio*, defined as the ratio of the volume of the non-dominated search space over the volume of the entire search space, as follows:

$$HVR(P) = 1 - \frac{HV(P)}{volS}$$



Jan

Feb

Mar

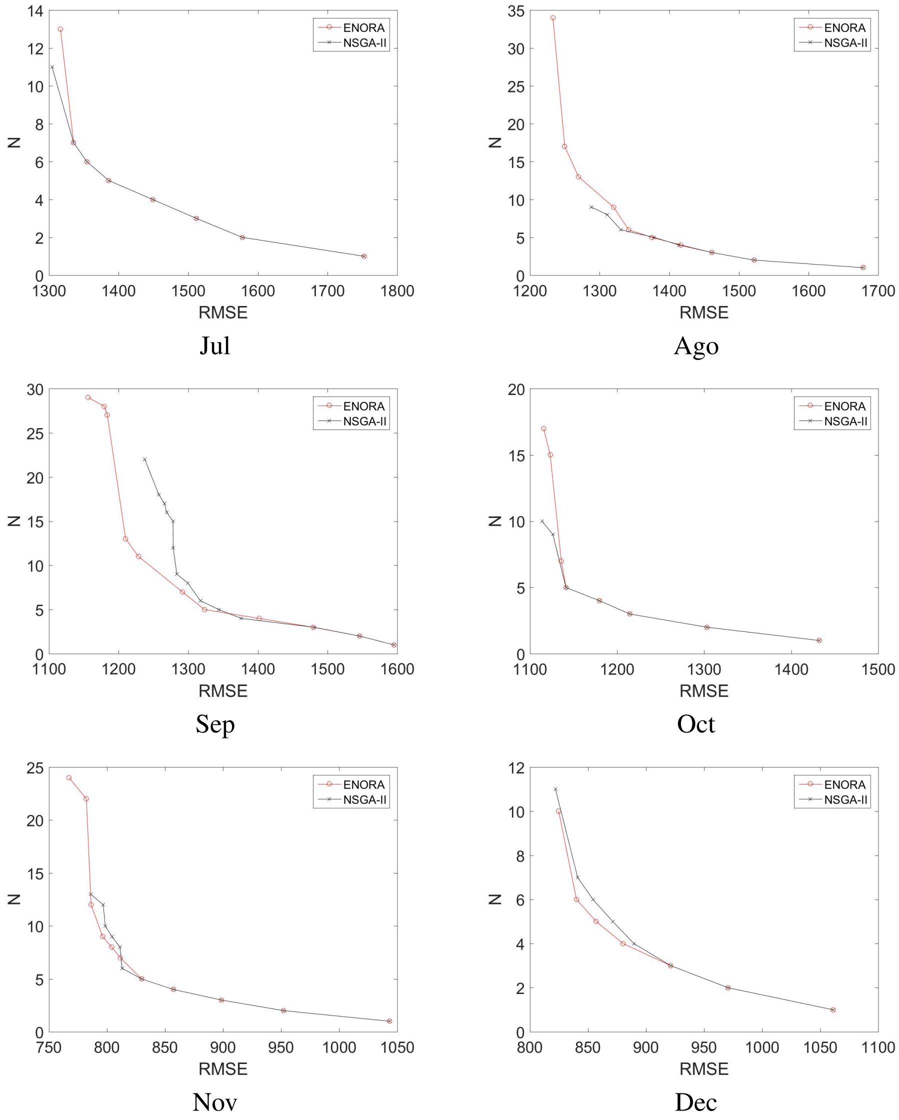
Apr

May

Jun

Figure 3: Pareto front of the best population in the last execution (from January to June).

where $volS$ is the volume of the search space. Other performance metrics can be used [21]; however, the hypervolume measures, simultaneously, both diversity and optimality of the non-dominated solutions, and it is therefore very convenient. Among other advantages, to use the hypervolume metric the *optimal* population, not always available (and not available for the problem at hand), is not needed. Conversely, other popular metrics, such as *error ratio*, *generational distance*, *maximum Pareto-optimal*



Jul

Ago

Sep

Oct

Nov

Dec

Figure 4: Pareto front of the best population in the last execution (from July to December).

front error, *spread*, *maximum spread*, or *chi-square-like deviation*, all require to know the structure and the individual of an optimal population. Additionally, other metrics such as *spacing* only measure the uniformity of the non-dominated solutions and do not take into account the extent of spread or the optimality.

Fig. 3 and Fig. 4 depict, for each month, the Pareto front of the last population

Problem	Algorithm	Minimum	Maximum	Mean	S.D.	C.I. Low	C.I. High
Jan	ENORA	0.4707	0.4983	0.4881	0.0061	**0.4859**	**0.4904**
	NSGA-II	0.4822	0.5260	0.5047	0.0101	0.5009	0.5085
Feb	ENORA	0.4791	0.5053	0.4934	0.0070	**0.4907**	**0.4960**
	NSGA-II	0.4847	0.5219	0.4998	0.0091	0.4964	0.5032
Mar	ENORA	0.6492	0.6738	0.6624	0.0062	**0.6601**	**0.6647**
	NSGA-II	0.6616	0.6872	0.6757	0.0071	0.6730	0.6783
Apr	ENORA	0.5869	0.6182	0.6058	0.0066	**0.6034**	**0.6083**
	NSGA-II	0.6015	0.6504	0.6259	0.0111	0.6217	0.6300
May	ENORA	0.4091	0.4415	0.4205	0.0063	**0.4181**	**0.4228**
	NSGA-II	0.4138	0.4754	0.4405	0.0171	0.4341	0.4468
Jun	ENORA	0.5768	0.6024	0.5874	0.0061	**0.5851**	**0.5897**
	NSGA-II	0.5823	0.6145	0.6005	0.0088	0.5972	0.6038
Jul	ENORA	0.6777	0.6957	0.6885	0.0046	**0.6868**	**0.6902**
	NSGA-II	0.6835	0.7000	0.6951	0.0044	0.6934	0.6967
Aug	ENORA	0.6653	0.6925	0.6810	0.0069	**0.6784**	**0.6836**
	NSGA-II	0.6736	0.7087	0.6935	0.0107	0.6895	0.6975
Sep	ENORA	0.6091	0.6500	0.6302	0.0109	**0.6261**	**0.6342**
	NSGA-II	0.6271	0.6799	0.6526	0.0139	0.6474	0.6578
Oct	ENORA	0.7078	0.7328	0.7209	0.0055	**0.7188**	**0.7229**
	NSGA-II	0.7218	0.7387	0.7339	0.0053	0.7319	0.7359
Nov	ENORA	0.6500	0.6652	0.6590	0.0040	**0.6575**	**0.6605**
	NSGA-II	0.6534	0.6769	0.6658	0.0055	0.6637	0.6679
Dec	ENORA	0.7288	0.7488	0.7386	0.0050	**0.7367**	**0.7405**
	NSGA-II	0.7338	0.7634	0.7460	0.0072	0.7433	0.7487

S.D = Standard Deviation of Mean
C.I. = Confidence Interval for the Mean (95%)

Table 2: Statistics for the hypervolume ratio obtained of 30 runs with algorithms ENORA and NSGA-II.

(the best one among the 30 executions), both for ENORA and for NSGA-II, so that their hypervolume can be visually compared. The statistical values of the hypervolume among the 30 runs are shown in Tab. 2, and the relative box-plots are shown in Fig. 5. In Tab. 2 numbers in bold indicates that the related algorithm is better with significant statistical difference because the confidence interval is non overlapping.

