# A Hybrid Method of Unsupervised Feature Selection Based on Ranking

Yun Li, Bao-Liang Lu
Department of Computer Science and Engineering
Shanghai Jiao Tong University
800 Dong Chuan Rd, Shanghai 200240, P. R. China
{liyun_mail, bllu}@sjtu.edu.cn

Zhong-Fu Wu
College of Computer Science
ChongQing University
174 Shazheng Rd, Chongqing, P. R. China
wzf@cqu.edu.cn

## Abstract

*Feature selection is a key problem to pattern recognition. So far, most methods of feature selection focus on sample data where class information is available. For sample data without class labels, however, the related methods for feature selection are few. This paper proposes a new way of unsupervised feature selection. Our method is a hybrid approach based on ranking the features according to their relevance to clustering using a new ranking index which belongs to exponential entropy. Firstly a candidate feature subset is selected using a modified Fuzzy Feature Evaluation Index (FFEI) with a new method to calculate the feature weight, which makes the algorithm to be robust and independent of domain knowledge. Then a wrapper method is used to select compact feature subset from the candidate feature set based on the clustering performance. Experimental results on benchmark data sets indicate the effectiveness of the proposed method.*

## 1. Introduction

One of the key problems that arise in a great variety of fields, including pattern recognition and machine learning, is the so-called feature selection. Feature selection not only obtains better accuracy of the predictor but also reduces training and inference time. When the class labels of sample data are available we use supervised feature selection, while in many pattern recognition application, class labels are unknown, thereby unsupervised feature selection is appropriate. Some methods of unsupervised feature selection have been developed [1, 2, 3, 4, 5]. All of the existing methods at least have one of the following shortcomings. 1) Only redundant features are removed; 2) Only irrelevant features are eliminated; 3) Their performance on high-dimensional data sets is not satisfactory; 4) They need expensive computation cost for high-dimensional data or they are sensitive to noisy data. To overcome the above deficiencies of the existing methods, we propose a new method for selecting a subset of important features by using unsupervised learning. The organization of the article is as follows. In the next section, we describe a ranking measure. In section 3, a hybrid method is described, In section 4, we provide experimental results along with comparisons. The paper ends with conclusions in section 5.

## 2. Feature ranking

We first rank the features according to their importance on clustering. Feature ranking is efficient since it requires only the computation of *n* scores and sorting the scores. Statistically, it is robust against overfitting because it introduces bias but it may have considerably less variance [6]. Here, the designed ranking index belongs to exponential entropy.

Let $S_{p,q}$ be the similarity between two instances $X_p$ and $X_q$, and let $N$ be the number of samples on which the feature ranking index is computed. The feature ranking index is defined as:

$$H = \sum_{p=1}^{N} \sum_{q=1}^{N} [S_{p,q} \times e^{(1-S_{p,q})} + (1 - S_{p,q}) \times e^{S_{p,q}}] \quad (1)$$

where $S_{p,q}$ takes value in [0.0-1.0] [1]. When $S_{p,q} \to 0(1)$, $H$ decreases. However, $S_{p,q} \to 0.5$, $H$ increases. In other words, the index decreases as the similarity (dissimilarity) between two patterns belonging to the same cluster (different cluster) in the feature space, increases. This is appropriate to character the clustering performance of the selected feature set.

For ranking of features we can use $H$ in the following way. Each feature is removed in turn and $H$ is calculated. If the removal of a feature results in the minimum $H$, the feature is the least relevant; and vice versa. The minimum $H$ indicates the removed feature has the least effect on the distribution of sample in the data set, so it has least influence on the cluster. For the data set with large number of data

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

points, we use a scalable method that is based on random sampling [1]. It should be noted that for *H* measure working well the cluster structure needs to be retained and should be largely independent of the number of data points. The ranking process is named as RANK.

## 3. Feature selection algorithm

In section 2 we propose a ranking scheme. Now, the remaining issue is how many features we should choose from the ranked feature list. We present a two-stage feature selection algorithm, which is a hybrid method. The first stage is to find a candidate feature set using a modified Fuzzy Feature Evaluation Index (FFEI) [3], which is a filter method. We adopt FFEI as the evaluation criterion because it has the following two main characteristics: 1) It has solid theory basis and can get better performance; and 2) The ranking index *H* and FFEI are all using Euclidean distance, so there is no bias. Since a mechanism to remove potentially redundant features from the already selected features has not been considered in the first stage, the second stage is to refine the results and use other more sophisticated approaches to search a compact feature subset from the candidate feature set by using Fuzzy C-Means (FCM), which is a wrapper.

