Journal of Southeast University (English Edition) Vol. 22, No. 2, pp. 169 – 175 June 2006 ISSN 1003—7985

# Heuristic feature selection method for clustering

Xu Junling[1] Xu Baowen[1,2] Zhang Weifeng[1,3] Cui Zifeng[1]

([1] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

([2] State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)

([3]College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** In order to enable clustering to be done under a lower dimension, a new feature selection method for clustering is proposed. This method has three steps which are all carried out in a wrapper framework. First, all the original features are ranked according to their importance. An evaluation function $E(f)$ used to evaluate the importance of a feature is introduced. Secondly, the set of important features is selected sequentially. Finally, the possible redundant features are removed from the important feature subset. Because the features are selected sequentially, it is not necessary to search through the large feature subset space, thus the efficiency can be improved. Experimental results show that the set of important features for clustering can be found and those unimportant features or features that may hinder the clustering task will be discarded by this method.

**Key words:** feature selection; clustering; unsupervised learning

Feature selection is an important task in data analysis. It is useful to limit redundancy of features, promote comprehensibility, and find clusters (or structures) hidden in high-dimensional data[1]. However, most research available is related to supervised learning, and little attention has been paid to unsupervised learning. Clustering is a form of unsupervised learning which clusters similar objects together[2−4]. It makes objects in the same cluster similar to each other and objects in different clusters dissimilar to each other. Objects are always represented as points in data space, and the similarity between two points is calculated by their features. Most clustering methods available have a presumption that every feature has the same importance to clustering, or they do not discriminate different features[5]. In fact, an important feature can help to form clusters, improve the quality of clusters, but unimportant features may have no effect on clustering or even affect the clustering algorithms adversely. That is why many clustering algorithms cannot handle high-dimensional data well. However, in practical application, data mining always concerns high-dimensional data sets, so selecting a feature subset to represent the data and clustering on it is an effective method to solve the above mentioned problem. Using the set of important features to represent the whole data can enhance the understandability of clusters, and the removal of unimportant features can improve the efficiency of clustering, and can reduce the size of data storage.

## 1 Related Work

According to whether the feature selection process depends on the induction algorithm which ultimately uses the selected features, the feature selection methods for supervised learning are divided into two types (filter vs. wrapper)[6]. Filter models are independent of concrete induction algorithms and they employ some metric dependent on intrinsic properties of the data. On the other hand, in the wrapper model, the feature selection algorithm works as a wrapper around the induction algorithm. Different feature subsets are evaluated by using the induction algorithm as a black box over the training data in order to obtain performance estimation. In unsupervised learning, we use clustering algorithms instead of induction algorithms.

In the past one or two decades, a number of methods for feature selection for clustering have been proposed, most of which are wrapper models, and few filter methods have been proposed in Refs. [7 − 8]. The characteristic of the filter model is its fast processing speed, and the wrapper model needs more time because of its calling clustering algorithms repeatedly, and it should be rerun if one wants to change for another clustering algorithm. But the wrapper model can often

**Received** 2005-11-14.

**Foundation items:** The National Natural Science Foundation of China (No. 60425206, 60503020, 60373066, 60403016), the National Basic Research Program of China (973 Program) (No. 2002CB312000), the Natural Science Foundation of Jiangsu Province (No. BK2005060), the Natural Science Foundation of Jiangsu Higher Education Institutions (No. 04KJB520096).

**Biographies:** Xu Junling (1984—), male, graduate; Xu Baowen (corresponding author), male, doctor, professor, bwxu@ seu. edu. cn.

achieve better results than the filter model, because the bias of the clustering algorithm is considered during the process of selecting features.

Wrapper methods can be categorized based on whether they select features for the whole data (global type) or just for a fraction of the data in a cluster (local type)[8]. Methods proposed in Refs. [3, 9] are both local wrapper methods. In Ref. [9], the projected clustering finds subsets of features important for (or used to define) each cluster. It first uses a $k$-medoid clustering algorithm to find clusters considering all features (using all features to represent the data) and then finds the most important features for each cluster using Manhattan distance. The algorithm called CLIQUE in Ref. [3] automatically finds subspaces of the highest dimensionality such that high-density clusters exist in those subspaces. It first divides each dimension into user given divisions, and then starts with finding dense regions (or clusters) in one-dimensional data and works upward to find $j$-dimensional dense regions using candidate generation algorithm Apriori [10].