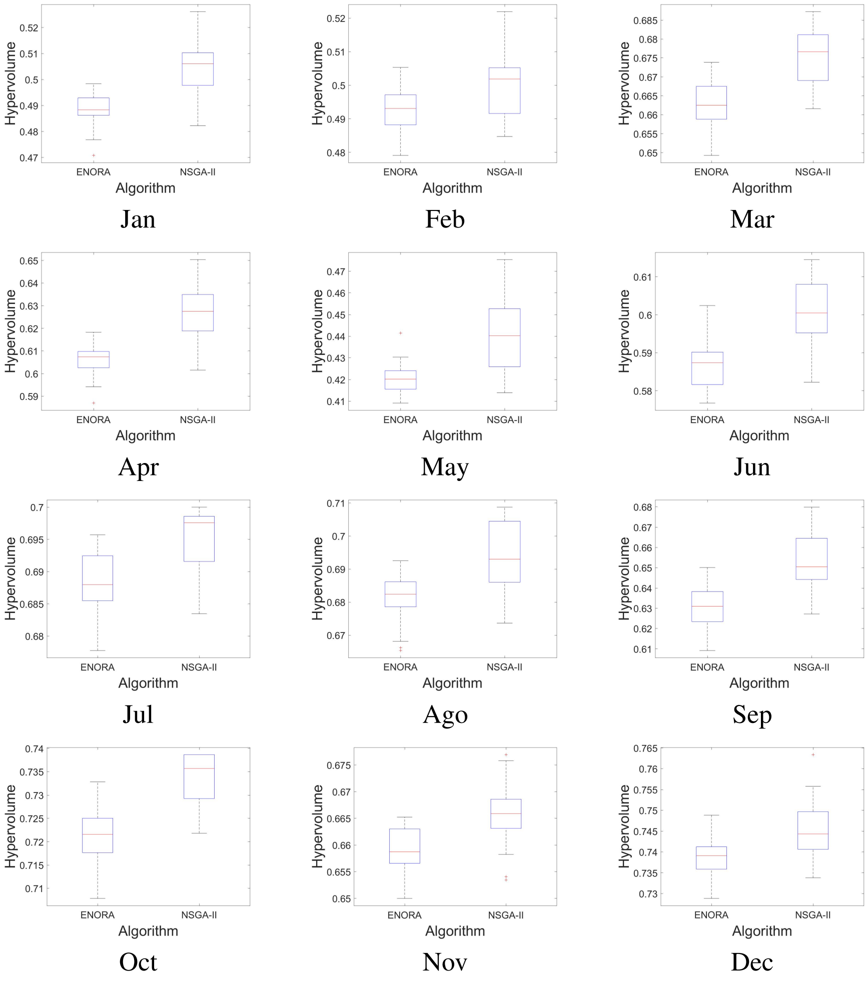
Hypervolume metric requires reference points which identify the maximum and minimum values for each objective. For feature selection in regression problems, as in this work, the worst $\mathcal{RMSE}$ value is not known a priori because it depends on the actual data base under consideration. We propose the following approach to establish the minimum (f_1^{lower}, f_2^{lower}) and maximum (f_1^{upper}, f_2^{upper}) values for each objective:

$$\begin{aligned} f_1^{lower} &= 0 \\ f_1^{upper} &= \max_i \ \mathcal{RMSE}\left(DB_i\right),\ i = 1, \ldots, N \\ f_2^{lower} &= 1 \\ f_2^{upper} &= N \end{aligned}$$

where DB_i is the data base composed by only one attribute i. Note that if any individual of the population has a worst value than f_1^{upper}, then that individual is not taken into account in the calculation of the hypervolume because it is dominated by the point with objective values (f_1^{upper}, f_2^{lower}).

4.5. Regression model(s) performance evaluation and statistical tests

The feature selection phase produced three reduced data sets, one for each of the (best) selections with ENORA (ENORA-DS), with NSGA-II (NSGA-II-DS), and with



Jan Feb Mar

Apr May Jun

Jul Ago Sep

Oct Nov Dec

Figure 5: Box-plot diagrams of the hypervolume ratio of 30 runs of ENORA and NSGA-II.

RFE (RFE-DS). Now, we evaluate the performances of the regression models that can be built over such reduced data sets and the original data set (Original-DS).

In a first evaluation, we use Random Forest to learn a regression model and to compare the performances of the three selections against the one of the original data set. This performance evaluation is particularly interesting given that our feature selection methods was based on Random Forest as evaluator of the wrapper method. The *Out of Bag error* measure was evaluated with full train set over 50 iterations. The result of this evaluation, specified by month, are shown in Tab. 3. We also measured the $\mathcal{RMSE}$, the *serialized train set size* (i.e., a measure of how complex is the train set when saved in a persistent form, e.g., on the hard-disk as a byte-stream), and the *training time*, with

Month	ENORA-DS	NSGA-II-DS	RFE-DS	Original-DS
Jan	**10758.51(168.97)**	10774.97(175.61)	11618.91(298.16)	12318.36(303.28)
Feb	**5907.25(103.84)**	5946.53(85.46)	6463.95(116.71)	6650.13(169.22)
Mar	**3005.84(37.75)**	3048.23(44.92)	3273.69(46.87)	3296.62(42.63)
Apr	1833.55(29.28)	**1823.53(27.22)**	1943.12(32.98)	2023.21(37.82)
May	**1400.43(22.87)**	1400.45(21.76)	1514.07(30.72)	1709.91(45.63)
Jun	963.53(8.86)	983.55(10.50)	**957.67(12.88)**	1062.39(15.87)
Jul	753.15(7.54)	**746.04(8.26)**	778.66(7.38)	854.07(10.27)
Aug	**691.89(9.18)**	700.01(7.43)	731.84(9.34)	756.79(10.09)
Sep	**639.79(7.52)**	683.75(6.38)	656.14(10.82)	707.42(9.08)
Oct	569.02(6.42)	**563.39(7.26)**	635.54(8.71)	635.09(6.84)
Nov	486.61(4.74)	**473.01(5.21)**	485.75(6.17)	550.82(6.43)
Dec	444.94(5.16)	**442.28(5.87)**	503.82(7.35)	506.59(6.78)

Table 3: *Out Of Bag error* (full training, 50 iterations), Random Forest.

Month	ENORA-DS	NSGA-II-DS	RFE-DS	Original-DS
Jan	**24889.83(11737.96)**	25084.47(11801.95)	27874.60(14502.84)	29461.60(15623.85)
Feb	**11843.42(3905.03)**	11923.77(3932.47)	14534.19(5529.57)	15002.61(5845.31)
Mar	**6264.64(1988.61)**	6368.28(2113.90)	6629.60(2208.75)	6691.59(2212.97)
Apr	**3812.46(1523.97)**	3840.19(1704.12)	4168.81(1893.88)	4315.07(1954.86)
May	**2635.12(957.58)**	2666.92(947.32)	3533.80(1403.60)	3937.23(1827.04)
Jun	**1591.74(394.09)**	1615.20(400.56)	1750.28(514.71)	1846.14(526.66)
Jul	1268.91(336.12)	**1258.10(342.97)**	1278.97(349.15)	1370.34(386.83)
Aug	**1226.30(382.12)**	1248.04(383.06)	1288.27(458.83)	1319.61(469.74)
Sep	**1121.66(355.65)**	1195.19(371.78)	1217.84(461.51)	1294.44(503.85)
Oct	**1004.53(428.59)**	1021.23(431.14)	1085.78(499.24)	1084.38(499.42)
Nov	**753.16(146.03)**	754.10(137.16)	762.37(143.18)	826.44(170.69)
Dec	**800.53(208.91)**	806.58(204.50)	855.83(244.04)	857.44(243.27)

Table 4: $\mathcal{RMSE}$ (10-fold cross-validation, 30 iterations), Random Forest.

10-folds cross-validation over 30 iterations. Tab. 4, Tab. 5, and Tab. 6, respectively, show the results of these three experiments. In all tables 3, 4, 5, and 6, the best values for each month are marked in bold.

