Feature Selection Library (MATLAB Toolbox)

Giorgio Roffo

giorgio.roffo@glasgow.ac.uk
University of Glasgow,
School of Computing Science

Abstract. Feature Selection Library (FSLib) is a widely applicable MATLAB library for Feature Selection (FS). FS is an essential component of machine learning and data mining which has been studied for many years under many different conditions and in diverse scenarios. These algorithms aim at ranking and selecting a subset of relevant features according to their degrees of relevance, preference, or importance as defined in a specific application. Because feature selection can reduce the amount of features used for training classification models, it alleviates the effect of the curse of dimensionality, speeds up the learning process, improves model's performance, and enhances data understanding. This short report provides an overview of the feature selection algorithms included in the FSLib MATLAB toolbox among: filter, embedded, and wrappers methods.

Keywords: Feature Selection Library, ILFS, FSLib, Feature selection, variable ranking, variable selection, dimensionality reduction, pattern discovery, filters, wrappers, embedded, Linear Regression, Deep Learning, Logistic Regression, Relief-F, Laplacian Score, mutinffs, Fisher Score, mrmr, fsv, mcfs, Infinite Feature Selection, Feature Selection via Eigenvector Centrality, SVM-RFE, Convolutional Neural Networks, File Exchange, MATLAB Central, MathWorks

1 Introduction

Feature selection (FS) for classification is a well-researched problem, aimed at reducing the dimensionality and noise in data sets. Adequate selection of features may improve accuracy and efficiency of classifier methods. Feature ranking and selection for classification aims at reducing the dimensionality and noise in data sets. FS performs information filtering since it removes redundant or unwanted information from an information stream. Sometimes in many learning domains, a human operator defines the potentially useful features. However, not all of these features may be relevant and some of them may be redundant. In such a case, automatic feature selection can be employed for removing irrelevant, redundant, and noisy information from the data, often leading to better performance in learning and classification tasks. Indeed, feature selection is a widely recognized important task in machine learning, artificial intelligence, computer vision, and data mining successfully applied in fields like information retrieval (e.g., feature-based user retrieval [24,19,3]), user re-identification by soft-biometrics [20,21], recommendation systems [25], visual object tracking [14,15,26,27] for real-time feature ranking and selection and many other domains. Feature selection improves algorithms performance and classification accuracy since the chance of overfitting increases with the number of features.

Feature selection techniques can be partitioned into three classes [11]: *wrappers* (see Fig. 2), which use classifiers to score a given subset of features; *embedded* methods (see Fig. 3), which inject the selection process into the learning of the classifier; *filter* methods (see Fig. 1), which analyze intrinsic properties of data, ignoring the classifier. Most of these methods can perform two operations, *ranking* and *subset selection*: in the former, the importance of each individual feature is evaluated, usually by neglecting potential interactions among the elements of the joint set [5]; in the latter, the final subset of features to be selected is provided. In some cases, these two operations are performed sequentially (ranking and selection) [12,1,7,33,16,32]; in other cases, only the selection is carried out [8]. Generally, the subset selection is always supervised, while in the ranking case, methods can be supervised or not.

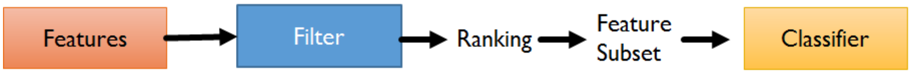

Fig. 1. Filter Methods: the selection of features is independent of the classifier used. They rely on the general characteristics of the training data to select features with independence of any predictor.

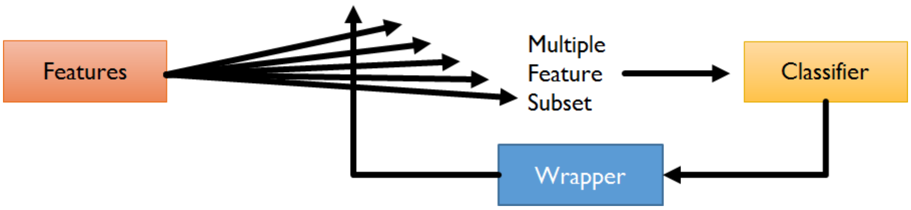

Fig. 2. Wrapper models involve optimizing a predictor as part of the selection process. They tend to give better results but filter methods are usually computationally less expensive than wrappers.

