Unsupervised Feature Selection for Noisy Data

Kaveh Mahdavi[1,2](✉), Jesus Labarta[1,2], and Judit Gimenez[1,2]

[1] Barcelona Supercomputing Center (BSC), Jordi Girona. 29, Barcelona, Spain
{kaveh.mahdavi,jesus,judit}@bsc.es
[2] Universitat Politcnica de Catalunya, Campus Nord, Barcelona, Spain

Abstract. Feature selection techniques are enormously applied in a variety of data analysis tasks in order to reduce the dimensionality. According to the type of learning, feature selection algorithms are categorized to: supervised or unsupervised. In unsupervised learning scenarios, selecting features is a much harder problem, due to the lack of class labels that would facilitate the search for relevant features. The selecting feature difficulty is amplified when the data is corrupted by different noises. Almost all traditional unsupervised feature selection methods are not robust against the noise in samples. These approaches do not have any explicit mechanism for detaching and isolating the noise thus they can not produce an optimal feature subset. In this article, we propose an unsupervised approach for feature selection on noisy data, called Robust Independent Feature Selection (RIFS). Specifically, we choose feature subset that contains most of the underlying information, using the same criteria as the Independent component analysis (ICA). Simultaneously, the noise is separated as an independent component. The isolation of representative noise samples is achieved using factor oblique rotation whereas noise identification is performed using factor pattern loadings. Extensive experimental results over divers real-life data sets have showed the efficiency and advantage of the proposed algorithm.

Keywords: Feature selection · Independent Component Analysis · Oblique rotation · Noise separation

1 Introduction

Data is often represented by high dimensional feature vectors in many areas, such as face recognition, image possessing and text mining. In practice, not all features are relevant and important to the learning task, many of them are often correlated, redundant, or even noisy sometimes, which may result in adverse effects such as over-fitting, low efficiency and poor performance. Moreover, high dimensionality significantly increases the time and space requirements for processing the data. Feature selection is one effective means to identify relevant features for dimension reduction [11]. Once a reduced feature subset is chosen, conventional data analysis techniques can then be applied.

From the perspective of label availability, feature selection algorithms can also be classified into supervised feature selection and unsupervised feature selection. Supervised feature selection methods, such as Pearson correlation coefficients [16], Fisher score [3], and Information gain [4], are usually able to effectively select good features since labels of training data, which contain the essential discriminative information for classification that can be used. However, in practice, there is usually no shortage of unlabeled data but labels are expensive. Hence, it is a great significance to develop unsupervised feature selection algorithms which can make use of all the data points. In this paper, we consider the problem of selecting features in unsupervised learning scenario which is more challenging task because of the lack of label information that would guide the search for relevant features.

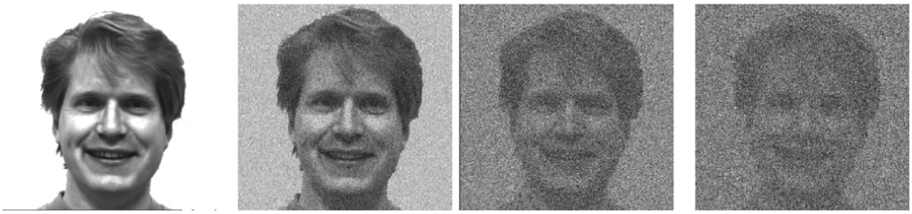


Fig. 1. Gaussian noisy versions of the image sample from COIL20 data set with different σ^2. From left to right σ^2 is: 0, 0.1, 0.4 and 0.7.

Another important factor which affects the performance of feature selection is the consideration of outliers and noise. Real data is not usually ideally distributed and outliers or noise often appear in the data, thus the traditional feature selection approach may work well on clean data. However, it is very likely to fail in noisy data sets. [15]. As an example, various types of noise are arisen during the image transmission and acquisition that Gaussian noise is one of them. It means noisy image pixel is the sum of the actual pixel value and a random Gaussian distributed noise value [18]. Figure 1 shows noisy versions of sample image from COIL20[1] data set with different σ^2 values. It can be seen that indeed, as the σ^2 value increases, the picture gets more and more ambiguous. In experimental part of this work we aim at applying our robust feature selection on cropped data set by Gaussian noise with $\sigma^2 \leq 0.7$.

In this paper, we introduce a new unsupervised feature selection algorithm, called Robust Independent Feature Selection (RIFS). We perform noise separation, isolation and robust feature selection simultaneously to select the most important and discriminative features for both unsupervised and supervised learning. Specifically, our purposed method exploits the structure of the latent independent components of a feature set and separates the noise as a component. By using independent component analysis, RIFS suggests a principled way to

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.

measure the similarity between different features and to rank each of them without label information. Thus, it imposes an oblige rotation on the independent factor indicator matrix to isolate the noise.

The rest of the paper is as follow: in Sect. 2, we present a brief review of the related work. Our proposed method, which we name Robust Independent Feature Selection (RIFS), is described in Sect. 3. The experimental results are illustrated in Sect. 4, followed by a summary in Sect. 5.

