Neural Comput & Applic (2019) 31 (Suppl 2):S989–S997
https://doi.org/10.1007/s00521-017-3062-0

ORIGINAL ARTICLE

Semi-supervised relevance index for feature selection

Frederico Coelho[1] · **Cristiano Castro**[1] · **Antônio P. Braga**[1] · **Michel Verleysen**[2]

Received: 25 August 2016 / Accepted: 13 June 2017 / Published online: 21 June 2017

Abstract This paper presents a new relevance index based on mutual information that is based on labeled and unlabeled data. The proposed index, which is based in Mutual Information, takes into account the similarity between features and their joint influence on the output variable. Based on this principle, a method to select features is developed to eliminate redundant and irrelevant features when the relevance index value is less then a threshold value. A strategy to set the threshold is also proposed in this work. Experiments show that the new method is capable of capturing important joint relations between input and output variables, which are incorporated into a new feature selection clustering approach.

Keywords Mutual information · Semi-supervised · Feature selection · Similarity criterion

✉ Frederico Coelho
fredgfc@ufmg.br

Cristiano Castro
crislcastro@gmail.com

Antônio P. Braga
apbraga@ufmg.br

Michel Verleysen
verleysen@dice.ucl.ac.be

[1] Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

[2] Université Catholique de Louvain, Louvain-la-Neuve, Belgium

1 Introduction

Most feature selection methods aim at minimum redundancy (mR) and maximum relevance (MR) [9] of features by considering two independent objective functions. Relevance is quantified according to the ability of features to discriminate target classes, while redundancy in a subset of features is estimated on a feature-to-feature basis, without necessarily considering class labels. Although the problem is not often treated with an Optimization approach in mind, it is clear that there are two objective functions involved, mR and MR, and that they can be conflicting, since their minima may not coincide. Minimum redundancy may not result on maximum relevance, and vice versa.

Since redundancy can be estimated even without knowing data labels, i.e., by considering only unlabeled data, and relevance generally requires label information, trading-off them can be seen as a semi-supervised feature selection (SSFS) problem. Methods that consider both labeled and unlabeled sources of information gained importance in recent years, especially due to the increasing ability of present machines to acquire data, while labeling is still limited. New methods could be developed by extending present semi-supervised learning principles to incorporate feature selection, or by developing new selection strategies that are robust to a reasonable amount of unlabeled data.

Redundancy is a concept that is associated with spatial relations of features, so it relies on a feature similarity index (FSI) in order to be estimated; however, most approaches, e.g., feature clustering [13], do not regard the influence on labels as part of FSI. We consider in this paper that both feature-to-feature similarity and feature-to-label relations could be combined in a single FSI index. We propose a FSI that can be considered a semi-supervised

feature similarity index, since unlabeled data are used to evaluate the *direct similarity* and the labeled data are used to evaluate how related are the influences of each feature on the output labels. The two components are linearly combined according to a trade-off parameter λ, which is responsible to balance the importance of the two quantities in the final index. A strategy for selecting λ is also presented in this paper.

The similarity index is a pairwise measure that is estimated in the subspace characterized by the corresponding pair of features. Mutual information [2] (MI) among samples was used in this paper to estimate the unlabeled portion of the index. Likewise, the supervised portion is estimated by a distance relation between the MI of each feature and the output labels. Its first component is an estimation of how input samples are related in the two subspaces, and the second one is an estimation of how they relate to the output variable.

In this work, first the similarity index will be defined and a strategy to set the balance parameter will be explained. After that, a significance criterion will be defined in order to set a threshold to the similarity values and then a semi-supervised feature selection method will be proposed. Finally, the experiments are performed and the results are presented and discussed.

2 Semi-supervised feature similarity index S

2.1 Related work

To reduce the number of features of a data set, there are three general strategies where, more or less, algorithms could be classified. In the first strategy there are the so-called filter methods whose objective is to rank the features according to some relation (or correlation) [12, 18] between the input variables and the target concepts [14, 31, 32]. This approach is intuitive and easy to implement, but it usually fails to consider the relevance of a given feature in the presence of others [8], since most filter methods are univariate [5, 18]. In the second strategy, the algorithms are called wrappers and they access the accuracy of a given model in order to find the best feature subset. Technically, they are the best approach, but they usually have to perform an exhaustive search over all possible subsets of features which is costly and, even for problems with relatively small dimension, unfeasible. The third category of methods refers to the embedded algorithms. As Wrapper methods, embedded ones use the model as part of the selection process, but perform feature selection during the training process [3, 25, 26].

Basically, labels and data are the available sources of information to perform FS. Many methods [12, 13] are able to deal only with labeled data, while others only deal with unlabeled data [15, 16]. However, in many real situations, the amount of labeled data is not sufficient to well characterize the relations between input data and output classes. Since labeling by human experts can be costly, it is common in many kinds of problems to have large unlabeled data sets available and very few labeled data.

