A *K*-means-based Algorithm for Projective Clustering

Mohamed Bouguessa
Department of Computer Science
University of Sherbrooke
Quebec, Canada, J1K 2R1
m.bouguessa@usherbrooke.ca

Shengrui Wang
Software School
Xiamen University
Fujian 361005, China
s.wang@usherbrooke.ca

Qingshan Jiang
Software School
Xiamen University
Fujian 361005, China
qjiang@xmu.edu.cn

Abstract

In this paper, a new algorithm for projective clustering is proposed. The algorithm consists of two phases. The first phase performs attribute relevance analysis by detecting dense regions in each attribute, thereby allowing irrelevant attributes and outliers to be captured and eliminated. Starting from the results of the first phase, the second phase aims to uncover clusters in different subspaces. The clustering process is based on the k-means algorithm, with the computation of distance restricted to subsets of attributes where object values are dense.

1. Introduction

One of the primary tasks in data mining is clustering, which aims to discover homogenous groups (clusters) of data according to a certain similarity measure [1]. In many practical clustering algorithms, the concept of similarity is based on the distance. Recent research [2] reveals that in high-dimensional data, the distance between any two data points becomes almost the same, making it difficult to differentiate similar data points from dissimilar ones.

Feature selection techniques are developed to reduce the dimensionality of the data by eliminating redundant and irrelevant features [3]. However, in real-life applications, different clusters may exist in different subspaces spanned by different dimensions. In such cases, dimension reduction using a conventional feature selection technique may lead to substantial information loss [4].

The following example provides an idea of the difficulties encountered by clustering algorithms and feature selection techniques. Figure 1 illustrates a data set composed of n data points $\{x_1, \ldots, x_n\}$ in 10-dimensional space, with four clusters that have their own correlated dimensions (e.g., cluster 1 exists in dimensions A_3, A_5, A_6, A_8, A_{10}). For every correlated dimension, cluster points are assumed to be distributed according to a normal distribution while the remaining data points are uniformly distributed. Finally, a small percentage of the data points are outliers.

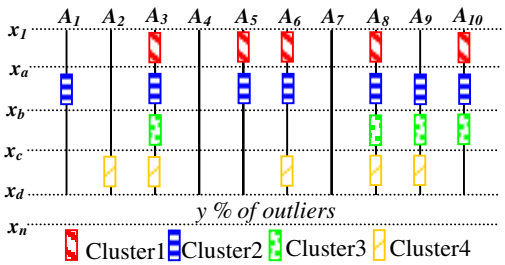

Figure 1. Example of high-dimensional data

For such an example, a traditional clustering algorithm is likely to fail to find the four clusters. Although feature selection techniques can reduce the dimensionality of the data by eliminating irrelevant attributes such as A_4 and A_7, there is an enormous risk that they will also eliminate relevant attributes such as A_1. This is due to the presence of many sparse data points in A_1, where cluster 2 is in fact present.

To address these problems, new classes of projective clustering methods have emerged. A projected cluster is a subset S of data points, together with a subspace of dimensions, D, such that the points in S are closely clustered in D [4]. For instance, the third cluster in the data set presented in Figure 1 is $(S_3, D_3) = (\{x_b, \ldots, x_c\}, \{A_3, A_8, A_9, A_{10}\})$.

The problem of finding clusters in different subspaces has been addressed in [4]. The algorithm PROCLUS, which is a variant of the k-medoid method, finds the subspace dimensions of each cluster via a process of evaluating the locality of the space near it. ORCLUS [5] is the extended version of PROCLUS that looks for non-axis parallel subspaces. Both algorithms require the user to provide the average

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

dimensionality of the subspace, which is very difficult to do in real-life applications

FINDIT [6] is another projective clustering algorithm which is similar to PROCLUS in structure but uses a unique measure called the *Dimension Oriented Distance* (*DOD*) rather than a classical distance function. In [7], Ng et al. propose an algorithm called EPCH. This algorithm performs projective clustering by histogram construction. For another survey of projective clustering algorithms, see [8].

In this paper, we present an algorithm named PCKA (Projective Clustering based on the *K*-means Algorithm), which is able to: 1) detect irrelevant attributes and outliers; and 2) discover clusters in subspaces spanned by different combinations of dimensions. In contrast to PROCLUS, our algorithm does not require the user to specify the dimensionality of the projected clusters.

The remainder of this paper is organized as follows. Our projective clustering algorithm is presented in Section 2. Section 3 presents the experiments and the performance results. Our conclusions are given in Section 4.

