Feature Selection: A Review and Comparative Study

Younes Bouchlaghem[1,*] *and Yassine Akhiat*[2] *and Souad Amjad*[3]

[1]Department of Informatics, UAE, Tetouan Morocco
[2]Department of Informatics, USMBA, Fez Morocco
[3]Department of Informatics, UAE, Tetouan Morocco

Abstract. Feature selection (FS) is an important research topic in the area of data mining and machine learning. FS aims at dealing with the high dimensionality problem. It is the process of selecting the relevant features and removing the irrelevant, redundant and noisy ones, intending to obtain the best performing subset of original features without any transformation. This paper provides a comprehensive review of FS literature intending to supplement insights and recommendations to help readers. Moreover, an empirical study of six well-known feature selection methods is presented so as to critically analyzing their applicability.

Keywords: Dimensionality Reduction, Feature Extraction, Feature Selection, Environment.

1 Introduction

The vast amount of data has bombarded machine learning researchers with a multitude of unparalleled problems, making the learning task more challenging and computationally difficult. When using data mining and machine learning algorithms to analyze high-dimensional data, the performance of the learning algorithms can deteriorate due to over-fitting problem [1]. Machine learning models become more difficult to understand as the number of features increases, resulting in a reduction in generalizability [2]. To reduce the high dimensionality of data, data mining may use dimensionality reduction techniques. Feature extraction and selection are two types of dimensionality reduction [3]. The main objective of feature extraction (FE) is to reduce the initial feature space to a smaller one, where features lose their meaning due to the transformation (see Fig.1). The commonly used feature extraction methods include multi-dimensionality scaling [7], Isomap [8], Local linear Embedding [9], and Principal Component Analysis [10]. Feature selection (FS), as opposed to feature extraction, is the task of selecting relevant attributes and deleting irrelevant and redundant ones in order to obtaining the highest best feature subset without transformation (See Fig.1). As a result, learning models built with the chosen subset of features are more readable and interpretable [3,4,5,25].The main reasons behind using feature selection are: Reducing the storage capacity and execution time, preventing the curse of dimensionality problem, minimizing the over-fitting issue, resulting in improved model generalization and increasing the performance attainability [6].

This paper review will provide a critical analysis of six feature selection methods from different categories to expose their advantages and disadvantages. Moreover, we will shed light on the recent advance in feature selection in order to provide some guidelines and recommendations to practitioners and researchers. Along with reviewing the existing feature selection methods, we will empirically evaluate the effectiveness, shortcomings and applicability of six well-known FS methods including ReliefF, MI, SVM-RFE, SFS, Lasso and Ridge. In section 2, six FS methods are well discussed as well as their strength and limitations. Section 3 is devoted the obtained results applying the previously discussed FS methods. As to the end of this paper, we present the conclusion.

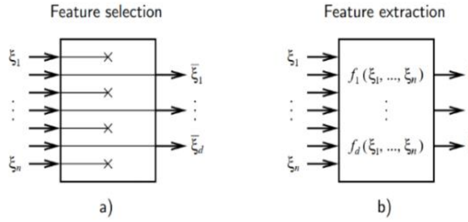

Fig. 1. Feature selection VS b) Feature extraction

Feature selection is categorized into three major approaches: Filter, Wrapper and Embedded approach.

- **Filters:** which perform feature selection without involving any modeling algorithm by relying on certain feature characteristics of training data, primarily its relationship with the class label. Filter methods are faster and more generalizable compared to Wrappers.
- **Wrappers:** instead of evaluating features individually as in filters, wrappers evaluate features taking into account the interaction between them. To determine and compare the relative output of each generated subset, they use learning algorithms which leads generally to make the learning process complex and expensive in terms of execution time.
- **Embedded:** Since feature selection is conducted and optimized during the classification process, this

[*] Corresponding author: younes011991@gmail.com

methodology is able to classify dependencies with less computing difficulty than wrapper.

