A Novel Outlook on Feature Selection as a Multi-Objective Problem

Pietro Barbiero[1][0000−0003−3155−2564], Evelyne Lutton[2][0000−0003−0889−4427], Giovanni Squillero[1][0000−0001−5784−6435], and Alberto Tonda[2][0000−0001−5895−4809]

[1] Politecnico di Torino, Torino, Italy
pietro.barbiero@studenti.polito.it, giovanni.squillero@polito.it
[2] UMR 782, Université Paris-Saclay, INRA, AgroParisTech, France
evelyne.lutton@inra.fr, alberto.tonda@inra.fr

Abstract. *Feature selection* is the process of choosing, or removing, features to obtain the most informative feature subset of minimal size. Such subsets are used to improve performance of machine learning algorithms and enable human understanding of the results. Approaches to feature selection in literature exploit several optimization algorithms. Multi-objective methods also have been proposed, minimizing at the same time the number of features and the error. While most approaches assess error resorting to the average of a stochastic K-fold cross-validation, comparing averages might be misleading. In this paper, we show how feature subsets with different average error might in fact be non-separable when compared using a statistical test. Following this idea, clusters of non-separable optimal feature subsets are identified. The performance in feature selection can thus be evaluated by verifying how many of these optimal feature subsets an algorithm is able to identify. We thus propose a multi-objective optimization approach to feature selection, EvoFS, with the objectives to i. minimize feature subset size, ii. minimize test error on a 10-fold cross-validation using a specific classifier, iii. maximize the analysis of variance value of the lowest-performing feature in the set. Experiments on classification datasets whose feature subsets can be exhaustively evaluated show that our approach is able to always find the best feature subsets. Further experiments on a high-dimensional classification dataset, that cannot be exhaustively analyzed, show that our approach is able to find more optimal feature subsets than state-of-the-art feature selection algorithms.

Keywords: Feature selection · Machine learning · Multi-objective optimization · Evolutionary algorithms · Multi-objective evolutionary algorithms.

1 Introduction

The field of *machine learning* (ML) deals with algorithms producing predictive models, that are able to improve their performance over time, given an increasing amount of data. *Supervised* ML, a category including the notable examples

of classification and regression, defines a set of problems for which training data is labeled. In ML terminology, data is organized in *samples*, each reporting measurements over a set of *features*; in other terms, samples can be seen as the rows in a dataset, while features can be seen as the columns. The aim of a supervised ML algorithm is to find relationships between features that can reliably predict the value of the *target*, a specific feature in the problem.

While ML algorithms can often be successful at a given task, they might face issues when dealing with large number of features, as an increase in dimensionality creates a corresponding increase in the search space of combinations of features to be explored. As the search space grows, it becomes harder for the ML algorithms to find good optima. Not only, but even when the results are satisfying, very often the predictive models obtained are *black boxes*, that cannot be interpreted by humans. Selecting the features involved in a problem can help not only to reduce the search space for the ML algorithms to explore, but also to make the models more human-readable.

Specialized literature on feature selection shows different approaches to scoring feature subsets, ranging from mutual information to analysis of variance. Evolutionary algorithms (EAs) have been successfully used for feature selection [1,2], even with multi-objective approaches, attempting to both minimize the number of features in a subset, and minimize error [3,4], with recent applications ranging from face recognition [5] to medicine [6].

Most of the proposed feature selection algorithms relying upon classifiers or regressors to obtain an evaluation of a feature subset, exploit a K-fold cross-validation to better assess average error. As this is a stochastic process that will return K error values, a K-fold cross-validation can be assimilated to sampling an unknown probability distribution K times; thus, comparing feature subsets on just their average error might be misleading. A more formally correct approach would be to assess the likelihood that the two sets of K error values have been sampled from two different distributions, using a statistical test. This statistical comparison can uncover to the existence of clusters of feature subsets whose performance is non-separable, and can thus be considered equally optimal. To the best of the author's knowledge, this analysis is usually not considered in feature selection literature.

In this work, we propose a novel approach to multi-objective feature selection, that we call EvoFS. The objectives to be optimized are: i. minimize feature subset size, ii. minimize test error on a 10-fold cross-validation using a specific classifier, iii. maximize the analysis of variance value of the lowest-performing feature in the set. The third objective improves human understanding of the results, as it pushes for feature subsets where all relationships between single features and the target are significant.

