Predictable Features Elimination: An Unsupervised Approach to Feature Selection

Predictable Features Elimination: An Unsupervised Approach to Feature Selection

Pietro Barbiero[1][0000−0003−3155−2564], Giovanni Squillero[2][0000−0001−5784−6435], and Alberto Tonda[3][0000−0001−5895−4809]

[1] Cambridge University, Cambridge, UK
pb737@cam.ac.uk
[2] Politecnico di Torino, Torino, IT
squillero@polito.it
[3] UMR 518 MIA, INRAE, Paris, FR
alberto.tonda@inrae.fr

Abstract. We propose an unsupervised, model-agnostic, wrapper method for feature selection. We assume that if a feature can be predicted using the others, it adds little information to the problem, and therefore could be removed without impairing the performance of whatever model will be eventually built. The proposed method iteratively identifies and removes predictable, or nearly-predictable, redundant features, allowing to trade-off complexity with expected quality. The approach do not rely on target labels nor values, and the model used to identify predictable features is not related to the final use of the feature set. Therefore, it can be used for supervised, unsupervised, or semi-supervised problems, or even as a safe, pre-processing step to improve the quality of the results of other feature selection techniques. Experimental results against state-of-the-art feature-selection algorithms show satisfying performance on several non-trivial benchmarks.

1 Introduction

The aim of most Machine Learning (ML) algorithms is to build a predictive model starting from the feature values of a given training set. State-of-the-art algorithms are usually quite effective at tacking problems with huge number of samples, yet they might face issues if the number of features is huge. An increase in the dimensionality of the problem, in fact, may correspond to a much steeper increase of the search space, impairing the optimization of the models, or creating other problems usually described with the vague expression "curse of dimensionality" [2].

A reduction in the number of variables may be obtained either by means of feature extraction or feature selection. Techniques in the former group, such as principal component analysis (PCA) or autoencoders, built a new, more compact set of features out of the original one. Feature selection techniques, on the other hand, aim at finding a subset of the original features, that still allows ML algorithms to build reliable predictive models. There might be practical reasons

to opt for the second class of techniques, for instance, if each single feature has a cost because it needs to be physically measured. Moreover, predictive models that use tens or hundreds of features are de facto black boxes, quite hard if not completely impossible to interpret. Identifying the key features involved in a problem can make the final model more human-readable: problems involving genomic data [3] would greatly benefit from the possibility to find understandable correlations; improvement could be noticeable in the field of visualization [23]; and the general need for explainable artificial intelligence (XAI) is only but increasing in present days.

Starting from the observation that the value of a redundant feature, by definition of redundancy, can be inferred using the information contained in the other features, we propose Predictable Features Elimination (PFE). PFE is an unsupervised, model-agnostic, wrapper approach for feature selection: In a first step, each feature is scored and ranked using a statistical measure; then, starting from the lowest-ranked feature, an auxiliary ML model is trained to predict that feature using all the others; if the performance of the model exceeds a given quality (for example, $R2 > 0.95$), the information it provides is assumed to be redundant and the feature is removed. The procedure then iterates to the next feature, and once all features have been analyzed in this way, those remaining represent the final subset.

Experimental results on several non-trivial benchmarks from the OpenML repository [26] show that the proposed approach is competitive with the state of the art in the field, obtaining feature subsets that are either more informative or smaller than competing feature selection algorithms. The main contributions of this paper are:

- We describe Predictable Features Elimination, a new unsupervised, model-agnostic, wrapper approach for feature selection.
- We compare the performance of PFE against state-of-the-art feature selection algorithms, showing the advantages of using our method especially on large datasets.
- We compare PFE against other algorithms on an artificial dataset, specifically designed for assessing feature selection algorithms. Results show that the features selected by PFE include almost all meaningful information.

2 Feature Selection in Machine Learning

Feature Selection (FS) is the process of identifying the features of a data set in order to obtain a minimal, informative subset. Features may not be part of this subset for two main reasons: they might be unrelated to the underlying nature of the problem, just adding noise; they might be heavily correlated with others features, adding no relevant information for the task.

Applications range from face recognition [28] to medicine [35], while approaches can be divided into two categories [11]: filters that score features according to a criterion (often a statistical test); and recursive procedures (forward or

backwards) that attempt to reduce the features to a small set of non-redundant ones [16, 5].