Examples of global methods are presented in Refs. [5, 11 – 14]. The method described in Ref. [11] uses a $k$-means clustering algorithm for evaluation of feature subsets. In Ref. [12], expectation-maximization (EM) and trace measure are used for evaluation. It also provides visual aids for the user to decide the optimal number of features in the feature subset. Ref. [13] applied sequential forward and backward search. To evaluate each candidate subset, they measured the category utility of the clusters found by applying COBWEB (a hierarchical clustering algorithm) in conjunction with the feature subset. Ref. [14] applied "blind" (similar to the filter) and "feedback" (analogous to the wrapper) approaches to COBWEB, and used a feature dependence measure to select features. But all those methods mentioned above need to search for the important feature subset through the large feature subset space, and thus decrease its efficiency. The approach described in Ref. [5] is a two-step method, it first ranks and then selects a subset of important features, but the feature subset acquired may contain redundant features. However, those disadvantages can be overcome by using our method.

Our method has three steps. First, we rank all the original features according to their importance. An evaluation function $E(f)$ used to evaluate the importance of a feature is introduced. Secondly, the set of important features is selected. Finally, the possible redundant features are removed from the important feature subset. These three steps are all carried out in a wrapper framework as shown in Fig. 1.

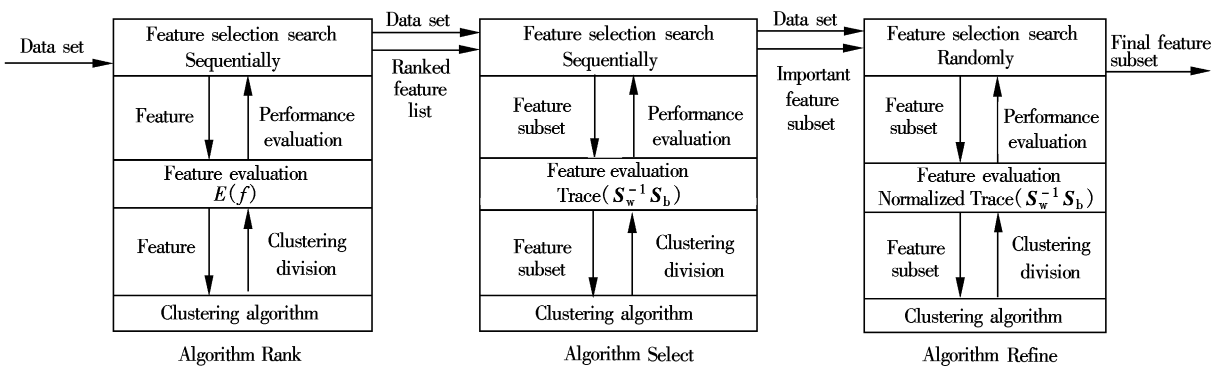


**Fig. 1** Feature selection method for clustering (wrapper model)

## 2 Evaluation of Feature Importance

To make the description more brief, the notations used in this paper are as follows: $n$ is the number of the objects in the data sets, $n_i$ is the number of the objects in cluster $i$, $t$ is the number of original features, $k$ is the number of clusters, $f$ or $F_i$ represents a feature, $m_i$ is the mean of cluster $i$, $m_{ip}$ is the value of the mean of cluster $i$ on feature $p$, $o_{jp}$ is the value of object $j$ on feature $p$, and $E(f)$ is the function value on the importance of feature $f$.

In a single-dimensional data set, clusters can be formed if the single feature takes values in separate ranges. In a multi-dimensional data set clusters can be formed from a combination of feature values although the single features by themselves alone may take uniform values[5]. That is to say, even if many related features are unimportant singly, they may become significant when combined together, so the evaluation of the feature importance should take the relationship of those features into consideration.