2 Feature selection methods classification

Feature selection is an active research filed in machine learning, as it is an important pre-processing, finding success in different real problem applications. In general, feature selection algorithms are categorized into supervised, Semi-supervised and Unsupervised feature selection [2,3,4,5,6]. Supervised feature selection methods usually come in three flavors: Filter, Wrapper and Embedded approach. In this review, we will focus mainly on feature selection in high dimensional classification tasks.

2.1 Filter Methods

Filters evaluate feature relevance based on data intrinsic characteristics. As it can be seen in Figure 2, this category is a pre-processing stage that is independent of the induction algorithm. There are two key stages in the filtering process. To cause the classification model, it first ranks features individually based on a particular criterion measure such as distance, Pearson correlation, and entropy [4]. Second, it selects the best-ranked features using a threshold value. The remaining features are deemed to be unnecessary and uninformative. After that, the induction classifier receives the selected subset of features as input. Filters are fast which makes them more suitable for high dimensional datasets [3,4,5]. Since the relations between the independent variables aren't considered, redundant features may be allowed to be chosen. The following sub-sections provide an in-depth analysis of two well-known filter methods.

2.1.1 Mutual Information

This technique shows how much information is exchanged between a feature and the class label (between two features). MI is a measure of the impact of one random variable on another [22, 23, 26]. The characteristic with the most shared information with the class target is the best, since it may effectively identify members of one class from those of another. MI is mathematically defined as follows:

$$\mathrm{I(X,Y)} \sum_{\mathrm{i=1}}^{\mathrm{m}} \sum_{\mathrm{j=1}}^{\mathrm{m}} \mathrm{p}\big(\mathrm{X(i),Y(j)}\big).\log\left(\frac{\mathrm{p(X(i),Y(j))}}{\mathrm{p(X(i).p(Y(j)}}\right) \quad (1)$$

MI is zero when X and Y are statistically independent $\big(p\big(X(i),Y(j)\big) = p(X(i).p(Y(j)\big)$.

2.1.2 Relief

This method is an instance-based method. It tries to find the nearest neighbor from the same class (nearest hit) and from the opposite class for each training example (nearest miss) [12]. The difference in each feature's score is the difference between them. Relief quantifies each feature with a score Wi, which can be used to rank features. Relief approach can be highly effective for high-dimensional datasets since the weighting Wi of each feature is iteratively modified for each selected case. If a feature weight from the feature in nearby instances of the same class rather than nearby instances of the other class, the feature weight decreases; otherwise, it increases. Each feature's weight is calculated as follows:

$$W_i = W_i - (x_i - nearHit_i)^2 + (x_i - nearMiss_i)^2 \quad (2)$$

This algorithm's limitations include the fact that it is only applicable to two-class classification problems and does not tackle redundancy. An extended version of Relief called Relief-A is proposed in [13] for solving the missing data issue in order to solve the previously described problems with Relief. Relief-F is implemented to solve multi-class issues [24].

2.2 Wrapper Methods

On the contrary, wrapper approach construct models from scratch for each generated subset [14]. Then, it uses prediction performance as a criterion function verify its efficiency. This category takes into account the interactions between features. Generally, Wrappers achieve a better performance than Filters. However, they are very expensive in terms of models complexity and resources requirements. Many wrapper methods have been proposed in the literature [1,2,3,4,5]. In following subsections, we will concentrate on the most widely used wrapper methods.

2.2.1 Support Vector Machine Recursive Feature Elimination (SVM-RFE)

In [15], Guyon introduced a hybrid feature selection method where a support vector machine is based on recursive feature elimination (SVM-RFE). It begins with the all features then iteratively it eliminates features without backtracking. This greedy top-down strategy computes the normal vector of a decision hyper-plane created by a linear support vector machine (SVM) classifier. Projections of the normal vector to the coordinate system are subsequently used to assess the ranking of individual features. The features with the highest ranking are retained, and those with the lowest ranking are removed. This procedure is repeated until the desired number of features is attained, or the performance cannot be improved any further.