Several different metrics that have been proposed to assess the content of information of a given feature subset: from mutual information [15] to analysis of variance [9]. However, only some of the information content of a feature subset can be assessed through such metrics, as taking into account the contribution of non-linear combinations of features would be too computationally expensive. Recursive procedures on the other hand frequently rely on more complex measurements, usually a goodness-of-fit [11] for a model wrapped inside the feature selection, sometimes combined with regularization terms [30]. In some cases, the number of features and the quality of the fit are evaluated separately, and each candidate subset is placed on a Pareto front [32].

Literature reports FS techniques as complex as exploiting evolutionary algorithms (EAs) [6, 32], with single- or multi-objective approaches [13, 31, 1]. Anyhow, the most popular approach in literature is still probably the 20-year old Recursive Feature Elimination (RFE) [12], a supervised, wrapper methodology that iteratively removes the worst features based on the performance of the target model.

3 Predictable feature elimination

The algorithm

Predictable Features Elimination, the approach we present in this work, stems from the observation that if the distribution of a feature f_r can be approximated by using the information of other features, then f_r is likely to be almost redundant and of little importance for whatever model will be eventually built. Algorithm 1 summarizes the main steps of the training process.

PFE requires the user to provide two parameters: an auxiliary machine learning model g and a threshold σ. The first is the model used to discriminate non-redundant features, the second, the acceptable loss of information. A suboptimal choice of g would cause some features to be erroneously marked as non-redundant, increasing the size of the feature set, but probably not affecting the quality of the final model. On the other hand, a low σ is likely to make PFE select a very small set of features, but also to impair the quality of the final model.

The algorithms is composed of two phases: an initialization and the main loop. In the former, features are ranked according to their mutual average linear correlation. First, the feature correlation matrix C is computed:

$$C = \Big(diag(K_{XX})\Big)^{1/2} K_{XX} \Big(diag(K_{XX})\Big)^{1/2} \tag{1}$$

where K_{XX} is the auto-covariance matrix of the input matrix X:

$$K_{XX} = E[(X - E[X])(X - E[X])^T)] \tag{2}$$

The correlation matrix C is used to estimate the amount of mutual information shared among features. By summing up the rows of C, we obtain for each feature an approximation of the amount of information which can be obtained using all the other features:

$$\kappa = \sum_i C(i, \cdot) \tag{3}$$

The more a feature is correlated with others, the lower the chances that the feature may contain exclusively useful information. Hence, by ranking features according to their mutual average linear correlation, we will obtain an ordered list of their significance:

$$\kappa_s = sort(\kappa) \tag{4}$$

Starting from the feature with the highest rank f, the ML model g is trained on the remaining features using the f-th feature as a target variable y:

$$y = X(\cdot, \kappa_s(f)) \tag{5}$$

The performance of g is assessed on a validation set X_{val} using the coefficient of determination $R2$. If the validation score is greater than the user defined threshold σ, then it means that the model g represents an accurate nonlinear association between the f-th feature and the other features. Hence, the chances that the f-th feature may contain exclusively useful information are low. Therefore, the f-th feature should be safely removed from the feature set and the process may continue using the following feature in the ranking. The algorithm stops when more than half of the features have been analyzed.

Algorithm 1: Predictable feature elimination

Input: data $X \in \mathbb{R}^{n,d}$, model g, threshold $\sigma \in [0, 1]$
Initialize $C = corr(X^T)$
Initialize $\kappa = \sum_i C(i, \cdot)$
Initialize $\kappa_s = sort(\kappa)$
for $f = 1$ **to** $\lfloor d/2 \rfloor$ **do**
  Initialize $y = X(\cdot, \kappa_s(f))$.
  Split data into train and validation sets
  Train model $g \leftarrow (X_{train}, y_{train})$
  Make validation predictions $\hat{y} = g(X_{val})$
  Evaluate predictions $score = R2(\hat{y}, y_{val})$
  if $score \geq \sigma$ **then**
    Remove current feature
  end if
end for

Theoretical foundation

The following theorems yield a theoretical justification for the proposed approach. Besides, they show how the performance loss can be formally estimated by providing upper bounds in worst case scenarios.

