Scalable Model-based Clustering by Working on Data Summaries [1]

Huidong Jin[‡*], Man-Leung Wong[‡], Kwong-Sak Leung[*]
[‡]Department of Computing and Decision Sciences
Lingnan University
Tuen Mun, N.T., Hong Kong
hdjin@ieee.org, mlwong@ln.edu.hk,

[*]Department of Computer Sci. & Eng.
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
ksleung@cse.cuhk.edu.hk

Abstract

The scalability problem in data mining involves the development of methods for handling large databases with limited computational resources. In this paper, we present a two-phase scalable model-based clustering framework: First, a large data set is summed up into sub-clusters; Then, clusters are directly generated from the summary statistics of sub-clusters by a specifically designed Expectation-Maximization (EM) algorithm. Taking example for Gaussian mixture models, we establish a provably convergent EM algorithm, EMADS, which embodies cardinality, mean, and covariance information of each sub-cluster explicitly. Combining with different data summarization procedures, EMADS is used to construct two clustering systems: gEMADS and bEMADS. The experimental results demonstrate that they run several orders of magnitude faster than the classic EM algorithm with little loss of accuracy. They generate significantly better results than other model-based clustering systems using similar computational resources.

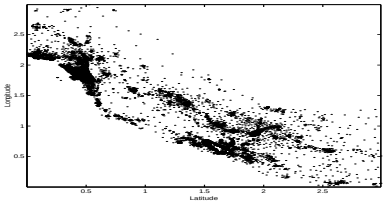
1. Introduction

With the explosive growth of data amassed from business and scientific disciplines, scalable data mining systems become more and more important. They bridge the gap between the limited computational resources and large databases. Their running time grows linearly or sub-linearly with data size, given computational resources such as main memory [1, 7]. Model-based clustering techniques can identify clusters of various shapes and handle complicated databases with different kinds of attributes [3, 4]. Furthermore, they have solid mathematical foundations from the statistics community [9]. These techniques have successfully been applied to numerous real-life applications [2, 3, 6, 9, 14]. Thus, the research on scalable model-based cluster analysis is significant.

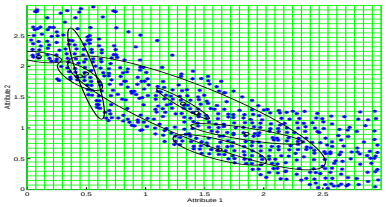
The Expectation-Maximization (EM) algorithm is an iterative procedure for finding maximum likelihood estimates of parameters in a mixture model. EM normally generates more accurate results than hierarchical model-based clustering [6, 10]. Though some attempts have been made to speed up EM [9, 11], EM and its extensions are still computationally expensive for large databases. The lazy EM algorithm [15] evaluates the significance of data items and then operates only on the significant ones. Comparing with EM, its speedup factor is less than three. The scalable EM algorithm [1] uses a heuristically extended EM algorithm to identify compressible regions of data. Then it retains their sufficient statistics in order to load another batch of data, and invokes EM again. Its speedup factor is up to ten [1, 7].

In this paper, we will present our scalable model-based clustering systems which can run several orders of magnitude faster than the classical EM algorithm on large databases. Moreover, there is no or little loss of accuracy. They also can generate much more accurate results than other scalable model-based clustering systems. The basic idea is to categorize a data set into sub-clusters first and then generate a mixture model from their summary statistics directly by a specifically designed EM algorithm. This new EM algorithm works on the summary statistics of the sub-clusters, and it is associated with a pseudo mixture model that is developed to approximate the aggregate behavior of each sub-cluster of data items under the original mixture model. Thus, the new EM algorithm can efficiently generate a good estimate of the original mixture model. For example, for the California housing data plotted in a scaled Latitude-Longitude space in Figure 1(a), our clustering systems generate two mixtures, respectively, from 551 and 780 data summaries of 20,640 data items. The two mixture models clearly describe the housing structure in California, as illustrated in Figures 1(b) and 1(c), where a data summary is indicated by '*', and a Gaussian distribution is indicated by an 'o' and its associated ellipse.