While wrapper models involve optimizing a predictor as part of the selection process, filter models rely on the general characteristics of the training data to select features with independence of any predictor. Wrapper models tend to give better results but filter methods are usually computationally less expensive than wrappers. So, in those cases in which the number of features is very large, filter methods are indispensable to obtain a reduced set of features that then can be treated by other more expensive feature selection methods.

Embedded methods, Fig. 3, differ from other feature selection methods in the way feature selection and learning interact. In contrast to filter and wrapper approaches, in embedded methods the learning part and the feature selection part can not be separated - the structure of the class of functions under consideration plays a crucial role. For example, Weston et al. [2000] measure the importance of a feature using a bound that is

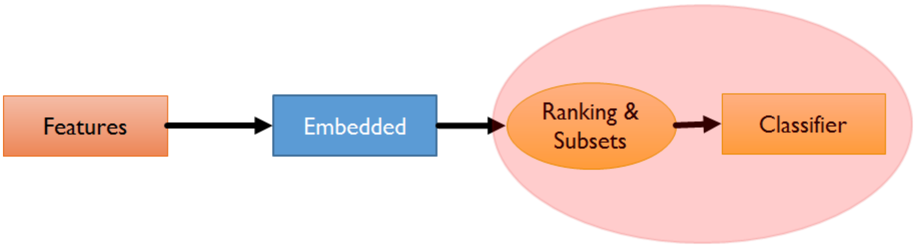

Fig. 3. In embedded methods the learning part and the feature selection part can not be separated.

valid for Support Vector Machines only thus it is not possible to use this method with, for example, decision trees.

Feature selection is *NP-hard* [11]; if there are n features in total, the goal is to select the optimal subset of $m \ll n$, to evaluate $\binom{n}{m}$ combinations; therefore, suboptimal search strategies are considered. With the filters, features are first considered individually, ranked, and then a subset is extracted, some examples are Inf-FS [30], (EC-FS) [22], MutInf [33], and Relief-F [16]. Conversely, with wrapper and embedded methods, subsets of features are sampled, evaluated, and finally kept as the final output, for instance, FSV [1,7], and SVM-RFE [12].

Each feature selection methods can be also classified as $Supervised$ or $Unsupervised$. For supervised learning, feature selection algorithms maximize some function of predictive accuracy. Because we are given class labels, it is natural that we want to keep only the features that are related to or lead to these classes. Generally, feature selection for supervised machine learning tasks can be accomplished on the basis of the following underlying hypothesis: "*a good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other* [6]". We may define a feature which is highly correlated with the class as "relevant", whereas a feature which is uncorrelated with the others as not "redundant". Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. Figure 4(Left) shows an example of feature redundancy. Note that the data can be grouped in the same way using only either feature x or feature y. Therefore, we consider features x and y to be redundant. Figure 4(Right) shows an example of an irrelevant feature. Observe that feature y does not contribute to class discrimination. Used by itself, feature y leads to a single class structure which is uninteresting. Note that irrelevant features can misguide classification results, especially when there are more irrelevant features than relevant ones.

In unsupervised learning, we are not given class labels. Unsupervised learning is a difficult problem. It is more difficult when we have to simultaneously find the relevant features as well.

The report is organized as follows. A brief overview of the existing methods is given in Section 2. Finally, conclusions are provided in Section 3.