Theorem 1 (Elimination for linear combinations). *Let $\mathcal{F}$ be a set of features $\mathcal{F} = \{f_1, \ldots, f_d\}$ and $\mathcal{F}'$ be a subset of $\mathcal{F}$ such that $f_i \notin \mathcal{F}'$. If the feature f_i is a linear combination of the feature set $\mathcal{F}'$, then fitting a linear model using $\mathcal{F}'$ is equivalent to fitting the same model using $\mathcal{F}' \cup \{f_i\}$.*

Proof. By definition f_i is a linear combination of $\mathcal{F}'$, hence:

$$f_i = \sum_{j \neq i} w_j f_j \tag{6}$$

which can be written as:

$$f_i = F'w \tag{7}$$

where $F' \in \mathbb{R}^{n \times d'}$ is a matrix whose columns are features in $\mathcal{F}'$ and $w \in \mathbb{R}^{d'}$ is a row vector containing the weights of the linear combination.

A linear model g trained using the matrix F' can be written as:

$$\widehat{y} = g(F') = F'w^g = \sum_{j \in d'} f_j w_j^g \tag{8}$$

Let F'' be the matrix whose columns correspond to the features in $\mathcal{F}' \cup \{f_i\}$, then the model g trained on F'' can be written as:

$$\begin{aligned}
\widehat{y} = g(F'') = F''w^g &= \sum_{k \in d''} f_k w_k^g \\
&= \sum_{j \in d'} f_j w_j^g + w_i^g f_i \\
&= \sum_{j \in d'} f_j w_j^g + w_i^g \sum_{j \in d'} f_j w_j \\
&= \sum_{j \in d'} f_j w_j^g + \sum_{j \in d'} f_j w_j w_i^g \\
&= \sum_{j \in d'} f_j w_j^g w_j w_i^g \\
&= \sum_{j \in d'} f_j \omega_j^g
\end{aligned} \tag{9}$$

Theorem 2 (Approximate elimination for linear combinations). *Let $\mathcal{F}$ be a set of features $\mathcal{F} = \{f_1, \ldots, f_d\}$ and $\mathcal{F}'$ be a subset of $\mathcal{F}$ such that $f_i \notin \mathcal{F}'$. If the feature f_i can be written as a linear combination of the feature set $\mathcal{F}'$ with an additional term ϵ, then the upper bound of the training error obtained by fitting a linear model on $\mathcal{F}'$ instead of $\mathcal{F}' \cup \{f_i\}$ is at most ϵ.*

Proof. By definition f_i can be written as a linear combination of the feature set $\mathcal{F}'$ with an additional term ϵ, hence:

$$f_i = \sum_{j \neq i} w_j f_j + \epsilon \tag{10}$$

which can be written as:

$$f_i = F' w + \epsilon \tag{11}$$

where $F' \in \mathbb{R}^{n \times d'}$ is a matrix whose columns are features in $\mathcal{F}'$ and $w \in \mathbb{R}^{d'}$ is a row vector containing the weights of the linear combination.

A linear model g trained using the matrix F' can be written as:

$$\widehat{y} = g(F') = F' w^g = \sum_{j \in d'} f_j w_j^g \tag{12}$$

Let F'' be the matrix whose columns correspond to the features in $\mathcal{F}' \cup \{f_i\}$, then the model g trained on F'' can be written as:

$$\begin{aligned}
\widehat{y} = g(F'') = F'' w^g &= \sum_{k \in d''} f_k w_k^g \\
&= \sum_{j \in d'} f_j w_j^g + w_i^g f_i \\
&= \sum_{j \in d'} f_j w_j^g + w_i^g \sum_{j \in d'} f_j w_j + \epsilon \\
&= \sum_{j \in d'} f_j w_j^g + \sum_{j \in d'} f_j w_j w_i^g + \epsilon \\
&= \sum_{j \in d'} f_j w_j^g w_j w_i^g + \epsilon \\
&= \sum_{j \in d'} f_j \omega_j^g + \epsilon
\end{aligned} \tag{13}$$

Theorem 3 (Approximate elimination). *Let $\mathcal{F}$ be a set of features $\mathcal{F} = \{f_1, \ldots, f_d\}$ and $\mathcal{F}'$ be a subset of $\mathcal{F}$ such that $f_i \notin \mathcal{F}'$. Let g and h be two nonlinear models with equivalent capacity. If the feature f_i can be written as a function of $\mathcal{F}'$ through h with an error term ϵ, then the upper bound of the training error obtained by fitting g on $\mathcal{F}'$ instead of $\mathcal{F}' \cup \{f_i\}$ is at most $\eta(g, \epsilon)$.*

Proof. By definition f_i can be written as a nonlinear function h of the feature set $\mathcal{F}'$ with an additional term ϵ, hence:

$$f_i = h(F') + \epsilon \tag{14}$$

where $F' \in \mathbb{R}^{n \times d'}$ is a matrix whose columns are features in $\mathcal{F}'$.

