Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 29 (2008) 10–18
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Localized feature selection for clustering

Yuanhong Li, Ming Dong *, Jing Hua

Department of Computer Science, Wayne State University, Detroit, MI 48202, United States

Received 3 March 2006; received in revised form 22 March 2007
Available online 25 August 2007

Communicated by A. Fred

Abstract

In clustering, global feature selection algorithms attempt to select a common feature subset that is relevant to all clusters. Consequently, they are not able to identify individual clusters that exist in different feature subspaces. In this paper, we propose a localized feature selection algorithm for clustering. The proposed algorithm computes adjusted and normalized scatter separability for individual clusters. A sequential backward search is then applied to find the optimal (maybe local) feature subsets for each cluster. Our experimental results show the need for feature selection in clustering and the benefits of selecting features locally.

Keywords: Clustering; Unsupervised learning; Feature selection; Scatter separability

1. Introduction

Clustering is a common unsupervised learning technique used to discover the nature groups of similar objects, represented by vectors of measurements, in multidimensional spaces. A clustering algorithm typically considers all features of the data in an attempt to learn as much as possible about the objects. However, with high dimensional data, such as in visual recognition and document classification cases, many features are redundant or irrelevant. The redundant features are of no help for clustering; even worse, the irrelevant features may hurt the clustering results by hiding clusters in noises. To alleviate this problem, one of the most extensively used methods is *feature selection*. The objective of feature selection is threefold: improving the performance of clustering, providing fast and cost-efficient solution, and providing a better understanding of the underlying process that generates the data.

Feature selection involves searching through various feature subsets, followed by the evaluation of each of them using some evaluation criteria (Blum and Langley, 1997; Dong and Kothari, 2003; Kohavi and John, 1997; Yu and Liu, 2004). The mostly used search strategies are greedy sequential searches through the feature space, either forwards or backwards. Different types of heuristics, such as sequential forward or backward search, floating search, beam search, bidirectional search, and genetic search, have been suggested to navigate the possible feature subsets (Caruana and Freitag, 1994; Kohavi and John, 1997; Pudil et al., 1994; Yang and Honavar, 1997). In supervised learning, classification accuracy is widely used as evaluation criterion (Blum and Langley, 1997; Kohavi and John, 1997; Motoda and Liu, 2002; Yang and Honavar, 1998; Yu and Liu, 2004). However, in unsupervised learning, feature selection is more challenging since the class labels are unavailable to guide the search.

Feature selection in supervised learning has been widely studied (Dong and Kothari, 2003; Kohavi and John, 1997; Yu and Liu, 2004). However, for unsupervised learning, the research is relatively recent (Dash et al., 2002; Dy and Brodley, 2004; Law et al., 2004; Mitra et al., 2002; Modha and Spangler, 2003). The objective is to select important features for clustering in the absence of class labels. In (Mitra et al., 2002), a maximum information compression index is

* Corresponding author.
E-mail address: mdong@cs.wayne.edu (M. Dong).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.08.012

used to measure feature similarity so that feature redundancy is detected. The algorithm described in (Dash and Liu, 2000) evaluates the clustering tendency of each feature by an entropy index. In (Modha and Spangler, 2003), weights are assigned to different feature spaces for k-means clustering based on within-cluster and between-cluster matrices. Feature saliency is integrated in EM algorithm in (Law et al., 2004) so that feature selection is performed simultaneously with clustering process. Dy and Brodley recently proposed a wrapper criterion for clustering (Dy and Brodley, 2004), which evaluates the quality of clusters using normalized cluster separability (for k-means) or normalized likelihood (for EM clustering). In their approach, the bias on the feature subsets with respect to dimensionality is ameliorated by cross-projection normalization.

In the aforementioned algorithms, the candidate feature subsets are evaluated globally. Regardless what the evaluation criteria are, global feature selection approaches compute them over the entire dataset. Thus, they can only find one relevant feature subset for all clusters. However, it is the local intrinsic properties of data counts during clustering (Ke and Kanade, 2004). Such a global approach cannot identify individual clusters that exist in different feature subspaces. An algorithm that performs feature selection for each individual cluster separately is highly preferred.

In this paper, we propose a localized feature selection algorithm for clustering. The proposed algorithm computes *adjusted and normalized scatter separability* for individual clusters. A sequential backward search is then applied to find the optimal (maybe local) feature subsets for each cluster. Our experimental results show that the proposed localized feature selection outperforms global approaches on various datasets.

The rest of the paper is organized as follows. The motivation and details of the proposed algorithm are described in Section 2. Our algorithm is evaluated using both a synthetic dataset and several real-world datasets in Section 3. In Section 4, some conclusions are provided.