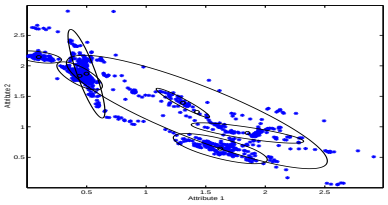
[1]The work is partially supported by Hong Kong RGC Grant CUHK 4212/01E and Lingnan University direct research grant (RES-021/200). We would like to thank the referees for their constructive suggestions.



(a) California housing data.



(b) Clusters generated by gEMADS.



(c) Clusters generated by bEMADS.

Figure 1. Gaussian mixture models generated for the California housing data by two model-based clustering systems.

In the next section, we describe our model-based clustering framework and then apply it to a Gaussian mixture model. Two possible data summarization procedures are described in Section 3. A pseudo mixture model and its associated EM algorithm are developed for the Gaussian mixture model in Section 4. In Section 5, comprehensive experiment results are given on both synthetic and real-life data sets. The conclusive comments come in the last section.

2 Scalable Model-based Clustering

Given a data set $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, the model-based clustering techniques assume that each data item $\mathbf{x}_i = [x_{1i}, \cdots, x_{Di}]^T$ is drawn from a K-component mixture model Φ with probability $p(\mathbf{x}_i|\Phi) = \sum_{k=1}^{K} p_k \phi(\mathbf{x}_i|\theta_k)$. Here p_k is the mixing proportion for the k^{th} cluster ($0 < p_k < 1$ for $k = 1, \cdots, K$, and $\sum_{k=1}^{K} p_k = 1$); $\phi(\mathbf{x}_i|\theta_k)$ is a *component density function* with parameter θ_k. Given Φ, one may get a crisp clustering by assigning the data items $\mathbf{x}_i$ to cluster k where $k = \arg\max_l \{p_l \phi(\mathbf{x}_i|\theta_l)\}$. Thus, a mixture model Φ can be viewed as a clustering solution. Expectation-Maximization (EM) is a widely used algorithm for finding maximum likelihood estimates of Φ iteratively. In each iteration, EM needs to scan the whole set, which prohibits EM from large databases.

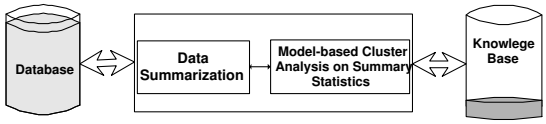


Figure 2. A scalable clustering framework.

There are three approaches to scale-up classic clustering algorithms such as EM. Random sampling is easy to implement, but often brings about inaccuracy [5]. The weighted sampling uses a weighted (pseudo) sample to represent a group of data item [5, 13], and requires slight modification on the classical clustering techniques. However, as shown in Section 5, its performance also depends heavily on the sampling procedures. The third strategy is to construct summary statistics of the large data set on which to base the desired analysis [1, 16].

Our scalable model-based clustering framework falls into the last category. It is motivated by the following observation. In a scalable system, a group of similar data items usually needs to be handled as an object in order to save computational resources. In model-based cluster analysis, a component density function essentially determines clustering results. Thus, for each group of similar data items, a new component density function can be defined in order to remedy the possible loss of clustering accuracy caused by the trivially homogeneous treatment of these data items. For example, a pseudo component density function for their summary statistics can be developed to approximate their aggregate behavior under the original component density function. Finally, its associated clustering algorithm, e.g., an algorithm derived from the general EM algorithm [9], can effectively generate a good mixture model from these summary statistics directly. Thus, as illustrated in Figure 2, our framework has two phases: *data summarization*, which partitions similar data items into exclusive sub-clusters and generates their summary statistics, and *in-memory model-based clustering analysis*, which generates mixture models using the new EM algorithm associated with the pseudo mixture model.