For instance, following the wrapper strategy the method proposed in [21] selects features accessing the accuracy of a given model. Like in other approaches, this method is based on the assumption that both labeled and unlabeled data are drawn from the same distribution and in the principle of margin maximization.

In the "filter type" approach, the existent semi-supervised methods are mostly based on the margin maximization principle and make the assumption that all labeled and unlabeled data are i.i.d.. The semi-supervised feature selection via spectral analysis [32], the locality sensitive semi-supervised feature selection method [31], the *FS-manifold* method [29] and the method in [33] apply, somehow, the spectral graph theory, but they differ in how to apply it. As an example, Zhao et al. in [32] developed a semi-supervised feature selection algorithm based on spectral graph theory (spectral analysis). It assumes that if patterns are in the same cluster, they are considered to belong to the same class. The method searches for clusters, trying to maximize both cluster separability and the consistency with labeled information at the same time. This is done by means of cluster indicators of the normalized min-cut method [22] defining good separable cluster structures applying the Graph Laplacian theory. There are also other filter methods, sharing the same basic assumptions, that were developed within the logistic I-RELIEF framework [14] and try to measure each feature discrimination power.

The semi-supervised version of the support vector machines (S^3VM) [30] is another way to perform feature selection. The idea is to use the S^3VM with 1-norm formulation which produces a weight matrix with a lot of zeros. The features whose weights go to zero are not useful and could be discarded. One of the drawbacks of this method is that it does not work very well for bigger datasets, and there are parameters to be set.

Some works also use information theory as we do in our proposed method but in a different way. The supervised method developed in [13] use mutual information as a similarity measure. All steps in [13] were performed only considering labeled data; however, unlabeled data could be used in the clustering step, where labels are not needed at all. It was done in [19] where a semi-supervised feature selection method within the feature clustering approach was developed. The method cluster features using a semi-supervised metric based on the mutual information. It uses the Ward

method [28] to cluster features according to the similarity measure between features X_i and X_j, described as $S(X_i, X_j) = \alpha_1\big(H(X_i/X_j) + H(X_j/X_i)\big) + \alpha_2\big(\mathrm{MI}(X_i, Y/X_j) + \mathrm{MI}(X_j, Y/X_i)\big)$, where H is the entropy, MI is the Mutual Information, X is the input data and Y is the output variable. The first term in this equation is also known by Mantara's distance. After the end of the clustering process, the representative feature of each cluster will be the one with the larger Mutual Information with the output Y.

In [4], the authors use an information theoretic framework similar to information bottleneck to derive a global criterion for feature clustering in order to capture the optimality of word clustering, that is based on the generalized Jensen–Shannon divergence. In order to minimize the objective function, a divisive algorithm is proposed. This algorithm is reminiscent of the k-means algorithm but uses Kullback Leibler divergences instead of squared Euclidean distances. Its divisive algorithm monotonically decreases the objective function value while minimizing "within-cluster divergence" and simultaneously maximizing "between-cluster divergence".

Still, in [24], the authors propose a supervised feature selection method which builds a dissimilarity space using information theoretic measures that can be applied to continuous and discrete data. The metric is the conditional mutual information between features with respect to the output variable that represents the class labels. The algorithm tries to find a compression of the information contained in the original set of features using hierarchical clustering.

2.2 Proposed metric

In this section, a novel similarity index that considers both labeled and unlabeled data, named here S, is presented. The main idea of the S index is to serve not only as feature-to-feature similarity, but also to consider the individual effects each feature has on the output labels. Thus, the general semi-supervised similarity criterion S between two random variables X_1 and X_2 is defined as

$$S(X_1, X_2) = \overbrace{(1-\lambda)A(X_1, X_2)}^{\text{Unsupervised}} - \lambda \overbrace{B(X_1, X_2)}^{\text{Supervised}} \quad (1)$$

with

$$A(X_1, X_2) = \Phi(X_1, X_2)\ , \{X^{(\ell)} \cup X^{(u)}\}$$
$$B(X_1, X_2) = abs(\Phi(X_1, Y) - \Phi(X_2, Y))\ , \{X^{(\ell)}\}$$

where $X^{(\ell)}$ is the labeled data set, $X^{(u)}$ is the unlabeled data set, Φ is a similarity function, *abs* is the absolute value function, Y is the output label vector (for $X^{(\ell)}$ only), and $0 \leq \lambda \leq 1$ is used to balance the contributions of A and B. A corresponds to the unsupervised term in Eq. 1 since it does not consider labels; however, it is based on the union, $X^{(\ell)} \cup X^{(u)}$, of the two input datasets available ($X^{(\ell)} \cup X^{(u)}$). Likewise, B is considered the supervised term since it is calculated according to the relation between input variables and output labels and, of course, considers only the labeled data set$X^{(\ell)}$.