In order to highlight possible significant statistical differences among ENORA-DS, NSGA-II-DS, RFE-DS and Original-DS (combined results over the entire year) with Random Forest method, we performed the non-parametric *Friedman test* [79] with significance level $\alpha = 0.05$ for the following measures: *Out of Bag Error*, $\mathcal{RMSE}$, *Serialized train set size*, and *Training time*. Previously, we used a *Saphiro-Wilk normality test* to make sure that data do not derive from a normal distribution. Since the Friedman test determined that significant statistical differences exist in all cases, we performed a non-parametric *Nemenyi multiple comparison post hoc test* to highlight the origin of such a difference. Tab. 7 shows the results of this analysis: boldfaced numbers indicate that statistical differences exist between the corresponding pair of data bases.

In a successive phase, we evaluated the subset of selected features by building other regression models over them, and we compared the performances of such models

Month	ENORA-DS	NSGA-II-DS	RFE-DS	Original-DS
Jan	**118890.70(51.99)**	**118889.70(51.99)**	544961.30(239.50)	915365.90(402.97)
Feb	107973.30(47.18)	**86132.50(37.56)**	680720.30(299.60)	915365.90(402.97)
Mar	129789.10(56.79)	**86126.50(37.56)**	773476.70(340.47)	915365.90(402.97)
Apr	293583.10(128.92)	**80648.30(35.16)**	451607.90(198.63)	915345.90(402.97)
May	**80825.30(35.16)**	97033.90(42.37)	124207.90(54.39)	915345.90(402.97)
Jun	**75189.10(32.75)**	75191.10(32.75)	113289.50(49.58)	915345.90(402.97)
Jul	**75193.10(32.75)**	86112.50(37.56)	167851.50(73.62)	915345.90(402.97)
Aug	151617.90(66.41)	**102485.10(44.77)**	735283.30(323.64)	915345.90(402.97)
Sep	173459.70(76.03)	**86118.50(37.56)**	462477.30(203.44)	915345.90(402.97)
Oct	102494.10(44.77)	**53357.30(23.14)**	909879.70(400.57)	915347.90(402.97)
Nov	206210.90(90.45)	**75186.10(32.75)**	146029.70(64.01)	915347.90(402.97)
Dec	97034.90(42.37)	**86108.50(37.56)**	882598.70(388.55)	915347.90(402.97)

Table 5: *Serialized train set size* (10-fold cross-validation, 30 iterations), Random Forest.

Month	ENORA-DS	NSGA-II-DS	RFE-DS	Original-DS
Jan	**0.18(0.01)**	**0.18(0.01)**	0.28(0.01)	0.34(0.01)
Feb	0.17(0.01)	**0.16(0.01)**	0.33(0.01)	0.39(0.01)
Mar	0.21(0.01)	**0.18(0.01)**	0.38(0.01)	0.41(0.01)
Apr	0.24(0.01)	**0.17(0.01)**	0.29(0.01)	0.38(0.01)
May	**0.14(0.01)**	**0.14(0.01)**	0.18(0.01)	0.32(0.01)
Jun	**0.14(0.01)**	**0.14(0.01)**	0.18(0.01)	0.31(0.01)
Jul	**0.13(0.01)**	**0.13(0.01)**	0.18(0.01)	0.31(0.01)
Aug	0.16(0.01)	**0.14(0.01)**	0.27(0.01)	0.29(0.01)
Sep	0.16(0.01)	**0.13(0.01)**	0.23(0.01)	0.29(0.01)
Oct	0.12(0.01)	**0.11(0.01)**	0.29(0.01)	0.29(0.01)
Nov	0.17(0.01)	**0.12(0.01)**	0.15(0.01)	0.28(0.01)
Dec	**0.11(0.01)**	0.12(0.01)	0.26(0.01)	0.27(0.01)

Table 6: *Training time* (10-fold cross-validation, 30 iterations), Random Forest.

against each other, and against the models that can be built over the original data set. The purpose of this test is to verify how robust is the selected data set at the moment of evaluating them with methods different from the one used in the wrapper method. We chose several representative regression methods, shown in Tab. 8, different from Random Forest.

For each month, each data set (ENORA-DS), (NSGA-DS) and (RFE-DS), and for each regression algorithm named above, we performed a 10-fold cross-validation 30 times with training data using the *train* function of Caret R, with the following control command:

trainControl(method="repeatedcv", repeats=30, number=10)

After re-sampling, the *train* function automatically chooses the tuning parameters associated with the best value. Tab. 9, Tab 10, Tab. 11, and Tab. 12 show the root mean squared error of this evaluation, in the case, respectively, of ENORA-DS, NSGA-II-DS, RFE-DS, and the Original-DS. Tab. 13 shows, for each month, a summary of

Meassure	Test de Friedmam p-value	Pairwise comparisons using Nemenyi multiple comparison post hoc test			
Out Of Bag error	$< 10^{-5}$		ENORA-DS	NSGA-II-DS	Original-DS
		NSGA-II-DS	0.98906	–	–
		Original-DS	$\mathbf{5.6e-05}$	$\mathbf{0.00023}$	–
		RFE-DS	0.11950	0.22910	0.11950
$\mathcal{RMSE}$	$< 10^{-5}$		ENORA-DS	NSGA-II-DS	Original-DS
		NSGA-II-DS	0.38940	–	–
		Original-DS	$\mathbf{4.6e-07}$	$\mathbf{0.00085}$	–
		RFE-DS	$\mathbf{0.00085}$	0.11950	0.38940
Serialized train set size	$< 10^{-5}$		ENORA-DS	NSGA-II-DS	Original-DS
		NSGA-II-DS	0.68534	–	–
		Original-DS	$\mathbf{0.00023}$	$\mathbf{1.1e-06}$	–
		RFE-DS	0.16794	$\mathbf{0.00851}$	0.16794
Training time	$< 10^{-5}$		ENORA-DS	NSGA-II-DS	Original-DS
		NSGA-II-DS	0.68534	–	–
		Original-DS	$\mathbf{0.00032}$	$\mathbf{1.7e-06}$	–
		RFE-DS	0.14218	$\mathbf{0.00653}$	0.22910

Table 7: Results of non-parametric Friedman and Nemenyi tests on *Out Of Bag error*, $\mathcal{RMSE}$, *Serialized train set size* and *Training time* measures with Random Forest classifier.

Method	Brief description	Tuning parameters
M5Rules	Model rules	pruned, smoothed
pcr	Principal component analysis	ncomp (#Components)
relaxo	Relaxed lasso	lambda (Penalty parameter), phi (Relaxation parameter)
foba	Ridge regression with variable selection	k (#Variables retained), lambda (L2 penalty)
leapForward	Linear regression with forward selection	nvmax (Maximum number of predictors)
leapSeq	Linear regression with stepwise selection	nvmax (Maximum number of predictors)
penalized	Penalized linear regression	lambda1 (L1 penalty), lambda2 (L2 penalty)
pprmm	Projection pursuit regression	nterms
ridge	Ridge regression	lambda (Weight decay)

Table 8: Regression algorithms used in the test phase.