2 Related Work

Feature selection algorithms can be grouped into two main families: filter and wrapper. Filter methods [7,20] select a subset of features by evaluating statistical properties of data. For wrapper methods [6], feature selection is wrapped in a learning algorithm and the performance on selected features is taken as the evaluation criterion. Wrapper methods couples feature selection with built-in mining algorithm tightly, which lead to less generality and extensive computation. In this paper, we are particularly interested in the filter methods which are much more affordable.

The majority of the existing filter methods are supervised. Perhaps, Max variance [5] is the simplest yet effective unsupervised assessing criterion for selecting features. This measure principally projects the data points along the dimensions of maximum variances. Although the maximum variance metrics detect features that are purposeful for descriptive analysis, there is no reason to assume that these features must be useful for discriminating between data in distinct classes.

The Principal Component Analysis (PCA) algorithm shares the same principle of maximizing variance. Thus, some feature selection algorithms [12,14] are available for selecting the features by means of Principal Component Analysis. However, its orthogonal constraint on the feature selection projection matrix is unreasonable since feature vectors are not necessarily orthogonal with each other in nature.

Currently, the Laplacian Score algorithm [7] and its extensions [2,20] have been proposed to select features by leverage of manifold learning. Laplacian Score algorithm utilizes a spectral graph to extract the local geometric structure of the data then it selects feature subset which is mapped perfectly to the graph.

Another important factor which affects the performance of feature selection is the consideration of outliers and noise. In reality, outliers and noise are corrupting the distribution of the data, thus it is important or even necessary to consider noise robustness for unsupervised feature selection. Zhai [15] purposed RUFS method which jointly performs robust label learning via local learning regularized robust orthogonal non-negative matrix factorization and robust feature learning via joint $l_{1,2}$-norms minimization. A remarkable drawback of the algorithm is that its performance is relatively sensitive to the number of selected features.

The intention of our work is to purpose an unsupervised feature selection technique that can choose better features subset across a noisy data set; thereby, we are proposing a hybrid algorithm to utilize feature selection along with the noises separation and isolation.

3 Background

We consider the canonical problem of unsupervised feature selection is the following. We use X to indicate a data set of N data points $X = (x_1, x_2, ..., x_N)$, $x_i \in R^M$. The objective is to find a feature subset with size d which includes the majority informative features. In preference to, the points $[x'_1, x'_2, ..., x'_N]$ mirrored in the reduced $d-$dimensional space $x'_i \in R^d$ can perfectly maintain the original geometric structure of data in $M-$dimensional space.

In the remaining part of this section, we discuss the main data mining techniques that we utilize in our feature selection approach.

Independent Component Analysis (ICA). To detect the latent structure of data, Independent Component Analysis (ICA) [9] tries to unmix some different sources (includes noise) that have been collected together. ICA is a statistical and computational technique for revealing the hidden sources/components that underlie sets of random variables, measurements or signals. The main ICA problem assumes that the observation X is an unknown linear mixture A of the M' unknown sources S:

$$X = AS, \qquad X \in \Re^M, \qquad S \in \Re^{M'}, \qquad A \in \Re^{M \times M'}$$

We assume that each component s_i of S is zero-mean, mutually independent $p(s_i, s_j) = p(s_i)p(s_j)$ and drawn from different probability distribution which is not Gaussian except for at most one. The goal of ICA is to find an approximation W (demixing matrix) of A^{-1} such that:

$$\hat{S} = WX \approx S, \qquad W \in \Re^{M' \times M}$$

ICA is a generative model since the model describes how X could be generated from A and S. ICA tries to find A by estimating the matrices of its SVD decomposition $A = U\Sigma V^T$ [17]. Ideally, W should be:

$$W = A^{-1} = V\Sigma^{-1}U^T$$

FastICA [19] is an algorithm that searches the optimal value of W, which estimates the sources S by approximating statistical independence. The algorithm starts from an initial condition, for example, random demixing weights w_0. Then, on each iteration step, the weights w_0 are first updated by:

$$w_0{}^+ = E\{x(w_0{}^T x)^3\} - 3||w_0||^2 w_0$$

so that the corresponding sources become more independent, and then $w_0{}^+/norm$ (normalized), so that w_0 stays orthonormal. The iteration is continued until the weights converge $|w_0{}^T w_0{}^+| \approx 1$. The w_0 is an optimal approximation of W.

$$\hat{S} = w_0 X \approx S \tag{1}$$

When one tries to perform feature analysis of the data, each row of S can reflect the data distribution on the corresponding hidden source. Thus, if the data is cropped by noise, the noise is remarked as an independent source.

Oblique Rotation. Preliminary result from a factor analysis is not easy to post-process (i.e. clustering, classification). Simply, rotation has been developed not long after factor analysis to help us to clarify and simplify the results of a factor analysis. Two main types of rotation are used: orthogonal when the new axes are also orthogonal to each other, and oblique when the new axes are not required to be orthogonal to each other. The Promax [8] is an oblige rotation technique which has the advantage of being fast and conceptually simple. Promax rotation has three distinct steps.

First, it extracts the Varimax [10] orthogonal rotated matrix $\Lambda_R = \{\lambda_{ij}\}$.