To evaluate the importance of features, we introduce a novel measure as follows. First, a feature $f$ is used each time to represent the data and the data set is divided into $k$ clusters by using the fuzzy $k$-means clustering algorithm (in fact, the fuzzy $k$-means clustering algorithm used in this paper can be replaced by other clustering algorithms, as long as they are based on distance), supposed to be $C_1, C_2, \ldots, C_k$, and the mean of each cluster is calculated, supposed to be $m_1, m_2, \ldots, m_k$. Then the intra-cluster similarity in each cluster and the inter-cluster dissimilarity are calculated, the former represented by the sum of the distance from each point to the mean of the same cluster, and the latter represented by the distance from the mean of one cluster to that of its nearest neighbor. Finally, the $E(f)$ value of the feature $f$ is calculated according to Eq. (1).

$$E(f) = \sum_{i=1}^{k} \frac{n_i}{n} \frac{\sum_{j=1}^{n_i} \sqrt{\sum_{p=1}^{t} (o_{jp} - m_{ip})^2}}{\min_{q=1,2,\ldots,k;\, q\neq i} \sqrt{\sum_{p=1}^{t} (m_{qp} - m_{ip})^2}} \tag{1}$$

In Eq. (1), the numerator represents the intra-cluster similarity, while the denominator represents the inter-cluster dissimilarity, and $E(f)$ is the weighted sum of the ratio between the numerator and denominator of every divided cluster, i. e., the weighted sum of the ratio between intra-cluster similarity and inter-cluster dissimilarity, representing the quality of clusters when using feature $f$ to represent the data and clustering. The smaller the $E(f)$ value of feature $f$ is, the better the clustering result achieved by clustering based on it will be, that is, the more important it will be, otherwise, its importance will not be so significant.

It seems that the evaluation function $E(f)$ in Eq. (1) considers a single feature each time while neglecting the relationship between those features. In fact, such a relationship is actually taken into consideration in $E(f)$ as illustrated in Fig. 2.

In Fig. 2(a), the values of the two features are uniformly distributed, and clustering on any single feature proves to be not so good. On the contrary, clustering in a two-dimensional space combined by feature $F_1$ and $F_2$ turns out to be of high quality. On the other hand, in Fig. 2(b), the two features are also uniformly distributed, but clustering neither in the one-dimensional nor in the two-dimensional space is ideal. Now we calculate $E(F_1)$ in Fig. 2(a) and that in Fig. 2(b). Here, the means of clusters are marked as dots. From Eq. (1), it can be seen that the greater the denominator, the farther the distance between clusters, and hence possibly the better the quality of clustering. Here, the distance between clusters in Fig. 2(a) is about $\sqrt{2}$ times

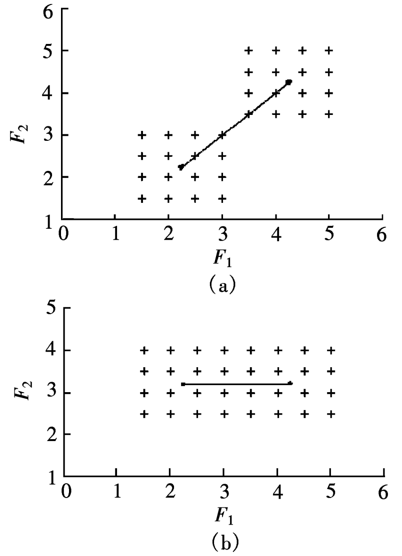


**Fig. 2** Related features and unrelated features. (a) Two related features; (b) Two unrelated features

of that in Fig. 2(b), while their numerators are almost the same, so $E(F_1)$ in Fig. 2(a) is smaller than that in Fig. 2(b), indicating that the importance of the former is more significant. So we can draw a conclusion that in evaluating the importance of feature $F_1$ in Fig. 2(a), the part that the related feature $F_2$ has played is really considered.

## 3 Feature Ranking and Selection

Our method first ranks all original features in descending order according to their importance, and then selects the important feature subset from the original feature set and removes the possible redundant features from the selected feature subset.