Month	M5Rules	pcr	relaxo	foba	leapForward	leapSeq	penalized	ppr	ridge
Jan	34603.87	39944.49	45966.71	37500.19	37415.92	37869.98	37379.13	**31598.47**	35275.13
Feb	17132.59	18612.82	19034.58	19286.81	18760.46	18593.09	18898.90	**15899.25**	18062.85
Mar	8159.44	8120.65	9372.00	8239.20	8118.29	7939.63	**7594.08**	8092.80	7801.88
Apr	5190.18	5361.07	5457.76	5246.77	5182.62	5046.50	5127.59	6873.05	**4879.28**
May	4507.22	5977.60	6942.10	5664.50	6119.83	5931.54	6251.55	**4076.10**	5847.55
Jun	2204.07	2498.04	2704.76	2465.42	2399.40	2570.92	2490.50	**2190.16**	2463.37
Jul	1617.91	1606.93	1592.59	**1532.73**	1570.04	1578.42	1561.18	1547.95	1537.96
Aug	1800.30	1584.01	1630.53	1596.17	1604.98	**1539.29**	1566.73	1650.14	1573.18
Sep	1823.91	1830.78	1947.19	1698.50	**1662.70**	1680.16	1705.14	1787.16	1666.77
Oct	1350.76	1332.05	1539.98	1433.86	1362.09	1317.14	1318.90	**1260.11**	1287.13
Nov	952.71	981.11	993.34	950.29	947.43	**947.08**	966.96	1021.80	963.59
Dec	989.73	1031.11	1053.37	991.17	1014.47	**956.25**	1023.27	1012.41	1032.53

Table 9: $\mathcal{RMSE}$ obtained with ENORA for selected regression methods (different from Random Forest).

the average $\mathcal{RMSE}$ obtained with the reduced data bases for the selected regression methods (different from Random Forest). Figure 6 shows, for each month, the $\mathcal{RMSE}$

Month	M5Rules	pcr	relaxo	foba	leapForward	leapSeq	penalized	ppr	ridge
Jan	37661.94	41049.32	49266.68	38883.05	37587.26	37498.97	39589.37	**29116.69**	35864.30
Feb	17625.63	18643.31	19252.90	18187.14	18641.96	19016.75	18858.16	**14491.27**	18012.17
Mar	8331.65	8313.57	8189.17	**7757.02**	8042.78	8319.33	8021.13	7770.78	8263.46
Apr	5048.76	5331.02	4839.14	5107.47	5015.22	5232.16	5104.78	**4575.82**	4820.10
May	4817.66	6226.26	6724.66	6143.54	6342.04	6171.63	5980.49	**3892.11**	5775.98
Jun	2407.59	2701.44	3133.42	2394.11	2503.58	2356.52	2424.85	**2241.70**	2519.73
Jul	1657.79	1567.14	1619.78	1519.98	1557.70	1600.84	1557.53	1616.61	**1490.93**
Aug	1527.50	1602.16	1649.77	**1443.50**	1524.98	1511.47	1534.44	1477.60	1517.24
Sep	1678.39	1621.86	2125.08	1652.55	**1603.14**	1659.52	1659.12	1646.77	1773.65
Oct	1313.75	1329.80	1419.01	1333.20	1300.71	1265.15	1290.56	1260.79	**1254.84**
Nov	933.20	978.27	992.91	974.74	955.68	919.76	959.27	**916.21**	979.23
Dec	986.53	1038.50	1084.35	968.62	1023.88	1073.27	1005.11	**943.87**	1002.81

Table 10: $\mathcal{RMSE}$ obtained with NSGA-II for selected regression methods (different from Random Forest).

Month	M5Rules	pcr	relaxo	foba	leapForward	leapSeq	penalized	ppr	ridge
Jan	35755.42	39855.98	41583.29	**35689.32**	36108.56	37682.69	41252.64	45814.33	40857.73
Feb	19259.82	19591.13	18606.47	18426.46	18757.90	**17718.72**	18625.31	24634.44	18630.90
Mar	9371.68	8359.13	8938.21	8242.65	8365.99	**8143.50**	8765.66	11283.30	8979.27
Apr	5545.57	**5107.87**	5142.34	5308.94	5548.14	5151.10	5534.55	6193.76	5344.75
May	5206.87	5791.20	5451.56	5229.36	**5103.85**	5642.37	5440.71	5391.23	5139.80
Jun	2290.28	2461.17	2440.11	2310.55	2283.62	2305.09	**2183.48**	2231.66	2274.90
Jul	1616.27	1537.93	1543.40	1506.39	1540.78	1530.69	1491.90	1537.00	**1479.78**
Aug	1720.56	**1597.32**	1676.43	1616.73	1610.66	1603.76	1597.38	1841.39	1618.78
Sep	1868.15	1546.99	**1546.67**	1575.85	1763.18	1599.15	1633.98	1800.92	1761.31
Oct	1391.19	1426.45	1396.30	**1333.17**	1510.83	1456.92	1494.97	1696.83	1405.26
Nov	960.20	1008.73	**905.27**	955.06	937.32	934.52	920.28	929.57	924.16
Dec	1115.10	1041.47	1042.13	987.63	971.29	**956.16**	1105.92	1360.84	1062.73

Table 11: $\mathcal{RMSE}$ obtained with RFE for selected regression methods (different from Random Forest).

Month	M5Rules	pcr	relaxo	foba	leapForward	leapSeq	penalized	ppr	ridge
Jan	37771.92	39781.05	51433.83	37832.75	39243.78	**36821.46**	39655.62	44928.11	39500.97
Feb	18385.42	18417.80	26642.44	18108.39	18719.91	**18107.89**	19615.01	22560.59	19638.13
Mar	8851.09	8454.30	9931.38	8467.71	**8152.61**	8312.55	8994.29	11045.26	8623.85
Apr	5864.49	5420.78	5421.95	**4989.79**	5229.75	5347.82	5786.63	5993.44	5591.12
May	5660.75	5583.07	5816.17	5810.09	5548.37	**5256.34**	6263.03	6514.27	5860.86
Jun	2388.08	2433.88	2422.85	**2341.31**	2531.53	2396.78	2539.45	3288.81	2468.42
Jul	1748.18	1599.28	1787.29	1597.30	1609.93	**1572.89**	1733.54	1910.15	1683.14
Aug	1732.78	1554.02	1832.42	1581.32	1565.83	**1550.48**	1698.90	1848.44	1719.04
Sep	1802.76	**1611.93**	1785.30	1754.48	1721.44	1751.51	1770.80	1884.32	1717.33
Oct	1584.69	1385.78	1426.35	**1256.98**	1291.12	1298.66	1506.65	1764.75	1408.93
Nov	1031.78	1047.78	1004.21	949.70	941.15	**935.78**	1043.80	1244.28	1007.89
Dec	1058.34	1064.30	1001.22	1011.96	**965.88**	973.72	1148.47	1358.90	1051.72

Table 12: $\mathcal{RMSE}$ obtained with original data bases for selected regression methods (different from Random Forest).

obtained by each method divided by the average $\mathcal{RMSE}$, providing a graphical vision of the deviation of each method with respect to the average.

Again, in order to highlight possible significant statistical differences among ENORA-DS, NSGA-II-DS, RFE-DS and Original-DS, in this case, using $\mathcal{RMSE}$ with the combined data for entire year and with respect to all regression methods, we performed the Saphiro-Wilk normality test and the Friedman test; in this case, as shown in Tab. 14, no statistical differences emerge ($p-value \geq \alpha$), so that we did not perform the Nemenyi

Month	ENORA-DS	NSGA-II-DS	RFE-DS	Original Data Base
Jan	**37505.99**	38501.95	39399.99	40774.39
Feb	18253.48	**18081.03**	19361.24	20021.73
Mar	8159.78	**8112.10**	8938.82	8981.45
Apr	5373.87	**5008.27**	5430.78	5516.20
May	5702.00	5786.04	**5377.44**	5812.55
Jun	2442.96	2520.33	**2308.99**	2534.57
Jul	1571.75	1576.48	**1531.57**	1693.52
Aug	1616.15	**1532.07**	1653.67	1675.92
Sep	1755.81	1713.34	**1677.36**	1755.54
Oct	1355.78	**1307.53**	1456.88	1435.99
Nov	969.37	956.58	**941.68**	1022.93
Dec	**1011.59**	1014.11	1071.47	1070.50
Average	**7143.21**	7175.82	7429.16	7691.27

Table 13: Summary of the average $\mathcal{RMSE}$ obtained with the reduced data bases for the selected regression methods (different from Random Forest).