Second, a target matrix is contrived to power matrix $P = (p_{ij})_{p\times m}$ by raising the factor structure coefficients to the power of Promax rotation $k > 1$,

$$p_{ij} = \left| \frac{\lambda_{ij}}{\sqrt{(\sum_{j=1}^{m} \lambda_{ij}^2)}} \right|^{k+1} \left(\frac{\sqrt{\sum_{j=1}^{m} \lambda_{ij}^2}}{\lambda_{ij}} \right)$$

Finally, it uses the matrix P to rotate the original matrix X by two levels approximation. Level one, it calculates the matrix $L = (\Lambda_R' \Lambda_R)^{-1} \Lambda_R' P$. Then, it normalizes the L by column to a transformation matrix $Q = LD$, where $D = 1/\sqrt{diag(L'L)}$ is the diagonal matrix that normalizes the columns of L. So, the preliminary rotated matrix is

$$f_{promax-temp} = Q^{-1} f_{varimax}$$

by reason of, $Var(f_{promax-temp}) = (Q'Q)^{-1}$ and the diagonal elements do not equal 1.

Level two, the rotated matrix is modified by matrix $C = \sqrt{diag((Q'Q)^{-1})}$ to $f_{promax} = C f_{promax-temp}$ the rotated factor pattern is

$$\Lambda_{Promax} = \Lambda_R Q C^{-1} \tag{2}$$

The coefficients in the rotated data is smaller, but the absolute distance between them significantly increased. It improves the quality of posterior analysis (i.e. clustering, classification).

4 RIFS Algorithm Description

In the this section, we will introduce our Robust Independent Feature Selection (RIFS) algorithm.

First of all, the independent components are computed from the X. Let $\mathbb{S}$ be a matrix whose rows are the independent decomposition vector of the matrix X and $V = [v_1, v_2, ..., v_M]$, $v_i \in R^{M'}$ is the columns of $\mathbb{S}$. Each vector

v_i represents the projection of the $i'th$ feature (variable) of the vector X to the new dimensional space, that is, the M' elements of v_i correspond to the weights of the $i'th$ factor on each axis of the new subspace. The key observation is that features that are highly correlated or have high mutual information will have extremely similar weight (changing the sign has no statistical significance). On the two extreme sides, two independent features have maximally separated weight vectors; while two fully correlated features have identical similar absolute weights vectors.

Technically, the ICA method decomposes a multivariate data into independent latent sources and white noise is an underlying source that is also drawn out as an independent component by ICA. Let $S = [s_1, s_2, ..., s_{m'}]$, $s_i \in R^M$ be the rows of $\mathbb{S}$. The s_{wn} is representing white noise when the M elements of s_{wn} have much the same absolute value with finite variance, because the white noise is randomly having equal intensity at different features [13].

In order to isolate the noise, we use the Promax method to rotate the projected feature vectors v_is to $RV = [rv_1, rv_2, ..., rv_m]$, $rv_i \in R^{M'}$ with power k. It forces the structure of the factors loadings to become bipolar that subsequently facilitates the noise isolation from the main hidden sources. It quite mitigates the drawback of the noise during discriminative analysis by uniforming the factors load of s_{wn}.

To find the best subset, we look for the profoundly cross-correlated features subset by using the underlying factor structure of the RV_i and k_mean. The features of random vector X are clustered to $C = [c_1, ..., c_d]$ when c_j represents $j'th$ cluster. We consider selecting d feature from M feature candidates.

In continue, the centroid of any cluster is computed:

$$C_j = \frac{1}{m_j} \sum_{rv_i \in c_j} rv_i \tag{3}$$

where m_j is the size of $j'th$ cluster.

Then, in any cluster the feature vectors rv_i are ranked based on their similarity with cluster centroid:

$$similarity(rv_i, C_j) = \frac{rv_i . C_j}{\|rv_i\| \times \|C_j\|} \tag{4}$$

Where values range between -1 and 1, where -1 is perfectly dissimilar and 1 is perfectly similar.

We select the highest ranked rv_i for each cluster as a corresponding vector and the corresponding feature x_i is chosen as an independent representative feature. The selected features depute each cluster properly in terms of escalated spread, independence and restoration.

We summarize the complete RIFS algorithm for feature selection in Algorithm 1.

Algorithm 1: RIFS for Feature Selection

```
Require:  $N$ data points with $M$ features;
          $d < M$ : the number of selected features ;
          $k$ : the power of Promax rotation;
Ensure: $d$ selected features
1: Compute the Independent Components as discussed in Section 3.1. Let $V = [v_1, v_2, ..., v_M]$, $v_i \in R^{M'}$ contain feature decomposition vectors and $M'$ is the number of hidden independent components.
2: Rotate the $V$ to $RV$ as discussed in Section 3.2, with power coefficient set to $k$. We get $RV = [rv_1, rv_2, ..., rv_M]$, $v_i \in R^{M'}$.
3: Cluster the vectors $rv_i$ to $d$ categories $C = [c_1, ..., c_d]$ by K-Means algorithm. Let $C_j$ be the centroid of cluster $c_j$ according to Eq. (3).
4: Compute the similarity score for each feature vectors $rv_i$ according to Eq.(4)
5: Return the corresponding feature $x_i$ of the most similar feature vector $rv_i$ to the cluster's centroids for each $d$ cluster.
```

4.1 Computational Complexity Analysis

The computational cost for the main steps of our algorithm can be computed as follows:

- The ICA computational cost is $O(NM(1 + M)d')$ where M is the number of features/dimensions, N is the number of samples, and d' is the number of iterations in fastICA algorithm.
- The K-Means and Promax algorithms are utilized on just lower dimension including M points with $M' - dimensional$ vectors, so their computational costs are negligible.