A nonlinear model g trained using the matrix F' can be written as:

$$\widehat{y} = g(F') \tag{15}$$

Let F'' be the matrix whose columns correspond to the features in $\mathcal{F}' \cup \{f_i\}$, then the model g trained on F'' can be written as:

$$\widehat{y} = g(F'') = g(F', f_i) = g(F', h(F') + \epsilon) \tag{16}$$

Since the information in $h(F')$ can be obtained from F' by applying the function h, then g can be fitted on F' without information loss by discarding $h(F')$:

$$\widehat{y} = g(F'') = g(F', \epsilon) = g(F') + \eta(g, \epsilon) \tag{17}$$

where η is a function of g and ϵ.

The following definition can be used to monitor in an unsupervised way the performance loss when multiple features are recursively eliminated. In some applications, this lemma may be used to derive alternative stopping conditions.

Definition 1 (Validity of feature elimination). *Let λ be an upper bound of the performance loss required for a specific application and let $\{\eta_1, \ldots, \eta_k\}$ be a sequence of training errors obtained by performing k steps of feature elimination. The sequence of k feature elimination steps is valid if and only if $\lambda \leq \sum_{i=1}^{k} \eta_i$.*

4 Experimental Evaluation

The proposed approach has been implemented from scratch in Python 3, using only open-source libraries [19, 17]; the source code, including all the parameter values used in the experiments, is available under the European Union Public Licence (EUPL) from GitHub[4].

All experiments have been run on the same machine equipped with an AMD EPYC 7301 16-core processor running at 2 GHz, and with 64 GiB memory.

Experimental setup

The performance of predictable feature elimination is compared over a 10-fold cross-validation against both supervised and unsupervised feature selection algorithms [19, 17]: Laplacian score for feature selection (`lap_score` [14]), spectral feature selection for unsupervised clustering (`SPEC` [34]), multi-cluster feature selection (`MCFS` [4]), non-negative spectral feature selection (`NDFS` [18]), regularised discriminative feature selection (`UDFS` [33]), and recursive feature elimination (`RFE` [12]).

A `Ridge` classifier has been used both as the internal estimator for RFE, and to discard redundant features in PFE. It must be noted that PFE is agnostic to the choice of the estimator g, as far it is not significantly superior to the one eventually used in the final model — the underlying assumption being that if a feature can be predicted by g, it may as well be inferred by the final model.

[4] https://github.com/glubbdubdrib/predictable-feature-elimination

`Ridge` has been chosen both for its high train speed and its generalization ability in a variety of experimental settings.

For all the experiments the σ parameter has been set to the default 0.9, a reasonable value that leads to the removal of at most half of the original features. For the sake of comparison, all the other feature selection algorithms are set in order to provide for each fold the same number of features chosen by PFE. For each fold, each algorithm is used to select a subset of the original features.

In order to generate reproducible results, all algorithms that exploit pseudo-random elements in their training process have been set with a fixed seed. Unless differently specified, each algorithm uses its default parameters as defined in [19, 17]. Each dataset has been standardized removing the mean and scaling to unit variance (`StandardScaler` [36]).

Redundancy detection

The MADELON dataset proposed in [10] was specifically designed to challenge feature selection algorithms in detecting redundant features. Features generated by MADELON can be informative (d_i), repeated (d_r), or redundant (d_c). The algorithm generates clusters of points normally distributed about vertices of an hypercube in a subspace of dimension d_i and assigns an equal number of clusters to each class. Then it stacks d_c linear combinations of the informative features followed by d_r duplicates, drawn randomly with replacement from the informative and redundant features. All the remaining features (d_n) are random noise ($d_n = d - d_i - d_c - d_r$). This benchmark dataset is used to assess the ability of PFE in detecting redundant features. For this experiment the number of informative features is set to 150 as well as the number of redundant and repeated features ($d_i = d_c = d_s = 150$). The total number of features is set to $d = 500$, thus $d_n = 50$ features are just random noise. The task is to detect informative, redundant, duplicate, and noisy features in order to correctly classify clusters of samples on hypercube vertices. Experimental results are shown in Figure 1. Once feature selection algorithms are fitted on a training fold, an instance of `RidgeClassifier` and of `DecisionTreeClassifier` are used to assess the quality of the selection on the test set. The resulting $F1$ score [25] is then compared to the one obtained by training on the same fold but using all the original features. In this way, the performance of all techniques can be evaluated with respect to a fair baseline (see Figure 1, top). Except for `lap_score`, PFE resulted as the fastest approach. The most interesting result of the simulation is represented by boxplots in Figure 1, bottom. They show for each kind of feature (informative, redundant, duplicate, and noisy) the percentage of features retained by each algorithm. Notably, PFE and `MCFS` preserve most of the informative features while discarding most of the redundancy. However, `MCFS` is the worst technique in terms of duplicate detection, whereas PFE is the best one together with `NDFS` and `lap_score`. It should be remarked, anyway, that PFE retains all noisy features. Yet, it is not surprising at all as the approach is not designed to get rid of random noise. In authors view, PFE is not meant to be used alone but combined with other complementary feature selection approaches.

