An Algorithm for High-Dimensional Traffic Data Clustering

Pengjun Zheng and Mike McDonald

Transportation Research Group, University of Southampton,
Highfield, Southampton, SO17 1BJ, United Kingdom
{p.zheng, mm7}@soton.ac.uk

Abstract. High-dimensional fuzzy clustering may converge to a local optimum that is significantly inferior to the global optimal partition. In this paper, a two-stage fuzzy clustering method is proposed. In the first stage, clustering is applied on the compact data that is obtained by dimensionality reduction from the full-dimensional data. The optimal partition identified from the compact data is then used as the initial partition in the second stage clustering based on full-dimensional data, thus effectively reduces the possibility of local optimum. It is found that the proposed two-stage clustering method can generally avoid local optimum without computation overhead. The proposed method has been applied to identify optimal day groups for traffic profiling using operational traffic data. The identified day groups are found to be intuitively reasonable and meaningful.

1 Introduction

Data clustering is a process of finding natural groupings in a dataset so that data points belonging to same group are more similar than those belonging to different groups [1]. A number of clustering algorithms have been proposed in the past [2]. The most widely used clustering methods are c-means (or K-means) and fuzzy c-means algorithms ([3]-[5]).

The c-means (CM) is a hard clustering method where each point of the dataset belongs to one of clusters exclusively. The fuzzy c-means (FCM) allows for partial membership of belonging to several clusters, i.e. a data point can belong to more than one cluster with different degrees of memberships. This allows for ambiguity in the data and provides a natural partition method compatible with human inaccurate reasoning. The optimal partition is achieved by minimising a specified objective function, usually the weighted sum of squared Euclidean distances between data points and cluster centres [4]-[5].

Both CM and FCM find the optimal partition using iterative procedures. The dataset is initially partitioned into c clusters randomly. The algorithm then iteratively updates the c centres that implicitly represent the partition. The c-means algorithms have been successfully used in many data clustering applications, mainly for its simplicity and efficiency.

The objective function of CM and FCM is non-convex so that the algorithm may converge to a local optimal solution that is significantly inferior to the desirable

L. Wang et al. (Eds.): FSKD 2006, LNAI 4223, pp. 59 – 68, 2006.

global optimum [6]-[7]. To overcome this drawback, clustering methods using Genetic Algorithms have been proposed to find global optimums [8]-[9]. However, GA-based clustering methods are usually computational expensive [10] which may take thousands of iterations to find a global optimum. The approach may only be suitable for low-dimensional and small data sets.

Traffic data refer to time-series data collected using traffic monitoring equipments. With the rapid development in the Intelligent Transportation Systems, large scale traffic monitoring has now become more and more a commonplace. A typical regional traffic surveillance system usually has more than one thousand sensors collecting data at 1-minute interval round the clock. Time-series databases are often extremely large. As traffic conditions evolve on a daily basis (1440 minutes), 1-min traffic data collected in one day is a time series with a length of 1440, which can usually be considered as a point in 1440-dimensional space. In this sense, traffic databases are characterized by high-dimension and large size.

There has been much interest in the Knowledge Discovery in Data (KDD) from traffic data through data clustering, i.e. identification of natural day groups for traffic profiling purpose so that accurate historical traffic profiles can be constructed from those days with similar traffic conditions. This can theoretically be realised using time-series traffic data from a prolonged time period (e.g. 1 year) based on standard data clustering methods. However, the result may suffer from local optimum if CM and FCM are used, and may not be computationally feasible for GA-based algorithms because of the dimension and size of the data.

In this paper, a two-stage fuzzy clustering method is proposed. Dimensionality reduction is first applied on the original data so that a compact representation of the data is derived which contains only the main features of the original data. By clustering the low-dimensional compact data, optimal partition of the compact data is found. The identified partition is then used as the initial partition for the clustering of the complete data, thus effectively reduces the possibility of local optimum. The proposed method has been applied to identify optimal day groups for traffic profiling using operational traffic data. It is found that the two-stage algorithm is able to find the global optimum without increasing computation demands.