Therefor, where $M' \ll N$ and d' is customarily fixed as a constant 200, the total computational cost of RIFS is roughly corresponding to the performance of fastICA. So the total cost of our RIFS algorithm is $O(NM(1 + M)d')$.

5 Empirical Study

In this section, we have carried out several experiments to show the robustness, efficiency and effectiveness of our proposed RIFS method for unsupervised feature selection. The experiments consider both unsupervised (clustering) and supervised (classification) study. In the experiments, we have compared the RIFS, Laplacian Score and Maximum Variance. Laplacian Score and Maximum Variance are both state-of-the-art feature selection algorithms (filter methods), so this comparison makes possible to examine the efficacy of our proposed RIFS method.

5.1 Parameter Selection

Our RIFS has only one parameter, which is the k in performing the Promax rotation. We carried out different experiments in order to estimate the optimum value of k. RIFS achieves stable good performance with the k between 2 and 4 on all the four data sets. When k is less than 2, the performance slightly decreases as the k decreases. We assume $k = 4$ entire all experiments (both unsupervised and supervised study), in order to bring into uniformity.

Table 1. Summary of four benchmark data sets

Data set	Instance	Feature	Classes
YALE	165	1024	15
ISOLET	1560	617	26
USPS	9298	256	10
COIL20	1440	1024	20

5.2 Data Sets

We used four real world data sets in our experiments. The basic statistics of these data sets are outlined below in Table 1:

- The first one is **YALE**[2] face database which contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. The original images are normalized (in scale and orientation) in order that the two eyes have been aligned at the same level. Then, we have cropped the face area into the final images for processing. The size of each cropped image is 32 × 32 pixels, with 256 gray levels per pixel. Thus, each face image can be represented by a 1024-dimensional vector.
- The second one is **ISOLET**[3] spoken letter recognition data. It contains 150 subjects who spoke the name of each letter of the alphabet twice. The speakers are grouped into sets of 30 speakers each, and are referred to as isolet1 through isolet5. In our experimentation, we use isolet1 which consists 1560 examples with 617 features.
- The third one is the **USPS** (see Footnote 3) handwritten digit database. A famous subset contains 9298 16 × 16 hand written digit images in total.
- The fourth one is **COIL20** (see Footnote 3) image library from Columbia which contains 20 objects. The images of each object were taken 5° apart as the object is rotated on a turntable and each object has 72 images. The size of each image is 32 × 32 pixels, with 256 gray levels per pixel.

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html.

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.

5.3 Study of Unsupervised Cases

In this subsection, we apply our feature selection algorithm to clustering. The k-means clustering is performed by using the selected features subset and compare the results of both different algorithms and noise varieties.

Evaluation Metric. We evaluate the clustering result by informative overlapping between the obtained label of each data point using clustering algorithms and the label provided by the data set. We use the normalized mutual information metric (NMI) [7] as a performance measure. Let C indicate the set of clusters collected from the ground truth and C' obtained from a clustering algorithm. Their mutual information metric MI(C, C') is defined as follows:

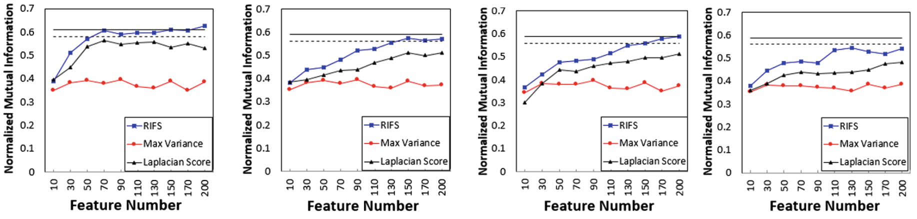
$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j).log_2 \frac{p(c_i, c'_j)}{p(c_i).p(c'_j)} \qquad (5)$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a data point arbitrarily selected from the data set belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected data point belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information NMI as follows:

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \qquad (6)$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $NMI(C, C')$ ranges from 0 to 1. $NMI = 1$ if the two sets of clusters are identical, and $NMI = 0$ if the two sets are independent.

