Journal of Intelligent & Fuzzy Systems 31 (2016) 927–937
DOI:10.3233/JIFS-169022
IOS Press

A clustering-based feature selection via feature separability

Shengyi Jiang[a,c] and Lianxi Wang[b,*]
[a]*School of Informatics, Guangdong University of Foreign Studies, Guangzhou, China*
[b]*School of Information Management, Sun Yat-Sen University, Guangzhou, China*
[c]*Laboratory of Language Engineering and Computing, Guangzhou, China*

Abstract. With the extensive increase of the amount of data, such as text categorization, genomic microarray data, bioinformatics and digital images, there are more and more challenges in feature selection. Recently, feature selection has been widely studied in supervised learning, but there is significantly less work in unsupervised learning because of the absence of class information and explicit search criteria. In this work, we introduce a new measure to assess the importance of features in terms of feature separability. A clustering-based feature selection algorithm is then introduced to conduct the feature selection. The proposed algorithm with nearly linear time complexity selects final feature subset through a ranking procedure based on the separabilities of features and it is applicable to datasets of mixed nature. Experimental results on UCI datasets show that our method, by retaining relevant features, can obtain similar or even better results of classification and clustering for most datasets, and it outperforms other traditional supervised and unsupervised feature selection methods in terms of dimensionality reduction and classification accuracy.

Keywords: Feature selection, feature separability, clustering, unsupervised learning

1. Introduction

In many real-world fields, such as genomic microarray analysis, digital images recognition, text classification and uncertain data management, it is not uncommon to have data sets with hundreds of thousands of features [1, 2]. Feature selection has become an important way to reduce the dimensionality, but it is still considered as an intractable problem in machine learning and data mining. High dimensional data may largely degrade the learning performance and compromise the quality of clustering. Therefore, to retain important features and remove irrelevant and redundant ones, various robust and effective feature selection algorithms have been introduced [3–33]. Generally, feature selection methods can be classified into two groups according to the evaluation criterion, i.e., filter methods [4–17] and wrapper methods [31–33]. The filter methods evaluate the quality of features by using the intrinsic properties of the data and are independent of any learning algorithm. Many existing feature relevance measures are designed to evaluate the quality of the candidate subsets, such as, distance [6–8], correlation [9–12], consistency [13], and rough set [14, 15]. Distance is known as separability, divergence, or discrimination measure [6, 7, 16, 17]. Correlation is widely used to measure the relevance between features and class feature [9, 10]. Consistency is treated as the ratio of the samples that can be used to evaluate the quality of categorical features [13]. On the contrary, the wrapper methods select features based on the results of learning algorithms. In general, the filter methods are generally much faster than the wrapper methods. In this paper, we will focus on the filter methods.

*Corresponding author. Lian-Xi Wang, School of Information Management, Sun Yat-Sen University, Guangzhou 510006, China. E-mail: wanglianxi2012@163.com.

Feature selection has been widely studied in supervised learning, but it is rarely studied in unsupervised learning. It is believed that unsupervised feature selection is less effective than supervised feature selection because of the absence of class information and explicit search criteria. It is not until very recently that several algorithms have been developed to address these issues for clustering. Briefly, there are two fundamentally different approaches for feature selection based on clustering: feature clustering [18–25] and object clustering [26–29]. The former approach removes redundant features by partitioning the original feature set into distinct clusters formed by similar features. After the features are grouped into clusters, such filters select a subset of features from each cluster, discarding other less discriminative ones. Covões et al. [18] and Mitra et al. [19] defined some new indices to measure feature similarity so that feature redundancy is detected. Although they obtained promising results, both of them still require data with continuous features only. Li et al. [22] proposed a localized feature selection algorithm for clustering, which computes adjusted and normalized scatter separability for individual clusters. However, the algorithm must simultaneously compute the number of clusters and the local feature subset. Recently, Zeng et al. [24] presented a new feature selection approach for Gaussian mixture clustering, in which a new feature relevance measurement index was introduced to identify the most relevant features.

The object clustering approach first partitioned a set of objects into clusters by a certain distance measurement, so that objects in the same cluster are more similar to each other than objects in different clusters. Then it selected discriminate features based on the importance of the features according to some defined criteria. Modha et al. [27] introduced a new method for feature weighting in k-means clustering based on intra-cluster and inter-cluster matrices. However, finding optimal weights from a predefined set of variable weights may not guarantee that the predefined set of weights would contain the optimal weights. Wang et al. [28] proposed a feature-weight learning approach to improve the performance of Fuzzy C-Means based on a defined similarity measurement and an evaluation function. However, they are complicated and difficult to interpret. Huang et al. [29] presented W-k-means algorithm that can calculate feature weights automatically. Based on the partition in the iterative k-means clustering process, the algorithm calculates a new weight for each feature based on the variance of the intra-cluster distances. The new weights are used to decide the cluster memberships of objects in the next iteration, and the optimal weights are found when the algorithm converges. This approach is dependent on the selection of k value. Unfortunately, many clustering algorithms, such as K-means, FCM, W-k-Means and so on, are only applicable to one type of features.