Mutual information (MI) was adopted in this paper to represent function Φ, especially due to its ability to identify nonlinear dependencies between random variables. It should also be noted that other *similarity* functions could be used such as Pearson correlation [18] or the Relief Index [12].

The unsupervised term of S is calculated according to the MI between the pair of features (X_1, X_2) and represents how much information one feature retains in relation to the other. It is maximum when the two features are identical, and null when they are unrelated. The similarity is then proportional to the MI between X_1 and X_2.

The supervised term of S represents the difference between the MI of each one of the random variables X_1 and X_2 and the output variable Y. In case their MI are the same in relation to Y, the supervised term is null.

The similarity between input variables X_1 and X_2 shall depend on their influence on the output labels. If two features are similar with respect to the output variable, then B is small and S is large. Notice that B should be larger if the two features have distinct relevance levels in relation to the output variable.

Many feature clustering methods are based on similarity measures between pairs of input variables, which do not usually consider the influence of individual variables on the output labels.

Feature selection via S index is able to select a group of features that best represents the problem in terms of all joint information available, which includes not only unlabeled but also labeled data.

3 Setting λ

The S index can be seen as the MI between two features, penalized by the difference of their MI in relation to the output variable. The weight of the "supervised" term B in the linear combination that results in S is controlled by the value of λ. Obviously, if $\lambda = 0$ then S corresponds to the usual MI between a pair of features, i.e., similarity index S takes into account only the "unsupervised" term in this case. In such a situation, with $\lambda = 0$ we have that $S > 0$, at least in theory.[1] In fact, choosing the value of λ requires

[1] Due to the numerical constraints of estimators, it may happen that estimation may yield negative values.

that a trade-off controlled by λ is accomplished between the supervised and unsupervised terms.

One extreme situation occurs when exists a pair of two identical features with the same relevance to the output, what leads A to be maximum and $B = 0$. In such a situation, S is *maximum* and equals to A. Here λ makes no difference in the calculation of S. Another extreme situation happens when two features are completely independent to each other, one of them is the same as Y, and the other one is independent of Y. In this case, A is equal to zero and B is maximum, resulting in a minimum, and negative, S index. The value of λ will, of course, determine the minimum value of similarity. The adopted strategy to determine the value of λ is presented in the next section.

3.1 Adopted strategy

As will be explained in details in Sect. 4, the pair of features is ranked by its S value in order to evaluate redundancy and relevance. Depending on the value of λ ranking positions can change. The behavior of S as a function of λ for different pair of features is shown in Fig. 1. Based on this behavior, it is possible to define three main ranges of values that this parameter can assume. There are two ranges of values where, despite changes in S, when increasing λ, the similarity ranking of the feature pairs does not change. For the range $0 \leq \lambda < \lambda_{\text{low}}$ the S value is more related to the unsupervised similarity measure (A term). The term λ_{low} is the value of λ that results on the first change in the ranking order of similarities. On the other hand, in the range $\lambda_{\text{hi}} < \lambda < 1$ there is no change in the similarity ranking, and the index is then more affected by the supervised term B. The term λ_{hi} is the value of λ that results in the final change in the similarity ranking.

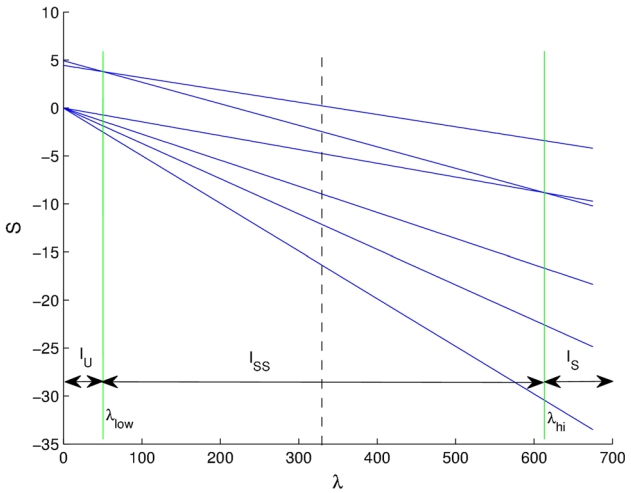

Fig. 1 λ ranges example. Each curve shows the behavior of the S index in function of λ for each pair of features. I_{U} is the preponderantly unsupervised interval, while I_{S} is the mainly supervised interval. I_{SS} is the mainly semi-supervised interval. The first inversion of the S ranking occurs at λ_{low}, and the last change occurs at λ_{hi}. The *dashed line* shows the chosen λ for a more balanced semi-supervised criterion

Nevertheless, intuitively, a more balanced semi-supervised index can be achieved within the range $\lambda_{\text{low}} < \lambda < \lambda_{\text{hi}}$. The idea is to set λ in such way that the S is neither dominated by A nor by B. In this paper, the value of λ was set to the mean value between λ_{low} and λ_{hi}, which yields a more balanced solution between the supervised and unsupervised components of the index. So, it can be defined as the mean value between these two limits, shown by the dashed vertical line in Fig. 1.