Clustering Results. In order to randomize the experiments, we evaluate the clustering performance with different number of clusters (K = 7, 11, 13, 15 on YALE; K = 3, 5, 7, 10 on USPS; K = 5, 10, 15, 20 on COIL20 and K = 10, 15, 20, 26 on ISOLET). For each given cluster number K (except using the



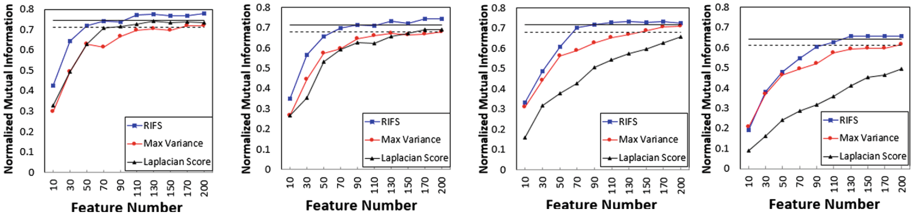
(a) Original data (No noise)

(b) Gaussian noisy ($\sigma^2 = 0.1$)

(c) Gaussian noisy ($\sigma^2 = 0.4$)

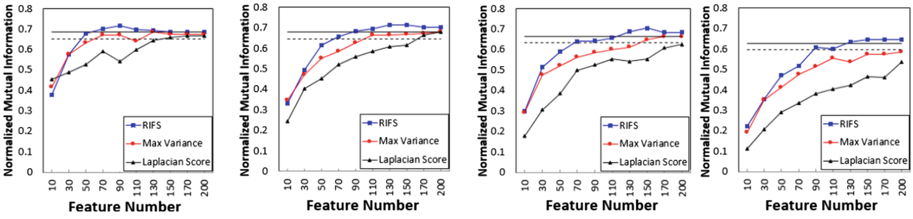
(d) Gaussian noisy ($\sigma^2 = 0.7$)

Fig. 2. Clustering performance vs. the number of selected features on YALE.



(a) Original data (No noise) *(b)* Gaussian noisy ($\sigma^2 = 0.1$) *(c)* Gaussian noisy ($\sigma^2 = 0.4$) *(d)* Gaussian noisy ($\sigma^2 = 0.7$)

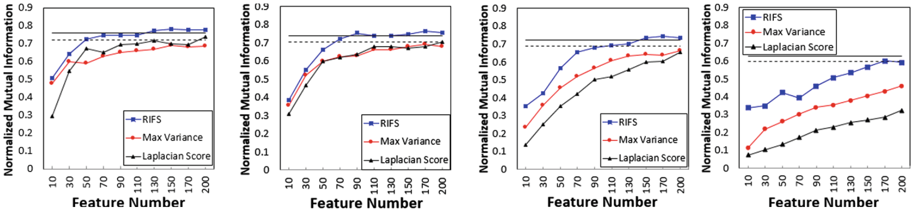
Fig. 3. Clustering performance vs. the number of selected features on Isolet.



(a) Original data (No noise) *(b)* Gaussian noisy ($\sigma^2 = 0.1$) *(c)* Gaussian noisy ($\sigma^2 = 0.4$) *(d)* Gaussian noisy ($\sigma^2 = 0.7$)

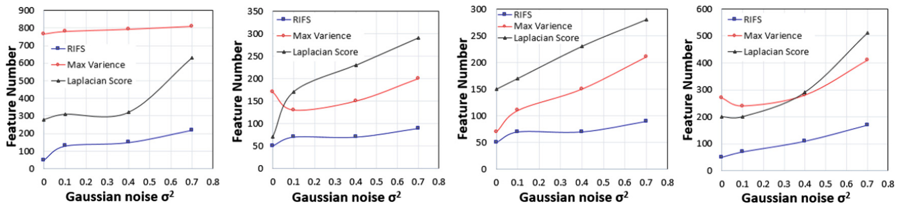
Fig. 4. Clustering performance vs. the number of selected features on USPS.

entire data set), 10 tests were conducted on different randomly chosen clusters. Then, for each data set, the overall average performance as well as the standard deviation was computed over all tests with different cluster number K. In each test, we applied different algorithms to select d features and applied k-means for clustering. In order to initiate the k-mean starting point, we applied the Hierarchical Clustering algorithm [1] then the obtained d clusters centroids are used as k-mean starting points. In principal, we performed the above procedure on clean data sets. Then, we added different Gaussian noise ($\sigma^2 = 0.1, 0.4, 0.7$)



(a) Original data (No noise) *(b)* Gaussian noisy ($\sigma^2 = 0.1$) *(c)* Gaussian noisy ($\sigma^2 = 0.4$) *(d)* Gaussian noisy ($\sigma^2 = 0.7$)

Fig. 5. Clustering performance vs. the number of selected features on COIL20.



(a) YALE *(b)* ISOLET *(c)* USPS *(d)* COIL20

Fig. 6. The noise level vs. the number of selected feature that is needed to achieve the 95% of clustering performance with all features.

Table 2. The proportion of features (# selected features/# all features%) that is needed to achieve the 95% of clustering performance with all features.