As far as continuous and categorical features are concerned, many approaches in supervised learning, such as rough set [14], neighborhood rough set [15] and correlation [4, 9], are employed to solve the mixture problems. However, there is very little work that can handle the mixed-type features for feature selection with object clustering in unsupervised learning. To solve this issue, we explore a feature selection method based on single-pass clustering algorithm that is applicable to categorical and continuous features by partitioning the similar objects into the same clusters. The proposed algorithm selects features according to variation of feature separability values. The final feature subset is determined by several iterative clustering results. Experimental comparisons with other supervised and unsupervised filter feature selection methods on UCI data of different dimensionalities shows that the proposed algorithm can obtain competitive results in terms of dimensionality reduction and classification accuracy.

This paper is organized as follows. Section 2 describes some basic concepts, and Section 3 presents the proposed feature selection approach based on single-pass clustering. The experimental results are discussed in Section 4. In Section 5, we conclude this paper by pointing out some issues for future research.

2. Basic concept

In this section, we begin with a brief introduction to some preliminaries in Section 2.1, and then describe single-pass clustering algorithm in Section 2.2.

2.1. Preliminaries

We suppose that dataset D consists of N instances, each instance with m features (m_C categorical and m_N continuous) where $F_i(1 \leq i \leq m)$ denotes the *i-th* feature. To reduce the effect of various measurement units, it is necessary to standardize numerical features.

Definition 1. For a cluster C and a feature value $a_i \in F_i$, the frequency of a in C with respect

to F_i is defined as: $Freq_{C|F_i}(a_i) = |\{object|object \in C, object.F_i = a_i\}|$.

Definition 2. For a cluster C, the cluster summary information (CSI) is defined as: $CSI = \{n, Summary\}$, where n is the size of the cluster $C\,(n = |C|)$, Summary is given as the frequency information for categorical feature values and the centroids for numerical features: $Summary = \{< Stat_i, Cen > |Stat_i = \{(a, Freq_{c|F_i}(a))|a \in F_i\},\ 1 \le i \le m_C, Cen = (C_{m_C+1}, C_{m_C+2}, \ldots, C_{m_C+m_N})$.

Definition 3. For a subcluster C of dataset D, let $p = (p_1, p_2, \ldots, p_m)$, and $q = (q_1, q_2, \ldots, q_m)$.

(1) The difference or distance between objects p and q on F_i is named as $dif(p_i, q_i)$. For categorical features, $dif(p_i, q_i) = \begin{cases} 1 & p_i \neq q_i \\ 0 & p_i = q_i \end{cases} = 1 - \begin{cases} 0 & p_i \neq q_i \\ 1 & p_i = q_i \end{cases}$, For numerical features, $dif(p_i, q_i) = |p_i - q_i|$.

(2) The distance between objects p and q, $d(p, q)$ is defined as $d(p, q) = \sqrt{\sum_{i=1}^{m} dif(p_i, q_i)^2/m}$.

(3) The distance between object p and cluster C, $d(p, C)$ is defined as $d(p, C) = \sqrt{\sum_{i=1}^{m} dif(p_i, C_i)^2/m}$, where C_i is the summary of cluster C on feature F_i, $dif(p_i, C_i)$ is the distance between object p and cluster C on feature F_i. For categorical features, $dif(p_i, C_i)$ is defined as the average difference between p and every object in C on F_i, that is, $dif(p_i, C_i) = 1 - Freq_{C|F_i}(p_i)/|C|$, while for numerical features $dif(p_i, C_i)$ is defined as $dif(p_i, C_i) = |p_i - c_i|$.

(4) The distance between clusters C_1 and C_2, $d(C_1, C_2)$ is defined as $d(C_1, C_2) = \sqrt{\sum_{i=1}^{m} dif_i(C_1, C_2)^2/m}$, where $dif_i(C_1, C_2)$ is the difference between C_1 and C_2 on feature F_i. For categorical features, $dif_i(C_1, C_2)$ is the average difference among any object p in C_1 and object q in C_2 and defined as $dif_i(C_1, C_2) = 1 - \frac{1}{|C_1|\cdot|C_2|}\sum_{P\in C_1} Freq_{C_1|F_i}(p_i) \cdot Freq_{C_2|F_i}(p_i) = 1 - \frac{1}{|C_1|\cdot|C_2|}\sum_{q\in C_2} Freq_{C_1|F_i}(q_i) \cdot Freq_{C_2|F_i}(q_i)$, while for the numerical feature, $dif_i(C_1, C_2)$ is defined as $dif_i(C_1, C_2) = \left|c_i^{(1)} - c_i^{(2)}\right|$.

2.2. Single-pass clustering algorithm

Before giving the feature selection algorithm, we first introduce the procedure of single-pass clustering algorithm [30]. The single-pass clustering algorithm employs the least distance principle to divide dataset. The single-pass clustering algorithm is described as follows:

1) Initialize the set of clusters S, as the empty set, and read a new object p.
2) Create a cluster with the object p.
3) If no object is left in the database, go to *(6)*, otherwise read a new object p, and find the cluster C_j in S which is closest to the object p. Namely, find a cluster C_j in S, such that $d(p, C_j) \le d(p, \overline{C})$ for all $\overline{C}$ in S.
4) If $d(P, C_j) > r$, go to *(2)*.
5) Merge object p into cluster C_j and modify the CSI of cluster C_j, go to *(3)*.
6) Stop.