2 Fuzzy c-Means Clustering

The methodology for partition a data set into c fuzzy clusters, the fuzzy c-means (FCM) clustering algorithm, has been developed by Dunn and generalised by Bezdek [4]-[5]. FCM is an iterative clustering method that produces an optimal c partition by minimising the weighted sum objective function J_{FCM}

$$J_{FCM} = \sum_{k=1}^{n} \sum_{i=1}^{c} [A_i(x_k)]^q d^2(x_k, v_i) \tag{1}$$

where $\boldsymbol{X=\{x_1, x_2, \ldots, x_n\}} \subset R^s$ is the data set in the s-dimension of the input variables, n is the number of data points, c is the number of clusters with $2 \le c < n$, $A_i(x_k) \in [0,1]$ is the degree of membership of $\boldsymbol{x_k}$ in the i^{th} cluster, q is the weighing

exponent, v_i is the prototype of the centre of cluster i, $d^2(x_j,v_i)$ is the Euclidean distance between object x_k and cluster centre v_i.

The optimal fuzzy set can be determined by an iterative process where J is successively minimised whilst $V=[v_1, v_2, \ldots, v_c]$ and $A=[A_1, A_2, \ldots, A_c]$ are updated using (2) and (3) at m^{th} iteration:

$$v^{(m)}{}_i = \frac{\sum_{k=1}^{n} [A_i{}^{(m)}(x_k)]^q x_k}{\sum_{k=1}^{n} [A_i{}^{(m)}(x_k)]^q} \tag{2}$$

$$A_i{}^{(m+1)}(x_k) = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}}, \text{ if for } 1 \le i \le c, \left\| x_k - v_i \right\| \ne 0 \tag{3}$$

$$= 0 \quad \text{otherwise}$$

The detailed iterative procedure can be described as follows:

1. initialise $A_i^{(m)}(x_j)$ for all j, $m=0$
2. calculate v_i based on $A^{(m)}$ using Eq. (2)
3. Compute $A^{(m+1)}(x_j)$ for all j, using Eq. (3)
4. If $A^{(m+1)}$ and $A^{(m)}$ are close enough, e.g. $\left\| A^{(m+1)} - A^{(m)} \right\| < \varepsilon$, stop. Else go to step 2.

The CM algorithm can be regarded as a special case of FCM only that the membership function A is a two-value function:

$$A_i(x_k) = 1, \text{ for } 1 \le i \le c, \left\| x_k - v_i \right\| \le \left\| x_l - v_i \right\|, \; l \ne k \tag{4}$$

$$= 0 \quad \text{, otherwise}$$

Some widely used implementations of FCM (e.g. Matlab) generate initial partition randomly. An outline of the FCM initialisation in Matlab is shown below:

```
function InitFCM(c,n)
     generate c(number of cluster) by n(number of data
     points) matrix of random number;
     calculate column sum;
     scale random numbers in each column by the column
     sum so that the column sum of the scaled partition
     matrix is always equal to one;
```

As FCM is sensitive to the initial partition, it may converge to a partition that is a local optimum under some initial partitions. This could be random in nature as the initial partition is generated randomly.

3 Dimensionality Reduction of High-Dimensional Data

A time series of length s can be considered as a point in s-dimensional space R^s. Dimensionality reduction is a technique of decomposition and representation of the data in a reduced parameter space R^s, where $S<s$. Major dimensionality reduction methods include Discrete Fourier Transform (DFT) [11], Singular Value Decomposition (SVD) and Discrete Wavelet Transform (DWT). Dimensionality reduction is achieved by ignoring some details in the source data. For instance, DFT decomposes a signal (time series) of length s into s sine/cosine waves that can be recombined into the original signal. Most of the Fourier coefficients have very low amplitude and can be discarded without much loss of information. The signal can still be approximately reconstructed from those few high amplitude Fourier coefficients. It is therefore possible to approximately represent the original time series using a few coefficients in another parameter space (e.g. frequency domain for DFT). A simple but rather useful technique for data dimensionality reduction is aggregate approximation [12], in which the consecutive data values within a time interval are collapsed by a single value:

$$\bar{x}_j = f(x \mid s_j, e_j) \tag{5}$$

where s_j and e_j be indices of dimensions in $\boldsymbol{x}$ for interval j such that $s_j \leq e_j$. $f()$ is a mapping (e.g. the average and weighted average).