In principle, the framework is applicable to many mixture models, but we focus on Gaussian mixture models in the paper because of their wide applications [1, 3, 6, 14]. In a Gaussian mixture model, each component is a multivariate

Gaussian distribution:

$$\phi(\mathbf{x}_i|\theta_k) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}_i-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_i-\mu_k)\right\}}{(2\pi)^{\frac{D}{2}}\,|\Sigma_k|^{\frac{1}{2}}}. \quad (1)$$

The parameter θ_k consists of a mean vector μ_k and a covariance matrix Σ_k. The classical EM algorithm estimates the parameters to maximize log-likelihood $L(\Phi) = \log\left[\prod_{i=1}^{N} p(\mathbf{x}_i|\Phi)\right]$ iteratively. It alternates between the following two steps.

1. **E-Step**: Given the mixture model parameters, compute the membership probability $t_{ik}^{(j)} = \frac{p_k^{(j)}\phi(\mathbf{x}_i|u_k^{(j)},\Sigma_k^{(j)})}{\sum_{l=1}^{K} p_l^{(j)}\phi(\mathbf{x}_i|u_l^{(j)},\Sigma_l^{(j)})}$.

2. **M-step**: Given $t_{ik}^{(j)}$, update the mixture model parameters from the total N data items for $k = 1, \cdots, K$:

$$p_k^{(j+1)} = \frac{1}{N}\sum_{i=1}^{N} t_{ik}^{(j)} \quad (2)$$

$$\mu_k^{(j+1)} = \frac{\sum_{i=1}^{N} t_{ik}^{(j)}\mathbf{x}_i}{\sum_{i=1}^{N} t_{ik}^{(j)}} = \frac{\sum_{i=1}^{N} t_{ik}^{(j)}\mathbf{x}_i}{N\cdot p_k^{(j+1)}} \quad (3)$$

$$\Sigma_k^{(j+1)} = \frac{\sum_{i=1}^{N} t_{ik}^{(j)}(\mathbf{x}_i-\mu_k^{(j+1)})(\mathbf{x}_i-\mu_k^{(j+1)})^T}{N\cdot p_k^{(j+1)}} \quad (4)$$

As the summary statistics of sub-clusters are the only information passed from the first phase onto the second one, they play an important role in clustering accuracy. They had better reflect the data distribution within the sub-clusters. For example, a Gaussian distribution embodies a covariance matrix, thus the covariance information should be included in the summary statistics for Gaussian mixture models. Thus, we define the following data summary as the summary statistics of a sub-cluster.

Definition 1 *For the m^{th} sub-cluster, the* **data summary** *is defined as a triple: $DS_m = \{n_m, \nu_m, \Gamma_m\}(m = 1, \cdots, M$, where M is the number of sub-clusters). Here, n_m is the number of data items in the m^{th} sub-cluster; $\nu_m = \frac{1}{n_m}\sum_{\text{the } m^{th} \text{ sub-cluster}} \mathbf{x}_i$ is the mean of the data in the m^{th} sub-cluster; and $\Gamma_m = \frac{1}{n_m}\sum_{\text{the } m^{th} \text{ sub-cluster}} \mathbf{x}_i\mathbf{x}_i^T$ is the average of the outer products of the n_m data items.*

The data summary DS_m comprises the zeroth, first, and second moments of the m^{th} sub-cluster of data items. It contains sufficient statistics when the data items within the sub-cluster follow a Gaussian distribution. Taking the data summary as summary statistics for a sub-cluster, a new pseudo mixture model and its associated EM algorithm, EMADS, will be derived in Section 4. Two scalable clustering systems can be developed by cooperating with the two data summarization procedures presented in the next section.