The threshold r may influence the quality of clustering and the time-efficiency of the algorithm. As r decreases, the number of produced clusters increases. On the contrary, if r is large enough, we can obtain only a small number of clusters. Therefore, in order to gain meaningful clustering results, we have to choose a proper threshold r. According to the process of clustering, threshold r should be typically greater than the inter-cluster distance and smaller than the intra-cluster distance. Hence, we can logically assume that r should be close to the average distance of any pair of objects. Because of the large dataset, we adopt sampling techniques to develop our strategy for determining the proper threshold. The details are described as follows:

a) Choose randomly N_0 pairs of objects in the dataset D.
b) Compute the distances between each pair of objects.
c) Compute the EX which is equal to the average of distances from (b).
d) Select r in $[0.5 * EX,\ EX]$.

3. Feature selection algorithm

In this section, we first propose a definition of feature separability for measuring the importance of features and a new feature selection algorithm based on the single-pass clustering algorithm. Then we analyze the time complexity.

3.1. Feature separability measure and feature selection algorithm

Now we discuss how to evaluate the importance of features for unsupervised data. A feature is

discriminative if it can separate the clusters well. The feature values of a good feature should be partitioned into groups of the same or similar values. Each group, called cluster, consists of values that are similar to each other and dissimilar to feature values of other groups. In other words, the good features are able to distinguish the clusters and should have greater separability values. Based on these considerations, we introduce a new definition to determine the important of features according to feature separability value.

Definition 4. Let $C = \{C_1, C_2, \ldots, C_k\}$ be the results of clustering on dataset D, $D = \bigcup_{i=1}^{k} C_i (C_i \bigcap C_l = \Phi, i \neq l)$. The separability values on the feature F_i is defined as: $f_i = \sum_{l=1}^{k} \sum_{t=1}^{k} dif_i(C_l, C_t)$, $(t \neq l <= k)$, where $dif_i(C_l, C_t)$ is the difference between cluster C_l and C_t on feature F_i.

Definition 4 describes the separability of a feature among the clusters. Therefore, it can be used to measure the importance of features.

Based on the aforementioned definitions and the single-pass clustering algorithm, a new feature selection algorithm, named $UFS - SC$ (Unsupervised Feature Selection Based on Single-pass Clustering) is formulated. $UFS - SC$ employs feature separability to measure the importance for the features. The whole procedure of the proposed algorithm is described in Fig. 1.

Inputs: Dataset D with a full feature set F, $F = \{F_1, F_2, \mathrm{L}, F_m\}$

Outputs: Selected feature subset S

1.(Initialization) Set $F \leftarrow$ "initial set of m features";
$S \leftarrow$ "empty set";

2.(Pre-computation) Cluster the dataset D based on the single-pass clustering;
For $\forall F_i \in F$ compute f_i;

3.(Choice of the threshold)
(a) Rank F_i in descending order according to f_i;
(b) Find the feature F_i that maximizes $f_i \big/ f_{i+1}$;

4.(Feature selection) Choose the feature F_j, where $j \geq i$; set $S \leftarrow S \cup \{F_j\}$;

5.(Output) Output the selected feature subset S;

Fig. 1. UFS-SC algorithm.

In the implementation of our algorithm, we observe experiments converge in fewer than 10 iterations, and typically within 3 to 5 iterations.

3.2. Time complexity analysis

To simplify the analysis, we start with the assumption that the final number of cluster summary information (CSI) is k in s $(3 \leq s \leq 5)$ iterations and each categorical feature consist of distinct values n_i. We analyze the time complexity for each step of the proposed algorithm. Step 1 of setting model depends on the size of the data set (N) and the number of features (m). In the worst case, the time complexity of step 1.1 is $O(n \cdot k(\sum_{i=1}^{m_C} n_i + m_N))$. In practice, the time complexity can be expected to be $O(n \cdot k \cdot m)$. Step 1.2 computes the feature separability values between any pair of clusters, and its time complexity is $O\left(k^2 \cdot \left(\sum_{i=1}^{m_C} n_i + m_N\right)\right)$. Since we have k << n, in the worst case, the time complexity of step 1 is $O\left(n \cdot k \cdot \left(\sum_{i=1}^{m_C} n_i + m_N\right)\right)$, which approximates to $O(n \cdot k \cdot m)$. The time complexity in step 2 is $O(m \log(m))$ due to use of quick sort method, and linear time is $O(m)$ in step 3.

The overall computational complexity of the algorithm is estimated as $O\left(s \cdot n \cdot k \left(\sum_{i=1}^{m_C} n_i + m_N\right)\right)$. The time complexity for each stage is nearly linear with the size of dataset, the number of features and the final number of clusters. This implies that our algorithm has good scalability and it can be used in feature selection for higher dimensional data.

4. Experiments

The evaluation of performance of feature selection algorithms includes three aspects: 1) the degree of dimensionality reduction; 2) accuracy; and 3) efficiency. However, it is inconvenient to compare efficiency because of the variation of experiment environment. In this section, we use experimental results to evaluate the effectiveness of the proposed feature selection algorithm with dimensionality reduction, classification and clustering accuracy.

4.1. Experimental design

To study the effect of the proposed algorithm, we perform the experiments on Wine data available from UCI database. Wine data set consists of 178 instances, each instance with thirteen continuous features.

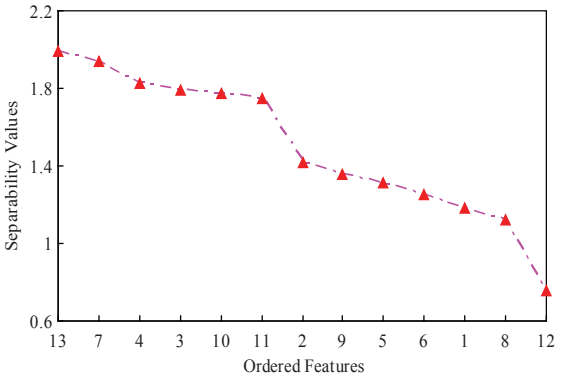

Fig. 2. Features are ranked in descending order by separability values.