Depending on the choice of $f()$ and $[s_j, e_j]$, s-dimensional data $\boldsymbol{x}= \{x^1, x^2, \ldots, x^s\}$ can be compacted into S-dimensional data $\bar{x} = \{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^S\}$, where $S<s$. In the simplest form, an equal-width aggregation scheme by averaging source data in equal aggregation intervals can effectively reduce data dimensionality by s/S times:

$$\bar{x}_j = \frac{S}{s} \sum_{i=s_j}^{e_j} x_i \tag{6}$$

The different between the original data and the compact data at any $[s_j, e_j]$ will be:

$$\Delta x_i = x_i - \bar{x}_j \tag{7}$$

Typical traffic flow data and its compact representation under reduced space are shown in Figure 1. The original data is in 1-min interval (1440 dimensional). By aggregating the data at one-hour interval, the compact data can be regarded as 24-dimensional. It can be observed that the compact data basically retain the global shape of the original data, while the differences are mainly short-term variations. It can also be found that variations within each time interval are significantly different; an indication that the equi-width aggregate approximation may not be optimal. It is therefore possible to aggregate data in varying intervals that can capture the global pattern of the data better. This is equivalent to finding optimal S pairs of aggregation intervals $\{(s_1, e_1), (s_2, e_2), \ldots, (sS, eS)\}$, to minimise the total:

$$SSE = \sum_{j=1}^{S} \sum_{i=sj}^{ej} (x_i - \bar{x}_j)^2 \tag{8}$$

where $s_{j+1} = e_j + 1$. This will result in an optimal aggregate approximation scheme in terms of approximation errors (by minimising SSE).

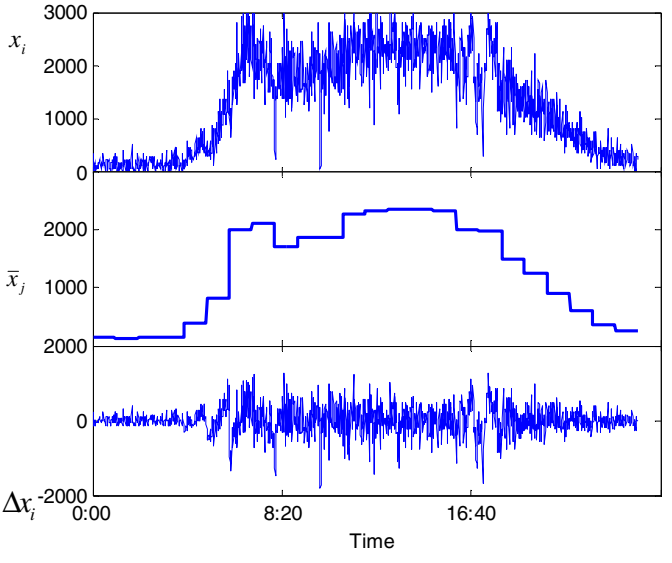

Fig. 1. Typical high-dimensional traffic data and its compact representation

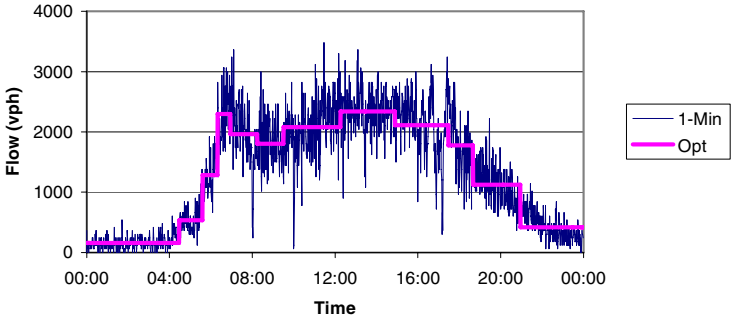

Fig. 2. Compact data based on optimal aggregation approximation

There exist many efficient algorithms to solve this kind of optimisation problem. In this research, the algorithm based on dynamic programming as proposed in [13] has been implemented to compute optimal aggregation intervals. The compact data at a reduced dimension of 12 is shown in Figure 2. It can be clearly observed that the compact data follow the global trends of the original data well. The SSE is even smaller than that based on equi-width aggregation approximation at dimension of 24.