2. The Algorithm PCKA

Let *DB* be a data set of *d*-dimensional points, where the set of attributes is denoted by $A=\{A_1, A_2, \ldots, A_d\}$. Let $X=\{x_1, x_2, \ldots, x_n\}$ be the set of *n* data points, where $x_i=(x_{i1}, \ldots, x_{ij}, \ldots, x_{id})$. Each x_{ij} *(i=1, …, n; j=1, …, d)* corresponds to the value of data point x_i on attribute A_j. In what follows we will call x_{ij} a *1-d point*. In order to uncover clusters in different subspaces, PCKA proceeds in two phases: attribute analysis and subspace clustering.

The main focus of the first phase of PCKA is to detect and eliminate irrelevant attributes and outliers. Irrelevant attributes in clustering contain noise and values that tend to be uniformly distributed while relevant ones are located in dense regions. These dense regions represent the 1-d projection of some clusters [7]. By detecting such regions in each attribute we are able to discriminate between relevant and irrelevant attributes.

In order to detect densely populated regions in each attribute we compute a sparseness degree λ_{ij} for each *1-d point* x_{ij} by measuring the variance of its *k* nearest (*1-d point*) neighbors.

Definition 1. The sparseness degree of x_{ij} is defined as

$$\lambda_{ij} = \frac{\sum_{y \in p_i^j(x_{ij})} (y - C_i^j)^2}{k+1}$$

where $p_i^j(x_{ij}) = \{nn_k^j(x_{ij}) \cup x_{ij}\}$ and $\left|p_i^j(x_{ij})\right| = k+1$.

$nn_k^j(x_{ij})$ denotes the set of *k*-nn of x_{ij} in attribute A_j and C_i^j is the centre of the set $p_i^j(x_{ij})$, i.e

$$C_i^j = \frac{\sum_{y \in p_i^j(x_{ij})} y}{k+1}.$$

Intuitively, a large value of λ_{ij} means that x_{ij} belongs to a sparse region, while a small one indicates that x_{ij} belongs to a dense region.

Calculation of the *k* nearest neighbors is, in general, an expensive task, especially when the number of data points *n* is very large. However, since we are searching for the *k* nearest neighbors in a one-dimensional space, we can perform the task in an efficient way by pre-sorting the values in each attribute and limiting the number of distance comparisons to a maximum of *2k* values.

In order to identify dense regions in each attribute, we are interested in all sets of x_{ij} having a small sparseness degree, determined by a pre-defined threshold $\varepsilon \in \Re$.

Definition 2. Let $\varepsilon \in \Re$, where ε is a density threshold.
If $\lambda_{ij} < \varepsilon$ then $z_{ij} = 1$ and x_{ij} belong to a dense region;
else $z_{ij} = 0$ and x_{ij} belong to a sparse region.

From definition 2, we obtain a matrix of binarized degrees of density $Z_{(n*d)}$ which contains the information on whether each data point falls into a dense region of an attribute. For example, Figure 2 illustrates the matrix *Z* for the data used in the example in Section 1.

		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
Cluster 1	x_1	0	0	1	0	1	1	0	1	0	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	1	0	1	1	0	1	0	1
Cluster 2	x_a	1	0	1	0	1	1	0	1	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		1	0	1	0	1	1	0	1	1	1
Cluster 3	x_b	0	0	1	0	0	0	0	1	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	1	0	0	0	0	1	1	1
Cluster 4	x_c	0	1	1	0	0	1	0	1	1	0
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	1	1	0	0	1	0	1	1	0
Outlier s	x_d	0	0	0	0	0	0	0	0	0	0
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	x_n	0	0	0	0	0	0	0	0	0	0

Figure 2. The matrix $Z_{(n*d)}$.

In order to capture irrelevant attributes and outliers, we use the properties of the matrix *Z*. For this purpose we propose the following rules:

Rule 1. An attribute A_j *(j=1,…, d)* is irrelevant if

$$\sum_{i=1}^{n} z_{ij} = 0.$$

Rule 2. A point x_i *(i=1,…, n)* is an outlier if

$$\sum_{j=1}^{d} z_{ij} = 0$$

These two rules suit our purpose of eliminating irrelevant attributes and outliers because they are based

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

on the fact that the sparseness degree λ_{ij} in such situations is much larger than ε. Figure 2 illustrates the case, corresponding to the example in Figure 1, in which attributes A_4 and A_7 are irrelevant and the points $x_d, \ldots, x_n$ are outliers.