### 3.1 Feature ranking algorithm

Based on $E(f)$, i. e., the evaluation function of the feature importance, in Eq. (1), the fuzzy $k$-means clustering algorithm is used to divide the data set into $k$ clusters according to each feature $f$. With the help of the division result, $E(f)$ can be worked out, with which to rank the features. The detailed algorithm is shown as follows:

**Algorithm Rank**

Input: Data set $D$, cluster number $k$, and original feature number $t$.

Output: Ranked feature list RF (in descending order according to their importance).

Method:

```
For (i = 1; i < t; i + +) {
    Fuzzy k-means (F_i, D);
    Eva[i] = E(F_i);
}
RF = Sort (Eva).
```

### 3.2 Feature selection algorithm

After ranking all the features according to their importance by using algorithm Rank, a ranked feature list RF comes into being. The next step is to determine which feature in RF should be selected. If $N$, the needed number of features, is known beforehand, the final feature subset can be fixed by selecting the first $N$ features from RF directly, and the task is finished. Otherwise, the feature subset has to be determined by reentering the wrapper framework. Now these features have all been ranked in accordance with their importance. There is no need to search the space of $2^t$ in order to select the feature subset; instead, it might as well add the features in RF one by one in order.

There are numerous criteria functions for the evaluation of the clustering quality in literature, and here, the criteria function Trace($\boldsymbol{S}_\mathrm{w}^{-1}\boldsymbol{S}_\mathrm{b}$) is taken. It is invariant under any nonsingular linear transformation of data[15]. $\boldsymbol{S}_\mathrm{w}$ is the within-cluster scatter matrix and $\boldsymbol{S}_\mathrm{b}$ is the between-cluster scatter matrix, and they are defined as follows:

$$\boldsymbol{S}_\mathrm{w} = \sum_{j=1}^{k} \pi_j E[(\boldsymbol{X}-\boldsymbol{\mu}_j)(\boldsymbol{X}-\boldsymbol{\mu}_j)^\mathrm{T} \mid \omega_j]$$

$$\boldsymbol{S}_\mathrm{b} = \sum_{j=1}^{k} \pi_j(\boldsymbol{\mu}_j - \boldsymbol{M}_\mathrm{o})(\boldsymbol{\mu}_j - \boldsymbol{M}_\mathrm{o})^\mathrm{T}$$

$$\boldsymbol{M}_\mathrm{o} = E[X] = \sum_{j=1}^{k} \pi_j \boldsymbol{\mu}_j$$

where $\pi_j$ is the probability that an instance belongs to cluster $\omega_j$; $\boldsymbol{X}$ is a $d$-dimensional random feature vector representing the data; $k$ is the number of clusters; $\boldsymbol{\mu}_j$ is the sample mean vector of cluster $\omega_j$; $\boldsymbol{M}_\mathrm{o}$ is the total sample mean across all data points in the data set; and $E[\cdot]$ is the expected value operator.

To select the important feature subset in the wrapper framework, the feature subset $F$ is initialized with the first feature in RF, and then the features in RF are added into it one by one. It goes like this: use the fuzzy $k$-means clustering algorithm to divide $D$ into $k$ clusters considering the feature subset $F$; work out the Trace($\boldsymbol{S}_\mathrm{w}^{-1}\boldsymbol{S}_\mathrm{b}$) and growth ratio (slope) of it. Each time a new feature is added, here $\boldsymbol{S}_\mathrm{w}$ and $\boldsymbol{S}_\mathrm{b}$ are $|F| \times |F|$ matrices. If at a certain point the slope is negative or less than a threshold $\theta$, it means the feature subset before this point is important. The detailed algorithm is shown as follows:

**Algorithm Select**

Input: Data set $D$, cluster number $k$, and original feature number $t$.
Output: Important feature subset FI.
Method:

```
Rank ();
For (i = 1; i < t; i + +) {
Fuzzy k-means (F_1…F_i, D);
Tr[i] = Trace(S_w^{-1} S_b);
Slope = Calculate the growth ratio of Tr;
    If (Slope < 0 or Slope < θ) {
        FI = F_1…F_{i-1};
        Break;
    }
}
Draw the curve of Tr according to features.
```

The algorithm Select is influenced by the parameter $\theta$ (a small nonnegative real number) in the selecting process, but its value is very difficult to determine, so it is the usual case to let it be zero and determine the important feature subset according to the changing curve of Trace($\boldsymbol{S}_\mathrm{w}^{-1}\boldsymbol{S}_\mathrm{b}$).