Taking into account the statistical considerations of non-separability of performance for feature subsets, it is possible to better assess experimental results. Experimental evaluations on simple datasets, that can be completely analyzed, show that the proposed approach reliably uncovers more feature subsets inside the clusters of the non-separable, optimal ones, when compared to state-of-

the-art feature selection algorithms. Further experiments on a high-dimensional dataset confirm the previous results.

2 Background

2.1 Machine learning

Given a set of samples $\bar{x}$, and a set of corresponding values for a target $\bar{y}$, generated by an unknown function f, a supervised ML algorithm has the objective to learn an approximation $\hat{f}$, so that the values predicted by $\hat{f}$ for $\bar{x}$ have the least possible error with respect to $\bar{y}$.

2.2 Feature selection

In ML, feature selection is the process of choosing (or eliminating) features from a dataset, reducing them to the minimal, most informative subset. Removing information might, at first glance, seem detrimental for the performance of ML algorithms: however, certain features might just add noise; or they might be redundant, for example being heavily correlated with others; and finally, eliminating features reduces the search space that ML algorithms have to explore, facilitating the task of finding effective models.

Besides improving the performance of ML algorithms (not only in terms of computation time but also regarding precision of results [7]), feature selection can also be used to reduce information and ultimately make it human-readable. For example, while reviewing the contributions of 1,000 different variables in a problem is impossible for human experts, a selection of 10 highly-informative features can usually be analyzed, even if relevant parts of the information are removed. This is particularly useful when dealing with genomic or other high-dimensional data [8]. More generally, feature selection is one facet of dimensionality reduction, which is an important domain in the field of data visualization [9].

Feature selection can be performed using various approaches [10], simple ones consist in filtering the features according to a criterion (often based on statistical tests), or in using recursive procedures (forward or backwards) to eliminate redundant features [11][12]. Subset selection methods are more complex and rely on the definition of a quality measurement of the subset. The problem is thus turned into an optimization one: selecting the best subset of features that maximizes an objective function (usually a "goodness-of-fit" combined with a regularization term, including a penalty for a large number of variables [10,13]). Several single-objective EAs have been proposed, exploiting similar scores for the fitness function [1,2]. Finally, feature construction and space dimensionality is another way to reduce information. Subsets made of combinations of features are built for a better representation of the dataset (dimensionality reduction methods, principal component analysis for instance).

Given a candidate subset of features, evaluating its efficacy is not trivial. Ideally, what would need to be measured is the *content of information* of the

feature subset, and several metrics have been proposed to assess it in literature: for example *mutual information* [14] or *analysis of variance* [15]. In practice, however, even the most popular metrics can only assess part of the information content of a feature subset, as taking into account the contribution of non-linear combinations of features is too computationally expensive.

A different way to assess efficacy for a feature subset is using it as input of a ML algorithm, and evaluate the difference in performance compared with the same algorithm, using all features, or a different feature subset. To avoid issues with overfitting, a K-fold cross-validation can be used, obtaining an average of its performance (for example, classification accuracy) on the test folds. As the cross-validation procedure is stochastic, comparing two feature subsets on just their average performance on test folds is not enough, because the variance of the results is not taken into account. A more robust approach is to consider the K performance results on test folds of the two feature subsets as samples drawn from two probability distributions, and exploit a statistical test to assess the likelihood that the two sets of samples are drawn from different distributions. If the two sets of samples are separable below an arbitrary confidence threshold, for example $p < 0.05$, the feature subset with the best average performance can be considered better than the other. The main issue of this methodology is that it is sometimes impossible, with the available data, to separate the performance of different feature subsets.

2.3 Multi-objective evolutionary algorithms and feature selection

Multi-objective optimization algorithms aim at finding the best compromises between conflicting criteria, ultimately delivering a set of non-comparable, non-dominated solutions to the users. Evolutionary Algorithms (EAs) currently represent the state-of-the-art in the field, with the Multi-Objective EA (MOEA), Non-Sorting Genetic Algorithm II (NSGA-II) [16] being one of the most widely adopted for real-world applications.