The semi-supervised interval I_{ss} can be determined finding the minimum and maximum values of λ that results on the first and last changes in S, as shown in Fig. 1. The possible intersection for each feature pair combination (P_k) is evaluated, and after that the mean value of λ in the interval between the first and the last crossing points is considered as the chosen λ. The formulation is shown in Eqs. 2 and 3, where λ_{ij} is the value where the similarity index for pairs i and j is the same.

$$\lambda_{ij} = \frac{A_j - A_i}{B_j - B_i} \tag{2}$$

and,

$$\lambda_{\text{mean}} = \frac{\min(\lambda_{ij}) + \max(\lambda_{ij})}{2} = \frac{\lambda_{\text{low}} + \lambda_{\text{hi}}}{2} \tag{3}$$

4 Semi-supervised feature selection

Taking into account the properties discussed in the definition of the S index, the proposed method first eliminates redundant features, searching for features sharing a certain amount of information and having similar *relevance* to the output labels. In a second moment, the proposed method searches for those features whose MI in relation to the output variable *means* zero discarding them. In both steps, we will need to assess the reliability of MI estimation, which is explained in Sect. 4.1 before entering into the method description itself. The method is named here SSFC, which stands for semi-supervised feature clustering.

4.1 Significance of S

Once two features are completely unrelated, their estimated mutual information has to be zero in theory. In practice, as we are dealing with mutual information estimators, their outcome may not be properly zero, but a value near zero *meaning* zero. It is, therefore, important to know if a similarity value is really meaningful or if it is only a

number *that means zero* to the estimator. In [6], the authors use the called *permutation test* to evaluate the reliability of the mutual information estimation, and it can be used here in order to assess the significance of index S formulating a statistical test with the following hypothesis:

> Given two random variables X_1 and X_2, let S be the similarity function between them. Let the null hypothesis be that the variables X_1 and X_2 have no similarity. In this case, we aim at evaluating the distribution of S under the null hypothesis of no similarity.

In order to accomplish the significance test, instances from X_1 were permuted n_p times (forming the new permuted feature $X_1{}^p$ while X_2 is kept unchanged. For each permutation, the similarity index S_p between features $X_1{}^p$ and X_2 was calculated and stored. Based on the n_p values, the cumulative distribution function (CDF) of S_p was obtained, and then the 95th percentile ($S_p^{95\text{th}}$) threshold was set in order to assess the significance of S [11]. In case $S > S_p^{95\text{th}}$, then S was considered significant, i.e., the null hypothesis was true.

4.2 First step: eliminating redundancies

In order to eliminate redundancies, SSFC accomplishes hierarchical feature clustering [28] according to the S index, described in Eq. 1. In other words, the method evaluates S for all possible pair of features and cluster them hierarchically from the highest value of S to the lowest. In case the value of S of a pair of features is significant, these two features can be clustered. In fact, one of them will be eliminated from dataset and the general rule for discarding a feature is based on the mutual information between the two features and the output variable Y, as described next:

> If $\text{MI}(X_i, Y) \geq \text{MI}(X_j, Y)$ then X_j is discarded, otherwise X_i is discarded,

where X_i and X_j are features from the considered pair of features.

Redundancy elimination continues iteratively until a stopping criteria is met. At the end, remaining features compose the initial selected feature subset F_s. The permutation test used to evaluate the significance of S as shown in Sect. 4.1 is the chosen stopping criterion to terminate the process of eliminating redundant features. If $S_{i,j}$ is less than or equal to the statistical zero value $S_p^{95\text{th}}$ defined by the permutation test, $S_{i,j}$ has no significant meaning and the evaluation process ends without any further feature elimination. The basic steps to eliminate redundant features are presented in Algorithm 1.

Algorithm 1: Basic algorithm to eliminate redundant features

input : Initial set F with all features
output: Final set F_s with selected features

1 $F_s \leftarrow F$;
2 **repeat**
3 **for** *each possible pair of features* (X_i, X_j) *from* F_s **do**
4 evaluate $S(X_i, X_j)$ using equation 1 ;
5 **end**
6 select the pair of features (X_i, X_j) with the higher S ;
7 apply permutation test and evaluate S_p^{95th} for features (X_i, X_j) ;
8 **if** S_{ij} *is significant* ($S_{ij} > S_p^{95th}$) **then**
9 evaluate MI between feature i and output Y - $MI(X_i, Y)$;
10 evaluate MI between feature j and output Y - $MI(X_j, Y)$;
11 discard from F_s the feature with smaller MI with relation to Y ;
12 **end**
13 **until** *stopping criterion*;

Note that Step 4 in Algorithm 1 is semi-supervised since it deals with labeled and unlabeled data. Step 11, in turn, is supervised because it uses only labeled data.