2.2.2 Sequential Forward Selection (SFS)

SFS is greedy search strategy [16]. It begins with an empty set and adds features in a greedy manner until the performance can no longer be improved. When the number of representative features (optimal subset) is minimal, SFS works well [17,18]. The disadvantage of

* Corresponding author: younes011991@gmail.com

this technique is that once a characteristic is included to the subset, it cannot be withdrawn, resulting in sub-optimal outcomes.

2.3 Embedded Methods

Unlike the wrapper strategy, which uses a heuristic search driven by the classifier's results, the embedded strategy is a compromise between the filter and wrapper approaches. It performs feature selection and creates an optimized classifier using the learning process itself. The embedded method, which is a middle ground between filters and wrappers, chooses features that emerge during the learning process based on the classifier's evaluation criteria, resulting in lower computational costs than wrappers. Regularization approaches are the most commonly used embedded methods.

2.3.1 LASSO (L1-Regulariezation)

The Least Absolte Shrinkage and Selection Operator, introduced by Robert Tibshirani in 1996 [19], is an efficient regularization function selection tool (LASSO). The LASSO approach penalizes the number of the machine learning algorithm parameters' absolute values. Any coefficients are reduced to zero by the additional limit (regularization). During the feature selection process, features with a non-zero coefficient after the shrinkage step are selected to be used in the model, whereas those with a coefficient of exactly zero are omitted. A tuning parameter (regularization parameter) is used to adjust the frequency of the regularization [20]. When λ is sufficiently enough, coefficients are forced to be precisely zero which results in reducing dimensionality. Indeed, the larger λ is, the more coefficients are shrinked to zero. On the other hand, if λ=0 this means that no regularization is applied (Ordinary Least Square). There are some benefits of using LASSO. Since the coefficients of the deceptive features are penalized and eliminated, it is effective at minimizing variance. As a result, it aids in preventing the over-fitting issue, resulting in better generalization ability. Furthermore, LASSO is very useful for improving interpretability by canceling irrelevant features. The LASSO approach is introduced in various mathematical forms; according to Robert Tibshirani, the lasso approximation is determined by the solution to the L1 regularization.

$$Minimize\left(\frac{\left|\left|Y - X\beta\right|\right|_2^2}{n}\right) Subject\ to \sum_{j=1}^{k} < t \qquad (3)$$

Where t is the upper bound for the sum of the coefficients. This optimization problem is equivalent to the parameter estimation that follows:

$$\hat{\beta}(\lambda) = \underset{\beta}{\operatorname{argmin}}\left(\frac{\left|\left|Y-X\beta\right|\right|_2^2}{n} + \lambda\left|\left|\beta\right|\right|_1\right) \qquad (4)$$

Where :

$$\left|\left|Y - X\beta\right|\right|_2^2 = \sum_{i=0}^{n}(Y_i - (X\beta)_i)^2, \left|\left|\beta\right|\right|_1 = \sum_{j=1}^{k}\left|\beta_j\right|$$

and λ ≥0 is the regularization parameter that control the amount of weight shrinkage.

The lasso method has some limitations:

- LASSO prefers to select one feature from each set of associated features while ignoring the others.
- In datasets of small n and large p (p is the number of features), LASSO selects no more than n attributes before saturating.

2.3.2 RIDGE (L2-Regularization)

RIGE regularization, also known as L2-Regularization, is a regularization technique. The objective function is given a squired magnitude of the model parameters as a penalty expression. Ridge shrinks model coefficients close to zero but not precisely zero, unlike LASSO, which offers a sparse collection of functions. The biggest difference between LASSO and Ridge strategy is that LASSO reduces the coefficients of unimportant to zero, effectively canceling them out. Ridge, on the other hand, produces non-zero coefficients, which is more useful for interpreting features.

The following Table (Table 1) provides a more detailed summary of the properties of the three feature selection categories, as well as the most prominent advantages and disadvantages.

Table 1. Advantages and disadvantages of Feature selection approaches.