2. Localized feature selection for clustering

2.1. Motivation

Our motivation for localized feature selection can best be illustrated using a synthetic dataset. We generate 400 data points with four clusters $\{C_1, C_2, C_3, C_4\}$ in four dimensional space $\{X_1, X_2, X_3, X_4\}$. Each cluster contains 100 points. Clusters C_1 and C_2 are created in dimensions X_1 and X_2 based on a normal distribution. X_3 and X_4 are white noise features in these two clusters. The means and standard deviations are: $\mu_{C_1} = [0.5, -0.5, 0, 0]$, $\mu_{C_2} = [-0.5, -0.5, 0, 0]$, and $\sigma_{C_1} = \sigma_{C_2} = [0.2, 0.2, 0.6, 0.6]$, respectively. Clusters C_3 and C_4 exist in dimensions X_2 and X_3 with white noise in X_1 and X_4, and are created in the same manner. The means and standard deviations are: $\mu_{C_3} = [0, 0.5, 0.5, 0]$, $\mu_{C_4} = [0, 0.5, -0.5, 0]$, and $\sigma_{C_3} = \sigma_{C_4} = [0.6, 0.2, 0.2, 0.6]$, respectively. Fig. 1 shows the data in different subspaces. A general clustering algorithm, such as k-means or EM, is unable to obtain satisfactory clustering results for this data, either on all features $\{X_1, X_2, X_3, X_4\}$, or on relevant feature subset $\{X_1, X_2, X_3\}$ (may be generated by a global feature selection algorithm, i.e. Law et al., 2004), because each cluster still has one irrelevant feature. For data in higher dimensional space, this problem becomes more prominent.

On the other hand, if we further remove X_3 from the feature subset $\{X_1, X_2, X_3\}$, we can completely separate C_1 and C_2, as shown in Fig. 1a. Similarly, C_3 and C_4 can be well separated by removing X_1 as shown in Fig. 1b. In addition, the clustering results of localized feature selection provide a better understanding of the underlying process that generates the data. For example, $C_1 \sim \{X_1, X_2\}$ clearly indicates that cluster C_1 is mainly generated by features X_1 and X_2.

Usually, there are two major components of a feature selection algorithm: evaluation criteria and feature subset search methods. In the following, we first discuss the evaluation criterion for the localized feature selection algorithm, then the search method.

2.2. Evaluation criteria

In this section, we first provide a brief introduction to scatter separability criterion, one of the well-known clustering criteria (Dy and Brodley, 2004), and then show how this criterion can be adapted to localized feature selection.

Let S_w and S_b denote within-class scatter matrix and between-class scatter matrix, respectively, we have,

$$S_w = \sum_{i=1}^{k} \pi_i E\{(X-\mu_i)(X-\mu_i)^T | C_i\} = \sum_{i=1}^{k} \pi_i \Sigma_i \tag{1}$$

$$S_b = \sum_{i=1}^{k} \pi_j (\mu_i - \mu_0)(\mu_i - \mu_0)^T \tag{2}$$

$$\mu_0 = E\{X\} = \sum_{i=1}^{k} \pi_i \mu_i \tag{3}$$

where π_i is the probability that an instance belongs to cluster C_i, X the d-dimensional input dataset, k the number of clusters, μ_i the sample mean vector of cluster C_i, μ_0 the total sample mean, Σ_i the sample covariance matrix of cluster C_i, and $E\{\cdot\}$ the expected value operator.

Since S_w measures how scattered the samples are from their cluster mean, and S_b measures how scattered the cluster means are from the total mean, the scatter separability is defined as

$$\mathrm{CRIT} = \boldsymbol{tr}(S_w^{-1} S_b) \tag{4}$$

Although there are a bunch of other separability criteria available, the measure CRIT enjoys a nice property that it is invariant under any nonsingular linear transformation (Fukunaga, 1990). However, this criteria requires a nonsingular within-class scatter matrix S_w. In the case that the S_w is singular, the following separability criteria can be used instead,

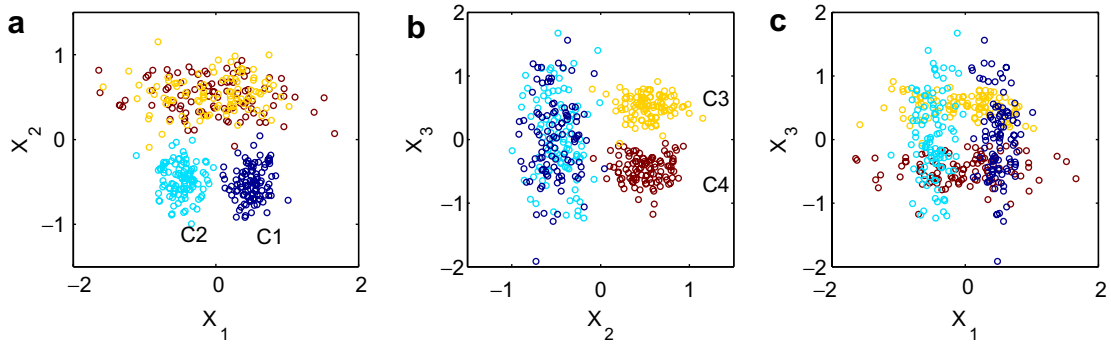

Fig. 1. Synthetic data plotted in different feature sets. Data from different clusters are marked with different colors. (a) in X_1 and X_2, (b) in X_2 and X_3, (c) in X_1 and X_3.