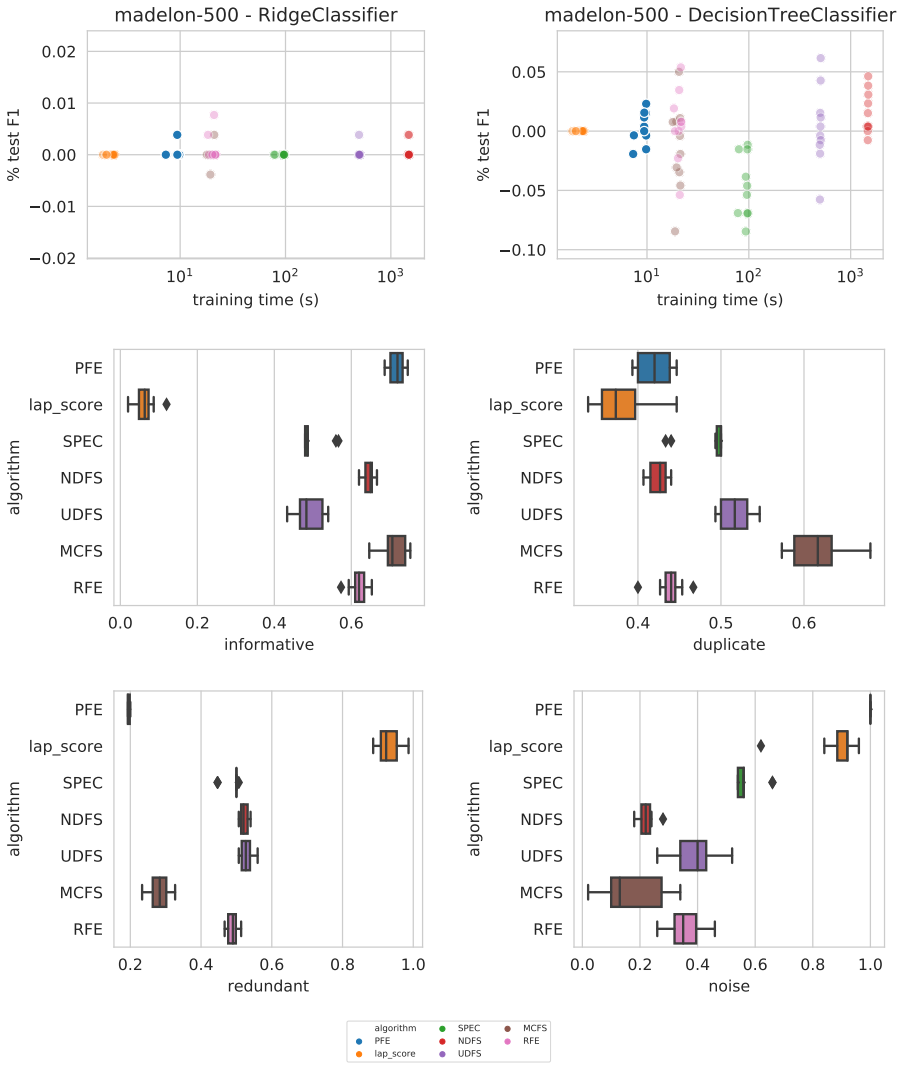

Fig. 1. Classification performances on the MADELON dataset (top two); feature selected for the MADELON dataset (bottom four).