4.3 Second step: eliminating irrelevant features

Now the objective is to identify any feature that does not contribute to the output variable. MI plays, again, a decisive role in the elimination of irrelevant features. Those features sharing more information with the output vector are maintained, while those not related to the output are discarded. Basically, the MI between each remaining feature in F_s and the output is evaluated. After that the "statistical significance" of the outcome is assessed. The relevance index R_i of feature X_i is estimated as

$R_i = \mathrm{MI}(X_i, Y)$, where Y is the vector of labels, $i = 1, 2, \ldots, \mathcal{N}$ and $\mathcal{N}$ is the number of features of F_s. If R is zero or statistically not significant (once we are dealing with MI estimators), feature is discarded.

In order to evaluate the significance of R, each feature has its elements permuted while the output vector remains unchanged, and then, the R index is evaluated again. This process is repeated n_p times in order to build a cumulative distribution function of the relevance index (R_p) for each feature. A threshold value is set to determine if R is significant or not. The threshold is set as the 95th percentile ($R_p^{95\text{th}}$) of the R_p CDF, meaning that 95% of the observed R_p values are smaller than this value in the CDF distribution. Therefore, if $R > R_p^{95\text{th}}$, then R can be considered as a significant (nonzero) relevance value and feature is not discarded, as described in Algorithm 2.

discriminant analysis (LDA). The first one because it is a nonlinear classifier and second one because it is robust and less sensitive to parameters in experiments with many trials. We also applied two supervised feature selection methods found in the literature (Relief [12] and Random Forest (RF) [10]) to the same training sets of each problem (now only over labeled data). The selected features are used to train the same classifiers, and the results are compared in order to show that the use of unlabeled data in the process can lead to better results. For each percentage of labeled data, the feature selection method was repeated 30 times, with training and test sets randomly selected at each trial.

In order to show that the proposed approach to select λ can achieve good results, experiments for a given data set were accomplished with different values of λ, according to the discussion presented in Sect. 3.

Algorithm 2: Basic algorithm to eliminate irrelevant features

input : Initial set F_s with selected features from Step 1
output: Final set F_s with selected features in Step 2

```
for feature i = 1 to 𝒩 do
    evaluate R_i = MI(X_i, Y) ;
    apply permutation test and evaluate R_{p_i}^{95th} for feature X_i ;
    if R_i is irrelevant (R_i < R_{p_i}^{95th}) then
        discard feature X_i from F_s;
    end
end
```

5 Experiments

Experiments were performed with the following classification problems, all of them from UCI Machine Learning Repository [27]:

- *SONAR*: sonar data set with 208 samples and 60 features;
- *PEN*: handwritten digits data set with 2111 samples and 16 features;
- *KDD*: knowledge discovery data mining competition data set with 600 samples and 41 features;
- *ILPD*: indian liver patients data set with 583 samples and 10 features;
- *IONO*: ionosphere data set with 351 samples and 34 features.

The feature selection method was applied considering different proportions of labeled and unlabeled data: 30 and 70% of the dataset. For each problem, we first perform feature selection with the proposed method and with λ value calculated as proposed in Sect. 3. Considering only the selected features in the training set, two classifiers were trained: multilayer perceptron (MLP) and linear

6 Results

For the IONO problem, we performed the proposed feature selection method for different values of λ including the one proposed in Sect. 3. The results are shown in Tables 1 and 2. For this problem, $\lambda_{\text{mean}} \approx 0.12$. As can be seen, the accuracies achieved by λ_{mean} for different proportions of labeled data and for the different classification methods using the selected feature subsets are the best ones except for the case where $P_l = 70\%$. The best results are marked in bold. There are reports in the literature [23] that show

Table 1 Final number of selected features for IONO problem

p_l (%)	n_s	λ					
		0	0.05	0.1218	0.2	0.5	1
30	$\min n_s$	1	2	4	2	2	4
	$\max n_s$	1	23	22	25	30	17
	$\overline{n}_s$	1	10.4	12	9.4	9.1	9.6
70	$\min n_s$	1	2	4	4	2	2
	$\max n_s$	2	6	12	8	10	15
	$\overline{n}_s$	1.1	4.6	6.9	5.6	5.4	5.5

Table 2 Accuracies achieved by classification models trained considering the final set of selected features for IONO problem