4 Two-Stage Clustering Method

A two stage clustering method has been proposed in this research for clustering high-dimensional traffic data. The clustering is initially performed using low-dimensional

data obtained through aggregation approximation. The identified optimal partition matrix is then used as the initial partition in the second stage clustering using full-dimensional data. The clustering based on compact data may be less likely to converge to local optimums. The second stage clustering based on full-dimensional data then looks for the further optimal partition by considering all details available in the data. An outline of the two-stage FCM is described as follows:

```
program 2stgFCM(c,n)
generate compact data by applying dimensionality reduc-
tion techniques;
apply FCM on the compact data (using random initial
partition matrix) to find the optimal partition matrix;
apply FCM on full-dimensional data using the optimal
partition matrix identified as the initial partition;
```

The optimal partition in data clustering is achieved by minimising the weighted sum of squared Euclidean distances between data points and cluster centres. It is not difficult to observe that the main components of Euclidean distance between two points in original space are preserved in the reduce space:

Let *x* and *y* are two points in *s*-dimensional space, *S* is the target dimensionality,

Let $\Delta x_i = x_i - \bar{x}$, $\Delta y_i = y_i - \bar{y}$ within each $[s_j, e_j]$, *j=1,2, …, S,* the squared Euclidian distance between *x* and *y* will be:

$$\begin{aligned}
\|x-y\|^2 &= \sum_{j=1}^{S} \sum_{i=sj}^{ej} (x_i - y_i)^2 = \sum_{j=1}^{S} \sum_{i=sj}^{ej} \left[(\bar{x}_j + \Delta x_i) - (\bar{y}_j + \Delta y_i)\right]^2 \\
&= \sum_{j=1}^{S} \left[(e_j - s_j + 1)(\bar{x}_j - \bar{y}_j)^2\right] + \sum_{j=1}^{S} \sum_{i=sj}^{ej} \left[(\Delta x_i - \Delta y_i)^2\right] \\
&= \|\bar{x}' - \bar{y}'\|^2 + \sum_{j=1}^{S} \sum_{i=sj}^{ej} \left[(\Delta x_i - \Delta y_i)^2\right] \qquad (9)
\end{aligned}$$

where $\bar{x}'_j = \sqrt{(e_j - s_i + 1)} \cdot \bar{x}_j$ and $\bar{y}'_j = \sqrt{(e_j - s_i + 1)} \cdot \bar{y}_j$ for *j=1,2,…,S*

It can be found that the optimal partition found by clustering data at reduced dimensionality (based on $\|\bar{x}' - \bar{y}'\|$) is equivalent to clustering the full-dimensional data by ignoring local variations of $\sum_{j=1}^{S} \sum_{i=sj}^{ej} \left[(\Delta x_i - \Delta y_i)^2\right]$, thus is a high-level partition that is unlikely to converge to local optimum. In addition, clustering based on the compact data is usually computational inexpensive as the reduced dimensionality is often much lower compared with the original dimensionality. Repeated clustering based on the compact data is possible to reduce the chances of converging to a local optimum.

5 Application

The proposed two-stage clustering algorithm has been applied to find natural day groups for traffic profiling purposes. The traffic profile refers to the historically

typical traffic conditions that represent the past experience and also the future expectations. The most common day groups for traffic profile are weekday, e.g. Monday - Sunday. This is intuitively reasonable as it can capture weekly patterns of traffic conditions. However, many other partitions are also possible, e.g. a partition between public holidays and working days. A promising approach of identifying these natural day groups is through learning from the operational data using data clustering techniques. Traffic flow data for each day can be regarded as a data point in s-dimensional space. By applying clustering on traffic data of more than one year (365 data points), the natural groups (clusters) can be identified from the optimal partition matrix using proper defuzzification process, e.g., the maximum membership procedure that assigns the data point (a day's traffic data) k to the cluster (day group) i with the highest membership

$$C_k = \arg_i \{\max(A_i(x_k)\}, i=1, 2, \ldots, c.$$

where $A_i(x_k) \in [0,1]$ is the degree of membership of $\boldsymbol{x_k}$ in the i^{th} cluster.