	Advantages	**Disadvantages**
Filters	– Independent of learning model , – Fast execution – Suitable for high dimensional data , – Good generalizability.	– Interactions between features are ignored , – Fail to handle redundancy problem, – No interaction with the learning algorithm
Wrappers	– Better performance attainability , – Take into account interaction between features , – Identify feature interactions of higher order .	– Very expensive in terms of execution times, – Prone to over-fit, – The learning algorithm is built from scratch for each subset.

[*] Corresponding author: younes011991@gmail.com

Embedded	– Faster than wrappers, – Accurate , – Take into account interaction between features , – Identify feature dependencies.	– Learning algorithm specific , – Classifier-dependent selection.

3 Experimental results and discussion

In this section, an extensive number of experiments have been conducted to evaluate the previously discussed feature selection methods from different categories IG and Relief for filters, RFE-SVM and RFE-RF for wrappers, Ridge and LASOO for embedded. The conducted experiments have been carried out using seven benchmarking datasets.

3.1 Datasets

To assess our approach, we used seven binary classification datasets, which are available on UCI machine learning repository [21], including Sonar, Spambase, Eighthr, Clean, Chess, Madelon, and Ds1.100. More details about the characteristics of each dataset are presented in Table 2.

Table.2: Datasets and their characteristics

Id	Datasets	Features	Instances	Distribution
1	Eighthr	306	200000	4% + /96% -
2	Spambase	57	4601	39% + /61% -
3	Sonar	61	208	47% + /53% -
4	Chess	36	3197	30% + /70% -
5	Madelon	500	4400	50% + /50% -
6	Ds1.100	100	26.733	3%+ /97%-
7	Clean	167	6598	15%+ /85%-

To assess the strengths and the weaknesses feature selection methods applicability, we carefully selected datasets to be different in terms of the number of data points, attributes, linearity and the distribution of class label.

3.2 Evaluation

In this experimental section, we tested the performance of six feature selection methods from different FS categories when used in conjunction with two classification algorithms (Random Forest, Support-vector machines) to obtained more reliable results. We use classification accuracy to evaluate the performance of each FS methods. Generally, the higher the classification accuracy, the better the selected feature subset is.

Table 3: Time execution of each feature selection method in seconds.

Id	Filter		Wrapper		Embedded	
	ReliefF	MI	SFS	RFE-SVM	LASSO	RIDGE
1	1.64	2.61	**6.29**	2.10	1.62	2.99
2	2.43	3.63	**5.82**	4.22	2.10	3.84
3	2.43	3.60	**5.81**	4.09	2.16	3.91
4	1.2	1.86	**5.19**	1.9	1.08	2.01
5	4.11	18.62	114.03	**591.42**	2.78	5.25
6	118.16	55.59	**251.88**	198.41	16.07	24.08
7	2.48	3.65	**7.28**	4.49	2.34	4.13

3.3 Discussion

By carefully analyzing the obtained accuracies of filter methods (see Table 4), it is clear that wrapper methods are able to increase the classification accuracy in the majority of datasets (5 out of 7) followed by embedded methods (4 out of 7) and filters have increased the classification accuracy of 3 datasets out of 7. Besides classification accuracy, time execution is another critical aspect in FS especially for high dimensional datasets. Table 3 presents the running time (in seconds) of the implemented FS methods. It is obvious that Filter methods are faster than other approaches (Wrapper and embedded). As wrappers make use of learning algorithms to assess each feature subset, there are very expensive compared to filter category. This is even clearly demonstrated when the size of the dataset becomes larger (see datasets ds1.100 and madelon). Embedded methods are good alternatives especially when the feature space is of high dimension.

4 Conclusion

In this review paper, we have provided a comprehensive survey of feature selection state-of-the-art methods and their categorization as well. Then, we have cited and discussed several fundamental algorithms of each category. Moreover, critical analysis and comparison have been conducted to expose the merits and demerits of each FS method. In addition, we have provided an empirical study to evaluated six feature selection methods including: ***RatlifF*** and ***IG*** as filters, ***SFS, RFE-SVM*** as wrappers, ***LASSO, RIDGE*** as embedded. Each method is evaluated in combination with two well-known classifiers (**SVM** and **RF**) to reliably test their ability to select the best subset. Seven benchmarking datasets have been employed to demonstrate the applicability of feature selection methods. The results clearly confirm that too much information does not always help machine learning algorithms as it implies that feature subset may include redundant or irrelevant features, and their presence can deteriorate the performance of the classifier. As a result, feature selection techniques are infeasible to reduce the data dimensionality.