Given their effectiveness, it is not surprising that MOEAs have been already applied to feature selection problems, where the conflicting objectives are usually: i. minimizing the number of features and ii. maximizing a quality metric for a feature subset. In [3] the authors apply NSGA-II for feature selection. In [4], differential evolution is used instead. MOEA approaches to feature selection have been recently applied to facial recognition [5] and medical imaging [6].

3 Proposed approach

We propose a novel approach to feature selection in ML, framing it as a multi-objective problem with three aims: i. minimizing the number of features; ii. minimizing error on a cross-validation; iii. maximizing mutual information content between each feature and the target. Feature selection can be seen as finding the best compromises between the number of features considered and the final result

for a ML algorithm. However, assessing the effectiveness of the selected features for the problem is far from trivial, and only indirect metrics are available.

It must be noted that analyzing all feature subsets for a given dataset is often impossible, as the total number of feature subsets of dimension d for a dataset with F features is:

$$\sum_{d=1}^{F} \binom{F}{d} = 2^F \tag{1}$$

3.1 Individual representation

Individuals represent feature subsets, and are internally stored as simple bit-strings of size equal to the number of features in the original dataset. A '1' in the i-th position of an individual means that the corresponding i-th feature is included in the subset; a '0' indicates that the i-th feature is not included in the subset.

3.2 Fitness functions

The first objective in the proposed approach is to minimize the number of features included in a subset:

$$O_1 = \sum_{i=1}^{F} I(i) \tag{2}$$

where I is an individual represented as a bit-string, $I(i)$ indicates the bit in i-th position, and F is the number of features in the problem, also corresponding to the size of an individual.

The second objective assesses the effectiveness of a candidate feature subset for a specific problem, through a K-fold cross-validation, a procedure where training data is divided into K parts, termed *folds*, that are alternatively used for training and test. This objective can be stated as:

$$O_2 = \frac{1}{K} \sum_{i=1}^{K} L_{k(i)} \tag{3}$$

where K is the number of folds; $k(i)$ is the i-th fold. $L_{k(i)}$ is defined as:

$$L_{k(i)} = L(y_{k(i)}, \hat{g}^{-k(i)}(x_{k(i)})) \tag{4}$$

where L is an error function, evaluating the differences between the values predicted by $\hat{g}^{-k(i)}$ and the known values $y_{k(i)}$; $\hat{g}^{-k(i)}$ is the function learned by a ML algorithm, trained on all data, except fold $k(i)$; in general, $\hat{g}$ is always considered to be an approximation of the real function g that generated the known values of y; $y_{k(i)}$ and $x_{k(i)}$ are the known values of the target and the

corresponding features for samples in $k(i)$, respectively. The error measured by L is averaged over the K folds to obtain the final value of O_2.

Finally, the third objective is a proxy for human readability of the candidate feature subset. Using the one-way Analysis of Variance (ANOVA) F-value procedure [17], that captures univariate relationships between a feature and the target. Indeed, the F-value of the i-th feature ϕ_i can be interpreted as the proportion of variance explained by the feature to the total variance in the data. If we make a reasonable assumption that an higher amount of explained variance may correspond to a higher discriminating capability, then we can rank features according to their ϕ, where the best feature will have the highest value. Finally, for each subset of features (i.e. the candidate solution), the third fitness objective is function of the worst ϕ in the feature subset:

$$O_3 = \frac{1}{\min(\phi_0, \phi_1, ..., \phi_f)} \tag{5}$$

where f is the number of features in the subset. This objective will force the evolutionary process to drop feature sets containing at least one variable whose univariate contribution is negligible. In fact, ML classifiers as well as other automatic FS algorithms (such as RFE) risk to retain a dramatic amount of features which are not a true causative source of the observed phenomenon (a.k.a. false positives). However, they are often selected as they might be slightly correlated with the target, providing a minor contribution to the classification accuracy.