The results with the proposed algorithm are shown in Fig. 2. After calculating the separability values of all features, we draw a figure with descending order according to their separability values. The ordered feature set is 13, 4, 3, 10, 11, 7, 2, 9, 5, 6, 1, 8, 12. From Fig. 2, a further observation is that the separability values of the features have the greatest descent point when ranking them in descending order. In this scenario, the greatest descent point which has the largest inclined rate is considered as the threshold. That is, the features of the first parts in which separability values are higher than the threshold are relevant features. Therefore, we choose the first six features (13, 7, 4, 3, 10, 11) as the goodness feature subsets on Wine data.

4.2. Experimental setup

All the experiments with eighteen datasets from UCI machine learning repository [34] are implemented by using Weka [35].

For each data set, we first run all the feature selection algorithms to obtain the newly selected features of each algorithm. The different feature selection methods have been validated using three different classifiers, i.e. C4.5, Ripper and Naïve Bayes. It should be noted that the nature of the decision and the learning process of each classifier are different. Thus, we are interested in checking the goodness of the subsets of selected features, independently from the type of classification rule applied.

Table 1 describes the summaries of each dataset, including the number of instances (Instances), continuous features (Con.), categorical features (Cat.) and classes (Classes). From the table, we can see that the datasets contain different number of instances, features and classes.

Table 1
Summary of bench-mark datasets

Dataset	Cat./Con.	Instances	Classes
Austra	8/6	690	2
Breast	0/9	699	2
Chess	36/0	3196	2
Credit	9/6	690	2
DNA	60/0	3190	3
German	13/7	1000	2
Heart	7/6	270	2
Horse-colic	19/7	300	2
Ionosphere	0/34	351	2
Iris	0/4	150	3
Liver	0/6	345	2
Mushroom	22/0	8124	2
Pima	0/8	768	2
Sonar	0/60	208	2
Spambase	0/57	4601	2
Wine	0/13	178	3
Vehicle	0/18	846	4
Vote	16/0	435	2

4.3. The compared methods

In order to assess the performance of the proposed method, an experimental comparison has been done with respect to other supervised and unsupervised feature selection methods. A brief description of the methods is as follows:

- CFS method [9]: The correlation feature subset method is a filter feature selection approach. It finds an optimal set of features by removing both irrelevant and redundant features and can be applicable to datasets of mixed nature.
- Consistency algorithm [13]: The study of consistency measure is capable of handling some noise and can be used to remove redundant and/or irrelevant features. In fact, this measure does not incorporate any search bias with regards to a particular classifier.
- FCBF algorithm [10]: Fast Correlation Based Filter determines feature subset by correlation measure. It can identify relevant features as well as redundancy among relevant features without pair-wise correlation analysis.
- FarVPKNN algorithm [14]: K-nearest-neighbour relation is computed with continuous features, and categorical features are computed as FarVPDN.
- NDEM algorithm [15]: Neighbourhood decision error minimization is defined and computed

decision positive regions and decision boundary in metric space by neighbourhood rough set.
– SSF algorithm [18] and MMP algorithm [19]: The two methods select feature subset by the feature similarity measure for continuous features in unsupervised learning.

Among them, the first five algorithms are supervised feature selection algorithms. *CFS*, *FarVPKNN*, and *NDEM* can deal with data with mixed types, while Consistency and FCBF algorithms require every continuous feature to be discretized. In the experiments, we keep special search strategy for each algorithm in the experiments.

4.4. Performance of dimensionality reduction

The selected features with different feature selection algorithms, such as *FCBF*, *NDEM*, *FarVPKNN* and *UFS-SC*, are presented on the order of selecting in Table 2. From the table, we can find that the selected features are distinct when different algorithms are applied.

Figure 3 visualizes a change of classification accuracy with respect to the number of selected features for four data sets. As shown in the figure, in most cases, the performance of our method is significantly better than those of *FCBF*, *FarVPKNN*, and *NDEM*. Generally, increasing the number of features greatly improves the accuracy on Heart and Sonar datasets by the proposed algorithm. Except for our method, the other algorithms climb slowly with increasing number of features in the selection process, and sometimes even drop rapidly. This phenomenon occurs because there are many redundant or noisy features in the feature sets.

4.5. Classification and clustering performance on selected features

Table 3 records the number of features selected by each feature selection algorithm and compares the size of selected feature subset of *UFS-SC* with those of supervised feature selection algorithms (*CFS*, *Consistency* and *FCBF*). The average dimensionality reduction of the datasets shows that all the methods are able to remove nearly 3/5 of the original features. From the average dimensionality reduction of all the datasets, we conclude that our method achieves promising results and its performance of dimensionality reduction is close to *FCBF*.

In the experiments, each dataset is split into the training set and test set (2/3 for training and the rest for testing). Each dataset is tested by 10 times for randomly shuffling to make sure that the results are not biased by the data sequences.

Tables 4–6 present the learning error rates of C4.5, Ripper and Naive Bayes respectively on different feature sets. From the averaged error rates of all datasets, we observe that, in general, (1) the proposed algorithm *UFS-SC* improves the accuracy on three learning algorithms compared with other methods on all the data sets; (2) almost all the feature selection algorithms classified on the newly selected features can obtain similar or even less classification error rates than that on the full set.