λ	$Acc \pm \sigma$			
	LDA		MLP	
	p_l = 30%	p_l = 70%	p_l = 30%	p_l = 70%
0	0.572 ± 0.078	0.547 ± 0.084	0.8 ± 0.061	0.796 ± 0.054
0.05	0.848 ± 0.051	0.799 ± 0.05	0.862 ± 0.042	0.874 ± 0.05
0.1218	**0.865 ±0.045**	**0.807 ± 0.046**	**0.867 ± 0.048**	0.879 ± 0.051
0.2	0.849 ± 0.056	0.8 ± 0.048	0.856 ± 0.057	0.876 ± 0.05
0.5	0.832 ± 0.058	0.807 ± 0.052	0.859 ± 0.05	0.873 ± 0.055
1	0.852 ± 0.045	0.792 ± 0.055	0.861 ± 0.051	**0.886 ± 0.046**

Bold results are the best ones

accuracy results from 90 to 92% for a classification task using a MLP trained with all 34 features of the IONO problem. SSFC selected from 4 to 12 features on average as can be seen in Table 1.

For the initial set of 41 features of KDD, SSFC resulted on feature set sizes ranging from 12 to 17 features. This result is consistent with the one reported in [17], which ended up with 12 features for the same classes that were considered in this work (normal use and smurf intrusion). Also, as shown in Table 4, all accuracy results were high, showing that the selected features are relevant for the discrimination problem. Tables 3 and 4 also show the outcomes of experiments with different percentages of labeled data. As more label information was available, feature set size and accuracy increased for the KDD dataset. The increase in classification accuracy due to the increase in the percentage of labeled data was expected, since more labeled data are available to induce the classifier. Its relation with the feature set size, however, shall depend of the problem. Nevertheless, it is clear that with a larger number of labeled data, the calculation of feature relevances is more accurate. This will affect the two basic steps of the proposed method as follows:

- in the first phase, when redundancies are eliminated, the B term of Eq. 1 affects the S index of each pair of features, which has a direct impact in the feature similarity rank and on the cluster sizes;
- in the second phase, when irrelevant features are discarded, ranking is also affected, since as more labeled data is included, data set distribution changes gradually and affects feature ranking.

The proposed SSFC method selected between 2 and 4 features from the initial set of 10 features in the ILPD problem, as shown in Table 3. Table 4 lists accuracies achieved by LDA and MLP classifiers. Due to the nonlinear nature of the problem, MLP performed much better when trained with the selected feature subset. Feature set reduction was more effective than reported in the literature for the same data set; however, the data set used in this work has less features than the one reported in [20]. Most relevant features reported in [20] were: total bilirubin, direct bilirubin, indirect bilirubin, albumin. With 70% less

Table 3 Final number of selected features

Problems	p_l (%)	n_s	Methods		
			SSFC	Relief	RF
IONO	30	min n_s	4	1	1
		max n_s	22	33	33
		$\overline{n}_s$	12	25.5	16.4
	70	min n_s	4	33	1
		max n_s	12	33	33
		$\overline{n}_s$	6.9	33	30.9
ILPD	30	min n_s	1	1	1
		max n_s	2	9	9
		$\overline{n}_s$	1.9	2.8	4.3
	70	min n_s	1	1	1
		max n_s	4	4	9
		$\overline{n}_s$	2.1	1.5	4.2
KDD	30	min n_s	12	1	1
		max n_s	13	16	3
		$\overline{n}_s$	12.7	4.7	1.9
	70	min n_s	16	1	1
		max n_s	17	15	2
		$\overline{n}_s$	16.7	3.8	1.1
SONAR	30	min n_s	4	1	1
		max n_s	6	59	6
		$\overline{n}_s$	5.3	19.4	2.2
	70	min n_s	9	1	1
		max n_s	13	9	59
		$\overline{n}_s$	11.6	2.9	4.2
PEN	30	min n_s	10	1	1
		max n_s	13	2	2
		$\overline{n}_s$	12.2	1.1	1.4
	70	min n_s	11	1	1
		max n_s	13	1	1
		$\overline{n}_s$	12.4	1	1

Table 4 Accuracies achieved by classification models trained considering the final set of selected features