Taking into account Eq. 2, 3, and 5, the multi-objective optimization problem can be described as:

$$argmin(O_1, O_2, O_3) \tag{6}$$

4 Experimental results

The experiments presented in this work deal with classification only, due to the greater availability of high-dimensional classification datasets in the public domain; but the proposed methodology can also be straightforwardly applied to regression problems. The experimental evaluation of the proposed approach is divided into two parts. Firstly, datasets that have a relatively low dimensionality (9-18 features) are analyzed: as all feature subsets for these datasets can be explored exhaustively, we can assess whether there is actually a single best feature subset, and whether different methodologies are able to find it. In a second batch of experiments, the proposed methodology tackles an artificial dataset with high dimensionality (500 features), that cannot be analyzed exhaustively, but whose characteristics are completely known.

4.1 Experimental setup

For the following experiments, the error function L (see Equation 4) is classification error, an established quality metric for classifiers, simply defined as the

ratio between incorrect predictions and total predictions. The closer classification error is to 0, the higher the quality of the predictive model. The classifier used to learn $\hat{f}$ in O_2 is Logistic Regression [18], a popular algorithm of proved effectiveness.

The MOEA selected for the experiments is NSGA-II [16], that currently represents the state of the art for multi-objective optimization with up to three objectives. After preliminary evaluations, NSGA-II's parameters are set to: $\mu = 100$, $\lambda = 100$, probability of crossover $p_c = 0.9$, probability of mutation $p_m = 1/l$, where l is the length of an individual, and a stop condition based on the maximum number of generations, set to 200.

The proposed approach, termed *EvoFS* in tables and figures, is compared against three popular state-of-the-art feature selection methods: recursive feature elimination (RFE) [19], that uses a classifier to score a feature set, then iteratively removes the lowest-performing feature and scores the subset again; greedy forward selection, that greedily adds features to a subset, using either their mutual information (MI) [14], or analysis of variance (ANOVA) [15] scores. All these methods need the user to specify the number of features to be selected, so in the experiments they have been called once for every possible size of feature subset in the problem, to have a fair comparison.

As previously stated, comparing the effectiveness of two feature subsets for classification using the error function L is not trivial, due to possible random effects in the classifier's training process, or in the way the training/test split of the data is performed. Randomly dividing the data in K folds and performing a K-fold cross-validation can help obtain a better average for L, but introduces further stochasticity in the process. When comparing results in this work, we consider the outcome of a K-fold cross-validation as K separate samples coming from an unknown statistical distribution. We then compare the results of two feature subsets as if assessing the likelihood that their accuracy scores have equal means. As we cannot assume that the two distributions have the same standard deviation, we use a Welch's T-test [20] with an arbitrary but commonly accepted threshold for the p-value ($p < 0.05$). Such a statistical test assumes that samples are drawn from populations that are normal in shape. As pointed out in [21], this assumption is quite easy to meet for a wide range of practical distributions at a significance level $\alpha = 0.05$ and a sample size of $K \geq 5$. In the following, $K = 10$. We will use this procedure to isolate clusters of feature subsets that are non-separable for their classification error, and can thus be considered all equally optimal with regards to this metric. For each considered dataset, running times for all algorithms are reported in Table 2.

All the code in the experiments has been implemented in Python v3, using the modules `scikit-learn` [22] for all ML and feature selection algorithms, `openml` [23] for accessing the datasets in the OpenML repository, and `inspyred` [24] for NSGA-II. The scripts are freely available in a Bitbucket repository[3]. Experiments have been run on a consumer-end laptop[4].

[3] `https://bitbucket.org/evomlteam/moea-feature-selection`

[4] Intel® Core™ i7-8750H 2.20 GHz, 8 GB RAM.

4.2 Simple datasets

In a first set of experiments, simple datasets with a limited number of features are examined. The advantage of dealing with such datasets is that all their feature subsets can be enumerated and analyzed, a task that becomes impossible if dealing with hundreds or thousands of features. The datasets are freely accessible on the OpenML repository [25], and their characteristics are summarized in Table 1.

Table 1. Characteristics of the datasets used in the experiments.

Dataset name	**Type**	**Features**	**Samples**	**Classes**	**Feature subsets**
diabetes [26]	Medical	9	768	2	512
australian [27]	Credit scores	14	690	2	16,384
vehicle [28]	Vehicle recognition	18	846	4	262,144
Madelon [29]	Artificial	500	4,400	2	10^{150}

In Figures 1, 2, and 3, we show how many non-separable feature subset that have size lower or equal to the best performing one each algorithm was able to find: ideally, these are the ones that human users should be interested in. Then, for each algorithm, the position of each non-separable solution found is mapped into the exhaustive exploration of all feature subsets.