3 Data Summarization Procedures

Our data summarization procedures sum up similar data items into data summaries. The grid-based data summarization procedure partitions a data set by imposing a multidimensional grid structure in the data space, and then incrementally sums up the data items within a cell into its data summaries. That is, the data items within a cell form a sub-cluster. For simplicity, each attribute is partitioned into several equidistant segments by grids. Thus, each cell has the same width in each attribute and has the same volume. For example, for the California housing data in Figure 1(a), we partition each attribute into 40 segments and obtain 551 non-empty sub-clusters, as shown in Figure 1(b). Thus, the cell widths specify the grid structure and the total number of sub-clusters.

To operate within the given main memory, we only store data summaries for the non-empty cells in a data summary array: DS-array. This DS-array has a fixed number of entries, M, according to the given amount of main memory. When a new data item is read, we calculate which cell it is located in. Then we efficiently search for its associated entry in the DS-array by using a hash function. If a corresponding entry is found, its data summary is updated to absorb the data item. Otherwise, a new entry will be allocated to store the data summary of the cell.

The grid-based data summarization procedure adaptively determines the cell width to better use the given main memory. At the beginning, the cell widths are initialized to reasonably small values. If the cell widths are quite small, then the number of non-empty cells may be greater than the number of entries in the DS-array. When the entries in the DS-array are used up, the cell width are increased and the DS-array is rebuilt. The grid-based data summarization procedure merges every two adjacent cells into a larger one along the dimension with the smallest width. Thus, a new data summary is calculated from two old ones without rereading the data.

If the Euclidean distance is used to define the similarity between two data items within sub-clusters, we may employ existing scalable distance-based clustering techniques [12, 13], such as BIRCH [16], to generate sub-clusters. BIRCH scans the data set to build an initial in-memory CF(Clustering Features)-tree, which can be viewed as a multilevel compression of the data set that tries to preserve its inherent clustering structure. Different from clustering

features, data summaries contain covariance information. Hence, in our experiments, BIRCH is modified to generate data summaries. The generated 780 data summaries for the California housing data are illustrated in Figure 1(c).

Both the BIRCH's and the grid-based data summarization procedures attempt to generate data summaries using restricted computational resources. They both read the data set only once. However, the former uses a tree indexing, while the later employs a hash indexing. The former makes better use of memory, while the later is simpler to implement and manipulate [8].

4 EMADS

Our in-memory model-based clustering algorithm directly generates a Gaussian mixture model from data summaries. Our design method is first to introduce a pseudo component density function on the data summaries. The pseudo density function can approximate the aggregate behavior of each sub-cluster of data items under the Gaussian distribution. We then define a pseudo mixture model and derive its associated EM algorithm — EM Algorithm for Data Summaries (EMADS) — according to the general EM algorithm [9].

In order to embody the covariance information in our pseudo component density function to better approximate the aggregate behavior, we simplify the data summary $DS_m = \{n_m, \nu_m, \Gamma_m\}$ into a **simplified data summary** $\mathbf{s}_m = \{n_m, \nu_m, \delta_m\}$. δ_m is the product of the square root of the largest eigenvalue of the covariance matrix $\left(\Gamma_m - \nu_m \nu_m^T\right)$ and its corresponding component vector. According to Theorem 6.1 in [8, p.120], δ_m is a good choice because its outer product best approximates the matrix. Now we introduce a new density function based on the sub-cluster to which a data item $\mathbf{x}_i$ belongs.

Definition 2 *For a single data item $\mathbf{x}_i$ within the m^{th} sub-cluster, its probability under the* **pseudo component density function** *ψ is*

$$\psi(\mathbf{x}_i \in \textit{the } m^{th} \textit{ sub-cluster}|\theta_k) \stackrel{\triangle}{=} \psi(\mathbf{s}_m|\theta_k) = \frac{\exp\left\{-\frac{1}{2}\left[\delta_m^T \Sigma_k^{-1} \delta_m + (\nu_m - \mu_k)^T \Sigma_k^{-1} (\nu_m - \mu_k)\right]\right\}}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}}, \quad (5)$$

where $\theta_k = (\mu_k, \Sigma_k)$ is the parameter for the k^{th} component of the ***pseudo mixture model*** *Ψ.*