Obviously, it is unavoidable that the important feature subset we have selected may contain some redundant features. To remove the possible redundant features in the feature subset, we normalize Trace($\boldsymbol{S}_\mathrm{w}^{-1}\boldsymbol{S}_\mathrm{b}$) by using the method taken in Ref. [1]. For brief narration, we take $T(\cdot)$ to represent Trace($\boldsymbol{S}_\mathrm{w}^{-1}\boldsymbol{S}_\mathrm{b}$). Let $T(F_i, C_j)$ be the criterion value using feature subset $F_i$ to represent the data and $C_j$ as the clustering assignment. The calculation of $\boldsymbol{S}_\mathrm{w}$ and $\boldsymbol{S}_\mathrm{b}$ is based on feature subset $F_i$, and they are both $d \times d$ matrices, where $d$ is the number of the features in $F_i$. The criteria value for cluster $C_1$ is normalized as $\mathrm{Nrt}(C_1, F_1, F_2) = T(F_1, C_1)T(F_2, C_1)$, and the criteria value for cluster $C_2$ is normalized as $\mathrm{Nrt}(C_2, F_1, F_2) = T(F_1, C_2)T(F_2, C_2)$. If $\mathrm{Nrt}(C_i) > \mathrm{Nrt}(C_j)$, we choose clustering $C_i$ and feature subset $F_i$. When the normalized criterion values are equal for $C_i$ and $C_j$, we favor the clustering from the lower dimensional feature subset. We can see that the value of $\mathrm{Nrt}(C_i)$ is only related to clustering assignment $C_i$. It gets rid of the bias against features, but concentrates on the quality of clustering; hence it can be used to remove the redundant features.

Now the feature subset selected beforehand is refined in order to remove the redundant features. Prev_s, Post_s are both feature subsets, Prev_c, Post_c are corresponding clustering assignments. The detailed algorithm is shown as follows:

**Algorithm Refine**

Input: Data set $D$ and cluster number $k$.
Output: Feature subset FVIP.
Method:

```
    FI = Select ();
    Prev_s = FI;
    Prev_c = Fuzzy k-means (Prev_s, D);
    Do {
        Post_s = Randomly select a subset from FI;
        Post_c = Fuzzy k-means (Post_s, D);
        Prev_ntr = Nrt(Prev_c, Prev_s, Post_s);
        Post_ntr = Nrt(Post_c, Prev_s, Post_s);
        If (Post_ntr > Prev_ntr) {
```

```
        Prev _ s = Post _ s;
        Prev _ c = Post _ c;
    }
    Else If (Post _ ntr = Prev _ ntr & | Post _ s | < | Prev _ s | )
    {
        Prev _ s = Post _ s;
        Prev _ c = Post _ c;
    }
} Until reaching the max times or you stop;
FVIP = Prev _ s.
```

As algorithm Refine is a Monte Carlo algorithm, it might be ended at any time, and the feature subset FVIP it gets when ended is the best feature subset available.

Using algorithm Select to do feature selection only needs to enter the wrapper framework $t$ times, which is smaller than many other algorithms that search in feature subset space. For methods similar to ours proposed in Ref. [5], the time complexity of algorithm Rank is $t\tau$ where $\tau$ is the time complexity of the clustering algorithm we used, which may be a little higher than that of the evaluation function used in Ref. [5], but the bias of the clustering algorithm is considered during the process of calculating feature importance. Algorithm Refine is an optional operation and can be stopped at anytime when it is necessary.