Figure 1 reports the results for the *diabetes* dataset. While all approaches are able to find feature subsets that are non-separable from the best ones, EvoFS finds the largest number. The same holds for the *australian* dataset, in Figure 2, where notably RFE seems unable to find good solutions of small size. For *vehicle*, that features the largest search space so far, results reported in Figure 3 show that, this time, RFE performs much better than the other two comparing methods, equalling the performance of EvoFS. Nevertheless, EvoFS is able to find a few non-separable solutions that are of smaller size than those uncovered by RFE.

An interesting general behavior that emerges from the plots, is that EvoFS is able to find non-separable feature subsets of lower size than the other algorithms. Notably, non-separable solutions of size larger than the best performing one are not included in its Pareto fronts.

4.3 High-dimensional datasets

The second set of experiments deals with a high-dimensional dataset, for which an exhaustive analysis of all feature sets is impossible. This datasets is artificial, taken from a classification competition focused on feature selection [29]. The datasets' characteristics are summarized in Table 1.

The targed dataset, named *Madelon*, is an artificial dataset that can be procedurally generated, with a few informative features, several features that are linear combinations of the informative features, and a large number of deceiving

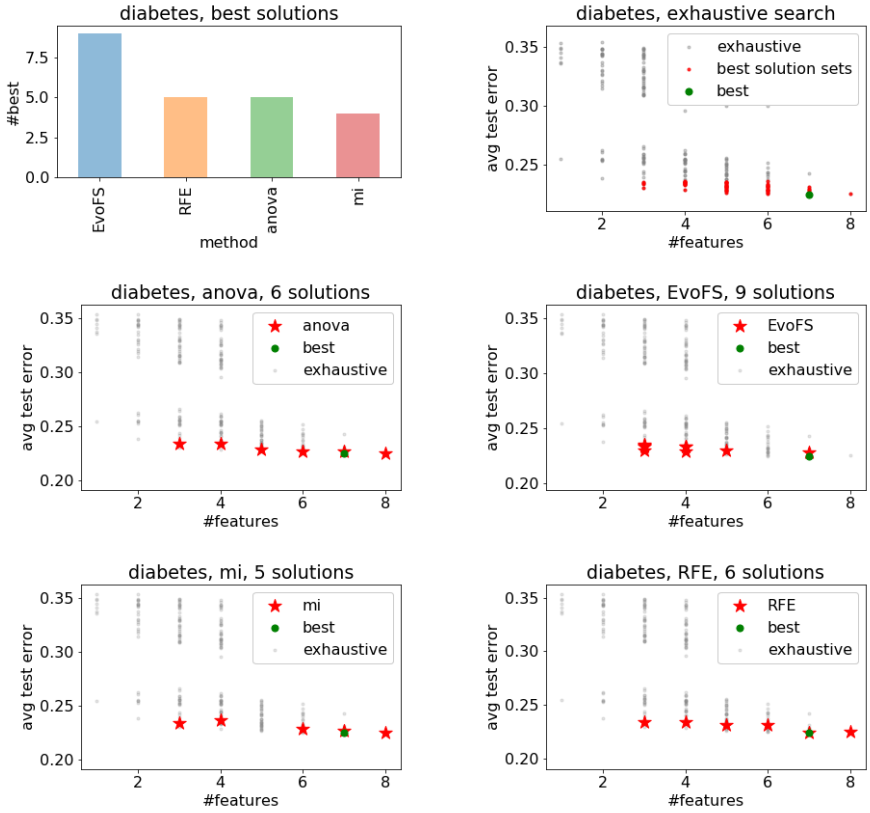

Fig. 1. (top left) Number of non-separable optimal feature subsets, of size less or equal to the one with the lowest error, found by each algorithm. **(top right)** All possible feature subsets for the dataset, identified exhaustively. In red, for each size, the ones that are non-separable. In green, the single feature subset with the lowest average error. **(middle-left to bottom-right)** Features subsets uncovered by the different approaches.