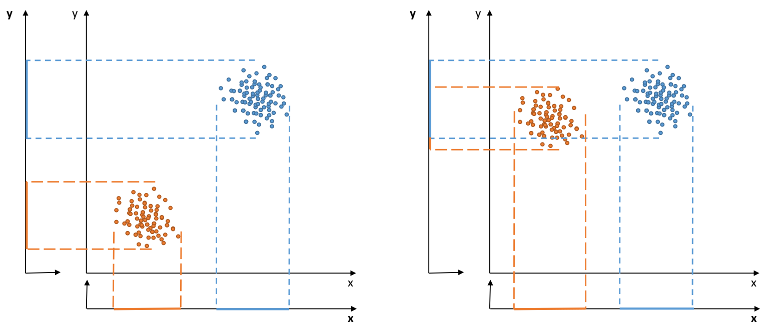

Fig. 4. (Left) In this example, features x and y are redundant, because feature x provides the same information as feature y with regard to discriminating the two clusters. (Right) In this example, we consider feature y to be irrelevant, because if we omit x, we have only one class, which is uninteresting.

2 Feature Selection Techniques

Feature selection can be understood as finding the feature subset of a certain size that leads to the largest possible generalization or equivalently to minimal risk. Among the most used feature selection strategies, *Relief-F* [16] is an iterative, randomized, and supervised approach that estimates the quality of the features according to how well their values differentiate data samples that are near to each other; it does not discriminate among redundant features, and performance decreases with few data. Similar problems affect SVM-RFE (RFE) [12], which is an embedded method that selects features in a sequential, backward elimination manner, ranking high a feature if it strongly separates the samples by means of a linear SVM.

An effective yet fast filter method is the *Fisher* method [8], it computes a score for a feature as the ratio of interclass separation and intraclass variance, where features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones. Other widely used filters are based on mutual information, dubbed *MI* here [33], which considers as a selection criterion the mutual information between the distribution of the values of a given feature and the membership to a particular class; Even in the last case, features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones.

The Infinite Latent Feature Selection (ILFS) [29] is a probabilistic latent feature selection approach that performs the ranking step by considering all the possible subsets of features bypassing the combinatorial problem. The most appealing characteristic of the ILFS is that it aims to model the features “relevancy” in a PLSA-inspired generative process. The derived mixing weights $P(z|f)$ are used to weight a graph of features. The weighted graph, serves to perform the ranking step providing a score of importance for each feature as a function of the importance of its neighbors.

Feature selection via Eigenvector Centrality (EC-FS) [22] is a filter method which maps the FS problem on an affinity graph - where features are the nodes - the solution is given by assessing the importance of nodes through some indicators of centrality, in particular, the *Eigenvector Centrality (EC)*. The gist of EC is to estimate the importance of a feature as a function of the importance of its neighbors. Ranking central nodes individuates candidate features, which turn out to be effective from a classification.

Selecting features in unsupervised learning scenarios is a much harder problem, due to the absence of class labels that would guide the search for relevant information. In this scenario, a recent unsupervised graph-based filter is the Inf-FS [30,27]. In the Inf-FS formulation, each feature is a node in the graph, a path is a selection of features, and the higher the centrality score, the most important (or most different) the feature. It assigns a score of importance to each feature by taking into account all the possible feature subsets as paths on a graph. Another unsupervised method is the Laplacian Score (LS) [13], where the importance of a feature is evaluated by its power of locality preserving. In order to model the local geometric structure, this method constructs a nearest neighbor graph. LS algorithm seeks those features that respect this graph structure.

Among the very recent wrapper approaches, the Dependence Guided Unsupervised Feature Selection (DGUFS) [9] considers the interdependence among original data, cluster labels, and selected features. In particular, DGUFS is a projection-free feature selection model based on l2,0-norm equality constraints. DGUFS performs feature subset selection by optimizing two terms: one term increases the dependence on original data, while the other term maximizes the dependence of selected features on cluster labels to guide the process of subset FS. Another unsupervised wrapper approach is the Feature Selection with Adaptive Structure Learning (FSASL) [4], that performs structure learning and FS simultaneously. FSASL is based on linear regression and its main drawback is the high computational complexity, which is costly for high dimensional data (see Table 1). The Unsupervised Feature Selection with Ordinal Locality (UFSOL) is proposed in [10] and it is a clustering-based approach. UFSOL preserves the relative neighborhood proximities and contributes to distance-based clustering.