## 4 Experimental Results and Analysis

In order to evaluate the performance of our method, some benchmark and synthetic data sets are used to take a test on it. Synthetic data sets Art-1 and Art-2 are generated randomly by Matlab. Feature $F_2$ and $F_3$ in Art-1 are related, and their values in the overall data set are uniformly distributed. Others are unimportant features unrelated to each other and their values in the whole data set are also uniformly distributed. In Art-2, the values of feature $F_2$, $F_9$ and $F_{10}$ obey the Gaussian distribution in different clusters and the clusters are overlapping to some extent, meanwhile, $F_9$ and $F_{10}$ are redundant to each other, and others are generated randomly. Each cluster is of equal size if not mentioned otherwise. Both of the above two data sets contain 5% noise data. The benchmark data sets are taken from UCI repository[16], though with its known class information, such information in the process of feature selection is ignored. Two data sets are chosen, namely Iris and Monks-3, the former containing four features of numerical value, among which $F_3$ and $F_4$ are of high importance, the latter containing six features of numerical value with 5% noise data in the class information, in which feature $F_2$ and $F_4$, feature $F_5$ and $F_4$ are related, respectively.

Fig. 3 shows the $E(f)$ value, the feature ranking algorithm Rank calculates for each feature $f$, with which to rank these features according to their importance. From Fig. 3(a), it can be seen that the $E(f)$ values of important features $F_3$ and $F_4$ do rank at the front. While in Fig. 3(c), though the two related features $F_2$ and $F_3$ are unimportant singly, they are ranking at the front, which indicates our evaluation function of the feature importance takes the relationship between features into consideration.

After ranking these features, the feature selection

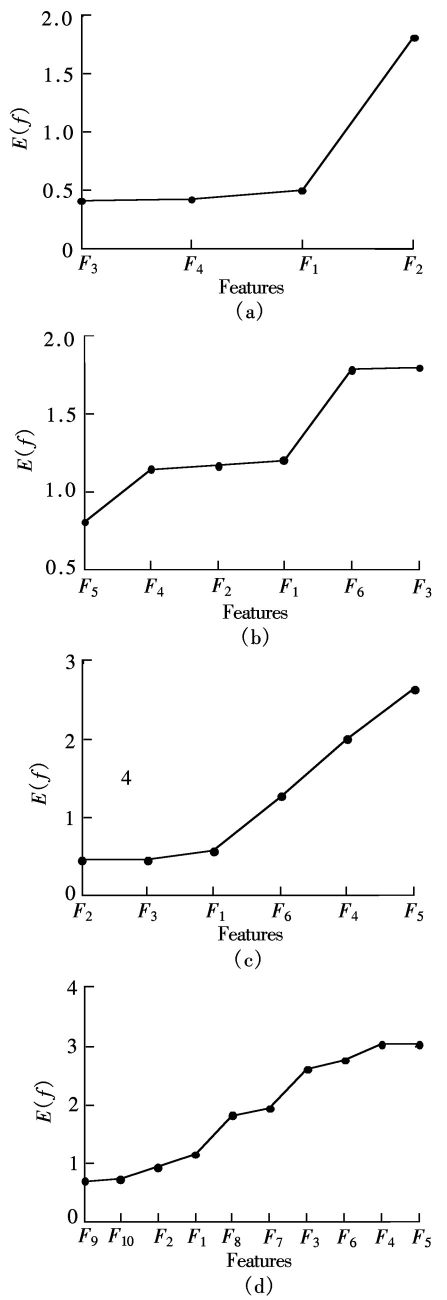


**Fig. 3** $E(f)$ value for features in each data set. (a) Iris; (b) Monks-3; (c) Art-1; (d) Art-2

algorithm Select is used to select important feature subsets for each data set. Fig. 4 shows the changing curve of Trace($\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$), which is obtained by adding the ranked features in the feature list into the feature subset (which initially is empty) one by one. Tab. 1 lists the results made by Rank and Select; where {$F_i$, $F_j$} represents these two features which are redundant to each other, [$F_i$, $F_j$] represents the feature subsets selected by the algorithm. From Tab. 1, we can see that features in the feature subsets chosen by this method are, with no exception, important ones. Only in the data set Art-2 two redundant features {$F_9$, $F_{10}$} fail to be distinguished.