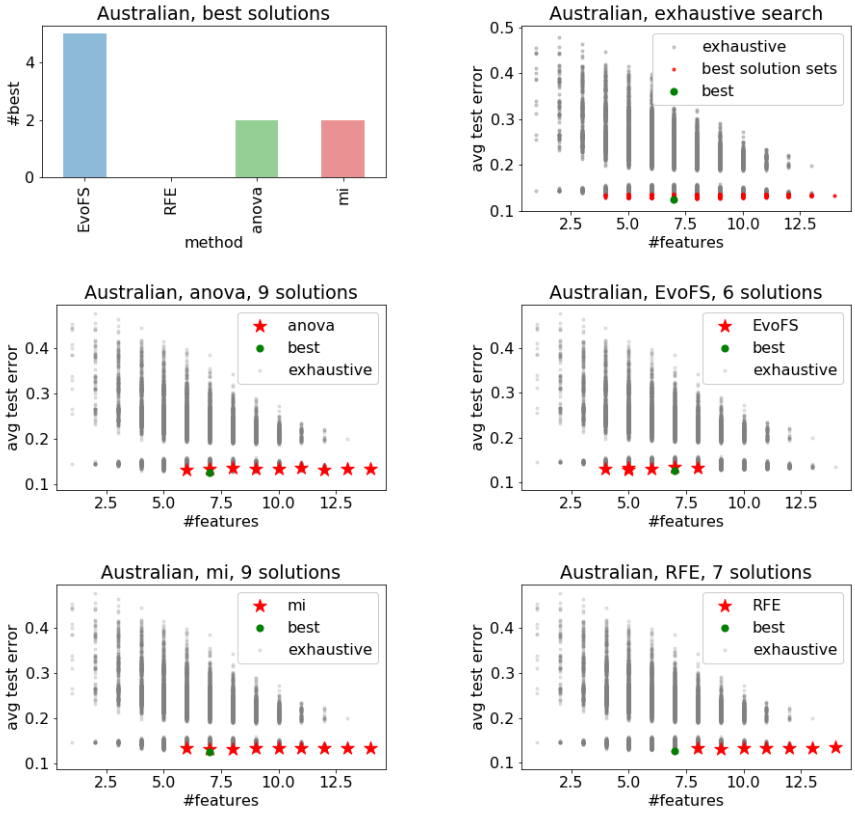

Fig. 2. (top left) Number of non-separable optimal feature subsets, of size less or equal to the one with the lowest error, found by each algorithm. **(top right)** All possible feature subsets for the dataset, identified exhaustively. In red, for each size, the ones that are non-separable. In green, the single feature subset with the lowest average error. **(middle-left to bottom-right)** Features subsets uncovered by the different approaches.