$$\mathrm{CRIT} = \boldsymbol{tr}(S_\mathrm{b})/\boldsymbol{tr}(S_\mathrm{w}) \tag{5}$$

In the rest of this paper, we use $\boldsymbol{tr}(S_\mathrm{w}^{-1}S_\mathrm{b})$ in our discussion. However, be aware that $\boldsymbol{tr}(S_\mathrm{b})/\boldsymbol{tr}(S_\mathrm{w})$ is used for a singular S_w. Similar to the definition of S_w, we define $S_\mathrm{w}^{(i)}$, the within-class matrix of an individual cluster C_i as,

$$S_\mathrm{w}^{(i)} = \frac{1}{n_i}E\{(X-\mu_i)(X-\mu_i)^\mathrm{T}|C_i\} = \frac{1}{n_i}\Sigma_i \tag{6}$$

where n_i is the number of points in cluster C_i. Now, we are ready to define the scatter separability of cluster C_i.

Definition 1. The scatter separability of cluster C_i is defined by,

$$\mathrm{CRIT}(C_i) = \boldsymbol{tr}(S_\mathrm{w}^{(i)-1}S_\mathrm{b}) \tag{7}$$

Assuming that identical clustering assignments are obtained when more features are added, the scatter separability CRIT prefers higher dimensionality since the criterion value monotonically increases as features are added (Fukunaga, 1990). The same conclusion could be drawn for the scatter separability of an individual cluster. Specifically, in (Fukunaga, 1990), it is shown that a criterion of the form $X_d^\mathrm{T}S_dX_d$, where X_d is d-column vector and S_d is a $d\times d$ positive definite matrix, monotonically increases with dimension. Based on this, we have:

Proposition 1. $\mathrm{CRIT}(C_i)$ *monotonically increases with dimensions as long as the clustering assignments remain the same.*

Proof. Since S_b can be expressed as $\sum_{j=1}^{k}Z_jZ_j^\mathrm{T}$ where Z_j is a column vector

$$\begin{aligned}\mathrm{CRIT}(C_i) = \boldsymbol{tr}(S_\mathrm{w}^{(i)-1}S_\mathrm{b}) &= \boldsymbol{tr}\left(S_\mathrm{w}^{(i)-1}\sum_{j=1}^{k}Z_jZ_j^\mathrm{T}\right)\\ &= \sum_{j=1}^{k}\boldsymbol{tr}(S_\mathrm{w}^{(i)-1}Z_jZ_j^\mathrm{T}) = \sum_{j=1}^{k}\boldsymbol{tr}(Z_j^\mathrm{T}S_\mathrm{w}^{(i)-1}Z_j)\\ &= \sum_{j=1}^{k}Z_j^\mathrm{T}S_\mathrm{w}^{(i)-1}Z_j\end{aligned} \tag{8}$$

Every term of Eq. (8) monotonically increases with dimension, thus the criterion for an individual cluster $\mathrm{CRIT}(C_i)$ monotonically increases with dimension. □

To alleviate this problem, normalization of the separability criterion with respect to dimensions is necessary for feature selection (Dy and Brodley, 2004). Moreover, for localized feature selection strategies, each cluster is associated with a distinct feature subset. It is usually impossible to compute S_b without proper normalization.

In the proposed algorithm, the normalization is performed using cross-projection over individual clusters. Suppose we have a cluster set C,

$$C = \{(C_1,S_1),\ldots,(C_i,S_i),\ldots,(C_k,S_k)\} \tag{9}$$

where S_i is the feature subset corresponding to cluster C_i. To calculate the scatter separability of (C_i,S_i) in cluster set C, we project all the clusters of C into feature subset S_i, and extend the scatter separability of cluster C_i as follows.

Definition 2. The scatter separability of cluster C_i in cluster set C on feature subset S_i is given by,

$$\mathrm{CRIT}(C_i,S_i)|_C = \boldsymbol{tr}(S_\mathrm{w}^{(i)-1}S_\mathrm{b})|_{C,S_i} \tag{10}$$

where $|_{C,S_i}$ denotes the project of cluster set C onto feature subset S_i.

Assume an iteration of search produces a new cluster set C' on subspace S_i',

$$C' = \{(C_1',S_i'),\ldots,(C_i',S_i'),\ldots,(C_k',S_i')\} \tag{11}$$

Let us also assume that cluster (C_i',S_i') corresponds to cluster (C_i,S_i), i.e., (C_i',S_i') is the cluster that has the largest overlap with (C_i,S_i) in set C'. We then generate a new cluster set, C^*, by replacing (C_i,S_i) in C with (C_i',S_i'),

$$C^* = \{(C_1,S_1),\ldots,(C_i',S_i'),\ldots,(C_k,S_k)\} \tag{12}$$

Note that $\mathrm{CRIT}(C_i,S_i)|_C$ and $\mathrm{CRIT}(C_i',S_i')|_{C^*}$ cannot be compared directly because of the dimension bias. We have to cross-project them onto each other,

$$\mathrm{NV}(C_i,S_i)|_C = \mathrm{CRIT}(C_i,S_i)|_C\cdot\mathrm{CRIT}(C_i,S_i')|_C \tag{13}$$

$$\mathrm{NV}(C_i',S_i')|_{C^*} = \mathrm{CRIT}(C_i',S_i')|_{C^*}\cdot\mathrm{CRIT}(C_i',S_i)|_{C^*} \tag{14}$$

After the cross-projection, the bias is eliminated and the normalized value NV can be used to compare two clusters

in different feature subspaces. A larger value of NV indicates larger separability, i.e., better cluster structures.