5.1 The Data

Operational traffic data collected using loop detectors are used to identify natural day groups for traffic profiling. The database consists of 1-min traffic flow data collected over a period of 640 days from 1 April 2004 to 31 December 2005. Data collected on 26 days are incomplete and are not included. In total, traffic data collected on 614 days (884160 readings) are used in the clustering.

The compact data are generated using two aggregate approximation schemes:

- Optimal aggregate approximation to a reduced dimension of 12
- Equi-width aggregate approximation to a reduced dimension of 24

Both direct and the proposed two-stage fuzzy clustering methods are used to cluster data into 3-10 clusters. The two-stage clustering method has also been used in a repeated initial clustering way where clustering over the compact data is repeated more than one time to find the optimal initial partition matrix. This is intended to prevent converging to local optimums in the clustering process of the compact data and is computational affordable as clustering on low-dimensional data demands little CPU time. Thus, four clustering runs are performed for a given pre-defined number of clusters:

- Direct clustering using full-dimensional data
- Two-stage clustering using compact data generated from optimal aggregate approximation method
- Two-stage clustering using compact data generated from Equi-width aggregate approximation method
- Two stage clustering based on the best initial partition obtained by repeatedly clustering using compact data generated using both aggregate approximation methods

5.2 Results

The identified day groups based on a cluster number of 5 are shown in Figure 3. Five natural day groups are dominated by Mon-Tue, Wed-Thu, Fri, Sat and Sun

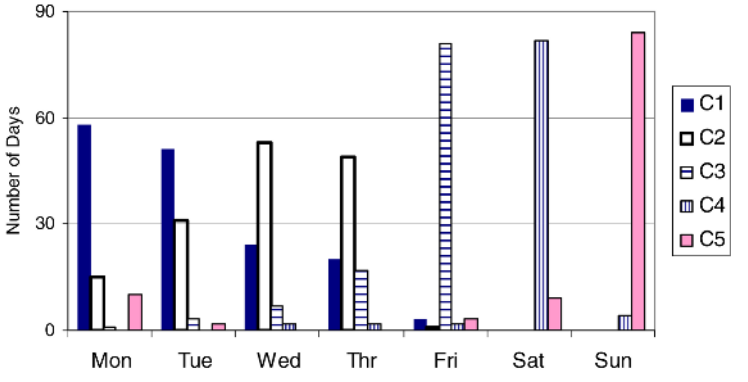

Fig. 3. Distribution of identified day groups over weekdays

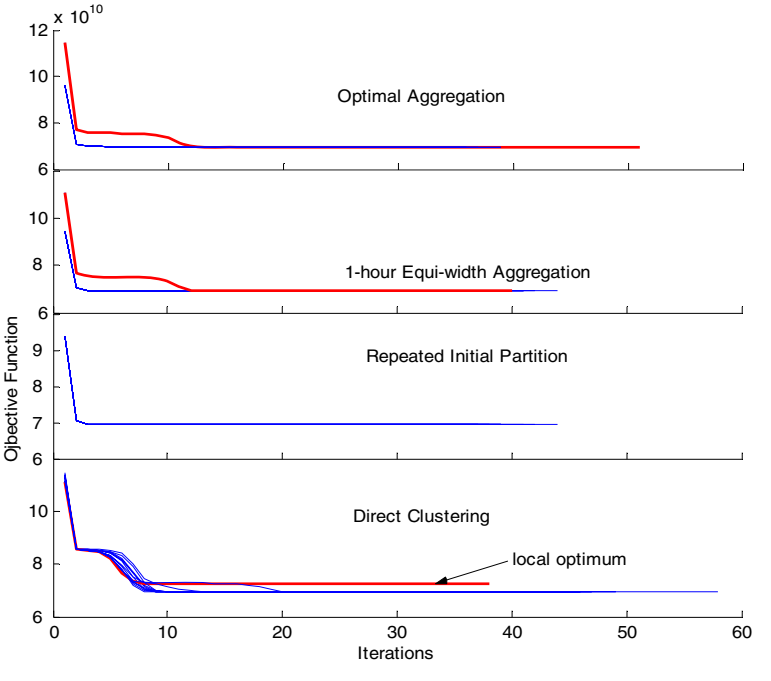

Fig. 4. Objective function values using different clustering procedures

respectively. This is intuitively reasonable as traffic patterns on Saturday and Sunday are usually different from working days. Traffic patterns on Friday have been identified as different from both other working days and non-working days. Traffic conditions on Monday and Tuesday are more alike, so do that on Wednesday and Thursday. It is worth noticing that 10 public holiday Mondays have all been successfully identified as belonging to Sunday group.