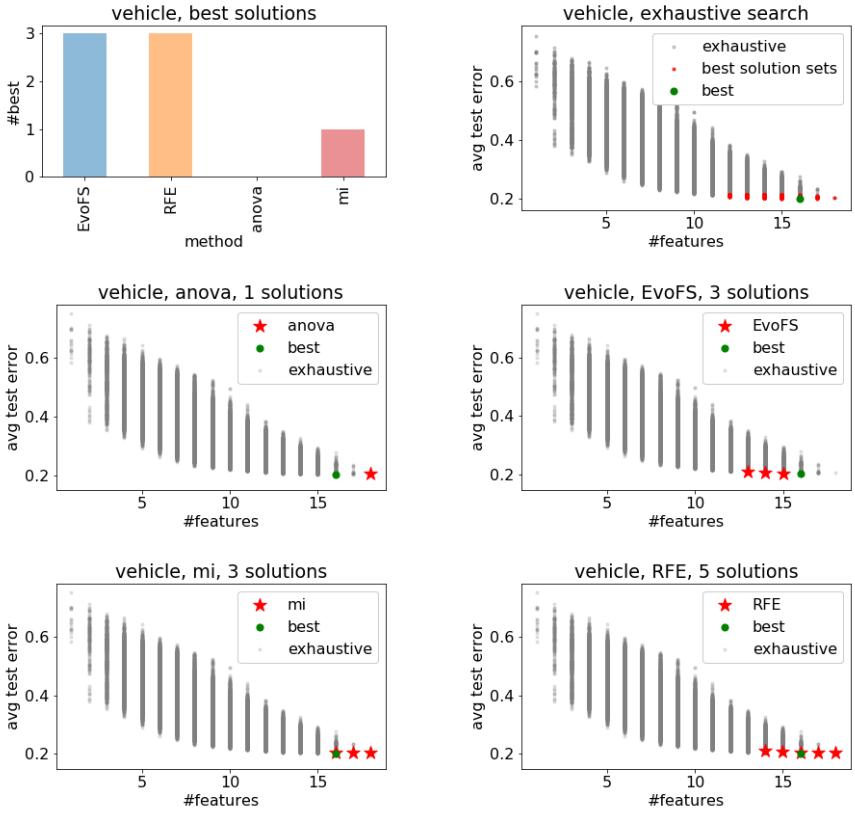

Fig. 3. (top left) Number of non-separable optimal feature subsets, of size less or equal to the one with the lowest error, found by each algorithm. **(top right)** All possible feature subsets for the dataset, identified exhaustively. In red, for each size, the ones that are non-separable. In green, the single feature subset with the lowest average error. **(middle-left to bottom-right)** Features subsets uncovered by the different approaches.

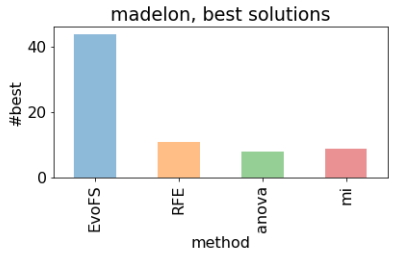

Fig. 4. Number of non-separable optimal feature subsets, of size less or equal to the one with the lowest error, found by each algorithm, on the *madelon* dataset.

features called *probes* [30]. For this work, we generated an instance of Madelon with the same parameters as the one featured in the competition [29]: 5 informative features, 15 linear combination features, 480 deceiving features/probes.

Figure 4 illustrates a summary of the results on the Madelon dataset. Remarkably, EvoFS is able to find a higher number of non-separable feature subsets having size lower or equal to the overall best solution. Moreover, EvoFS is also the only algorithm able to identify a non-separable solution of size 3, that includes only informative features (in positions 2, 3, 18).

While the greedy algorithms continue to be extremely fast on the high-dimensional dataset, it is noticeable from the running times reported in Table 2, how now RFE, with its iterative process, scales much worse than EvoFS.

Table 2. Running time (seconds) of the feature selection algorithms.

Dataset	**EvoFS**	**anova**	**MI**	**RFE**
diabetes	421.28 s	0.02 s	0.05 s	0.10 s
australian	579.38 s	0.03 s	1.47 s	0.24 s
vehicle	819.43 s	0.03 s	2.17 s	1.88 s
madelon	3,549.57 s	0.05 s	12.97 s	18,925.29 s

5 Conclusions

Feature selection is an important task in ML, to obtain feature subsets of limited size that provide excellent performance. However, measuring performance is not trivial: commonly used metrics, such as the average error on a K-fold cross-validation, have been shown to mislead when comparing feature subsets that are, in fact, statistically non-separable. Using statistical tests, we uncover clusters of non-separable feature subsets in simple datasets, that can be exhaustively analyzed. Armed with this knowledge, we can then re-evaluate the performance of a feature selection methodology by estimating the number of optimal, non-separable feature subsets that the algorithm is able to discover.

The multi-objective feature selection algorithm we propose is shown able to find large numbers of feature subsets in such optimal clusters, when compared to other state-of-the-art algorithms in literature.

Future works will focus on further statistical comparisons with other evolutionary approaches to feature selection, and eventually introducing a human-interactive factor in the algorithm, in order to further promote human understanding of the results.