2.2.1. Penalty of overlapping and unassigned points

Localized feature selection implicitly creates overlapping and/or unassigned data points. Overlapping points are the data which belongs to more than one cluster, while unassigned points are the data which belongs to noncluster. Specifically, the overlapping measure O can be computed as,

$$O = \sum_{i \neq j}^{k} \frac{|C_i \cap C_j|}{\text{mean}(|C_i|, |C_j|)} \tag{15}$$

where C_i and C_j are two different clusters. Unassigned measure U can be computed as,

$$U = \frac{n_u}{n} \tag{16}$$

where n and n_u are the total number of data and the number of unassigned points, respectively. Overlapping and/or unassigned data are allowed in some applications, and may be forbidden by other applications. Depending on the domain knowledge, we could adjust the impact of overlapping and unassigned points by introducing a penalty and to obtain the adjusted normalized value ANV.

Definition 3. The adjusted and normalized scatter separability pair of cluster C_i in cluster set C on feature subset S_i, and cluster C_i' in cluster set C^* on feature subset S_i' is given by,

$$\text{ANV}(C_i, S_i)|_C = \text{NV}(C_i, S_i)|_C \cdot e^{(-\alpha \Delta O - \beta \Delta U)} \tag{17}$$

$$\text{ANV}(C_i', S_i')|_{C^*} = \text{NV}(C_i', S_i')|_{C^*} \cdot e^{(\alpha \Delta O + \beta \Delta U)} \tag{18}$$

where ΔO and ΔU are the changes on the overlapping and unassigned measure, respectively, if cluster (C_i, S_i) is replaced by cluster (C_i', S_i'). α and β are two constants.

In Definition 3, α and β are used to control the sensitivity with respect to overlapping points and unassigned points. Large values of α and β discourage the occurrence of overlapping and unassigned data. On the other hand, if α or β is zero, the corresponding effect of overlapping or unassigned data will be ignored when two clusters are compared. The values of α and β depend on the given application and have to be determined empirically. For example, if a large portion of data is unassigned after clustering, β needs to be increased.

When two clusters (C_i, S_i) and (C_i', S_i') are compared, if $\text{ANV}(C_i, S_i)|_C > \text{ANV}(C_i', S_i')|_{C^*}$, we choose (C_i, S_i). If $\text{ANV}(C_i, S_i)|_C = \text{ANV}(C_i', S_i')|_{C^*}$, we prefer the cluster in the lower dimensional space. In addition, when two identical clusters are obtained in two different feature subsets, they have equal adjusted normalized value ANV, which is exactly what we want. More formally,

Proposition 2. *Given two identical clusters $C_1 = C_2$, and the corresponding feature subspaces S_1 and S_2, the adjusted normalized value* $\text{ANV}(C_1, S_1) = \text{ANV}(C_2, S_2)$.

Proof. Since $C_1 = C_2$, we have $C = C^*$. Thus,

$$\begin{aligned}\text{NV}(C_1, S_1) &= \text{CRIT}(C_1, S_1) \cdot \text{CRIT}(C_1, S_2) \\ &= \text{CRIT}(C_2, S_2) \cdot \text{CRIT}(C_2, S_1) = \text{NV}(C_2, S_2)\end{aligned}$$

And $\Delta O = \Delta U = 0$. Thus,

$$\text{ANV}(C_1, S_1) = \text{ANV}(C_2, S_2) \qquad \square \tag{19}$$

2.2.2. Unassigned/new data

In case that some new data are obtained or unassigned data are not allowed by an application, assignments have to be made after clustering for these new/unassigned points. The similarity of an instance and a cluster could be measured by either distance (k-means clustering), or likelihood (EM algorithm). The additional difficulty introduced by localized feature selection algorithm is that clusters are associated with different feature subsets, making the direct comparison between clusters meaningless. For distance based similarity, a straightforward solution is to normalize the distance measure over its variance within each cluster, and assign the instance to a cluster that minimizes the normalized distance,

$$\arg\min_{C_j} d = \arg\min_{C_j} \left(\frac{\|X_i|_{S_j} - \mu_j\|}{\sigma_j^2} \right) \tag{20}$$

where X_i is an unassigned point, μ_j the cluster mean vector of C_j, S_j the feature subset of C_j, $X_i|_{S_j}$ the projection of X_i into S_j, and $\|\cdot\|$ is the norm of a vector. A similar method can be developed for likelihood-based similarity measure.

2.3. Search methods

The cross-projection normalization scheme assumes that the clusters to be compared should be consistent in the structure of the feature space (Dy and Brodley, 2004). Consequently, we select sequential backward search instead of the sequential forward search adopted in (Dy and Brodley, 2004). The tradeoff is the slower clustering speed.