Cross-task benchmarks

The ability of feature selection algorithms in tackling different kind of machine learning problems is assessed using four benchmark datasets taken from the OpenML[26]. Table 1 highlights the main characteristics of the four datasets. The first two (gas-drift and isolet) are used to test classification and clustering performances while the latters are employed for regression. The quality of the

Table 1. Benchmark datasets.

dataset	samples	features	classes	reference
gas-drift	13,910	128	6	[27]
isolet	7,797	617	26	[8]
Mercedes	4,209	377	-	[7]
crime	1,994	127	-	[24]

selections is assessed over a 10-fold cross-validation. Once fitted on a training fold, each algorithm provides a selection of the original features to an instance of a machine learning model on the filtered training set only. `RidgeClassifier` and `DecisionTreeClassifier` are used to evaluate classification performance in terms of $F1$ score. `AgglomerativeClustering` [29] and `KMeans` [22] are employed to assess clustering performances through the Silhouette coefficient [20]. The number of centroids for k-means is chosen as twice as much as the number of classes. For regression `Ridge` and `DecisionTreeRegressor` are used to measure the coefficient of determination ($R2$ score, [21]). Once collected, performance scores are compared to the ones obtained by training on the same folds but using all the original features. In this way, the performance of all techniques can be evaluated with respect to a fair baseline. Figures 2, 3, and 4 show the results in terms of performance metrics and training time. As mentioned before, all the other feature selection algorithms are set in order to provide for each fold the same number of features chosen by PFE, thus yielding a fair comparison.

Compared to state-of-the-art techniques, PFE is among the fastest solutions together with `RFE` and `lap_score`. Notably, `RFE` is not used for clustering as it is a supervised algorithm, thus it cannot be employed for unsupervised tasks. Despite its unsupervised nature, PFE often matches `RFE` performances and sometimes provides even better solutions (i.e., Mercedes dataset using `DecisionTreeRegressor`). The efficiency of PFE with respect to other unsupervised approaches is revealed on the largest dataset (gas-drift) where it is faster by a few order of magnitudes. Moreover, even when PFE performances appear to be slightly worse than others (i.e., Mercedes using `Ridge`), it may be sufficient to change the downstream predictor (i.e., the performance looks much better when `DecisionTreeRegressor` is used). Indeed, by construction, PFE performs feature selection such that the information loss is almost negligible.

5 Conclusions

In this paper, a novel feature selection approach named Predictable Features Elimination has been introduced. At the heart of the methodology lies the idea that features whose value can be easily predicted based on the values of other features of the same sample, probably contain mostly redundant information. The algorithm iteratively trains a machine learning model using one of the features as a target, and if the quality of the model is above a user-defined threshold

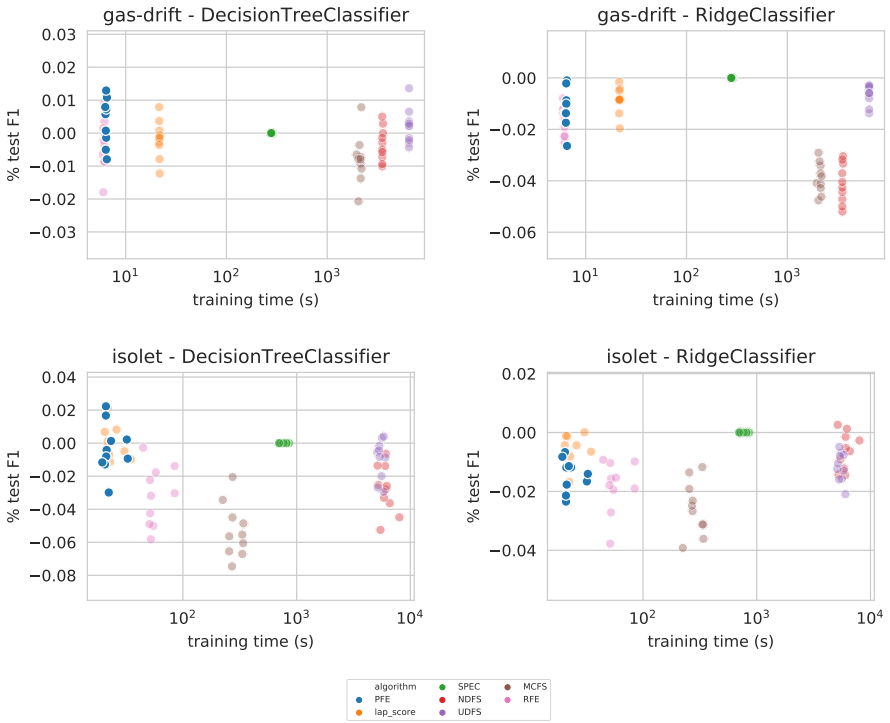

Fig. 2. Classification results on benchmark datasets.