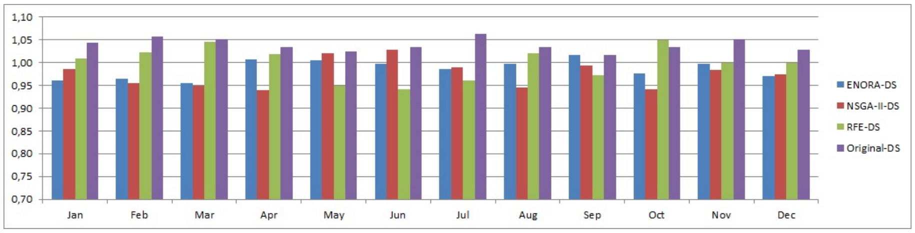


Figure 6: $\mathcal{RMSE}$ divided by average $\mathcal{RMSE}$ obtained with the reduced data bases for the selected regression methods.

multiple comparison post hoc test.

Measure	Test de Friedman p-value	Pairwise comparisons using Nemenyi multiple comparison post hoc test
$\mathcal{RMSE}$	0.5062	–

Table 14: Results of non parametric Friedman and Nemenyi tests over $\mathcal{RMSE}$ over all months and all selected regression methods (different from Random Forest).

4.6. Discussion

The above evaluations allow us to build a regression model by choosing, month by month, the best data set. Notice that our choice is based on the performances in terms of the $\mathcal{RMSE}$ obtained with all evaluated regression methods. It turns out that

Random Forest gave us the best result at each month, and that ENORA gave us the best data set in each month except July (where the best data set is given by NSGA-II). Tab. 15 shows the final regression model, where, for each month, we indicate the set of chosen attributes, the search strategy that generated it, and the regression method that gave the best results.

Month	RMSE	Method	Algorithm	Selected Attributes
Jan	24889.83	ENORA	Random Forest	Quan_2, Quan_4, Quan_11, Cat_6, Quan_17, Cat_126, Cat_183 Cat_198, Cat_205, Cat_209, Cat_223, Cat_238, Cat_374, Cat_381 Cat_398, Cat_450, Cat_454
Feb	11843.42	ENORA	Random Forest	Cat_1, Quan_4, Cat_4, Cat_13, Cat_164, Cat_171, Cat_240 Cat_311, Cat_334, Cat_371, Cat_375, Quan_19, Quan_21, Quan_25 Cat_494
Mar	6264.64	ENORA	Random Forest	Cat_1, Date_1, Quan_2, Quan_4, Quan_6, Cat_2, Cat_3 Cat_4, Cat_114, Cat_121, Cat_168, Cat_233, Cat_235, Cat_238 Cat_295, Cat_313, Cat_425, Cat_452, Cat_463
Apr	3812.46	ENORA	Random Forest	Quan_2, Quan_4, Quan_6, Cat_2, Cat_6, Cat_10, Cat_23 Cat_84, Quan_17, Cat_114, Cat_119, Cat_131, Cat_132, Cat_157 Cat_162, Cat_172, Cat_183, Cat_203, Cat_209, Cat_213, Cat_216 Cat_238, Cat_240, Cat_259, Cat_260, Cat_296, Cat_311, Cat_312 Cat_318, Cat_334, Cat_336, Cat_340, Cat_341, Cat_346, Cat_353 Cat_363, Cat_366, Cat_367, Cat_376, Cat_388, Cat_397, Cat_403 Cat_430, Cat_435, Cat_445, Cat_452, Cat_454, Quan_25, Cat_467
May	2635.12	ENORA	Random Forest	Quan_2, Cat_3, Cat_6, Cat_178, Cat_191, Cat_214, Cat_215 Cat_388, Cat_400, Quant_24,
Jun	1591.74	ENORA	Random Forest	Quan_2, Quan_8, Cat_3, Cat_4, Cat_6, Cat_12, Cat_205 Cat_303, Cat_491
Jul	1258.10	NSGA-II	Random Forest	Date_1, Quan_2, Quan_4, Quan_9, Quan_14, Cat_210 Cat_258, Cat_323, Cat_425, Cat_481, Cat_494
Aug	1226.30	ENORA	Random Forest	Date_1, Quan_2, Quan_3, Quan_4, Quan_8, Quan_12, Cat_3 Cat_11, Cat_15, Quan_16, Cat_178, Cat_205, Cat_235, Cat_297 Cat_311, Cat_313, Cat_316, Cat_346, Quan_21, Cat_450, Cat_451 Cat_454, Cat_483
Sep	1121.66	ENORA	Random Forest	Date_1, Quan_2, Quan_5, Quan_7, Quan_9, Cat_84, Quan_16 Quan_17, Cat_115, Cat_122, Cat_148, Cat_151, Cat_209, Cat_221 Cat_235, Cat_260, Cat_295, Cat_300, Cat_308, Cat_334, Cat_371 Cat_374, Quan_21, Cat_442, Cat_459, Cat_469, Cat_483
Oct	1004.53	ENORA	Random Forest	Date_1, Quan_2, Quan_4, Quan_8, Cat_157, Cat_191, Cat_209 Cat_213, Cat_304, Cat_376, Cat_397, Cat_403, Cat_443, Cat_481
Nov	753.16	ENORA	Random Forest	Date_1, Quan_2, Quan_3, Quan_4, Quan_8, Quan_9, Cat_2 Cat_3, Cat_12, Quan_16, Cat_114, Cat_157, Cat_168, Cat_178 Cat_183, Cat_198, Cat_203, Cat_213, Cat_214, Cat_215, Cat_226 Cat_249, Cat_300, Cat_311, Cat_340, Cat_341, Cat_397, Cat_403 Cat_436, Cat_442, Cat_454, Cat_459, Quant_24
Dec	800.53	ENORA	Random Forest	Date_1, Quan_2, Quan_4, Cat_12, Cat_223, Cat_226, Cat_235 Cat_323, Cat_371, Cat_381, Quan_21, Cat_462, Cat_498

Table 15: Final regression model, obtained with the best model for each month.

We can now analyze the obtained solutions based on the results of the tests. In terms of hypervolume, comparing the performances of ENORA and NSGA-II to the task of feature selection, and evaluated with Random Forest, we can conclude that ENORA provided better values than NSGA-II, and, therefore, the former performed better than the latter. The 95% confidence intervals of the mean based upon the t-distribution have been performed with samples of 30 individuals, and so the results are significant, leading us to conclude that the differences between the hypervolume values obtained with the algorithms are statistically significant. The values obtained by ENORA are (significantly) better than those obtained by NSGA-II in each of the 12 months. As far as evaluating the performances of the reduced data sets with Random Forest is concerned, we observed that all feature selection methods (ENORA, NSGA-II, and RFE) improved the performances of all examined indexes, namely $\mathcal{RMSE}$, Out-of-Bag error, Serialized train set size and Training time, over the original data set, and all of them effectively reduced the number of features. ENORA and NSGA-II provided smaller Out-of-Bag errors, in general, than RFE; in term of Out-of-Bag error, moreover, NSGA-II is worse than ENORA in 7 out of 12 months, and RFE is worse than ENORA in 10 cases. ENORA always obtains better $\mathcal{RMSE}$ values than NSGA-II in 11 cases and better $\mathcal{RMSE}$ values than RFE in all 12 months. The serialized train set size and training time obviously depends on the number of selected features. In general, NSGA-II obtains reduced data bases with fewer or an equal number of attributes than ENORA (at the expenses of the error). The main reason for this is that NSGA-II maintains worse diversity than ENORA, and so individuals with a greater number of attributes are eliminated more quickly.