As for the feature subsets selected by using the algorithm Select, which may contain some redundant features, we can use algorithm Refine to remove the redundant ones. Therefore, the four data sets mentioned above can get feature subsets shown in Tab. 2 with the processing of algorithm Refine. The result proves that such a feature selection method is very effective.

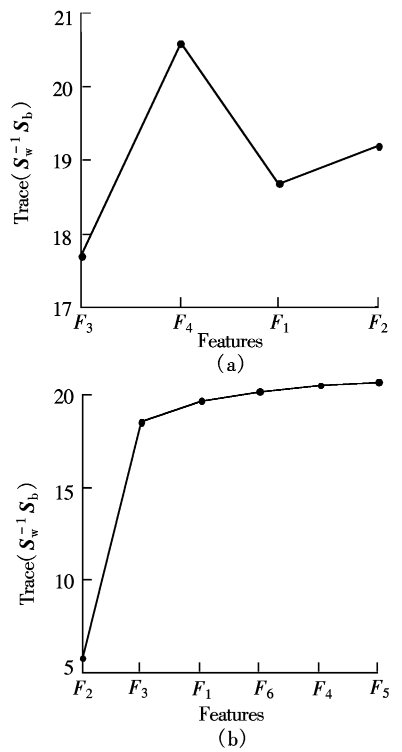


**Fig. 4** Trace($\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$) in data sets. (a) Iris; (b) Art-1

**Tab. 1** Feature ranking and selection results

| Data set | Feature number | Cluster number | Important features | Ranking result (in descending order) |
|---|---|---|---|---|
| Iris | 4 | 3 | $F_3, F_4$ | $[F_4, F_3], F_1, F_2$ |
| Monks-3 | 6 | 2 | $F_2, F_4, F_5$ | $[F_5, F_4, F_2], F_1, F_6, F_3$ |
| Art-1 | 6 | 2 | $F_2, F_3$ | $[F_2, F_3], F_1, F_6, F_4, F_5$ |
| Art-2 | 10 | 4 | $F_2, \{F_9, F_{10}\}$ | $[F_9, F_{10}, F_2], F_1, F_8, F_7, F_3, F_6, F_5, F_4$ |

**Tab. 2** Final feature subset

| Data set | Feature number | Cluster number | Important features | Feature subset |
|---|---|---|---|---|
| Iris | 4 | 3 | $F_3, F_4$ | $F_4, F_3$ |
| Monks-3 | 6 | 2 | $F_2, F_4, F_5$ | $F_5, F_4, F_2$ |
| Art-1 | 6 | 2 | $F_2, F_3$ | $F_2, F_3$ |
| Art-2 | 10 | 4 | $F_2, \{F_9, F_{10}\}$ | $F_2, F_9$ |

## 5 Conclusion

This paper introduces a feature selection method for clustering. It is a method of the wrapper model. Other than traditional methods of the wrapper model which handle the input just within one step and then attain the output, our method can be considered as entering the wrapper framework 3 times (as shown in Fig. 1). First, all the features are ranked in descending order according to their importance, and a function for evaluating the feature importance is introduced. Secondly, the important feature subsets are selected. Finally, a further step is taken to remove possible redundant features in the feature subset selected beforehand where no irrelative features exist. In the first two steps, the features are selected sequentially. In the last step only the feature subset space is needed to be searched through; however, at this point, the space is relatively small. So our method can improve the efficiency tremendously. The result shows that our method can select feature subsets of high importance for clustering tasks and discard those unimportant features or features that may hinder the clustering task.

As for even larger data sets, the sampling method can be used to select features. The clustering algorithm used to evaluate the feature importance in the wrapper framework is based on distance, and it may also be replaced by other clustering algorithms. But the evaluation function $E(f)$ should be changed accordingly with our continuing efforts.