The Least Absolute Shrinkage and Selection Operator (LASSO) [17], minimizes the prediction error while maintaining the sum of the absolute values of the model parameters smaller than a fixed value. This method applies a regularization process that penalizes the coefficients of the regression variables while setting the less relevant to zero to respect the constraint on the sum. FS is a consequence of this process when all the variables that still have non-zero coefficients are selected to be part of the model.

Finally, for the wrapper method, we include the *feature selection via concave minimization* (*FSV*) [1], where the feature selection process is injected *into* the training of an SVM by a linear programming technique.

2.1 The Feature Selection Library (FSLib) Approaches

For each algorithm[1], Table 1 reports its *type*, that is, f = filters, w = wrappers, e = embedded methods, and its *class*, that is, s = supervised or u = unsupervised (using or not using the labels associated with the training samples in the ranking operation). Additionally, we report computational complexity (if it is documented in the literature);

[1] The Feature Selection Library (FSLib) for MATLAB is publicly available on File Exchange - MATLAB Central - MathWorks

ID	Acronym	Type	Cl.	Comp. Complexity
1	CFS [12]	f	u	$\mathcal{O}(\frac{n^2}{2}T)$
2	DGUFS[9]	w	u	N/A
3	ECFS [22,28]	f	s	$\mathcal{O}(Tn+n^2)$
4	Fisher [8]	f	s	$\mathcal{O}(Tn)$
5	FSASL [4]	w	u	$\mathcal{O}(n^3+Tn^2)$
6	FSV [1]	e	s	$\mathcal{O}(T^2n^2)$
7	ILFS [29]	f	s	$\mathcal{O}(n^{2.37}+in+T+C)$
8	LASSO [17]	e	s	$\mathcal{O}(T^2n^2)$
9	LLCFS [34]	f	u	N/A
10	LS [13]	f	u	N/A
11	MCFS [2]	f	u	N/A
12	MI [33]	f	s	$\mathcal{O}(T^2n^2)$
13	Relief-F [16]	f	s	$\mathcal{O}(iTnC)$
14	RFE [12]	w	s	$\mathcal{O}(T^2nlog_2n)$
15	UDFS [31]	f	u	N/A
16	UFSOL[10]	w	u	$\mathcal{O}(iTCn^3)$
17	L0[12]	w	s	N/A
18	**Inf-FS** [23]	f	u	$\mathcal{O}(n^{2.37}(1+T))$
19	mRMR [18]	f	s	$\mathcal{O}(n^3T^2)$

Table 1. List of the feature selection approaches provided with the *Feature Selection Library (FSLib)*. The table reports their *Type*, class (*Cl.*), and complexity (*Compl.*). As for the complexity, T is the number of samples, n is the number of initial features, i is the number of iterations in the case of iterative algorithms, and C is the number of classes. The complexity of FSV cannot be specified since it is a wrapper (it depends on the chosen classifier).

Table 1 lists the proposed methods, reporting their *type*, that is, *f* = filters, *w* = wrappers, *e* = embedded methods, and their *class*, that is, *s* = supervised or *u* = unsupervised (using or not using the labels associated with the training samples in the ranking operation). Additionally, we report their computational complexity (if it is documented in the literature).

3 Concluding Remarks

We integrated the proposed set of feature selection methods with uniform input and output formats in our code library to facilitate large scale performance evaluation and application. The Feature Selection Library (FSLib) for MATLAB is publicly available on File Exchange - MATLAB Central - MathWorks.