It is clear that the computation of z_{ij} depends on the two input parameters ε and k. Although, it is difficult to formulate and obtain optimal values for these parameters, it is easy for us to propose guidelines for their estimation. According to our experiments, $\lambda_{ij} \leq 0.1$ is a good indication that x_{ij} belongs to a dense region. Therefore, setting $0<\varepsilon \leq 0.1$ is a reasonable choice. In practice, since the sparseness degree λ_{ij} is the indicator for dense regions and its values vary significantly depending on the attributes, we first normalize all the λ_{ij} for each attribute A_j by mapping them onto the interval [0, 1] before applying the thresholding. The choice of k can be made more in a more straightforward fashion. To gain a clear idea of the sparseness of the neighborhood of a point, we use $k \leq \sqrt{n}$ as a general guideline. Further investigation into the selection of these parameters is needed.

Based on the properties of the matrix Z, we can eliminate irrelevant attributes and outliers from the data set *DB*, and the corresponding columns and rows from the matrix Z, to obtain a reduced data set *RDB* and its associated matrix of binarized degrees of density, U. The dimension of *RDB* is *dim = d - number of irrelevant attributes* and the number of data points *nb = n – number of outliers.*

Phase 1 of PCKA is summarized as follows.

Input: Data set *DB*, ε, k.

Output: Data set *RDB*, matrix U, set of irrelevant attributes *IA*, set of outliers *OUT*, number of irrelevant attributes *ni*, number of outliers *no*.

1. Based on Def. 1 and Def. 2, compute the matrix Z.
2. Based on Rule 1 and Rule 2:
   - 2.1. *IA* = set of irrelevant attributes; *ni*= |*IA*| and *RDB* = *DB-IA*
   - 2.2. *OUT* = set of outliers; *no* = |*OUT*| and *RDB* = *RDB – OUT*
   - 2.3. Compute the matrix $U_{(nb*dim)}$

The main focus of Phase 2 of PCKA is to cluster the data points of *RDB*. For this purpose we use the *k*-means algorithm and exploit the properties of the matrix $U_{(nb*dim)}$. *K*-means partitions the data into a number of clusters, each of which is represented by a center. A data point is assigned to a cluster using a distance function, e.g., Euclidian distance, to calculate its distance from the center of the cluster. However, this is not an effective approach with high-dimensional data, because each dimension is equally weighted when computing the distance between two points. To solve this problem, we associate the binarized degrees of density u_{lm} *(l = 1,..., nb; m = 1, ..., dim)* in the matrix U to the Euclidian distance. This makes the distance measure more effective because the computation of distance is restricted to subsets where the object values are dense.

Formally, this weighted Euclidean distance between a point x_l and the cluster center v_c *(c = 1, ..., nc; nc is the number of clusters in data)* is defined as:

$$dist(x_l, v_c) = \sqrt{\sum_{m=1}^{\dim} u_{lm} \times (x_{lm} - v_{cm})^2} \qquad (1)$$

Phase 2 of PCKA is summarized as follows:

Input: Data set *RDB*, number of clusters *nc*, matrix U, termination threshold: *0 < s =small positive value*.

Output: cluster centers v and matrix $W_{(nb*nc)}$ of the membership degrees of each data point in each cluster.

1. Initialize the cluster centers v_c^0 *(c = 1, ..., nc)* randomly.
2. Compute the membership matrix $W_{(nb*nc)}$
   for *l = 1, ..., nb and j = 1, ..., nc*
   if $dist(x_l, v_c) < dist(x_l, v_j)$ then $w_{ij} = 0$
   else $w_{ij} = 1$
3. Compute the cluster centers
   $$v_c^1 = \frac{\sum_{l=1}^{nb} (w_{lc} \times u_l \times x_l)}{\sum_{l=1}^{nb} w_{cl}}; c = 1, \ldots, nc$$
4. If $\| v_c^0 - v_c^1 \| < s$ then go to *step 5* ($\|.\|$ is the usual Euclidean distance);
   else let $v_c^0 = v_c^1$ *(c=1, ..., nc);* go to *step 2*.
5. End of the algorithm.

As we can see, there are two significant modifications to the basic *k*-means algorithm. The first is the use of the weighted Euclidean distance in step 2; the second, the use of the matrix U in step 3 for computing the cluster centers. The use of the matrix U restricts the computation of distance and cluster centers to those subsets of attributes in which the data points belong to dense regions. The efficiency of PCKA is demonstrated in the following section.

3. Experimental Results

In this section we evaluate the suitability and accuracy of PCKA. For this purpose we generated four different data sets based on the method suggested by Aggarwal et al. [1]. Each data set contains clusters of different sizes in different subspaces of varying dimensionality. The number of data points n, the number of clusters *nc* and the average number of relevant dimensions per cluster *av* differ for each data

set. There are a certain percentage of outliers in each data set. Figure 3 summarizes the main characteristics of the generated data.