By observing the behaviour of the reduced data sets in other model learners, we can conclude that Random Forest presented better performances than every other algorithm, which was to be expected given that the wrapper-based feature selection mechanisms were trained with Random Forest as evaluator. The purpose of this evaluation was to establish the robustness of the data sets chosen by ENORA and NSGA-II when they are evaluated with regression methods different from the one used in the wrapper. Despite the fact that ENORA does not always give the best result in this comparison, it gives the best results in average. This implies that the data set chosen by ENORA is effective even for model learners different from the one in which it is trained, at least for the problem at hand.

Concerning the non-parametric statistical tests (for Random Forest), we can conclude that: in terms of Out Of Bag Error, ENORA-DS and NSGA-II-DS are statistically better than Original-DS; in terms of $\mathcal{RMSE}$, ENORA-DS is better than Original-DS and RFE-DS, while NSGA-II-DS is better than Original-DS; in terms of Serialized train set size, ENORA-DS is better than Original-DS, while NSGA-II-DS is better than Original-DS and RFE-DS; finally, in terms of Training time, ENORA-DS is better than Original-DS, while NSGA-II-DS is better than Original-DS and RFE-DS. Concerning the non-parametric statistical test performed with other regression methods, there are no significant statistical differences among the tested data sets for the $\mathcal{RMSE}$.

Separating the original data set into 12 months and performing feature selection to each one of them separately allows us to perform a meta-analysis of the results that goes beyond the construction of a precise regression model. We can do so by further analysing Tab. 15, and, in particular, by relating features with the month(s) in which they have been chosen. In this way, we can observe that there are particularly important features such as Quan_2 and Quan_4, chosen, respectively, 11 and 9 times out of 12. In Tab 16, we reported the 10 most chosen attributes. Although features are encoded, so that no common-sense or domain-related analysis can be performed, it can be very useful to understand which attributes influence most the outcome, and why. Moreover, by thinking of the attributes and the months as meta-data, we can perform simple analysis; months can be clustered by the attributes that have been selected in each of them,

Feature Name	Sum	Jan	Feb	Mar	Apr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dec
Quan_2	11	●		●	●	●	●	●	●	●	●	●	●
Quan_4	9	●	●	●	●			●	●		●	●	●
Date_1	7			●				●	●	●	●	●	●
Cat_3	5			●		●	●		●			●	
Cat_209	4	●			●					●	●		●
Cat_235	4			●					●	●			
Cat_311	4		●		●				●			●	
Cat_454	4	●			●				●			●	
Cat_6	4	●			●	●	●						
Quan_8	4						●		●		●	●	

Table 16: Most selected attributes.

to obtain (not surprisingly) precisely two clusters: one of them is composed by a single month, April, and the other one contains all remaining months. This clearly indicates that, in terms of sales, April differs from every other month. Similarly, we can cluster the attributes based on the months in which they have been chosen. Doing so reveals 5 distinguished clusters (with a high log likelihood), the most populated of which is the one that contains the features that have never been chosen, or have been chosen only once; a second well-defined cluster contains precisely the attributes Quan_2, Quan_4, and Date_1 (the three most chosen attributes); the other clusters contain attributes simply grouped by (roughly) the frequency of their appearance.

5. Conclusions

In this paper, we have applied the multi-objective evolutionary algorithm ENORA to the task of feature selection for sales prediction in online advertising. We proposed a methodology to integrate feature selection for regression, model evaluation, and decision making in order to choose the most satisfactory model according to a *a posteriori* process in a multi-objective context. To the best of our knowledge, there exists no previous attempt to integrate a multi-objective search strategy in a wrapper-based feature selection process for regression (similar applications have been previously attempted for supervised, and, less frequently, unsupervised classification). We compared the performances of ENORA to this task against those of the well-known multi-objective search strategy NSGA-II, and we concluded that the former gives better hypervolume values, and returns more precise data sets, than the latter. We also compared the selection that is the result of the application of our methodology against a selection from the application of the well-known wrapper method Recursive Feature Elimination, concluding that the proposed method ENORA performs better than RFE, both in terms of means error of the resulting regression model and in terms of cardinality of the chosen subset. Finally, we compared, under several indexes, the performances of the data sets chosen by ENORA and NSGA-II in a wide range of regression model learners different from the one used for training, that is, Random Forest, concluding that the data set chosen by ENORA performs better than the others in average. Our data were borrowed from the online competition site known as Kaggle (see *https://www.kaggle.com*). Data are encoded to protect the source, and our aim was not to build a regression model that was specific for this particular data set, but, rather, to present a novel - general - technique that can be applied to the problem of predicting online sales. However, it is interesting to compare the performances of our regression model with those of the winners (as retrieved on the web page as of May 2016) of the competition itself. Such a comparison cannot be done in a precise way, given that the test data set is not annotated. Nevertheless, we computed the *root mean logarithmic squared error* ($\mathcal{RMLSE}$) for each training data sets in 10-fold cross-validation (500 trees in Random Forest), obtaining the following results. In average, our regression model scores in the best 42% (with a $\mathcal{RMLSE}$ of 0.7003), in the best 12% in 5 months out of 12 (with a $\mathcal{RMLSE}$ of less than 0.6050), and it scores the best in 2 cases out of 12 (with a $\mathcal{RMLSE}$ of less than 0.5325). This proves that our methodology, although not specific for this particular data set, can be successfully applied to it.

By selecting suitable variables, more efficient models for sales forecasting can be constructed. Taking this into account, one interesting line of research consists of applying the variable selection to the CTR (Click Through Rate) prediction problem, as well as the problem of predicting the relationship between the number of clicks and the number of advert impressions (i.e., the probability that an advert will receive a click). Advertising networks tend to give priority to the most profitable adverts in order to increase their income; accurately estimating the likelihood of clicks allows online networks to choose the most profitable adverts. Another, closely related, line of research concerns computing the probability of an advert being treated as spam. A spam advert occurs when the advertiser has a malicious intent and pretends to defraud users with a product or by installing a virus in their computer. This kind of advertisement is particularly harmful, as users lose trust, making the value of advertising to decline. Variable selection can be applied to this problem, as well. In the same context, one of the biggest challenges of online advertising is to eradicate click fraud. In the Pay per Click model, publishers receive their income according to the number of clicks. Therefore, many fraudulent publishers try to cheat by false clicks platforms. While there exist several models specifically devoted to recognize false clicks, it would be interesting to see in which measure feature selection can improve such models. Other, more technical, research directions include the integration of ENORA as search strategy in multivariate filters, and the implementation of other heuristic search algorithms (such as multi-objective Particle Swarm Optimization) for feature selection enhanced with ENORA individuals' selection strategy.

Acknowledgements

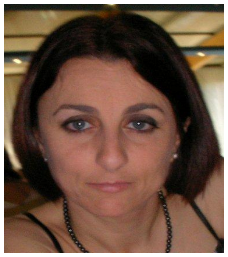
This work was supported by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA-CIEMAT belongs to CIEMAT and the Government of Spain. This research was also partially supported by Ministerio de Economía y Competitividad (Spain) and the ERDF program of the European Union: CAPAP-H5 network (TIN2014-53522-REDT), the Spanish projects TIN2013-45491-R and TIN2015-66972-C5-3-R supported by Ministerio de Economía y Competitividad, and the Italian INdAM GNCS "Project 2016 Logic, automata and games for self-adaptive systems".

Finally, the authors would like to thank prof. José T. Palma Méndez (University of Murcia - Spain) for his valuable help in designing and performing the non-parametric statistical tests to our results.