It is obvious that *UFS-SC* can maintain or improve the accuracy for most of the data sets according to the individual error rate value. To summarize, *UFS-SC* provides competitive results to other feature selection algorithms, especially when the cardinalities of the feature subsets are taken into account.

Three imbalanced data sets are selected from Table 1 to compare with the Naïve Bayes accuracy of *UFS-SC*, *SSF*, *MMP* and full sets. Table 7 summarizes the newly selected features of each unsupervised feature selection algorithm (In Tables 7 and 8, the results of *MMP* and *SSF* reported here are the best on the three imbalanced datasets).

Table 8 shows the accuracy of Naïve Bayes learning algorithm on the final feature subset with 10-fold cross validation. The results on *UFS-SC* and full features are also tested 10 times. The acronyms Max, Min and Aver refer to the maximum, minimum and average value of the ten results with *UFS-SC* respectively.

Compared with *MMP* and *SSF*, *UFS-SC* has two advantages. Firstly, the computational complexity of the developed algorithm is less than those of *MMP* and *SSF*, when the number of features is less than the number of instances in the datasets. More precisely, the time complexities of *MMP* and *SSF* are estimated at $O(m^2 \cdot n)$, and the computational complexity of *UFS-SC* is $O\left(s \cdot n \cdot k \left(\sum_{i=1}^{m_C} n_i + m_N\right)\right)$. Secondly, from Tables 7 and 8, we also observe that UFS-SC not only achieves higher degree of the dimensionality reduction than MMP, but also obtains similar classification accuracy. At the same time, although the performance of dimensionality reduction of the proposed algorithm is less effective than SSF, it has better performance in classification.

From the comparisons, another interesting point deserves our attention is that the classification

Table 2
Features that were sequentially selected when using different algorithms

Data	FCBF	FarVPKNN	NDEM	UFS-SC
Credit	9,11,15,8,6,14,4	3	9,13,6,10,7,1,2,12,3,4,8	9,10,7,2,13
Heart	13,12,3,9,10,8,7	13,5,10,12,3,1,4	13,12,3,11,7,4,1,8,2,5,6	13,6, 9,1,7,12,4
Sonar	11,48,44,51,54,28,36,21,4,5	11,12,49, 15,1,4	11,16,25,22,47,32,35,53,30,1,10	5,6,29,28,11,15,27,53,31,36
Wine	7,12,10,13,1,11,2,5,4,3	10,13,7,6,2,1	7,1,11,13,3,10	13,7,4,3,10,11

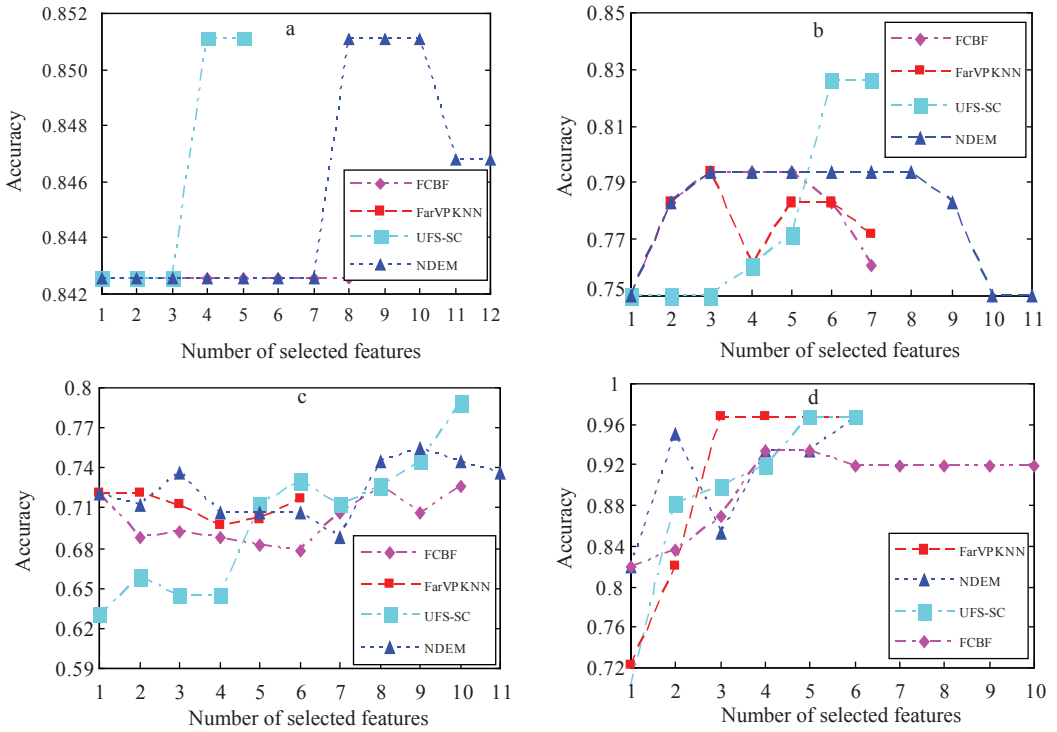

Fig. 3. C4.5 classification accuracies vs. different number of selected features on four data sets: (a) Credit; (b) Heart; (c) Sonar; (d) Wine.