References

1. Cilia, N.D., De Stefano, C., Fontanella, F., di Freca, A.S.: Variable-length representation for ec-based feature selection in high-dimensional data. In: International

Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 325–340. Springer (2019)
2. Xue, B., Zhang, M., Browne, W.N., Yao, X.: A survey on evolutionary computation approaches to feature selection. IEEE Transactions on Evolutionary Computation **20**(4), 606–626 (2015)
3. Hamdani, T.M., Won, J.M., Alimi, A.M., Karray, F.: Multi-objective feature selection with nsga ii. In: International conference on adaptive and natural computing algorithms. pp. 240–247. Springer (2007)
4. Xue, B., Fu, W., Zhang, M.: Multi-objective feature selection in classification: a differential evolution approach. In: Asia-Pacific Conference on Simulated Evolution and Learning. pp. 516–528. Springer (2014)
5. Vignolo, L.D., Milone, D.H., Scharcanski, J.: Feature selection for face recognition based on multi-objective evolutionary wrappers. Expert Systems with Applications **40**(13), 5077–5084 (2013)
6. Zhou, Z., Li, S., Qin, G., Folkert, M., Jiang, S., Wang, J.: Multi-objective based radiomic feature selection for lesion malignancy classification. IEEE journal of biomedical and health informatics (2019)
7. Fan, Y.J., Kamath, C.: On the selection of dimension reduction techniques for scientific applications (01 2012). https://doi.org/10.2172/1036865, part of the Annals of Information Systems book series (AOIS, volume 17)
8. Bermingham, M., Pong-Wong, R., Spiliopoulou, A., Hayward, C., Rudan, I., Campbell, H., F. Wright, A., F. Wilson, J., Agakov, F., Navarro, P., Haley, C.: Application of high-dimensional feature selection: Evaluation for genomic prediction in man. Scientific Reports **5**, 10312 (05 2015). https://doi.org/10.1038/srep10312
9. Tsai, F.S.: Dimensionality reduction for computer facial animation. Expert Systems with Applications **39**(5), 4965 – 4971 (2012). https://doi.org/10.1016/j.eswa.2011.10.018
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (Mar 2003)
11. Lewis, P.: The characteristic selection problem in recognition systems. IRE Transactions on information theory **8**(2), 171–178 (1962)
12. Chien, Y., Fu, K.S.: On the generalized karhunen-loève expansion (corresp.). IEEE Transactions on Information Theory **13**(3), 518–520 (1967)
13. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature selection for svms. In: Advances in Neural Information Processing Systems 13. pp. 668–674. MIT Press (2000)
14. Kozachenko, L., Leonenko, N.N.: Sample estimate of the entropy of a random vector. Problemy Peredachi Informatsii **23**(2), 9–16 (1987)
15. Fisher, R.A.: Xv.the correlation between relatives on the supposition of mendelian inheritance. Earth and Environmental Science Transactions of the Royal Society of Edinburgh **52**(2), 399–433 (1919)
16. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
17. Heiman, G.W.: Understanding research methods and statistics: An integrated introduction for psychology. Houghton, Mifflin and Company (2001)
18. Cox, D.R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological) **20**(2), 215–232 (1958)
19. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine learning **46**(1-3), 389–422 (2002)

20. Welch, B.L.: The generalization of Student's' problem when several different population variances are involved. Biometrika **34**(1/2), 28–35 (1947)
21. Krzywinski, M., Altman, N.: Points of significance: Comparing samples-part I. Nature methods **11**(3), 215 (2014)
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)
23. Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., Bischl, B.: OpenML: An R package to connect to the machine learning platform openml. Computational Statistics **32**(3), 1–15 (2017). https://doi.org/10.1007/s00180-017-0742-2
24. Garrett, A.: inspyred (version 1.0.1) inspired intelligence. `https://github.com/aarongarrett/inspyred` (2012)
25. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: Openml: Networked science in machine learning. SIGKDD Explorations **15**(2), 49–60 (2013). https://doi.org/10.1145/2641190.2641198
26. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
27. Quinlan, J.R.: Simplifying decision trees. International journal of man-machine studies **27**(3), 221–234 (1987)
28. Siebert, J.P.: Vehicle recognition using rule based methods (1987)
29. Guyon, I., Gunn, S., Ben-Hur, A., Dror, G.: Result analysis of the nips 2003 feature selection challenge. In: Advances in neural information processing systems. pp. 545–552 (2005)
30. Guyon, I.: Design of experiments of the nips 2003 variable selection benchmark. In: NIPS 2003 workshop on feature extraction and feature selection (2003)