References

1. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: ICML. pp. 82–90. Morgan Kaufmann (1998)
2. 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)
3. Cristani, M., Roffo, G., Segalin, C., Bazzani, L., Vinciarelli, A., Murino, V.: Conversationally-inspired stylometric features for authorship attribution in instant messaging. In: Proceedings of the ACM International Conference on Multimedia. pp. 1121–1124 (2012)

4. Du, L., Shen, Y.D.: Unsupervised feature selection with adaptive structure learning. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 209–218. ACM (2015)
5. Duch, W., Wieczorek, T., Biesiada, J., Blachnik, M.: Comparison of feature ranking methods based on information entropy. In: IJCNN. vol. 2. IEEE (2004)
6. Gennari, J.H., Langley, P., Fisher, D.: Models of incremental concept formation. Artif. Intell. 40(1-3), 11–61 (1989), `http://dx.doi.org/10.1016/0004-3702(89)90046-5`
7. Grinblat, G.L., Izetta, J., Granitto, P.M.: Svm based feature selection: Why are we using the dual? In: IBERAMIA. pp. 413–422 (2010)
8. Gu, Q., Li, Z., Han, J.: Generalized fisher score for feature selection. CoRR abs/1202.3725 (2012), `http://dblp.uni-trier.de/db/journals/corr/corr1202.html#abs-1202-3725`
9. Guo, J., Zhu, W.: Dependence guided unsupervised feature selection. In: Proc. AAAI Conf. Artificial Intell. (AAAI). pp. 2232–2239. New Orleans, Louisiana (Feb 2018)
10. Guo, J., Quo, Y., Kong, X., He, R.: Unsupervised feature selection with ordinal locality. In: Multimedia and Expo (ICME), 2017 IEEE International Conference on. pp. 1213–1218. IEEE (2017)
11. Guyon, I.: Feature extraction: foundations and applications, vol. 207. Springer Science & Business Media (2006)
12. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. 46(1-3), 389–422 (2002), `http://dx.doi.org/10.1023/A:1012487302797`
13. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing Systems 18 (2005)
14. Hua, G., Jégou, H. (eds.): Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II, Lecture Notes in Computer Science, vol. 9914 (2016), `http://dx.doi.org/10.1007/978-3-319-48881-3`
15. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R.P., Cehovin, L., Vojír, T., Häger, G., Lukezic, A., Fernández, G., Gupta, A., Petrosino, A., Memarmoghadam, A., García-Martín, A., Montero, A.S., Vedaldi, A., Robinson, A., Ma, A.J., Varfolomieiev, A., Alatan, A.A., Erdem, A., Ghanem, B., Liu, B., Han, B., Martínez, B., Chang, C., Xu, C., Sun, C., Kim, D., Chen, D., Du, D., Mishra, D., Yeung, D., Gundogdu, E., Erdem, E., Khan, F.S., Porikli, F., Zhao, F., Bunyak, F., Battistone, F., Zhu, G., Roffo, G., Subrahmanyam, G.R.K.S., Bastos, G.S., Seetharaman, G., Medeiros, H., Li, H., Qi, H., Bischof, H., Possegger, H., Lu, H., Lee, H., Nam, H., Chang, H.J., Drummond, I., Valmadre, J., Jeong, J., Cho, J., Lee, J., Zhu, J., Feng, J., Gao, J., Choi, J.Y., Xiao, J., Kim, J., Jeong, J., Henriques, J.F., Lang, J., Choi, J., Martínez, J.M., Xing, J., Gao, J., Palaniappan, K., Lebeda, K., Gao, K., Mikolajczyk, K., Qin, L., Wang, L., Wen, L., Bertinetto, L., Rapuru, M.K., Poostchi, M., Maresca, M.E., Danelljan, M., Mueller, M., Zhang, M., Arens, M., Valstar, M.F., Tang, M., Baek, M., Khan, M.H., Wang, N., Fan, N., Al-Shakarji, N., Miksik, O., Akin, O., Moallem, P., Senna, P., Torr, P.H.S., Yuen, P.C., Huang, Q., Nieto, R.M., Pelapur, R., Bowden, R., Laganière, R., Stolkin, R., Walsh, R., Krah, S.B., Li, S., Zhang, S., Yao, S., Hadfield, S., Melzi, S., Lyu, S., Li, S., Becker, S., Golodetz, S., Kakanuru, S., Choi, S., Hu, T., Mauthner, T., Zhang, T., Pridmore, T.P., Santopietro, V., Hu, W., Li, W., Hübner, W., Lan, X., Wang, X., Li, X., Li, Y., Demiris, Y., Wang, Y., Qi, Y., Yuan, Z., Cai, Z., Xu, Z., He, Z., Chi, Z.: The visual object tracking VOT2016 challenge results. In: Hua and Jégou [14], pp. 777–823, `http://dx.doi.org/10.1007/978-3-319-48881-3_54`
16. Liu, H., Motoda, H. (eds.): Computational Methods of Feature Selection. Chapman & Hall (2008)