Table 3
Number of feature selected on UCI datasets

Dataset	Full set	CFS	Consistency	FCBF	UFS-SC
Austra	12	7	13	7	3
Breast	9	9	7	8	4
Chess	36	7	6	7	18
Credit	14	7	13	6	5
DNA	60	22	10	22	8
German	20	3	14	4	6
Heart	13	7	11	7	7
Horse-colic	26	2	7	2	8
Ionosphere	34	14	7	5	9
Iris	4	2	2	2	2
Liver	6	1	1	1	3
Mushroom	22	4	5	4	8
Pima	8	4	8	3	3
Sonar	60	19	14	10	10
Spambase	57	15	25	14	14
Wine	13	11	5	10	6
Vehicle	18	11	18	4	8
Vote	16	4	10	3	3
Average	23.78	8.28	9.78	6.61	6.94
Average dimensionality reduction		65.39%	58.87%	72.31%	70.82%

Table 4
The averaged classifier error rates (%) of C4.5 on UCI datsets

Dataset	Full set	CFS	Consistency	FCBF	UFS-SC
Austra	16.09	16.68	16.17	16.51	15.83
Breast	5.13	5.13	5.29	4.62	4.87
Chess	0.58	5.61	5.37	5.97	2.69
Credit	14.17	14.72	15.32	15.23	15.02
DNA	6.64	6.47	6.67	6.55	6.53
German	29.74	29.53	28.47	28.68	30.88
Heart	25.00	23.91	25.87	22.58	19.29
Horse-colic	34.21	29.6	32.96	31.51	32.86
Ionosphere	10.92	11.26	11.76	9.92	12.78
Iris	4.90	4.79	4.79	4.79	4.71
Liver	38.73	40.49	40.49	40.49	33.98
Mushroom	0	0.95	0	0.97	0.89
Pima	27.52	26.28	27.52	33.13	25.95
Sonar	29.58	30.42	23.66	30.42	25.35
Spambase	7.74	7.43	8.82	7.36	10.11
Wine	9.84	7.54	5.90	9.67	8.03
Vehicle	29.41	35.21	29.41	39.03	33.47
Vote	4.05	4.05	4.19	4.66	4.26
Average	16.35	16.67	16.26	17.38	15.97

Table 5
The averaged classifier error rates (%) of Ripper on UCI datasets

Dataset	Full set	CFS	Consistency	FCBF	UFS-SC
Austra	14.94	14.30	14.30	14.30	15.15
Breast	4.92	4.92	4.87	4.33	4.66
Chess	1.26	5.56	5.37	5.64	2.74
Credit	13.38	15.53	14.38	13.62	13.87
DNA	6.57	6.41	6.43	5.97	5.25
German	29.15	28.76	29.65	28.56	31.12
Heart	18.48	21.74	22.39	21.74	21.74
Horse-colic	16.19	31.20	13.60	14.52	14.84
Ionosphere	12.08	11.09	11.76	11.33	11.78
Iris	7.84	3.92	3.92	3.92	3.92
Liver	34.15	38.31	38.31	38.31	32.20
Mushroom	0	0.95	0	0.97	1.08
Pima	25.95	25.98	25.95	32.52	25.31
Sonar	30.85	28.73	26.48	29.15	25.53
Spambase	7.85	8.15	8.43	7.99	10.17
Vehicle	32.88	40.28	32.88	57.64	36.39
Vote	4.32	4.32	4.05	4.26	4.26
Wine	9.18	11.48	10.49	8.85	7.70
Average	15.00	16.76	15.18	16.87	14.87

accuracy is easily affected by the distribution of data.

From the previous empirical study, we can conclude that *UFS-SC* can achieve higher degree of dimensionality reduction efficiently and enhance or maintain predictive accuracy on imbalanced datasets with selected features.

In the following part, we evaluate the efficiency of our method *UFS-SC* in clustering form the perspectives of number of clusters and the clustering accuracy. In general, 1) with similar or the same numbers of clusters, the higher clustering accuracy is, the better the performance; 2) with the same clustering accuracy, the smaller the number of clusters the better performance. Table 9 presents the clustering performance on full features and the newly selected features based on single-pass clustering algorithm. Each dataset is tested with 10 times of random shuffling to make sure that the results are not biased by the data sequences.

Table 9 shows that the clustering efficiency on UFS-SC is better than on the full feature sets in most cases. Specifically, in ten datasets out of eighteen, the effectiveness is improved on *UFS-SC*, not only in terms of the enhancement of the clustering accuracy but also in the reduction of the number of clusters. In the other two datasets (Breast, German), although

Table 6
The averaged classifier error rates (%) of Naive Bayes on UCI datasets

Dataset	Full set	CFS	Consistency	FCBF	UFS-SC
Austra	23.79	24.68	25.11	26.38	14.62
Breast	3.32	3.32	4.12	3.32	4.41
Chess	13.55	7.27	5.37	7.43	14.65
Credit	22.04	25.19	24.68	24.94	21.57
DNA	4.90	4.24	5.40	4.21	5.86
German	25.59	26.06	25.65	26.32	29.37
Heart	15.32	16.30	17.39	15.76	21.19
Horse-colic	34.76	34.88	35.84	35.71	35.02
Ionosphere	17.98	10.59	14.12	11.31	8.42
Iris	3.57	3.39	3.39	3.39	3.39
Liver	46.10	44.41	44.41	44.41	45.85
Mushroom	4.63	1.43	1.83	1.42	1.09
Pima	25.57	23.98	25.57	25.02	26.54
Sonar	29.83	29.30	31.55	26.63	30.99
Spambase	20.53	21.15	12.88	22.97	20.70
Vehicle	54.23	52.98	54.23	58.65	51.48
Vote	9.46	4.46	9.32	4.32	5.47
Wine	2.30	2.30	4.26	1.80	3.09
Average	19.86	18.66	19.17	19.11	19.09