### 3.1. Selecting the candidate feature set

***Computation of weight coefficient***

The FFEI is defined as [3]:

$$FFEI = \frac{2}{s(s-1)} \sum_{p=1}^{s} \sum_{q=1, p \neq q}^{s} \frac{1}{2} [\mu_{pq}^{T}(1-\mu_{pq}^{O}) + \mu_{pq}^{O}(1-\mu_{pq}^{T})] \tag{2}$$

$$\mu_{pq} = \begin{cases} 1 - \frac{d_{pq}}{D} & if \quad d_{pq} \leq D \\ 0 & otherwise \end{cases} \tag{3}$$

where $D = \beta d_{max}$, $d_{max} = [\sum_k (max_k - min_k)^2]^{(1/2)}$, $d_{pq} = [\sum_k w_k^2 (X_{pk} - X_{qk})^2]^{(1/2)}$, *D* is a parameter that reflects the minimum separation between a pair of patterns belonging to two different clusters, $\beta \in (0, 1)$ is a user defined constant parameter which determines the degree of flattening of the membership function in Eq. (2), and it is always difficult to be determined, and $w_k$ represents weighting coefficient corresponding to *k*th feature. Other definitions are presented in [3].

The index decreases as the similarity (dissimilarity) between two patterns belonging to the same cluster (different clusters) in the original feature space increases. This means, if the inter-cluster/intra-cluster distances in the transformed space increase/decrease, the feature evaluation index of the corresponding set of feature decreases [3].

Now the remained problem is how to get the value of $\omega_k$. We propose an algorithm (CalWeight) to determine $w_k$ as follows. Suppose we get a ranked feature set $(RF_1, RF_2, \cdots, RF_m)$, which is ordered by the descending *H* value, where *m* is the number of features. Let $RH_k$ be the *H* value of feature $RF_k$ in the ranked feature set and $RH_m$ is the minimum value; set overall difference of *H* value, ODH=0, and the difference of *H* values between feature $RF_k$ and $RF_m$, $DH_k = 0$

**Algorithm 1** CalWeight

**for** $k = 1$ to $m - 1$ **do**
  $DH_k = RH_k - RH_m$;
  $ODH = ODH + DH_k$
**end for**
$DH_m = 1$;
$ODH = ODH + DH_m$;
**for** $k = 1$ to *m* **do**
  $w_k = DH_k / ODH$;
**end for**

On one hand, $RH_k$ represents the clustering performance of the feature $RF_k$, which depends on the calculation of Euclidean distance. In FFEI, the Euclidean distance is also used to compute the similarity of samples. On the other hand, feature weight $w_k$ influences the similarity of samples, and then influences the degree of samples belonging to the same cluster. The feature weight $w_k$ also indicates the clustering performance of feature and has the resemble meaning to *H* value. Therefore, we can use *H* value to calculate the feature weight in (CalWeight) algorithm.

***Procedure of selecting candidate feature subset***

Now, we present an algorithm to select candidate feature set using FFEI below. In this algorithm we use the shorthand notation $FFEI(fs)$ to denote the value of FFEI as computed with all the features in feature subset *fs* and we use $\varepsilon$ as a user specified parameter representing the minimum acceptable decreasing in FFEI with each added feature. The algorithm for selecting candidate feature set can be described as follows.

The above algorithm returns *fs* as the candidate subset. The threshold $\varepsilon$ is very difficult to be determined for different data sets without prior knowledge. However, based on theoretic analysis above and experiment results below, we can see that FFEI values decrease initially and once all important features are added, it either goes up or remains relatively unchanged for any addition of unimportant features. Hence we can manually determine the stopping criterion through finding this point on the plot of FFEI value versus the number of relevant features, instead of pre-specified the value of threshold $\varepsilon$. This makes the algorithm easy to be implemented in practice.