17. Liu, H., Motoda, H.: Computational methods of feature selection. CRC Press (2007)
18. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE TPAMI 27, 1226–1238 (2005)
19. Roffo, G., Cristani, M., Bazzani, L., Minh, H., Murino, V.: Trusting Skype: Learning the way people chat for fast user recognition and verification. In: Proceeedings of the IEEE Conference on Computer Vision Workshops. pp. 748–754 (2013)
20. Roffo, G., Giorgetta, C., Ferrario, R., Cristani, M.: Proceedings of the Workshop on Human Behavior Understanding, chap. Just the Way You Chat: Linking Personality, Style and Recognizability in Chats, pp. 30–41. Springer Verlag (2014)
21. Roffo, G., Giorgetta, C., Ferrario, R., Riviera, W., Cristani, M.: Statistical analysis of personality and identity in chats using a keylogging platform. In: Proceedings of the International Conference on Multimodal Interaction. pp. 224–231 (2014)
22. Roffo, G., Melzi, S.: Features selection via eigenvector centrality. In: Proceedings of New Frontiers in Mining Complex Patterns (NFMCP 2016) (Oct 2016)
23. Roffo, G., Melzi, S., Cristani, M.: Infinite feature selection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4202–4210 (2015)
24. Roffo, G., Segalin, C., Vinciarelli, A., Murino, V., Cristani, M.: Reading between the turns: Statistical modeling for identity recognition and verification in chats. In: Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on. pp. 99–104 (Aug 2013)
25. Roffo, G., Vinciarelli, A.: Personality in computational advertising: A benchmark. In: Proceedings of ACM RecSys - International Workshop on Emotions and Personality in Personalized Systems (Sept 2016)
26. Roffo, G., Melzi, S.: Object tracking via dynamic feature selection processes. arXiv preprint arXiv:1609.01958 (2016)
27. Roffo, G., Melzi, S.: Online feature selection for visual tracking. In: In Conf. The British Machine Vision Conference (BMVC) (September 2016)
28. Roffo, G., Melzi, S.: Ranking to Learn: Feature Ranking and Selection via Eigenvector centrality, pp. 19–35 (2017)
29. Roffo, G., Melzi, S., Castellani, U., Vinciarelli, A.: Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1398–1406 (2017)
30. Roffo, G., Melzi, S., Cristani, M.: Infinite feature selection. In: IEEE International Conference on Computer Vision, ICCV. pp. 4202–4210 (2015)
31. Yang, Y., Shen, H.T., Ma, Z., et Al: L2,1-norm regularized discriminative feature selection for unsupervised learning. In: In Conf. International Joint Conference on Artificial Intelligence. pp. 1589–1594 (2011)
32. Yu, L., Han, Y., Berens, M.E.: Stable gene selection from microarray data via sample weighting. IEEE/ACM TCBB 9(1), 262–272 (2012)
33. Zaffalon, M., Hutter, M.: Robust feature selection using distributions of mutual information. In: UAI. pp. 577–584 (2002)
34. Zeng, H., Cheung, Y.m.: Feature selection and kernel learning for local learning-based clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(8), 1532–1547 (2011)