Authors' Biographies

Fernando Jiménez. Fernando Jiménez is Associate Professor at the Department of Communications and Information Engineering of the University of Murcia (Spain). He received an M.D. in Computer Science in 1991 from the University of Granada. He obtained a PhD in 1996 from the University of Murcia for his research on Evolutionary Computation applied to Fuzzy Transportation Problems. His research interests include Evolutionary Computation, Multi-objective Constrained Optimization, Soft Computing, Evolutionary Fuzzy Systems and Data Mining.

Gracia Sánchez. Gracia Sánchez is an Associate Professor at the Department of Communications and Information Engineering of the University of Murcia (Spain). He received an M.D. in Computer Science in 1996 from the Polytechnic University of Valencia and a PhD in 2002 from the University of Murcia for his research on Multi-Objective Evolutionary Algorithms and Fuzzy models applied to Non-Linear optimization problems. His research interests include Multi-objective Constrained Optimization, Soft Computing, Evolutionary Fuzzy Systems and Data Mining.



José M. García. José M. García is Full Professor of Computer Architecture at the University of Murcia (Spain), and also the Head of the Parallel Computer Architecture Research Group. He served as the Dean of the School of Computer Science from 2006 to 2012, and he specializes in Computer Architecture and Interconnection Networks. His current research interests focus in the design of power-efficient heterogeneous systems, and the development of data-intensive applications for those systems, especially bio-inspired evolutionary algorithms, and bio-informatics applications.

Guido Sciavicco. Guido Sciavicco holds a Master Degree in Computer Science and a PhD in Computer Science, both from the University of Udine (Italy). He is now Associate Professor at the University of Ferrara (Italy). He has been researching in temporal, spatial, and modal logic for over 10 years, and, recently, he has also focused on data mining and temporal data mining problems with applications.

Luis Miralles. Luis Miralles graduated from the University of Murcia (Spain) in Computer Engineering, and specialising in Artificial intelligence. He also holds a Master Degree in Computer Security from the International University of La Rioja (Spain). Currently, he is working full-time as a Research Professor at the Panamerican University (Mexico), while completing his PhD on online advertising and machine learning at the University of Murcia (Spain).

References

[1] J. Armstrong, Principles of forecasting: a handbook for researchers and practitioners, Kluwer, 2001.

[2] M. Falk, E. Hagsten, E-commerce trends and impacts across europe, International Journal of Production Economics 170 (A) (2015) 357 – 369.

[3] M. Braun, W. Moe, Online display advertising: Modeling the effects of multiple creatives and individual impression histories, Marketing Science 32 (5).

[4] F. Thiesing, O. Vornberger, Sales forecasting using neural networks, in: Proc. of the IEEE International Conference on Neural Networks, Vol. 4, 1997, pp. 2125–2128.

[5] C. Chen, W. Lee, H. Kuo, C. Chen, K. Chen, The study of a forecasting sales model for fresh food, Expert Systems with Applications 37 (12) (2010) 7696 – 7702.

[6] C. Lu, Sales forecasting of computer products based on variable selection scheme and support vector regression, Neurocomputing 128 (2014) 491 – 499.

[7] A. Gelman, J. Hill, Data Analysis Using Regression and Multilevel/Hierarchical Models, Cambridge University Press, 2007.

[8] V. Kumar, S. Minz, Feature selection: A literature review, Smart CR 4 (3) (2014) 211–229.

[9] R. Caruana, D. Freitag, Greedy attribute selection, in: In Proceedings of the Eleventh International Conference on Machine Learning, Morgan Kaufmann, 1994, pp. 28–36.

[10] A. Arauzo-Azofra, J. Benitez, J. Castro, Consistency measures for feature selection, Journal of Intelligence Information Systems 30 (3) (2008) 273–292.

[11] X. Zhang, Y. Hu, K. Xie, S. Wang, E. Ngai, M. Liu, A causal feature selection algorithm for stock prediction modeling, Neurocomputing 142 (2014) 48 – 59.

[12] B. Blesser, T. Kuklinski, R. Shillman, Empirical tests for feature selection based on a psychological theory of character recognition, Pattern Recognition 8 (2) (1976) 77 – 85.

[13] J. Tang, H. Liu, Feature selection for social media data, ACM Trans. Knowl. Discov. Data 8 (4) (2014) 1–27.

[14] G. Nandi, An enhanced approach to las vegas filter (lvf) feature selection algorithm, in: Emerging Trends and Applications in Computer Science (NCETACS), 2011 2nd National Conference on, 2011, pp. 1–3.

[15] H. Vafaie, K. D. Jong, Genetic algorithms as a tool for feature selection in machine learning, in: Proc. of the 4th International Conference on Tools with Artificial Intelligence (TAI), 1992, pp. 200–203.

[16] S. Dreyer, Evolutionary feature selection, Ph.D. thesis, Norvegian University of Science and Technology (2013).

[17] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[18] K. Deb, A. Pratab, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii., IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182 – 197.

[19] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (1-3) (2002) 389–422.

[20] C. Coello, V. D.V., L. G.B., Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic/Plenum publishers, New York, NY, USA, 2002.

[21] K. Deb, Multi-objective optimization using evolutionary algorithms, Wiley, London, UK, 2001.

[22] F. Jiménez, A. Gómez-Skarmeta, G. Sánchez, K. Deb, An evolutionary algorithm for constrained multi-objective optimization, in: Proc. of the 2002 IEEE Congress on Evolutionary Computation (CEC), Vol. 2, IEEE, 2002, pp. 1133–1138.

[23] F. Jiménez, G. Sánchez, J. M. Juárez, Multi-objective evolutionary algorithms for fuzzy classification in survival prediction, Artificial Intelligence in Medicine 60 (3) (2014) 197–219.

[24] F. Jiménez, E. Marzano, G. Sánchez, G. Sciavicco, N. Vitacolonna, Attribute selection via multi-objective evolutionary computation applied to multi-skill contact center data classification, in: Proc. of the 2015 IEEE Symposium on Computational Intelligence in Big Data (IEEE CIBD), 2015, pp. 488–495.

[25] F. Jiménez, R. Jodár, G. Sánchez, M. Martín, G. Sciavicco, Multi-objective evolutionary computation based feature selection applied to behaviour assessment of children, in: Proc. of the 2016 International Conference on Educational Data Mining (ICEDM), Vol. 2(6), 2016, pp. 1888–1897.

[26] I. Rechenberg, Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution, Frommann-Holzboog, Stuttgart, Germany, 1973.

[27] H. Schwefel, Numerical Optimization of Computer Models, John Wiley & Sons, Inc., New York, NY, USA, 1981.

[28] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1994) 221–248.

[29] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[30] A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. Cortina-Januchs, D. Andina, Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network, in: Proc. of the 36th Annual Conference on IEEE Industrial Electronics Society (IECON), 2010, pp. 2845–2850.

[31] S. Cotter, K. Kreutz-Delgado, B. Rao, Backward sequential elimination for sparse vector subset selection, Signal Processing 81 (9) (2001) 1849 – 1864.

[32] P. M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. 26 (9) (1977) 917–922.

[33] P. Gupta, D. Doermann, D. DeMenthon, Beam Search for Feature Selection in Automatic SVM Defect Classification, in: Proc. of the 16th International Conference on Pattern Recognition, 2002, pp. 212–215.

[34] H. Vafaie, K. De Jong, Genetic algorithms as a tool for feature selection in machine learning, in: Tools with Artificial Intelligence, 1992. TAI '92, Proceedings., Fourth International Conference on, 1992, pp. 200–203.

[35] S. Dreyer, Evolutionary feature selection, in: Norwegian University of Science and Technology, Department of Computer and Information Science, Institutt for datateknikk og informasjonsvitenskap, 2013, p. 76.

[36] L. Cervante, B. Xue, M. Zhang, L. Shang, Binary particle swarm optimisation for feature selection: A filter based approach, in: Proc. of the 2012 IEEE Congress on Evolutionary Computation (CEC), 2012, pp. 1–8.