[*] Corresponding author: younes011991@gmail.com

At the end of this paper, we present some overlooked and open issues that must be discussed and analyzed further:

- Ensemble techniques should be exploited in feature selection field to enhance the stability selection.
- Since Wrapper approach is mostly avoided in the literature, the focus should be spotlighted on hybrid and ensemble approaches.
- Due to the immense volume of generated datasets in many domains including image recognitions, text classification and biomedical, we intend to apply deep learning models to estimates the goodness of features relying on the existed hidden feature interaction that may be well-detected using deep learning instead of machine learning models.
- Using feature selection methods in environmental sound recognition could be helpful for extracting a large amount of information to understand our environment.

References

1. Roelofs, R., Fridovich-Keil, S., Miller, J., Shankar, V., Hardt, M., Recht, B., & Schmidt, L. A meta-analysis of overfitting in machine learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp. 9179-9189). . (2019, December).
2. Yassine, A., Mohamed, C., & Zinedine, A.. Feature selection based on pairwise evalution. In 2017 *Intelligent Systems and Computer Vision (ISCV)* (pp. 1-6). IEEE. (2017, April)
3. Akhiat, Y., Asnaoui, Y., Chahhou, M., & Zinedine, A. A new graph feature selection approach. In 2020 *6th IEEE Congress on Information Science and Technology (CiSt)* (pp. 156-161). IEEE. (2021, June).
4. Akhiat, Y., Chahhou, M., & Zinedine, A. Feature selection based on graph representation. In 2018 IEEE 5th *International Congress on Information Science and Technology (CiSt)* (pp. 232-237). IEEE. (2018, October).
5. Akhiat, Y., Chahhou, M., & Zinedine, A. Ensemble feature selection algorithm. *International Journal of Intelligent Systems and Applications*, 11(1), 24. (2019).
6. Akhiat, Y., Manzali, Y., Chahhou, M., & Zinedine, A. A New Noisy Random Forest Based Method for Feature Selection. Cybernetics and Information Technologies, **21**(2), 10-28. (2021).
7. Cox, M. A., & Cox, T. F. Multidimensional scaling. In Handbook of data visualization (pp. 315-347). Springer, Berlin, Heidelberg. (2008).
8. Tenenbaum, J. B., De Silva, V., & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. science, **290**(5500), 2319-2323. (2000).
9. Roweis, S. T., & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. science, **290**(5500), 2323-2326. (2000).
10. Barshan, E., Ghodsi, A., Azimifar, Z., & Jahromi, M. Z. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. Pattern Recognition, **44**(7), 1357-1371. (2011).
11. Quinlan, J. R. Induction of decision trees. Machine learning. (1986).
12. Kononenko, I. Estimating attributes: Analysis and extensions of RELIEF. *In European conference on machine learning* (pp. 171-182). Springer, Berlin, Heidelberg. (1994, April).
13. Yu, L., & Liu, H. Efficient feature selection via analysis of relevance and redundancy. The Journal of Machine Learning Research, 5, 1205-1224. (2004).
14. Kohavi, R., & John, G. H. Wrappers for feature subset selection. Artificial intelligence, **97**(1-2), 273-324. (1997).
15. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. Gene selection for cancer classification using support vector machines. Machine learning, **46**(1), 389-422. (2002).
16. Raman, B., & Ioerger, T. R. Instance-based filter for feature selection. Journal of Machine Learning Research, **1**(3), 1-23. (2002).
17. Tang, J., Alelyani, S., & Liu, H. Feature selection for classification: A review. Data classification: Algorithms and applications, **37**. (2014).
18. Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. Cybernetics and Information Technologies, **19**(1), 3-26.
19. Tibshirani, R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), **58**(1), 267-288. (1996).
20. Fonti, V., & Belitser, E. Feature selection using lasso. VU Amsterdam Research Paper in Business Analytics, 30, 1-25. (2017).
21. Lichman, M. UCI Machine Learning Repository http://archive.ics.uci.edu/ml. UCI Machine Learning Repository, 2013. (2013).
22. Battiti, R. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on neural networks, **5**(4), 537-550. (1994).
23. Guyon, I., & Elisseeff, A. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182. (2003).
24. Robnik-Šikonja, M., & Kononenko, I. Theoretical and empirical analysis of ReliefF and RReliefF. Machine learning, **53**(1), 23-69. (2003).
25. Akhiat, Y., Manzali, Y., Chahhou, M., & Zinedine, A. A New Noisy Random Forest Based Method for Feature Selection. Cybernetics and Information Technologies, **21**(2), 10-28. (2021).
26. Asnaoui, Y., Akhiat, Y., & Zinedine, A. Feature selection based on attributes clustering. In 2021 *Fifth International Conference On Intelligent Computing in Data Sciences* (ICDS) (pp. 1-5). IEEE. (2021, October).