## References

[1] Dy J G, Brodley C E. Feature subset selection and order identification for unsupervised learning [A]. In: *Proceedings of the 17th International Conference on Machine Learning* [C]. Stanford, 2000. 247 − 254.

[2] Han J, Kamber M. *Data mining: concepts and techniques* [M]. San Francisco: Morgan Kaufmann, 2001.

[3] Agrawal R, Gehrke J, Gunopulos D, et al. Automatic subspace clustering of high dimensional data for data mining applications [A]. In: *Proceedings of ACM SIGMOD International Conference on Management of Data* [C]. Seattle, Washington, DC, 1998. 94 − 105.

[4] Ganti V, Gehrke J, Ramakrishnan R. CACTUS-clustering categorical data using summaries [A]. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [C]. San Diego, 1999. 73 − 83.

[5] Dash M, Liu H. Feature selection for clustering [A]. In: *Proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining* [C]. Kyoto, Japan, 2000. 110 − 121.

[6] Kohavi R, John G H. Wrappers for feature subset selection [J]. *Artificial Intelligence*, 1997, **97**(1, 2): 273 − 324.

[7] Talavera L. Dependency-based feature selection for clustering symbolic data [J]. *Intelligent Data Analysis*, 2000, **4**(1): 19 − 28.

[8] Dash M, Choi K, Scheuermann P, et al. Feature selection for clustering—a filter solution [A]. In: *Proceedings of the* 2002 *IEEE International Conference on Data Mining* [C]. Maebashi City, Japan, 2002. 115 − 122.

[9] Aggarwal C C, Procopiuc C M, Wolf J L, et al. Fast algorithms for projected clustering [A]. In: *Proceedings of ACM SIGMOD International Conference on Management of Data* [C]. Philadelphia, 1999. 61 − 72.

[10] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases [A]. In: *Proceedings of the* 20*th International Conference on Very Large Data Bases* [C]. Santiago, Chile, 1994. 487 − 499.

[11] Kim Y S, Street W N, Menczer F. Feature selection in unsupervised learning via evolutionary search [A]. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [C]. Boston, 2000. 365 − 369.

[12] Dy J G, Brodley C E. Visualization and interactive feature selection for unsupervised data [A]. In: *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [C]. Boston, 2000. 360 − 364.

[13] Devaney M, Ram A. Efficient feature selection in conceptual clustering [A]. In: *Proceedings of the* 14*th International Conference on Machine Learning* [C]. Nashville, Tennessee, USA, 1997. 92 − 97.

[14] Talavera L. Feature selection as a preprocessing step for hierarchical clustering [A]. In: *Proceedings of the* 16*th International Conference on Machine Learning* [C]. Bled, Slovenia, 1999. 389 − 397.

[15] Fukunaga K. *Statistical pattern recognition*. 2nd ed [M]. San Diego: Academic Press, 1990.

[16] Blake C, Merz C. UCI repository of machine learning database [EB/OL]. http://www.ics.uci.edu/~mlearn/MLRepository.html. 1998/2005-10-05.

# 一种启发式聚类特征选择方法

徐峻岭[1]　徐宝文[1,2]　张卫丰[1,3]　崔自峰[1]

([1]东南大学计算机科学与工程学院,南京 210096)
([2]武汉大学软件工程国家重点实验室,武汉 430072)
([3] 南京邮电大学计算机学院,南京 210003)

**摘要:**为了使聚类可以在低维数据空间中进行,提出了一种新的聚类特征选择方法. 该方法分 3 个步骤,每个步骤都在一个 wrapper 框架中执行. 首先,将所有原始特征都按照重要性进行排序,引入一个特征重要性评价函数 $E(f)$;然后,顺序地选择特征组成重要特征子集;最后,去除重要特征子集中可能存在的冗余特征. 由于是顺序选择特征而不是在巨大的特征子集空间中进行搜索,因此算法效率很高. 实验结果表明该方法可以找出有助于聚类的重要特征子集,并且可以去掉那些不利于聚类的特征.

**关键词:**特征选择;聚类;无监督学习

**中图分类号:**TP391