In the algorithm, Let *s* be the number of samples on

**Algorithm 2** selecting candidate feature subset

Run RANK to get ranked feature set $OR = (RF_1, RF_2, \cdots, RF_n)$ and $RH_k$ for $k = 1, 2, \cdots, n$. Let $fs = \{RF_1\}$;
**for** $k = 2$ to $n$ **do**
  Calculate $FFEI(fs')$, where $fs' = fs \cup RF_k$;
  **if** $(FFEI(fs) - FFEI(fs')) > \varepsilon$
    Continue;
  **else**
    Break;
  **end if**
**end for**

which the feature evaluation index is computed. For one evaluated feature subset with *m* features $(m = 1, 2, \cdots, n)$ the distance between all pairs of samples should be calculated, and the time complexity is $O(s^2m)$. For every evaluated feature subset, the total time complexity is $O(s^2) + O(2s^2) + \cdots + O(ns^2) \approx O(n^2s^2)$. In practice, the size of the selected subset is much smaller than *n* and the complexity is significantly low. In addition, when the distance computation is done in parallel, the actual time cost is low.

### 3.2. Selecting compact feature subset

After using FFEI for feature selection in the first stage, we intend to find a small set of candidate features in the second stage. We use the wrapper method to search the compact feature subsets because the wrapper method has a much lower cost. We consider sequential forward selection (SFS) scheme [5] as search strategy and select FCM as clustering algorithm. Once the clustering is done we need to measure the cluster quality. We select scattering criterion, which is invariant under nonsingular transformation of data. One invariant criterion is $trace(P_W^{-1}P_B)$, where $P_W$ is within-cluster scatter matrix and $P_B$ is the between-cluster scatter matrix. The higher the $trace(P_W^{-1}P_B)$, the higher the ratio of between-cluster scatter to within-cluster one, and hence the higher the cluster quality. We use $trace(P_W^{-1}P_B)$ to compare the cluster quality for different subsets of the candidate features, and the compact subset gets the highest $trace(P_W^{-1}P_B)$.

## 4. Experiments

We empirically evaluate our feature selection method on different data sets. First, experiments are conducted on both benchmark problems and synthetic data sets to check the correctness of the ranking index and to see whether the proposed method can rank the important features at the top ranking features. Experiments are also conducted on synthetic and real-world data sets to evaluate the performance of candidate feature set selected by FFEI and compact feature set determined by FCM. We used a MATLAB random function to generate synthetic data. For synthetic data sets a few number features are chosen as important features and these features follow Gaussian distribution. Each cluster is roughly equal in size. Clusters are usually overlapping. Unimportant features are added which take uniformly random values. Two synthetic data sets, $S_1$ and $S_2$, are generated with different numbers of clusters and features. Benchmark and real-world data sets are selected from UCI machine learning repository. The details of these data sets are shown in Table 1. For Iris and Monk data sets the prior information is available regarding relevance of features and their numbers in original data set ( See Table 1).

**Table 1. Details of data sets**

| Data Sets | No. Features | No. Classes | Known Important Features |
|---|---|---|---|
| $S_1$ | 11 | 3 | 1-6 |
| $S_2$ | 22 | 6 | 1-7 |
| Iris | 4 | 3 | 3, 4 |
| Monk | 6 | 2 | 2, 4, 5 |
| Ionosphere | 34 | 2 | - |
| Sonar | 60 | 2 | - |

The experimental results for ranking and selection are shown in Table 2. From this table, we can see that our method is able to rank the relevant features in the top ranks and important features are selected for these data sets. For Ionosphere and Sonar data sets, we only compare the performance of different selected feature subsets instead of listing the concrete selected features.

**Table 2. Ranking and feature selection results**

| Data Sets | Ranking (Descending) | Selected Features |
|---|---|---|
| $S_1$ | {2,3,6,1,4,5}, 8, 11, 10, 7, 9 | 2, 3, 6, 1, 4, 5 |
| $S_2$ | $\{5, 6, 1, 2, 7, 3, 4\}, 9, 10, \cdots$ | 5, 6, 1, 2, 7, 3, 4 |
| Iris | {3,4}, 2, 1 | 3, 4 |
| Monk | {5,4,2}, 1, 6, 3 | 5, 4, 2 |

After ranking the features, the candidate feature set is selected, the changes of FFEI value are shown in Fig. 1. From Fig. 1, we can see that the FFEI values decrease with the addition of relevant features in a fast rate but slow down to almost a halt after all the relevant features are added. For practical applications, we can determine the selected feature subset by finding an approximate halt point in the curve.