	Method	No noise	$\sigma^2 = 0.1$	$\sigma^2 = 0.4$	$\sigma^2 = 0.7$	Average
YALE	RIFS	**4.9 ± 3.8**	**12.7 ± 3.2**	**14.6 ± 3.7**	**21.5 ± 2.5**	**13.4**
	Laplacian Score	27.3 ± 2.3	30.3 ± 3.9	31.3 ± 5.1	61.5 ± 8.6	37.6
	Max Variance	74.7 ± 1.1	76.2 ± 3.2	77.4 ± 3.1	79.1 ± 1.4	76.9
ISOLET	RIFS	**8.1 ± 2.2**	**11.3 ± 3.1**	**11.3 ± 4.3**	**14.6 ± 4.4**	**11.3**
	Laplacian Score	11.3 ± 3.6	27.6 ± 11.2	37.3 ± 7.8	47.0 ± 9.0	30.8
	Max Variance	27.6 ± 8.1	21.1 ± 6.2	24.3 ± 4.3	32.4 ± 11.1	26.3
USPS	RIFS	**19.5 ± 1.7**	**27.3 ± 2.3**	**27.3 ± 1.8**	**35.2 ± 4.6**	**27.3**
	Laplacian Score	58.6 ± 8.1	66.4 ± 17.2	89.8 ± 9.8	100.0 ± 0.0	78.7
	Max Variance	27.3 ± 5.2	43.0 ± 7.9	58.6 ± 15.1	82.0 ± 12.0	52.7
COIL20	RIFS	**4.9 ± 3.3**	**6.8 ± 2.3**	**10.7 ± 5.1**	**16.6 ± 7.7**	**9.8**
	Laplacian Score	19.5 ± 8.8	19.5 ± 6.3	28.3 ± 9.1	50.0 ± 12.8	29.3
	Max Variance	26.4 ± 5.4	23.4 ± 6.2	27.3 ± 8.7	40.0 ± 3.2	29.3

to the original data sets and repeated the above clustering producer. For each σ^2 value, 10 random noise generated and tests executed, and both the average performance and standard deviation recorded over these 10 tests.

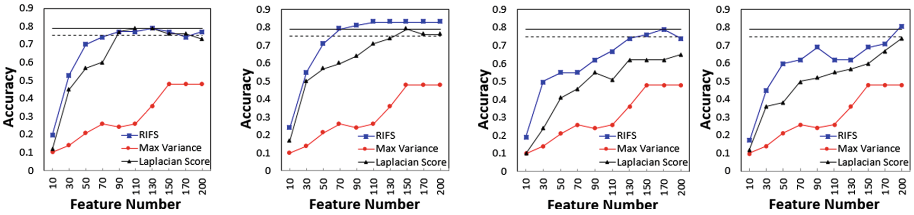
Figures 2, 3, 4 and 5 present the plots of clustering performance versus the number of selected features d on ISOLET, USPS, COIL20 and YALE, successively, without and with different level of Gaussian noises. As shown in the plots, our proposed RIFS algorithm persistently surpasses both competitors on all the four data sets and noise levels. From the plot (a) of each Figs. 2, 3, 4 and 5 (noise less), we can see RIFS converges to the best result in double quick time, with approximately 50 features. Meanwhile, both other methods mostly require more than 100 features (in average) to achieve 95% of the best result. When we add Gaussian noise with higher standard variance, we need to select more features to achieve reasonable clustering performance, as it can be seen in the plot $(b \sim c)$ of each Figs. 2, 3, 4 and 5. However, in RIFS case, this trend is very slightly pronounced when the performance of the other methods is reduced quickly by increasing the Gaussian noise standard variance, as it can be seen in Fig. 6. It would be worth mentioning that, on the ISOLET data set, our proposed RIFS algorithm performs strangely robust against the noise by selecting few more features. For example, in $\sigma^2 = 0.4$ case only 70 features are selected by RIFS and the clustering normalized mutual information is 70.3%, which is almost equal to the clustering result by using all the 617 features (71.7%). However, the Max Variance and Laplacian Score perform comparably to one another on original ISOLET data set but the Laplacian Score shows higher sensitivity to the noisy data. On COIL20 data set the Max Variance and Laplacian Score perform comparably to one another while Max Variance becomes obviously better than Laplacian Score On USPS data set. On YALE data set, Laplacian Score completely performs better than Max Variance, roughly, Max Variance does not have any function on YALE data set, possibly, due to the fact that sample size is small. The most surprising aspect of the result is that Max Variance slightly performs worse on original than data with light noise ($\sigma^2 = 0.1$) on COIL2 and ISOLET data sets.

The main objective of our experiment is to reduce the dimensionality of the data by taking to account the robustness against the noise, in Table 2, we report the selected feature proportion for achieving to at least 95% of the best clustering performance by using all features for each algorithm and Gaussian noise standard variance. The last column of each table records the average selected feature proportion over different standard variance of Gaussian noise. As it can be seen, RIFS significantly outperforms both other methods on all the four data sets. Laplacian Score performs the second best on YALE data set. Max Variance performs the second best on USPS and ISOLET data sets. Max Variance and Laplacian Score perform comparably to one another on COIL20 data set. Comparing with the second best method, RIFS selects 24.2%, 15.0%, 25.4% and 19.5% less proportion of features in average for reaching to the at least 95% of clustering performance with all features, when measured by normalized mutual information on the YALE, ISOLET, USPS and COIL20 data sets, respectively.