Specifically, the data are first clustered based on all available features. Then, for each cluster, the algorithm determines if there exists a redundant or noisy feature based on the adjusted normalized value ANV defined in Eqs. (17) and (18). If so, it will be removed. The above process is repeated iteratively on all clusters until no change is made, at which time the clusters with the associated feature subsets will be returned. The sequence of steps shown in Fig. 2 illustrates our algorithm in detail.

The complexity is $O(ndik)$ for the conventional k-means algorithm, and $O(nd^2ik)$ for the GFS-k-means algorithm, where n is the number of points, d the number of features, i the number of iteration (usually unknown), and k the number of clusters. The complexity of our approach, in worst case, is $O(nd^3k^2i)$ with backward sequential search. It shows that for datasets with very high dimensions and large number of clusters, the proposed algorithm is slow

```
input  : Dataset X_{n×d}
output: Clusters C = {(C_i, S_i)|i = 1, ..., k}
initialize C with all features;
repeat
    for i = 1 to k do
        Create a new subset S'_i by removing one feature from S_i ;
        Generate a new cluster set C' on S'_i ;
        Compare clusters in C' with corresponding clusters in C;
        if Better cluster found then
            Replace the corresponding cluster in C
        end
    end
until No change made ;
if Desired then
    Process unassigned data points
end
```

Fig. 2. The proposed localized feature selection algorithm.

compared to general k-means and global feature selecting algorithms. However, the complexity is in polynomial form, and thus is still acceptable in practice.

3. Experiment and results

We evaluate the localized feature selection algorithm using both synthetic and real-world datasets. The experimental results are obtained by choosing k-means as the clustering algorithm. However, note that the adjusted normalized value ANV is not restricted to k-means. It can be used together with any general clustering algorithm.

In general, it is difficult to evaluate the performance of a clustering algorithm on high dimensional data. Localized feature selection presents an additional layer of complexity by associating clusters with different feature subsets. Therefore, we take a gradual approach for our evaluation. We first test the proposed algorithm on a small synthetic dataset with known data distribution along each feature dimension. Then, we investigate five real-world datasets downloaded from UCI repository (Blake and Merz, 1998). On all UCI datasets, we perform a semi-supervised learning strategy for evaluation purpose. This makes it possible for us to compute a pseudo-accuracy measure for easy comparison among different algorithms. However, one should be aware that the "true" class labels are not always consistent with the nature grouping of the underlying dataset. Thus, the quality of clusters should be further analyzed in addition to the pseudo-accuracy. For this purpose, we also illustrate our results by visually examining the clusters in the selected feature subspace on synthetic data and Iris data.

For each dataset, we compare our localized feature selection algorithm (with k-means, denoted by LFS-k-means) with global feature selection algorithm (also with k-means, denoted by GFS-k-means), and k-means without feature selection. GFS-k-means is implemented in a similar fashion as Dy and Brodley (2004). The only difference is that we adopted the backward search strategy due to the reason discussed in Section 2.3.

In the experiments described above, the number of clusters k is set to the "true" number of classes. This is not always applicable in real world applications. How to determine the value of k is a common problem in unsupervised learning. It may strongly interact with the predicted cluster structures, as well as the selected feature subset in feature selection algorithms (Figueiredo and Jain, 2002; Fukunaga, 1990; Law et al., 2004). Another typical problem associated with clustering is how to initialize cluster centroids. Bad initial clusters/centroids might lead to low quality clusters. Techniques, such as preliminary clustering and choosing the best from several independent runs, are frequently used to alleviate the chance of bad initial clusters. In the proposed algorithm, bad initial clusters for backward searching may occur, particularly when many noise features are presented, and thus lead to unsatisfactory final clusters and feature subsets. This problem can be alleviated by preliminary clustering with a global feature selection algorithm. To this end, we perform another set of experiments as an example solution for unknown k and preliminary clustering in Section 3.4. Specifically, we evaluate our algorithm over three UCI datasets with a large number of features and examples, assuming the k is unknown. We first employ the global feature selection algorithm proposed in (Law et al., 2004) to estimate the number of clusters, global feature saliency and cluster centroids. Then, we use them as initial parameters and run our localized algorithm. Clusters obtained are labeled to its majority portion of true classes, and errors are calculated accordingly.

3.1. Synthetic data

The synthetic data is described in Section 2.1, and illustrated in Fig. 1. Penalties of overlapping and unassigned points (α and β) are set at 1.