If $\delta_m = 0$, the density function in Eq.(5) is equivalent to a Gaussian density function. In general, however, it is not a genuine density function because its integral over the whole data space is often less than 1. Roughly speaking, if the sub-cluster variance $n_m \left(\Gamma_m - \nu_m \nu_m^T\right)$ is small, say, in 1-D case, then the item $\delta_m^T \Sigma_k^{-1} \delta_m$ is also small. In other words, this density function has a higher probability for a data item in a dense area, which accords with the Gaussian density function. Furthermore, its associated EM algorithm, EMADS, can well approximate the aggregate behavior of each sub-cluster of data items, as explained in [8]. Hence, the pseudo component density function is practicable.

With Eq.(5), we get a density function under the pseudo mixture model Ψ for $\mathbf{x}_i$ within the m^{th} sub-cluster, $p(\mathbf{x}_i|\Psi) \stackrel{\triangle}{=} p(\mathbf{s}_m|\Psi) = \sum_{k=1}^{K} p_k \psi(\mathbf{s}_m|\mu_k, \Sigma_k)$.

The pseudo mixture model Ψ has the same parameters as the Gaussian mixture model Φ. In addition, the pseudo component density function approximates the aggregate behavior of each sub-cluster of data items under the Gaussian distribution. Thus, we can find a good Gaussian mixture model Φ by finding a maximum likelihood estimate of Ψ. Given the number of clusters K, we derive a new EM algorithm according to the general EM algorithm [9]. It efficiently gets an estimate of Ψ by maximizing the log-likelihood $L(\Psi) = \sum_{m=1}^{M} n_m \log\left(\sum_{k=1}^{K} p_k \psi(\mathbf{s}_m|\mu_k, \Sigma_k)\right)$ iteratively as follows.

Algorithm 3 EMADS

1. **Initialization***: Set the current iteration $j = 0$ and initialize the parameters, $p_k^{(j)}, \mu_k^{(j)}$ and $\Sigma_k^{(j)}$, such that $\sum_{k=1}^{K} p_k^{(j)} = 1$, and $\Sigma_k^{(j)}$ is symmetric and positive definite.*

2. **E-step***: Given the mixture model $\Psi^{(j)}$, compute the membership probability $r_{mk}^{(j)}$ for all $\mathbf{s}_m$,*

$$r_{mk}^{(j)} = \frac{p_k^{(j)} \psi(\mathbf{s}_m|u_k^{(j)}, \Sigma_k^{(j)})}{\sum_{i=1}^{K} p_i^{(j)} \psi(\mathbf{s}_m|u_i^{(j)}, \Sigma_i^{(j)})}. \quad (6)$$

3. **M-step***: Given $r_{mk}^{(j)}$, update the model parameters using $\mathbf{s}_m$ for all k,*

$$p_k^{(j+1)} = \frac{1}{N} \sum_{m=1}^{M} n_m r_{mk}^{(j)}, \quad (7)$$

$$\mu_k^{(j+1)} = \frac{\sum_{m=1}^{M} n_m r_{mk}^{(j)} \nu_m^{(j)}}{\sum_{m=1}^{M} n_m r_{mk}^{(j)}} = \frac{\sum_{m=1}^{M} n_m r_{mk}^{(j)} \nu_m^{(j)}}{N \cdot p_k^{(j+1)}}, \quad (8)$$

$$\Sigma_k^{(j+1)} = \frac{\sum_{m=1}^{M} n_m r_{mk}^{(j)} \left[\delta_m \delta_m^T + (\nu_m - \mu_k^{(j)})(\nu_m - \mu_k^{(j)})^T\right]}{N \cdot p_k^{(j+1)}}. \quad (9)$$

4. **Termination:** *If $\left|L(\Psi^{(j+1)}) - L(\Psi^{(j)})\right| \geq \epsilon \left|L(\Psi^{(j)})\right|$, set j to $j+1$ and go to step 2.*

Different from the Extended EM algorithm in [1] and the classical EM algorithm for Gaussian mixture models,

EMADS embodies the covariance information explicitly in both E-step and M-step. Thus, more accurate results can be expected. EMADS is also easy to implement because it involves only several equations (Eqs.(6)-(9)). Furthermore, EMADS can surely terminate according to the following theorem.