Problem	Method	$Acc \pm \sigma$			
		LDA		MLP	
		$p_l = 30\%$	$p_l = 70\%$	$p_l = 30\%$	$p_l = 70\%$
IONO	SSFC	0.865 ± 0.045	0.807 ± 0.046	0.867 ± 0.048	0.879 ± 0.051
	Relief	0.810 ± 0.037	0.867 ± 0.030	0.855 ± 0.051	0.917 ± 0.027
	RF	0.811 ± 0.045	0.864 ± 0.034	0.837 ± 0.053	0.901 ± 0.040
ILPD	SSFC	0.517 ± 0.068	0.524 ± 0.054	0.701 ± 0.025	0.706 ± 0.026
	Relief	0.710 ± 0.011	0.717 ± 0.027	0.713 ± 0.017	0.718 ± 0.029
	RF	0.711 ± 0.015	0.712 ± 0.028	0.713 ± 0.015	0.701 ± 0.081
KDD	SSFC	0.998 ± 0.005	0.995 ± 0.008	0.997 ± 0.007	0.997 ± 0.006
	Relief	0.994 ± 0.004	0.978 ± 0.094	0.739 ± 0.253	0.654 ± 0.218
	RF	0.994 ± 0.003	0.991 ± 0.006	0.718 ± 0.262	0.562 ± 0.183
SONAR	SSFC	0.737 ± 0.107	0.780 ± 0.071	0.719 ± 0.108	0.722 ± 0.091
	Relief	0.649 ± 0.080	0.732 ± 0.048	0.661 ± 0.084	0.735 ± 0.065
	RF	0.678 ± 0.061	0.714 ± 0.046	0.634 ± 0.091	0.738 ± 0.048
PEN	SSFC	1.000 ± 0.001	1.000 ± 0.001	0.999 ± 0.002	0.999 ± 0.002
	Relief	0.999 ± 0.001	0.399 ± 0.498	0.585 ± 0.240	0.209 ± 0.270
	RF	0.999 ± 0.001	0.999 ± 0.001	0.699 ± 0.253	0.524 ± 0.094

data samples, the method proposed in this work selected the following features: total bilirubin, direct bilirubin, sgot alamine aminotransferase, sgot aspartate aminotransferase. The last two features selected by SSFC are classified as 5th and 6th relevant features in [20]; however, *indirect bilirubin* is not present in the ILPD data set used in our experiments. The data set used in [20] contains 12 features, while ours contains only 10 features. Feature *albumin* was initially selected by the first step of our method; however, it was discarded since it had lower relevance in relation to the output variable. As can be seen, the MLP classifiers trained with features selected by SSFC achieved almost the same accuracy when compared with results for Relief and RF methods in this problem. As can be seen in Table 3, on average, with 30% of labeled instances SSFC selects 2 features and with 70% of data the number of selected features increased to approximately 4. For higher proportions of labeled data, there is no relevant difference in the final number of selected features. Nevertheless, the accuracy on the test set is not substantially affected by the increase in the data set size. This problem has 583 instances with 10 dimensions with two unbalanced classes. Data were randomly selected for training in order to maintain the original class proportions.

For the SONAR problem, the SSFC method selects between 4 and 13 features in a set of 60 original features. This problem has much less data than the total number of features, which may result in a higher sensitivity to the curse of dimensionality problem. In [7], the authors achieved 81% accuracy on average, for different numbers of hidden neurons in a MLP network, considering all 60 features. LDA yielded 78% accuracy using feature subsets composed of about 10 features, chosen when considering 70% of labeled data. Also, results achieved by SSFC are better than those achieved by Relief and RF except for the MLP with $P_l = 70\%$.

For the PEN problem, SSFC was able to select, on average, 12 features from the original set containing 16 features and for different proportions of labeled data, as can be seen in Table 3. In all experiments, both LDA and MLP achieved up to 99% accuracy on average (see Table 4). In [1], the authors achieved, according to the network configuration, from 92 to 98% accuracy on a test set, using MLP and considering all 16 features. Since SSFC achieved at least 98% accuracy on average, using about 12 features we can conclude that the discarded features are irrelevant, and, of course, SSFC was able to identify the most relevant features and performed much better than Relief and RF.

For the IONO problem, as can be seen in Table 3 the SSFC method select, in average, less features than supervised methods Relief and RF. We also can note in Table 4 that for $P_l = 30\%$, features selected by SSFC resulted in better accuracies than Relief and RF. In other words, with less labeled data, supervised methods loss accuracy, while SSFC do not in this problem.

7 Conclusions

This work presented a new method for feature selection that is capable of considering sources of information from labeled as well as from unlabeled data. In this semi-supervised feature selection framework, the method is based on the idea of eliminating redundancy by feature

clustering. The method adopts a novel semi-supervised approach, since labeled and unlabeled data are taken into account in the new similarity index, which is also proposed in this work. SSFC can be directly applied to multiple variables by incorporating them to the MI estimation. Stopping criterion for feature clustering can also incorporate further overall performance strategies, since it is based only on the significance level of *S*. The method, however, achieved competitive results with less features then previous works in the literature with the same data sets. It is interesting to highlight that the proposed method performed well even with a small number of labeled data.