Table 1 shows the confusion matrix and error rate of k-means with full feature set, k-means without the totally irrelevant feature X_4, GFS-k-means, and LFS-k-means, and Table 2 shows the selected feature subsets. Clearly, by employing all four available features, k-means performs poorly with a error rate of 0.225, which indicates that irrelevant features greatly reduce the clustering performance. Meanwhile, GFS-k-means does a terrible job with an unacceptable error rate of 0.708. The output feature subset contains only the noisy feature X_4. This surprising result could be explained as follows: since each feature is irrelevant to at least two clusters and each cluster has at least two irrelevant features, no feature subset is relevant to all clusters. We also evaluated k-means algorithm on the feature subset X_1, X_2, X_3, which are the globally relevant features that can probably be obtained by a *smart* global feature selection algorithm, as shown in Table 2. The error rate is as high as 0.428, indicating that the group structures cannot be recognized with globally relevant feature subset. The reason is that the structures are buried not only by the irrelevant feature X_4, but also by the relevant features X_1 and X_3. On the other hand, the proposed localized feature selection

Table 1
Confusion matrix and error rate on the synthetic data

Label	k-Means				k-Means w/o X_4				GFS-k-means				LFS-k-means			
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
T1	77	22	1	0	59	40	1	0	37	17	46	0	99	0	0	1
T2	3	76	0	21	45	49	0	6	33	22	45	0	0	100	0	0
T3	1	7	89	3	0	3	69	28	26	16	58	0	0	1	98	1
T4	23	0	9	68	3	0	45	52	35	14	51	0	2	0	0	99
Error	0.225				0.428				0.708				0.01			

C1–C4 are the output cluster labels, and T1–T4 are the true cluster labels.

Table 2
Feature subset distribution on the synthetic data

Algorithm	Feature subset(s)			
	C1	C2	C3	C4
k-Means	{1,2,3,4}			
GFS-k-means	{4}			
LFS-k-means	{1,2}	{1,2}	{2,3}	{2,3}

C1–C4 are the output cluster labels.

Table 4
Feature subset distribution on iris data

Algorithm	Feature subset(s)		
	C1	C2	C3
k-Means	{1,2,3,4}		
GFS-k-means	{3}		
LFS-k-means	{4}	{3,4}	{3,4}

C1–C3 are the output cluster labels.

algorithm produces an excellent result with an error rate of 0.01. From Table 2, we can see clearly that the relevant features for each cluster are selected correctly, and the clusters are well separated in the corresponding feature subspaces (Fig. 1a and b). This result confirms that selecting features locally is meaningful and necessary in clustering.

3.2. Iris data

Iris dataset from UCI is a widely used machine learning benchmark dataset for both supervised learning and unsupervised learning. This dataset has three classes, four features, and 150 instances. In this experiment, we set α and β to be 1 and 6, respectively.

Table 3 shows the confusion matrix and error rate of k-means, GFS-k-means, and LFS-k-means, respectively, and Table 4 shows the corresponding feature subsets. k-means, with all four features, is able to successfully identify cluster 1, "iris-setosa". However, it does not perform well on cluster 2, "iris-versicolor", with an error rate of 0.22, and cluster 3, "iris-virginica", with an error of 0.28. The GFS-k-means discards feature 1, 2, and 4, and recognizes the structure of the dataset much better with only feature 3. The proposed LFS-k-means results in the best pseudo-accuracy. The selected feature subsets show that cluster 1 can be separated along feature 4, clusters 2 and 3 can be separated along features 3 and 4. The right panel of Fig. 3 shows the scatter plot of iris data along features 3 and 4. Clearly, cluster 1 can be separated either by feature 3 or by feature 4. In other words, one of the features is redundant to cluster 1. The proposed algorithm keeps feature 4 and removes feature 3 from the subset. The selected features for clusters 2 and 3 (features 3 and 4) are also consistent with our visual inspection. The left panel of Fig. 3 clearly shows that features 1 and 2 are not helpful to differentiate these two clusters.

The experimental results on iris dataset show that the proposed algorithm is capable of reducing redundant/noisy features for each individual cluster. It can also provide us a better understanding of the date generation.

3.3. Other UCI data

We also evaluate LFS-k-means and compare the results with k-means and GFS-k-means on four other UCI datasets, Wine, Ion, Sonar, and Glass, which are more complicated than Iris dataset in terms of number of features and number of classes. From Wine to Ion to Sonar, the number

Table 3
Confusion matrix and error rate on iris data

Label	k-Means			GFS-k-means			LFS-k-means		
	C1	C2	C3	C1	C2	C3	C1	C2	C3
T1	50	0	0	50	0	0	50	0	0
T2	0	39	11	0	46	4	0	48	2
T3	0	14	36	0	3	47	0	4	46
Error	0.167			0.0467			0.04		

C1–C3 are the output cluster labels, and T1–T3 are the true cluster labels.

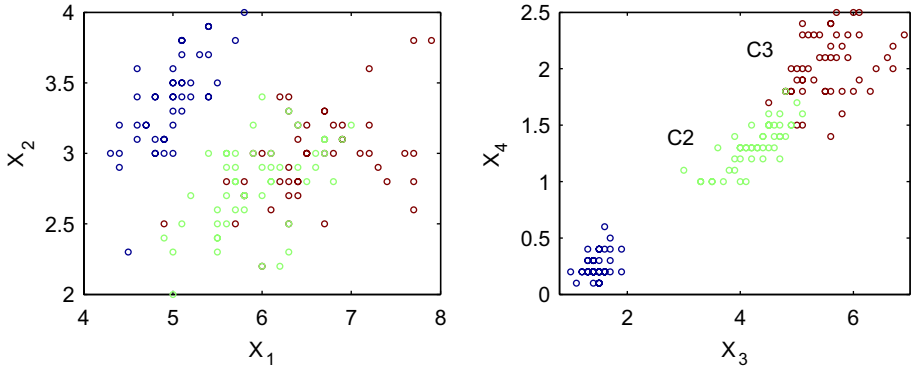

Fig. 3. Scatterplots on iris data using features 1 and 2 (left panel), and using features 3 and 4 (right panel). Data from different classes are marked with different colors.

of features increases from 13 to 32 to 60 with two or three classes. From Ion to Wine to Glass dataset, the number of classes increases from 2 to 3 to 5. Table 5 shows the experiment results.