(e.g., $\overline{V} > 0.9$), the feature is removed. After all features have been treated in this way, the remaining ones will constitute the final feature set. Not relying on target labels or values, PFE can be used for supervised, unsupervised, or semi-supervised machine learning problems.

Experimental results prove PFE to be competitive with state-of-the-art feature selection algorithms, on a set of non-trivial classification, regression, and clustering benchmarks. The main drawback of the approach is the impossibility of removing uninformative, but hard-to-predict features, for example those including completely random values: In most cases, however, such features are filtered out by subsequent machine learning algorithms applied to the data, as they have no correlation with the objectives.

Future works will investigate the performance on PFE on a wider range of benchmarks, and explore the possibility of using a similar idea on samples, to uncover coresets and potentially perform dataset compression. Finally, particular focus

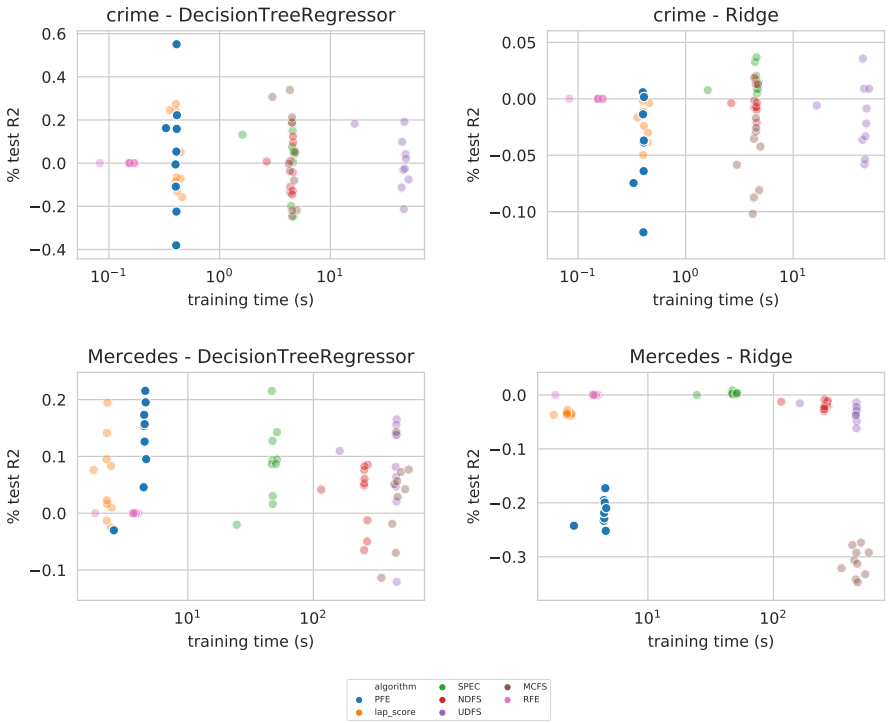

Fig. 3. Regression results on benchmark datasets.

References

1. Barbiero, P., Lutton, E., Squillero, G., Tonda, A.: A novel outlook on feature selection as a multi-objective problem. In: Proceedings of the 14th Biennial International Conference on Artificial Evolution. pp. 1–10. Springer (2019)
2. Barbiero, P., Squillero, G., Tonda, A.: Modeling generalization in machine learning: A methodological and computational study. arXiv preprint arXiv:2006.15680 (2020)
3. 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
4. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 333–342 (2010)
5. Chien, Y., Fu, K.S.: On the generalized karhunen-loève expansion (corresp.). IEEE Transactions on Information Theory **13**(3), 518–520 (1967)
6. 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)