Theorem 4 *If the matrix* $[\nu_1, \nu_2, \cdots, \nu_M]$ *is full rank, then the log-likelihood* $L(\Psi)$ *for EMADS converges monotonically to a log-likelihood value* $L^* = L(\Psi^*)$ *for a stationary mixture model* Ψ^*.

A proof can be found in [8]. The computational complexity of EMADS is $O(MKD^2I)$, where I is the number of iterations. It is linear with the number of data summaries. The total memory requirement of EMADS is $2MD + MK + KD^2 + KD + K + M$ floating point numbers. Thus, given a large data set, we can choose an appropriate number of sub-clusters, M, to sum up the data set into the given main memory and then generate Gaussian mixture models.

EMADS may be simplified into Weighted Expectation Maximization (WEM) when the component density function ψ is replaced by a Gaussian density function ϕ. Thus, WEM handles each data item in the same way as its corresponding sub-cluster mean vector, and the weights are the cardinality of the sub-clusters.

5 Performance of EMADS

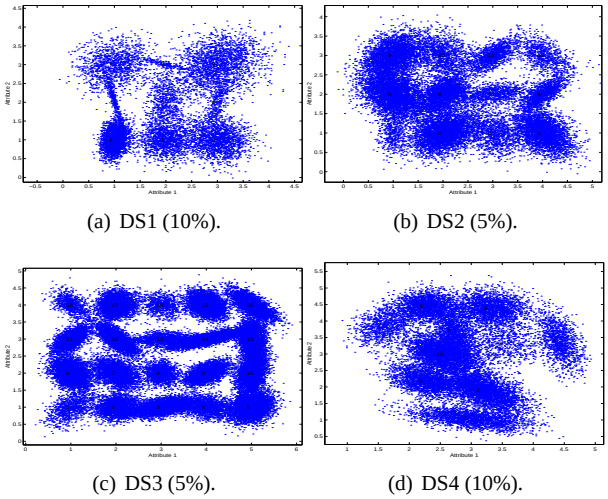
5.1 Methodology and Synthetic Data

Working on the data summaries generated by the grid-based and the BIRCH's data summarization procedures, EMADS is used to construct two clustering systems. We call them gEMADS and bEMADS, respectively. The gEMADS system is mainly specified to examine the sensitivity of EMADS to different data summaries. To highlight their performance, we compare them with several model-based clustering systems designed according to the other scaling-up strategies. They are

The EM algorithm: It is the classical EM algorithm for Gaussian mixture models.

The sampling EM algorithm: It is EM working on 5% random samples. It is referred to as sampEM hereafter.

The gWEM and the bWEM systems: They are WEM working on the data summaries generated by two data summarization procedures, respectively. These two systems may be viewed as density-biased sampling clustering techniques [13].



(a) DS1 (10%). (b) DS2 (5%).

(c) DS3 (5%). (d) DS4 (10%).

Figure 3. Data samples of four data sets.

All these systems are coded in MATLAB and experiments are conducted on a Sun Enterprise E4500 server. EMADS, WEM and EM are initialized with the cluster centers generated by K-means from M random samples. They terminate if the successive log-likelihood modification is within 10^{-5} of current value as used in [10, 15]. All experimental results reported are averaged on 10 independent runs. The data summarization procedures are set to generate at most 4,000 data summaries and use about 8 megabytes main memory. In contrast, there is no restriction on the amount of the main memory used for both EM and sampEM in our experiments.

Since all systems finally generate Gaussian mixture models, the natural evaluation metric is their log-likelihood value. For convenience, we average the log-likelihood over the samples. We also use *the clustering accuracy* to measure the generated mixture models for the synthetic data sets. The clustering accuracy is defined as the proportion of samples which are correctly clustered [10].