For Wine dataset, GFS-*k*-means keeps 12 out 13 features with accuracy of 0.039. On the other hand, LFS-*k*-means selects 10, 13 and 8 features for different clusters with a better accuracy of 0.023.

For the Ion dataset, GFS-*k*-means selects 10 features for both clusters. Compared to GFS-*k*-means, our proposed algorithm results in 1 feature for cluster C1 and 10 features for the other cluster C2. Notice that these 10 features for C2 are identical to those selected by GFS-*k*-means. This implies that localized feature selecting algorithm performs at least the same as global feature selecting algorithm. Furthermore, it also shows that it is often unsuitable to only select one feature subset for all the clusters in unsupervised learning.

Experiments on Glass and Sonar datasets give similar results. In summary, LFS-*k*-means leads to variant feature subsets for different clusters, and provide best (on Wine and Sonar) or similar (on Ion and Glass) pseudo-accuracy compared to conventional *k*-means algorithm and GFS-*k*-means. In addition, the feature subsets selected by LFS-*k*-means are usually much shorter than GFS-*k*-means. These results confirm that clusters do exist on localized feature subsets for certain problems.

3.4. UCI data with estimation of k and initial clusters

In this section, we evaluate our algorithm on three UCI datasets, WDBC, Image, and Zernike. WDBC is the Wisconsin Diagnostic Breast Cancer dataset with 30 features and 576 patterns from two classes (benign or malignant). Image is the image segmentation dataset with 2310 patterns and 19 features (18 of them are nonsingular) from seven categories (brickface, sky, foliage, cement, window, path, and grass). Zernike contains 47 Zernike moments extracted from 2000 handwriting numerals (0–9), 200 for each digit. These datasets with large number of features and examples present a more difficult problem for the proposed localized feature selection algorithm.

We suppose that the number of clusters k is unknown. We first run global feature selection algorithm in (Law et al., 2004) with 30 initial clusters, and obtain the estimated value of k, cluster centroids, and global feature sal-

Table 5
Comparison of *k*-means, GFS-*k*-means and LFS-*k*-means on other UCI datasets

Dataset				Subfeature			Error		
Name	Patt.	Feat.	Clas.	*k*-Means	GFS	LFS	*k*-Means	GFS	LFS
Wine	178	13	3	13	{1 2 3 4 5 6 8 9 10 11 12 13}	C1: {1 3 4 5 7 8 10 11 12 13} C2: {1 2 3 4 5 6 7 8 9 10 11 12 13} C3: {3 4 5 9 10 11 12 13}	0.034	0.039	0.023
Ion	351	32	2	32	{3 7 11 13 15 17 19 29 30 31}	C1: {13} C2: {3 7 11 13 15 17 19 29 30 31}	0.288	0.296	0.296
Glass	214	9	5	9	{2 3 5 6 7 8 9}	C1: {4 5 7 9} C2: {2 3 4 5 7 8 9} C3: {3 5 7 9} C4: {6 8} C5: {5 6}	0.192	0.201	0.196
Sonar	208	60	2	60	{35 36 37 38 41 42 44 46 47 51 55 56 57 58 59 60}	C1: {9 10 49 50 51 56 58} C2: {9 10 49 50 51 56 58}	0.452	0.466	0.375

Table 6
UCI datasets with estimated number of clusters and initial centroids

Dataset				GFS		LFS	Error	
Name	Patt.	Feat.	Clas.	$\hat{k}$	Salient feat.	Feat. subset	GFS	LFS
WDBC	576	30	2	8	{29 features}	C1: {24 features} C2: {25 features} C3: {13 14 16 17 23 26 29} C4: {26 features} C5: {25 features} C6: {4 13 14 16 23 26} C7: {4 13 14 16 23 26 29} C8: {4 14 16 23 26 29}	0.09	0.10
Image	2310	18	7	18	{17 features}	C1:{7 8 14 17} C2: {12 13} C3: {2 3 9 11 13 14 15 16 18} C4: {3 4 5 9 10 13 16 18} C5: {5 18} C6: {18} C7: {18} C8: {17 features} ...	0.19	0.28
Zernike	2000	47	10	17	{45 features}	C1:{16 features} C2: {22 features} C3: {13 features} C4: {13 features} C5: {2 features} C6: {44 features} C7: {16 features} C8: {44 features} ...	0.49	0.48

iency. Only features with saliency greater than 0.5 are kept. We then run the proposed algorithm using the estimated k and corresponding centroids. The experimental results are reported in Table 6. For Image and Zernike datasets, only the first eight clusters are shown.