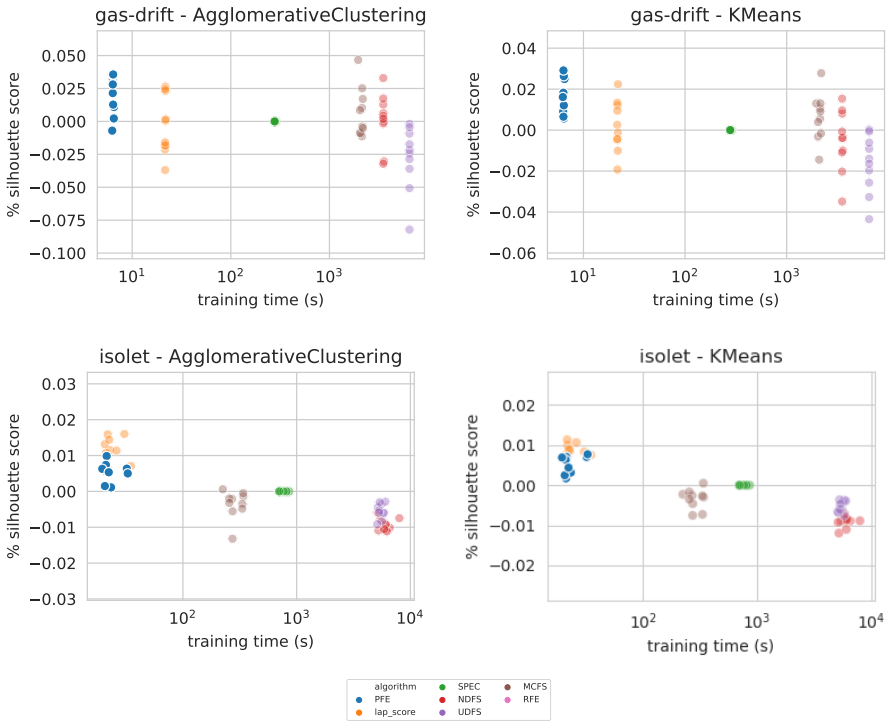

Fig. 4. Clustering results on benchmark datasets.

7. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A.: Autogluon-tabular: Robust and accurate automl for structured data. arXiv preprint arXiv:2003.06505 (2020)
8. Fanty, M., Cole, R.: Spoken letter recognition. In: Advances in Neural Information Processing Systems. pp. 220–226 (1991)
9. 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)
10. Guyon, I.: Design of experiments of the nips 2003 variable selection benchmark. In: NIPS 2003 workshop on feature extraction and feature selection (2003)
11. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (Mar 2003)
12. 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)
13. 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)
14. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in neural information processing systems. pp. 507–514 (2006)
15. Kozachenko, L., Leonenko, N.N.: Sample estimate of the entropy of a random vector. Problemy Peredachi Informatsii **23**(2), 9–16 (1987)

16. Lewis, P.: The characteristic selection problem in recognition systems. IRE Transactions on information theory **8**(2), 171–178 (1962)
17. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: A data perspective. ACM Computing Surveys (CSUR) **50**(6), 94 (2018)
18. Li, Z., Yang, Y., Liu, J., Zhou, X., Lu, H.: Unsupervised feature selection using nonnegative spectral analysis. In: Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
19. 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)
20. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
21. Steel, R.G.D., Torrie, J.H., et al.: Principles and procedures of statistics. Principles and procedures of statistics. (1960)
22. Steinhaus, H.: Sur la division des corp materiels en parties. Bull. Acad. Polon. Sci **1**(804), 801 (1956)
23. 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
24. Turner, M.C., Krewski, D., Pope III, C.A., Chen, Y., Gapstur, S.M., Thun, M.J.: Long-term ambient fine particulate matter air pollution and lung cancer in a large cohort of never-smokers. American journal of respiratory and critical care medicine **184**(12), 1374–1381 (2011)
25. Van Rijsbergen, C.J.: Information retrieval. 2nd. newton, ma (1979)
26. 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, http://doi.acm.org/10.1145/2641190.2641198
27. Vergara, A., Vembu, S., Ayhan, T., Ryan, M.A., Homer, M.L., Huerta, R.: Chemical gas sensor drift compensation using classifier ensembles. Sensors and Actuators B: Chemical **166**, 320–329 (2012)
28. 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)
29. Ward Jr, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American statistical association **58**(301), 236–244 (1963)
30. 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)
31. 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)
32. 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)
33. Yang, Y., Shen, H.T., Ma, Z., Huang, Z., Zhou, X.: L2, 1-norm regularized discriminative feature selection for unsupervised. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)

34. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th international conference on Machine learning. pp. 1151–1157 (2007)
35. 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)
36. Zill, D., Wright, W.S., Cullen, M.R.: Advanced engineering mathematics. Jones & Bartlett Learning (2011)