DB	n	Dimension	nc	av	Outliers
Data1	4000	20	4	8	10%
Data2	5000	30	4	12	15%
Data3	6000	40	5	17	20%
Data4	8000	60	6	25	30%

Figure 3. Generated data set characteristics

We compared the performance of our algorithm, in terms of clustering quality, with the recently published algorithm named EPC2[1] [7]. For this purpose, we measured the accuracy of clustering by matching the points in input and output clusters. We adopted the definition of accuracy given in [5], which measures the percentage of correctly partitioned data points. Due to space limitations, we restrict our discussion to this quality measure. In all our experiments, we set $\varepsilon = 0.1$ and $k = \sqrt{n}$. Table 1 illustrates the results of PCKA and EPC2.

Table 1. Results of PCKA and EPC2

	PCKA	EPC2
Data set	% Accuracy	% Accuracy
Data1	99.58	98.77
Data2	95.86	71.24
Data3	94.97	75.00
Data4	90.85	60.57

As we can see from Table 1, PCKA is able to achieve highly accurate results in different situations involving data sets with different characteristics. EPC2 succeeded in achieving high accuracy with Data1, but its accuracy drops significantly with the other three data sets. Here, we believe that the accuracy of EPC2 is greatly affected by the increasing value of the data's dimensionality and the change in the values of *av*. On the other hand, we remark that the accuracy of both PCKA and EPC2 tends to decrease as the percentage of outliers increases. This is because, since the outliers are randomly generated throughout the entire space, it is highly probable that some of them have been placed inside clusters. Therefore, both algorithms consider these points as points that belong to clusters.

We conclude with an experiment on a real data set, the Wisconsin Diagnostic Breast Cancer data set, obtained from the UCI Machine Learning Repository. In this data, $n = 569$, $d = 30$ and $nc = 2$. The accuracy is 93.49% for PCKA, while EPC2 fails completely, returning two empty clusters and classifying all data points as outliers. This experiment confirms the suitability of PCKA observed on the generated data sets.

[1]The implementation of the algorithm is available at: http://appsrv.cse.cuhk.edu.hk/~kdd/clustering/epc.html

4. Conclusion

We have proposed an efficient distance-based clustering algorithm for the challenging problem of high-dimensional clustering. Experiments show that PCKA significantly improves the quality of clustering with high-dimensional data. The accuracy achieved by PCKA results from the restriction to subsets of attributes imposed on the distance computation, and the initial selection of these subsets. Using this approach, we believe that many distance-based clustering algorithms could be adapted to cluster high-dimensional data sets.

There are some issues that need be explored in order to enhance the performance of PCKA. We plan to propose a more systematic way to set the parameter ε. We also plan to extend the scope of Phase 1 of PCKA from attribute relevance analysis to attribute relevance and redundancy analysis. This seems to have been ignored by all of the existing projective clustering algorithms.

References

[1] A. K. Jain, M. N. Mutry, and P.J. Flynn, "Data Clustering: a review", ACM Computing surveys, vol. 31, no. 3, pp. 264-323, 1999.

[2] K. Beyer, J. Goldstein, R. Ramakrishan, and U. Shaft, "When Is Nearest Neighbor Meaningful", Proc. Int'l Conf. Database Theory, 1999.

[3] H. Liu and L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering", IEEE Trans. Knowledge and Data Eng., vol. 17, no. 3, pp. 1-12, 2005.

[4] C.C. Aggarwal, C. Procopiuc, J. L. Wolf, P.S. Yu, and J.S. Park, "Fast Algorithm for Projected Clustering", Proc. SIGMOD Conf., 1999.

[5] C.C. Aggarwal and P.S. Yu, "Refining Clustering for High Dimensional Applications", IEEE Trans. Knowledge and Data Eng., vol. 14, no. 2, pp 210-225, 2002.

[6] K. G. Woo, J. H. Lee, M. H. Kim, and Y. J. Lee, "FINDIT: A fast and intelligent subspace clustering algorithm using dimension voting", Information and Software Technology, vol. 46, no. 4, pp. 255-271, 2004.

[7] E. K. K. Ng, A. W. F, and R C. W. Wong, "Projective Clustering by Histograms", IEEE Trans. Knowledge and Data Eng., vol. 17, no. 3, pp. 369-383, 2005.

[8] L. Parsons, E. Haque, and H. Liu, "Subspace Clustering for High Dimensional Data: A Review", ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90-105, 2004.