5.4 Study of Supervised Cases

In this experiment, we examine the discriminating capability of the different feature selection methods. The 1-Nearest Neighbor (1NN) classifier is used and we assume that well-selected feature subset should yield more accurate classifier [2]. We perform leave-one-out cross validation as follows: For each data point x_i, we find its nearest neighbor x_i'. Let $c(x_i)$ be the class label of x_i. The nearest neighbor classification accuracy rate (AR) is thus defined as

$$AR = \frac{1}{N}\sum_{i=1}^{N} \delta(c(x_i), c(x_i')) \tag{7}$$



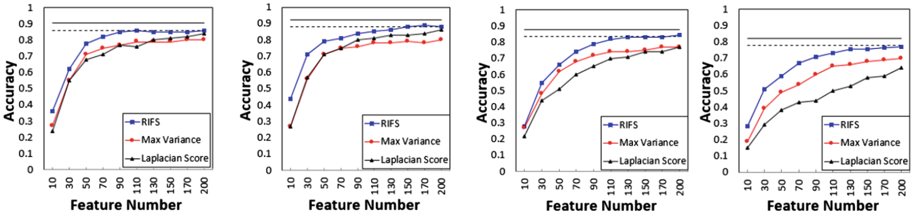
(a) Original data (No noise)

(b) Gaussian noisy ($\sigma^2 = 0.1$)

(c) Gaussian noisy ($\sigma^2 = 0.4$)

(d) Gaussian noisy ($\sigma^2 = 0.7$)

Fig. 7. Classification accuracy vs. the number of selected features on YALE.



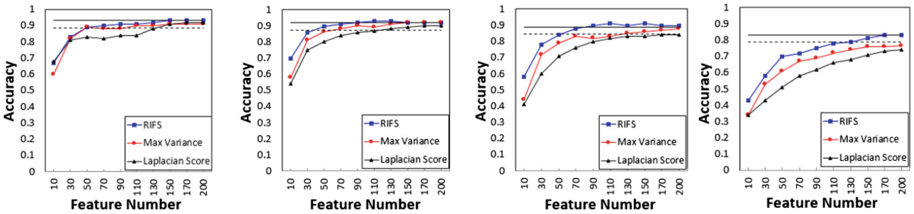
(a) Original data (No noise)

(b) Gaussian noisy ($\sigma^2 = 0.1$)

(c) Gaussian noisy ($\sigma^2 = 0.4$)

(d) Gaussian noisy ($\sigma^2 = 0.7$)

Fig. 8. Classification accuracy vs. the number of selected features on Isolet.



(a) Original data (No noise)

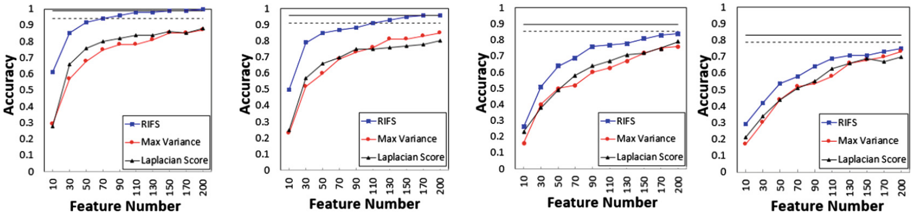
(b) Gaussian noisy ($\sigma^2 = 0.1$)

(c) Gaussian noisy ($\sigma^2 = 0.4$)

(d) Gaussian noisy ($\sigma^2 = 0.7$)

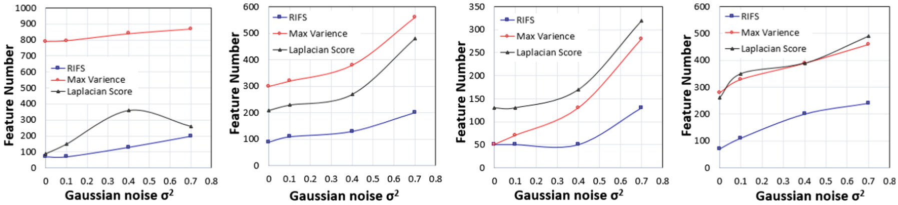
Fig. 9. Classification accuracy vs. the number of selected features on USPS.

where N is the number of data points and $\delta(a, b) = 1$ if $a = b$ and 0 otherwise. All results reported in this paper are obtained by averaging the accuracy from 10 trials of experiments. Figures 7, 8, 9 and 10 represent the plots of 1-nearest neighbor classification accuracy rate versus the number of selected features. As it can be seen, roughly on all the four data sets, RIFS at every turn goes one better than both other methods. Almost identical to clustering or even better, RIFS converges to the best result quickly on original (No noise) data set, with less than 90 features (in average) and it shows strange robustness against the noises. For example in case of $\sigma^2 = 0.7$, RIFS selects approximately 100 more



(a) Original data (No noise) *(b)* Gaussian noisy ($\sigma^2 = 0.1$) *(c)* Gaussian noisy ($\sigma^2 = 0.4$) *(d)* Gaussian noisy ($\sigma^2 = 0.7$)

Fig. 10. Classification accuracy vs. the number of selected features on COIL20.