Table 7
Number of feature selected on three imbalanced datasets

Dataset	Full set	UFS-SC	MMP	SSF
Breast	9	4	4	3.8
Ionosphere	34	9	32	2.2
Spambase	57	14	56	2.7

Table 8
The accuracy (%) of Naive Bayes on three imbalanced datasets

Dataset	Full set	UFS-SC			MMP	SSF
		Max	Min	Aver		
Breast	96.07	96.14	95.42	95.49	96.48	94.88
Ionosphere	82.36	83.19	77.78	79.23	83.78	75.50
Spambase	79.62	79.92	78.70	78.97	80.36	58.60

Table 9
Clustering results of selected and full features on single-pass clustering

Dataset	Before FS		After UFS-SC	
	Number of clusters	Clustering accuracy (%)	Number of clusters	Clustering accuracy (%)
Austra	10–66	80.72–86.23	6–28	85.94–86.52
Breast	13	96.99	5–6	94.70–96.57
Chess	22–98	66.93–73.75	22–87	72.56–82.79
Credit	45–73	85.07–85.79	2–51	85.51–86.23
DNA	6–45	55.05–70.69	6–25	67.87–83.48
German	28–95	71.3–75	28–92	71.4–73.7
Heart	2-3	81.48–82.59	2	82.96
Horse-colic	10–60	66.84–72.01	8–60	70.11–80.97
Ionosphere	60–67	81.2–83.19	36–65	81.2–86.32
Iris	22	97.33	22	96.67
Liver	14–49	59.71–67.25	14–49	59.13–66.09
Mushroom	15–30	93.83–99.61	11–30	96.06–99.01
Pima	10–53	69.14–73.96	10–53	67.71–70.96
Spambase	12–16	60.60–61.68	12–16	64.59–69.68
Sonar	21–30	74.03–76.44	21–30	67.79–68.75
Vehicle	23–99	55.2–70.45	23–99	58.27–67.49
Vote	3-4	92.34–93.27	4–5	91.71–91.73
Wine	11–49	93.82–98.88	11–49	89.89–95.51

the clustering accuracies on the selected features are lower than those on the full feature sets, the number of clusters is also smaller than that of the original. At the same time, the clustering accuracy of selected features becomes lower in the rest of seven datasets out of twenty, while the number of clusters after *UFS-SC* is the same as that before feature selection. Therefore, we conclude that the quality of clustering is improved for most datasets on selected features obtained with *UFS-SC*.

4.6. Analysis and discussion

We evaluate the proposed feature selection algorithm on UCI datasets in terms of dimensionality reduction and accuracy. The results of our method are obtained with single-pass clustering as the clustering algorithm. The experimental results discussed above reveal that the new filter method for feature selection introduced in this paper is practical for feature selection and is independent of classification and clustering learning algorithms. The proposed algorithm *UFS-SC* can handle data with mixed types of continuous and categorical features. It can also efficiently and effectively achieve higher degree of dimensionality reduction and enhance classification accuracy with the selected features. In addition, it can improve the quality of clustering for most benchmark data sets. Empirical results demonstrate that the proposed algorithm is superior to other supervised and unsupervised feature selection algorithms on most datasets in our experiments.

5. Conclusion

Data with high dimensionality and without class information may pose great challenges to machine learning and data mining. In this paper, we present an unsupervised feature selection algorithm based on single-pass clustering and by distinguishing important features through a ranking procedure by computing feature's separability values. The proposed feature selection algorithm, called *UFS-SC*, can remove the irrelevant features by calculating separability value of each feature. The feature separability values are calculated after clustering within selecting of the clustering threshold r by sampling techniques. The final results are determined by clustering for s iterations. The proposed algorithm with nearly linear time complexity can be applied to datasets containing a mixture of categorical and continuous features. Experimental results on UCI datasets have shown that our method, by retaining relevant features, can achieve promising classification and clustering results for most datasets. Compared with other traditional feature selection approaches, the proposed algorithm can obtain similar or even better performance in terms of dimensionality reduction and accuracy in both classification and clustering.

Despite the promising results achieved in this paper, there still exist limitations in our work. Firstly, the comparability for the values of separability on different types of features is worth further analysis. Secondly, the performance of the proposed approach may be dependent on the distribution of data. Thirdly,

it cannot remove all the redundant features as much as possible. We will introduce feature clustering method to solve these issues in our future work. We will also interested in examining our approach with other clustering algorithms and extend the application of our algorithm to data with higher dimensionality for parallel computation.

Acknowledgments

This research was supported by the Humanities and Social Sciences Research Youth Foundation of Ministry of Education of China (No.14YJC870021), the National Natural Science Foundation of China (No. 61572145, No. 61402119), the Science and Technology Planning Project of Guangdong Province (No. 2014A040401083, No. 2015A030401093), the Major Program of National Social Science Foundation of China (No. 12&ZD222).