On WDBC, the GFS algorithm leads to 29 salient features out of 30. Our approach produces different feature subsets for each cluster. The size of feature subsets varies from 6 to 26, with an average value of 15.8, which is much less than the size of feature subset obtained by GFS. Same results are observed on both Image and Zernike datasets: On Image dataset, feature subset size varies from 1 to 17 with an average value of 6.3, while the size of GFS's is 17. On Zernike dataset, feature subset size varies from 2 to 45 with an average value of 22.7, while the size of GFS's is 45.

The error rates of GFS and LFS on WDBC are almost the same (0.09 and 0.10, respectively), as well as the error rate on Zernike (0.49 and 0.48, respectively), which implies that our clustering results are comparative to GFS over those datasets. Note that the error rate on Image is different: 0.19 for GFS and 0.28 for LFS. However, one cannot conclude that the clustering quality of LFS is much worse than that of GFS on this dataset, since the cluster structures may be ambiguous between the true classes in this dataset. The benefit of LFS here is a much smaller subset of features for individual clusters.

4. Conclusions and future work

In clustering, global feature selection algorithms attempt to select a common feature subset that is relevant to all clusters, which may not be feasible for many high dimensional datasets with many clusters. In order to identify individual clusters that exist in different feature subspaces, we proposed a localized feature selection algorithm. We developed adjusted and normalized scatter separability (ANV) for individual clusters, based on which our algorithm was capable of reducing redundant/noisy features for each cluster separately. The proposed algorithm can also provide us a better understanding of the underlying process that generates the data. Our experimental results on both synthetic and real-world datasets showed the need for feature selection in clustering and the benefits of selecting features locally.

In this paper, we employed the cross-projection method to evaluate the quality of an individual cluster, which made it impracticable to change the number of clusters during clustering and feature selection process. Thus, a fixed k estimated in advance was required to perform localized feature selection with our approach. However, in the area of unsupervised learning with feature selection, algorithms that simultaneously compute the number of clusters and the local feature subset will be more desirable. In future work, this is the direction that we are actively pursuing.

Acknowledgements

This research was partially funded by the 21st Century Jobs Fund Award, State of Michigan, under Grant: 06-1-P1-0193.

References

Blake, C.L., Merz, C.J., 1998. UCI repository of machine learning databases.

Blum, Avrim, Langley, Pat, 1997. Selection of relevant features and examples in machine learning. Artif. Intell. 97 (1–2), 245–271.

Caruana, Rich, Freitag, Dayne, 1994. Greedy attribute selection. In: Internat. Conf. Machine Learning, pp. 28–36.

Dash, M., Choi, K., Scheuermann, P., Liu, H., 2002. Feature selection for clustering – a filter solution. In: IEEE Internat. Conf. on Data Mining, pp. 115–122.

Dash Manoranjan, Liu, Huan, 2000. Feature selection for clustering. In: Pacific-Asia Conf. on Knowledge Discovery and Data Mining, pp. 110–121.

Dong, M., Kothari, R., 2003. Feature subset selection using a new definition of classifiability. Pattern Recognition Lett. 23, 1215–1225.

Dy, Jennifer G., Brodley, Carla E., 2004. Feature selection for unsupervised learning. J. Machine Learning Res. 5, 845–889.

Figueiredo, M., Jain, A.K., 2002. Unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Machine Intell. 24, 381–396.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, Boston.

Ke, Qifa, Kanade, Takeo, 2004. Robust subspace clustering by combined use of KNND metric and SVD algorithm. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2004). IEEE, pp. 592–599.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artif. Intell. 97, 273–324.

Law, M.H.C., Figueiredo, M.A.T., Jain, A.K., 2004. Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Machine Intell. 26 (9), 1154–1166.

Mitra, P., Murthy, C.A., Pal, S.K., 2002. Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Anal. Machine Intell. 24 (4).

Modha, Dharmendra, Spangler, Scott, 2003. Feature weighting in *k*-means clustering. Machine Learning 52, 217–237.

Motoda, Hiroshi, Liu, Huan, 2002. Data Reduction: Feature Selection. Oxford University Press Inc., pp. 208–213.

Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Lett. 15, 1119–1125.

Yang, Jihoon, Honavar, Vasant, 1997. Feature subset selection using A genetic algorithm. In: Koza, John, R., Deb, Kalyanmoy, Dorigo, Marco, Fogel, David B., Garzon, Max, Iba, Hitoshi, Riolo, Rick L. (Eds.), Genetic Programming 1997: Proceedings of the Second Annual Conference, Stanford University, CA, USA, 13–16 July. Morgan Kaufmann, p. 380.

Yang, Jihoon, Honavar, Vasant G., 1998. Feature subset selection using a genetic algorithm. IEEE Intell. Systems 13 (2), 44–49.

Yu, Lei, Liu, Huan, 2004. Efficient feature selection via analysis of relevance and redundancy. J. Machine Learning Res. 5, 1205–1224.