[*] Corresponding author: younes011991@gmail.com

Table 4: The summarized results for all evaluated feature selection methods in terms of accuracy metric

	Without feature selection		With feature selection(Filter approach)				With feature selection(Wrapper approach)				With feature selection(Embedded approach)			
			ReliefF		MI		SFS		RFE-SVM		LASSO		RIDGE	
Id	SVM	RF	SVM	RF	SVM	RF	SVM	RF	SVM	RF	SVM	RF	SVM	RF
1	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	0.93 ±0.01	**0.94 ±0.00**	**0.94 ±0.00**	0.93 ±0.01	0.93 ±0.01
2	0.65 ±0.02	0.89 ±0.03	0.58 ±0.00	0.88 ±0.04	0.58 ±0.00	0.89 ±0.03	0.58 ±0.00	**0.90 ±0.03**	0.65 ±0.01	**0.90 ±0.03**	0.61±0.00	0.64 ±0.02	0.60 ±0.04	0.86±0.02
3	0.62 ±0.07	0.67 ±0.09	0.62 ±0.07	0.53 ±0.22	0.62 ±0.07	**0.74 ±0.18**	0.62 ±0.07	0.69 ±0.10	0.62 ±0.07	0.66 ±0.21	**0.63 ±0.16**	0.68 ±0.07	0.62 ±0.07	0.67 ±0.15
4	0.92 ±0.02	0.91 ±0.02	0.84 ±0.00	0.88 ±0.01	0.84 ±0.00	0.89 ±0.01	0.84 ±0.00	**0.92 ±0.03**	**0.85 ±0.01**	0.91 ±0.02	**0.85 ±0.00**	0.88 ±0.01	0.84 ±0.00	0.88 ±0.00
5	0.49 ±0.03	0.56 ±0.08	0.49 ±0.06	0.45 ±0.09	0.50 ±0.08	0.60 ±0.07	**0.51 ±0.04**	0.58 ±0.12	0.50 ±0.07	**0.66 ±0.08**	0.54 ±0.10	0.60 ±0.09	0.50 ±0.04	0.58 ±0.09
6	0.89 ±0.13	0.88 ±0.19	0.81 ±0.18	0.81 ±0.18	**0.90 ±0.16**	0.87 ±0.18	0.89 ±0.20	0.87 ±0.18	0.89 ±0.16	**0.89 ±0.15**	0.67±0.24	0.69 ±0.26	0.62 ±0.19	0.59 ±0.19
7	0.97 ±0.05	0.97 ±0.02	**0.97 ±0.02**	**0.97 ±0.00**	0.97 ±0.00	**0.97 ±0.00**	0.97 ±0.04	**0.97 ±0.00**	0.97 ±0.00	**0.97 ±0.00**	0.97 ±0.00	**0.97 ±0.00**	0.97 ±0.00	**0.97 ±0.00**

[*] Corresponding author: younes011991@gmail.com