The corresponding objective function values evolving over each clustering iteration have been shown in Figure 4. This is based on 12 runs to find the optimal partition of 5-day groups. Convergences to an inferior local optimum have only been observed two times based on direct clustering using full-dimensional data.

The average CPU times based on 12 runs are shown in Table 1. It can be found that the direct clustering method actually takes slightly longer time. This is because the clustering requires more iterations to converge than that of other three methods. The proposed two-stage algorithm can achieve better optimum while not increasing computational demands.

Table 1. CPU times under different clustering schemes (Pentium4 3GHz CPU)

Schemes	1-hour Equi-width aggregation		Optimal Aggregation		Direct Clustering	Repeated Initial partitions	
CPU Time (s)	Initial Partition	Total	Initial Partition	Total	Total	Initial Partition	Total
AVG	0.58	11.82	0.36	11.73	12.53	0.89	12.14
STD	0.06	0.55	0.07	1.34	1.70	0.17	0.72

6 Conclusions

A two-stage algorithm for clustering high-dimensional traffic data has been described in this paper. It has been compared with direct clustering method. The performance of the proposed algorithm has been shown to outperform the standard implementation. The algorithm is less likely to converge to local optimums while the computational demands have not been increased (or even slightly reduced). The application of the algorithm to cluster operational traffic data has produced promising results. The identified natural day groups are intuitively reasonable and meaningful. However, one thing remains un-guaranteed, that is, the identified clusters using the proposed method could still not be the global optimum. The absolute optimum may only be guaranteed by exhaustive enumeration, which is not yet practically feasible for large database due to extremely high computational demands.

References

1. Jain, A. K., Dubes, R. C.: Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ (1988)
2. Jain, A. K., Murty, M. N., Flynn, P. J.: Data Clustering: A Review, ACM Computing Surveys, vol. 31(3) (1999) 264-323
3. Hand, D. J., Krzanowski, W. J.: Optimising k-means Clustering Results with Standard Software Packages, Computational Statistics & Data Analysis, vol. 49(4) (2005) 969-973
4. Dunn, J. C.: A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters, Journal of Cybernetics, Vol. 3 (1974) 32-57
5. Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms, New York, Plenum Press, (1981)
6. Selim, S. Z. Ismail, M. A.: K-means Type Algorithms: A Generalised Convergence Theorem and Characterisation of Local Optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 6(1) (1984) 81-87.
7. Pollard, D.: A Central Limit Theorem for k-Means Algorithm, Annals of Probability, vol. 10 (1982) 919-926.
8. Murthy, C. A. Chowdhury, N.: In Search of Optimal Clusters using Genetic Algorithms. Pattern Recognition Letters, vol. 17 (1996) 825-832
9. Jones, D. Beltramo, M. A.: Solving Partitioning Problems with Genetic Algorithms. Proceedings of Fourth International Conference of Genetic Algorithms, (1991) 442-449.
10. Laszlo, M, Mukherjee, S. A Genetic Algorithm Using Hyper-Quadtrees for Low-dimensional K-means clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (4) (2006) 533-543
11. Vanloan, C. F.: Generalizing Singular Value Decomposition, SIAM Journal on Numerical Analysis, vol. 13 (1) (1976) 76-83
12. Keogh, E.J. Pazzani, M.J.: A simple dimensionality reduction technique for fast similarity search in large time series databases. Lecture Notes in Artificial Intelligence, Vol. 1805. Springer-Verlag, Berlin Heidelberg New York (2000), 122-133
13. Jagadish, H. V. Koudas, N., Muthukrishnan, S., Poosala, V., Sevcik, K. Suel, T.: Optimal Histograms with Quality Guarantees. Proceedings of the 24th VLDB Conference, New York, USA, (1998)