Acknowledgements This work was developed with financial support of CAPES and CNPq from Brazil.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Alimoglu F, Alpaydin E (1996) Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In: Proceedings of the fifth Turkish artificial intelligence and artificial neural networks symposium (TAINN 96)
2. Cover TM, Thomas JA (1991) Elements of information theory. Wiley, New York
3. Cun YL, Denker JS, Solla SA (1990) Optimal brain damage. In: Advances in neural information processing systems. Morgan Kaufmann, pp 598–605
4. Dhillon IS, Mallela S, Kumar R (2003) A divisive information theoretic feature clustering algorithm for text classification. J Mach Learn Res 3:1265–1287. http://dl.acm.org/citation.cfm?id=944919.944973
5. Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7:179–188
6. François D, Wertz V, Verleysen M (2006) The permutation test for feature selection by mutual information. In: ESANN 2006, European symposium on artificial neural networks, pp 239–244
7. Gorman PR, Sejnowski TJ (1988) Analysis of hidden units in a layered network trained to classify sonar targets. Neural Netw 1(1):75–89
8. Guyon I, Gunn S, Nikravesh M, Zadeh LA (2006) Feature extraction: foundations and applications (studies in fuzziness and soft computing). Springer, Secaucus. http://portal.acm.org/citation.cfm?id=1208773
9. Hanchuan Peng CD, Long F (2005) Minimum redundancy maximum relevance feature selection. IEEE Intell Syst 20:70–71
10. Feature selection for intrusion detection using random forest (2016) Hasan M., N.M.A.S., Molla, K. J Inf Sec 7:129–140
11. Kay SM (2006) Intuitive probability and random processes using MATLAB. Springer, Berlin
12. Kira K, (1992) Rendell LA A practical approach to feature selection. In: ML92: proceedings of the ninth international workshop on Machine learning, pp 249–256. Morgan Kaufmann Publishers Inc., San Francisco. http://portal.acm.org/citation.cfm?id=142034
13. Krier C, François D, Rossi F, Verleysen M (2007) Feature clustering and mutual information for the selection of variables in spectral data. In: Neural networks, pp 25–27
14. Li J (2008) Semi-supervised feature selection under logistic I-RELIEF framework. In: 2008 19th international conference on pattern recognition pp 1–4. doi:10.1109/ICPR.2008.4761687. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4761687
15. Llobet E, Gualdron O, Brezmes J, Vilanova X, Correig X (2005) An unsupervised dimensionality-reduction technique. In: Sensors, 2005. IEEE
16. Mitra P, Murthy CA, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 24(3):301–312. doi:10.1109/34.990133
17. Olusola AA, Oladele AS, Abosede, DO (2010) Analysis of kdd 99 intrusion detection dataset for selection of relevance features
18. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C: the art of scientific computing, vol 2. Cambridge University Press, New York
19. Quinzan, I., Sotoca, J.M., Pla, F.: Clustering-based feature selection in semi-supervised problems. In: International Conference on Intelligent Systems Design and Applications, vol 0, pp 535–540 (2009). doi:10.1109/ISDA.2009.211
20. Ramana BV, Babu MP, Venkateswarlu NB (2011) A critical study of selected classification algorithms for liver disease diagnosis. Int J Database Manage Syst 3:101–114
21. Ren J, Qiu Z, Fan W, Cheng H, Yu PS (2008) Forward semi-supervised feature selection. In: PAKDD'08: proceedings of the 12th Pacific-Asia conference on advances in knowledge discovery and data mining. Springer, Berlin, pp 970–976
22. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905. http://citeseer.ist.psu.edu/shi97normalized.html
23. Sigillito VG, Wing SP, Hutton LV, Baker KB (1989) Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Tech Dig 10(3):262–266
24. Sotoca JM, Pla F (2010) Supervised feature selection by clustering using conditional mutual information-based distances. Pattern Recognit 43(6):2068–2081. doi:10.1016/j.patcog.2009.12.013. http://www.sciencedirect.com/science/article/pii/S0031320309004828
25. Stoppiglia H, Dreyfus G, Dubois R, Oussar Y (2003) Ranking a random feature for variable and feature selection. J Mach Learn Res 3:1399–1414
26. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc B 58:267–288. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.7574
27. Uci machine learning repository. http://archive.ics.uci.edu/ml/
28. Ward JH (1963) Hierarchical grouping to optimize an objective function. J Am Stat Assoc 58(301):236–244. doi:10.2307/2282967
29. Xu Z, Jin R, Lyu MR (2009) King I discriminative semi-supervised feature selection via manifold regularization. In: IJCAI'09: proceedings of the 21st international joint conference on artificial intelligence. Morgan Kaufmann Publishers Inc., San Francisco, pp 1303–1308
30. Yang L, Wang L (2007) Simultaneous feature selection and classification via semi-supervised models. International conference on natural computation 1:646–650. doi:10.1109/ICNC.2007.666
31. Zhao J, Lu K, He X (2008) Locality sensitive semi-supervised feature selection. Neurocomputing 71(10–12):1842–1849. doi:10.1016/j.neucom.2007.06.014
32. Zhao Z, Liu H (2007) Semi-supervised feature selection via spectral analysis. In: SDM
33. Zhong E, Xie S, Fan W, Ren J, Peng J, Zhang K (2008) Graph-based iterative hybrid feature selection. In: IEEE international conference on data mining 1133–1138. doi:10.1109/ICDM.2008.63