References

[1] V. Bolón-Canedo, N. Sánchez-Maroño and A. Alonso-Betanzos, Recent advances and emerging challenges of feature selection in the context of big data, *Knowledge-Based Systems* **86** (2015), 33–45.

[2] G. Xiao, K. Li, K. Li, et al., Efficient top-(k,l) range query processing for uncertain data based on multicore architectures, *Distributed and Parallel Databases* **33** (2015), 381–413.

[3] C. Shang, M. Li, S. Feng, et al., Feature selection via maximizing global information gain for text classification, *Knowledge-Based Systems* **54** (2013), 298–309.

[4] S. Jang and L. Wang, Efficient feature selection based on correlation measure between continuous and discrete features, *Information Processing Letters* **116** (2016), 203–215.

[5] M. Last, A. Kandel and O. Maimon, Information-theoretic algorithm for feature selection, *Pattern Recognition Letters* **22** (2001), 799–811.

[6] I. Kononenko, Estimating attributes: Analysis and extensions of RELIEF, *In Proceedings of European Conference on Machine Learning (ECML)*, Catania, Italy, 1994, pp. 171–182.

[7] M. Sikonja and I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, *Machine Learning* **53** (2003), 23–69.

[8] T. Ho and M. Basu, Complexity measures of supervised classification problems, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002), 289–300.

[9] M. Hall, Correlation-based feature selection for categorical and numeric class machine learning, *In Proceedings of the 17th International Conference on Machine Learning*, San Francisco, USA, 2000, pp. 359–366.

[10] L. Yu and H. Liu, Efficient feature selection via analysis of relevance and redundancy, *Journal of Machine Learning Research* **5** (2004), 1205–1224.

[11] S. Kannan and N. Ramaraj, A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm, *Knowledge-Based Systems* **23** (2010), 580–585.

[12] H. Liu and L. Yu, Toward integrating feature selection algorithms for classification and clustering, *IEEE Transactions on Knowledge and Data Engineering* **17** (2005), 1–12.

[13] M. Dash and H. Liu, Consistency-based search in feature selection, *Artificial Intelligence* **151** (2003), 155–176.

[14] Q. Hu, J. Liu and D. Yu, Mixed feature selection based on granulation and approximation, *Knowledge based Systems* **21** (2008), 294–304.

[15] Q. Hu, W. Pedrycz and D. Yu, Selecting discrete and continuous features based on neighborhood decision error minimization, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* **40** (2010), 137–150.

[16] D. Zhang, S. Chen and Z. Zhou, Constraint score: A new filter method for feature selection with pair-wise constraints, *Pattern Recognition* **41** (2008), 1440–1451.

[17] J. Sotoca and F. Pla, Supervised feature selection by clustering using conditional mutual information-based distances, *Pattern Recognition* **43** (2010), 2068–2081.

[18] T. Covões, E. Hruschka and L. de Castro, A Cluster-based Feature Selection Approach, *In Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems, Salamanca*, Spain, 2009, pp. 169–176.

[19] P. Mitra, C. Murthy and S. Pal, Unsupervised feature selection using feature similarity, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002), 301–312.

[20] J. Dy and C. Brodley, Feature selection for unsupervised learning, *Journal of Machine Learning Research* **5** (2004), 845–889.

[21] W. Au, K. Chan and A. Wong, Attribute clustering for grouping, selection, and classification of gene expression data, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **2** (2005), 83–101.

[22] Y. Li, M. Dong and J. Hua, Localized feature selection for clustering, *Pattern Recognition Letters* **29** (2008), 10–18.

[23] L. Nanni, Cluster-based pattern discrimination: A novel technique for feature selection, *Pattern Recognition Letters* **27** (2006), 682–687.

[24] H. Zeng and Y. Cheung, A new feature selection method for Gaussian mixture clustering, *Pattern Recognition* **42** (2009), 243–250.

[25] J. Sotoca and F. Pla, Supervised feature selection by clustering using conditional mutual information-based distances, *Pattern Recognition* **43** (2010), 2068–2081.

[26] W. Hung, M. Yang and D. Chen, Bootstrapping approach to feature-weight selection in fuzzy C-means algorithms with an application in color image segmentation, *Pattern Recognition Letters* **29** (2008), 1317–1325.

[27] D. Modha and W. Spangler, Feature weighting in K-means clustering, *Machine Learning* **52** (2003), 217–237.

[28] X. Wang, Y. Wang and L. Wang, Improving fuzzy C-means clustering based on feature-weight learning, *Pattern Recognition Letters* **25** (2004), 1123–1132.

[29] J. Huang, M. Ng and H. Rong, Automated variable weighting in K-means type clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27** (2005), 657–668.

[30] S. Jiang and X. Song, A clustering-based method for unsupervised intrusion detections, *Pattern Recognition Letters* **5** (2006), 802–810.

[31] C. Huang and C. Wang, A GA-based feature selection and parameters optimization for support vector machines, *Expert Systems with applications* **31** (2006), 231–240.
[32] R. Kohavi and G. John, Wrappers for feature subset selection, *Artificial Intelligence* **97** (1997), 273–324.
[33] G. Chen and J. Chen, A novel wrapper method for feature selection and its applications, *Neurocomputing* **159** (2015), 219–226.
[34] A. Asuncion and D. Newman, UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.
[35] H. Witten and E. Frank, Data mining: Practical machine learning tools and techniques, Morgan Kaufmann, San Francisco. http://www.cs.waikato.ac.nz/ml/weka/, 2005.