[37] Z. Yong, G. Dun-wei, Z. Wan-Qiu, Feature selection of unreliable data using an improved multi-objective PSO algorithm, Neurocomputing 171 (2016) 1281 – 1290.

[38] C. Chen, On information and distance measures, error bounds, and feature selection, Information Sciences 10 (2) (1976) 159 – 173.

[39] L. Sun, J. Xu, Y. Tian, Feature selection using rough entropy-based uncertainty measures in incomplete decision systems, Knowledge-Based Systems 36 (2012) 206 – 216.

[40] A. Al-Ani, M. Deriche, Feature selection using a mutual information based measure, in: Pattern Recognition, 2002. Proceedings. 16th International Conference on, Vol. 4, 2002, pp. 82–85 vol.4.

[41] S. Das, Feature selection with a linear dependence measure, IEEE Transactions on Computers C-20 (9) (1971) 1106–1109.

[42] R. Kohavi, J. George, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324.

[43] A. Khalili, An overview of the new feature selection methods in finite mixture of regression models, Journal of the Iranian Statistical Society 10 (2) (2011) 201 – 235.

[44] W. Siedlecki, J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recognition Letters 10 (5) (1989) 335 – 347.

[45] M. ElAlami, A filter model for feature subset selection based on genetic algorithm, Knowledge-Based Systems 22 (5) (2009) 356 – 362.

[46] R. Anirudha, R. Kannan, N. Patil, Genetic algorithm based wrapper feature selection on hybrid prediction model for analysis of high dimensional data, in: Proc. of the 9th International Conference on Industrial and Information Systems (ICIIS), 2014, pp. 1–6.

[47] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognition Letters 28 (13) (2007) 1825 – 1844.

[48] A. F. Gómez-Skarmeta, F. Jiménez, J. Ibánez, S. Paredes, Evolutionary variable identification, in: Proc. of 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99), 1999.

[49] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems and their Applications 13 (2) (1998) 44–49.

[50] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. C. Coello, A survey of multiobjective evolutionary algorithms for data mining (part I), IEEE Transactions on Evolutionary Computation 18 (1) (2014) 4–19.

[51] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. C. Coello, A survey of multiobjective evolutionary algorithms for data mining (part II), IEEE Transactions on Evolutionary Computation 18 (1) (2014) 20–35.

[52] H. Ishibuchi, Multi-objective pattern and feature selection by a genetic algorithm, in: Proc. of Genetic and Evolutionary Computation Conference (GECCO'2000, Morgan Kaufmann, 2000, pp. 1069–1076.

[53] C. Emmanouilidis, A. Hunter, J. MacIntyre, C. Cox, A multi-objective genetic algorithm approach to feature selection in neural and fuzzy modeling, Journal of Evolutionary Optimization, An International Journal on the Internet 3 (1) (2001) 1–26.

[54] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[55] O. Cordon, F. Herrera, M. del Jesus, P. Villar, A multiobjective genetic algorithm for feature selection and granularity learning in fuzzy-rule based classification systems, in: IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, Vol. 3, 2001, pp. 1253–1258 vol.3.

[56] J. Liu, H. Iba, Selecting informative genes using a multiobjective evolutionary algorithm, in: Proceedings of the 2002 IEEE Congress on Evolutionary Computation (CEC), Vol. 1, 2002, pp. 297–302.

[57] G. Pappa, A. Freitas, C. Kaestner, Attribute selection with a multi-objective genetic algorithm, in: Proc. of the 16th Brazilian Symposium on Artificial Intelligence (SBIA), Vol. 2507 of Lecture Notes in Computer Science, Springer, 2002, pp. 280–290.

[58] S. Shi, P. Suganthan, K. Deb, Multiclass protein fold recognition using multiobjective evolutionary algorithms, in: Proc. of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2004, pp. 61–66.

[59] H. Chen, X. Yao, Evolutionary multiobjective ensemble learning based on bayesian feature selection, in: Proc. of the 2006 IEEE Congress on Evolutionary Computation (CEC), 2006, pp. 267–274.

[60] Y. Jin (Ed.), Multi-Objective Machine Learning, Vol. 16 of Studies in Computational Intelligence, Springer, Warsaw, Poland, 2006.

[61] J. García-Nieto, E. Alba, L. Jourdan, E. Talbi, Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis, Information Processing Letters 109 (16) (2009) 887 – 896.

[62] Z. Zhu, Y. Ong, J. Kuo, Feature selection using single/multi-objective memetic frameworks, in: Multi-Objective Memetic Algorithms, Vol. 171 of Studies in Computational Intelligence, Springer, 2009, pp. 111–131.

[63] M. Venkatadri, K. Srinivasa Rao, A multiobjective genetic algorithm for feature selection in data mining, International Journal of Computer Science and Information Technologies 1 (5) (2010) 443–448.

[64] A. Ekbal, S. Saha, C. Garbe, Feature selection using multiobjective optimization for named entity recognition, in: Proc. of the 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 1937–1940.

[65] A. Reynolds, D. Corne, M. Chantler, Feature selection for multi-purpose predictive models: a many-objective task, in: Proc. of the 11th International Conference on Parallel Problem Solving from Nature (PPSN), Vol. 6238 of Lecture Notes in Computer Science, Springer, 2010, pp. 384–393.

[66] A. Gaspar-Cunha, Feature selection using multi-objective evolutionary algorithms: Application to cardiac spect diagnosis, in: Proc. of the 4th International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB), Vol. 74 of Advances in Intelligent and Soft Computing, Springer, 2010, pp. 85–92.

[67] L. Li, M. Li, Y. Lu, Y. Zhang, A new multi-objective genetic algorithm for feature subset selection in fatigue fracture image identification, Journal of Computers 5 (7) (2010) 1105–1111.

[68] P. A. Castro, F. J. Von Zuben, Multi-objective feature selection using a bayesian artificial immune system, International Journal of Intelligent Computing and Cybernetics 3 (2) (2010) 235–256.

[69] B. Krishna, B. Kaliaperumal, Efficient genetic-wrapper algorithm based data mining for feature subset selection in a power quality pattern recognition application, International Arab Journal of Information Technology. 8 (4) (2011) 397–405.

[70] H. Karshenas, P. Larrañaga Múgica, Q. Zhang, C. Bielza, An interval-based multiobjective approach to feature subset selection using joint modeling of objectives and variables, Tech. rep., Facultad de Informática, Universidad Politécnica de Madrid (2012).

[71] J. Zhao, V. B. Fernandes, L. Jiao, I. Yevseyeva, A. Maulana, R. Li, T. Bäck, M. T. M. Emmerich, Multiobjective optimization of classifiers by means of 3-d convex hull based evolutionary algorithm, CoRR abs/1412.5710.

[72] T. Fawcett, An introduction to roc analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

[73] D. Kimovski, J. Ortega, A. Ortiz, R. Baos, Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection, Expert Systems with Applications 42 (9) (2015) 4239 – 4252.

[74] H. Xia, J. Zhuang, D. .Yu, Multi-objective unsupervised feature selection algorithm utilizing redundancy measure and negative epsilon-dominance for fault diagnosis, Neurocomputing 146 (2014) 113 – 124.

[75] C. Tan, C. Lim, Y. Cheah, A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models, Neurocomputing 125 (2012) 217 – 228.

[76] T. Ho, Random decision forest, in: Proc. of the 3rd International Conference on Document Analysis and Recognition, 1995, pp. 278–282.

[77] K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Anaysis and Machine Intellicence 20 (8) (1998) 832–844.

[78] M. Srinivas, L. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Transactions on Systems, Man, and Cybernetics 24 (4) (1994) 656–667.

[79] D. Zimmerman, B. Zumbo, Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks, Journal of Experimental Education 62 (1993) 75–86.