(a) YALE *(b)* ISOLET *(c)* USPS *(d)* COIL20

Fig. 11. The noise level vs. the number of selected feature that is needed to achieve the 95% of classification accuracy with all features.

features in average to converge to the best result on all data sets, as it can be seen in Fig. 11. Remarkably, on the USPS data set with moderate additive Gaussian noise ($\sigma^2 \leqslant 0.4$), RIFS consistently can achieve 95% of the best classification accuracy by using no more than 50 features. On this data set, the Max Variance algorithm performs comparably to our algorithms and much better than Laplacian Score, and Laplacian Score's performance is defected more by additive noise. On the COIL20 data set, the Laplacian Score and Max Variance algorithms perform comparably to each other. On the ISOLET data set, the Laplacian Score and Max Variance algorithms perform comparably to each other, and Laplacian Score performs the worst. Surprisingly, on the YALE data sets, Max Variance algorithm performs quite bad in both with and without noise, unless selecting approximately 80% of features and Laplacian Score performs better on noisy data with Gaussian noise $\sigma^2 = 0.7$ than $\sigma^2 = 0.4$, possibly, due to the fact that sample size is small. The same as unsupervised study, in Table 3, we report the average (over all Gaussian noise $\sigma^2 \in \{0, 0.1, 0.4, 0.7\}$) selected feature proportion for achieving to at least 95% of the best classification performance by using all features for each algorithm. As it can be seen, RIFS achieves to the 95% of the best classification performance with approximately two times less numbers of features than the second best competitor on all data sets.

Table 3. The average proportion of features (# selected features/# all features%) that is needed to achieve the 95% of classification accuracy rate with all features.

	YALE	ISOLET	USPS	COIL20
RIFS	**11.7**	**21.5**	**27.3**	**15.1**
Laplacian Score	21.0	48.2	67.4	36.4
Max Variance	80.4	63.2	51.8	35.6

5.5 Conclusion

In this paper, we present a new robust unsupervised feature selection approach called, *Robust Independent Feature Selection* (RIFS). We propose to make the best use of the independent components structure of a set of features, which is defined on the mixing matrix, both to select the feature subset and to decouple the noise as an latent independent source, simultaneously. Thus, RIFS isolates the noise by rotating the mixing matrix obliquely. When we have compared our RIFS method with two state-of-the-art methods, namely, Laplacian Score and Max Variance, the empirical results on different real world data sets validate that the proposed method obtains considerably higher effectiveness for both clustering and classification. Our proposed RIFS algorithm performs well on original data and it is strongly resists noises.

Acknowledgements. We thankfully acknowledge the support of the Comision Interministerial de Ciencia y Tecnologa (CICYT) under contract No. TIN2015-65316-P which has partially funded this work.

References

1. Arai, K., Barakbah A. R.: Hierarchical K-means: an algorithm for centroids initialization for K-means. Reports of the Faculty of Science, Saga University, **36**(1), pp. 25–31 (2007)
2. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD (2010). https://doi.org/10.1145/1835804.1835848
3. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn., p. 394. Wiley-Interscience (2005). https://doi.org/10.1002/047174882x
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn., pp. 400–401. Wiley-Interscience, Hoboken (2000)
5. Dy, J.G., Brodley, C.: Feature selection for unsupervised learning. JMLR **5**, 845–889 (2004)
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. JMLR **3**, 1157–1182 (2003)
7. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing System, vol. 18, pp. 507–514 (2005)
8. Hendrickson, A.E., White, P.O.: PROMAX: a quick method for rotation to oblique simple structure. J. Stat. Psychol. **17**(1), 65–70 (1964). https://doi.org/10.1111/j.2044-8317.1964.tb00244.x

9. Hyvrinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural Netw. **13**(4–5), 411–430 (2000)
10. Kaiser, H.F.: The varimax criterion for analytic rotation in factor analysis. Psychometrika **23**(3), 187–200 (1958). https://doi.org/10.1007/bf02289233
11. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. **17**(4), 491–502 (2005). https://doi.org/10.1109/tkde.2005.66
12. Lu, Y., Cohen, I., Zhou, X.S., Tian, Q.: Feature selection using principal feature analysis. In: Proceedings of the 15th ACM International Conference on Multimedia, pp. 301–304 (2007). https://doi.org/10.1145/1291233.1291297
13. Mancini, R., Carter, B.: Op Amps for Everyone. Texas Instruments, pp. 10–11 (2009). https://doi.org/10.1016/b978-1-85617-505-0.x0001-4
14. McCabe, G.P.: Principal variables. Technometrics **26**(2), 137–144 (1984)
15. Qian, M., Zhai, C.: Robust unsupervised feature selection. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 1621–1627 (2013)
16. Rodgers, J.L., Nicewander, W.A.: Thirteen ways to look at the correlation coefficient. Am. Stat. **42**(1), 59–66 (1988). https://doi.org/10.2307/2685263
17. Shlens, J.: A tutorial on principal component analysis. ArXiv preprint arXiv:1404.2986 (2014)
18. Shukla, H., Kumar, N., Tripathi, R.P.: Gaussian noise filtering techniques using new median filter. IJCA **95**(12), 12–15 (2014). https://doi.org/10.5120/16645-6617
19. Zarzoso, V., Comon, P., Kallel, M.: How fast is FastICA? In: Proceedings of the 14th European Signal Processing Conference, pp. 1–5 (2006)
20. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th ICML, pp. 1151–1157 (2007). https://doi.org/10.1145/1273496.1273641