Here, we only plot two curves with two different $\beta$ values, 0.3 and 0.6, and similar results are also obtained for other $\beta$ values. These results show that our method is robust against $D$.

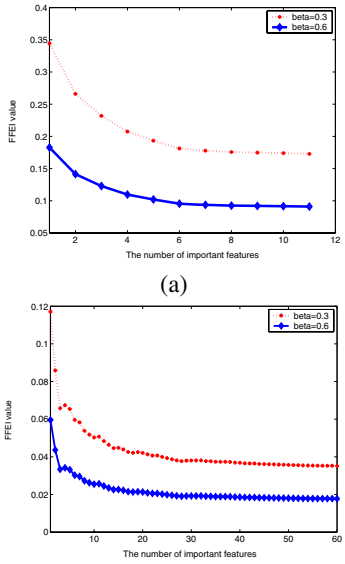


**Figure 1. Changes of FFEI values for (a)$S_2$ and (b)Sonar**

In order to compare the performance of selected features with the original features for classification, we use K-NN algorithm with K=3. Cross-validation is performed in the following manner: randomly select some data as training set to rank the original feature set and then select some data as testing set to determine the candidate subset of features. Subsequently some data are selected as training data from the remaining data to determine the compact subset. In the end, K-NN classifier will classify the final remaining data to compare the performance of candidate subset and compact subset. Ten such independent runs are performed and the average classification accuracy is used. The experimental results are shown in Table 3. The candidate set and compact set both can get higher dimensionality reduction rate. And the classification accuracy of candidate subset is better than compact subset. We can use different subsets according to different applications.

## 5. Conclusion

We have presented a hybrid unsupervised method for feature selection based on ranking using a new index. We have also considered the Fuzzy Feature Evaluation Index (FFEI) as filter criterion to select candidate feature set from ranked feature set and present a method to calculate the weight of feature in FFEI, which has been proved to be very efficient and robust against different $D$ values in FFEI. An important characteristic of the proposed method is that neither domain specialist nor prior knowledge of the problem is required, and therefore our method is very convenient for the user. For selection of compact feature set from candidate sets, we have used FCM based on the clustering performance. The experimental results on several data sets show its good performance.

**Table 3. Experimental results on Ionosphere and Sonar data sets**

| Data Sets | Selected Feature Set | Accuracy Rate Mean/SD | No. Features |
|---|---|---|---|
| Ionosphere | Candidate | 84.03/0.29 | 10 |
| | Compact | 82.00/0.30 | 5 |
| | OFS | 83.94/0.31 | 34 |
| Sonar | Candidate | 77.30/0.26 | 20 |
| | Compact | 70.63/0.33 | 6 |
| | OFS | 78.62/0.30 | 60 |


### Acknowledgments

This research was partially supported by the National Natural Science Foundation of China via the grants NSFC 60375022 and NSFC 60473040, and Shanghai Jiao Tong University and Microsoft Research Asian Joint Laboratory for Intelligent Computing and Intelligent Systems.


## References


[1] M. Dash and H. Liu, "Feature selection for clustering", Proc. Pacific Asia conf. KDD 2000, pp. 110-121.

[2] P. Mitra, C. A. Murthy and S. K. Pal, "Unsupervised feature selection using feature similarity", IEEE Trans. PAMI 24(3), 2000, pp. 301-312

[3] S. K. Pal, R. K. De and J. Basak, "Unsupervised feature evaluation: a neuro-fuzzy approach", IEEE Trans. NN, 11(3), 2000, pp. 366-376.

[4] H. C. Mart, A. T. Mario, Figueiredo and A. K. Jain, "Simultaneous feature selection and clustering using mixture models", IEEE Trans. PAMI, 9(26), 2004, pp. 1-13.

[5] M. Dash, K. Choi, P. Scheuermann and H. Liu, "Feature selection for clustering-A filter solution", International Conf. DM'02, Maebashi City, Japan.

[6] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection", JMLR, 3, 2003, pp. 1157-1182.