Table 1. The parameters of seven synthetic data sets.

DataSet	N(1000)	D	K	M
DS 1	108	2	9	2986
DS2	500	2	12	2499
DS3	1100	2	20	3818
DS4	200	2	10	2279
DS5	200	3	10	3227
DS6	240	4	12	3982
DS7	280	5	14	2391

We generate three groups of synthetic data sets based on random mixture models. The first group has three data sets, and their mean vectors of Gaussian components are located on 2-D grid, as illustrated in Figures 3(a), 3(b), and 3(c), respectively. In the second group of four data sets, two mean vectors of a mixture model are generated together to ensure that their Euclidean distance is 1.0. Hence, these two clusters are close and not well separated. A typical data set is illustrated in Figure 3(d). The parameters of these seven data set are listed in Table 1, where N, D, K, and M indicate the number of data items, the data dimensionality, the number of clusters, and the number of sub-clusters, respectively. The third group of 8 data sets is generated according to a 10-component Gaussian mixture model in 4-dimensional space. They differ only in their data sizes, which increase exponentially from 6,250 to 800,000.

5.2 Sensitivity

The data set (DS1) shown in Figure 3(a) is taken to examine the sensitivity of gEMADS to different data summaries. Figure 4 summaries the clustering accuracy of three clustering systems. The data summarization results are determined by different grid structures. For example, for the first 56*56 grid structure, we partition two attributes into 56 segments respectively and obtain 3,136 cells. These sub-clusters usually do not follow a Gaussian distribution. Here, sampEM(M) refers to sampEM working on M random samples where M is the number of sub-clusters.

For the first 7 grid structures, the segment numbers for each attribute are 56, 48, 40, 32, 24, 16, and 12, respectively. The cell granularity increases gradually. As shown in Figure 4, although the clustering accuracy of the three systems decreases, the accuracy of gEMADS decreases slowly and is normally higher than its two counterparts. Especially, it ranges from 95.4% to 91.4% for the first five grid structures, and the generated mixture models are very close to the original one. The last three grid structures in Figure 4 are used to generate very skew sub-clusters. For example, in the 12*86 grid structure, the cell width is 7.2 times longer than the cell height. Although the clustering accuracy of gEMADS system decreases from 92.0% to 83.6% for the last three grid structures, the decrease is much slower than that of gWEM. The one-tailed paired Student's t-Test for the 10 grid structures indicates that gEMADS significantly outperforms gWEM and sampEM at the 0.01 level. The performance of gEMADS is not so sensitive to the data summarization procedures, and acceptable when the sub-clusters are not too skew and large.

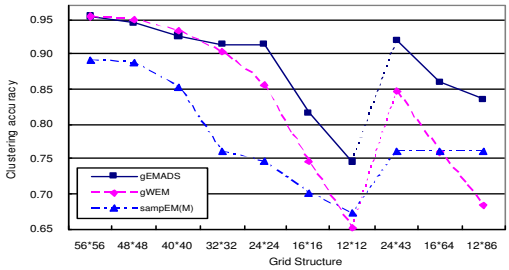


Figure 4. The clustering accuracy of three clustering systems for different data summarization or sampling procedures.

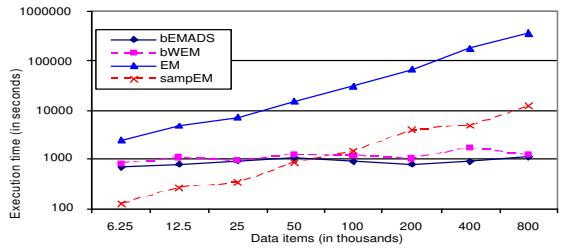
5.3 Scalability and Accuracy

The second set of experiments is conducted on the third group of data sets in order to analyze the scalability of bEMADS. Figure 5(a) illustrates the execution time of bEMADS, EM, bWEM, and sampEM.

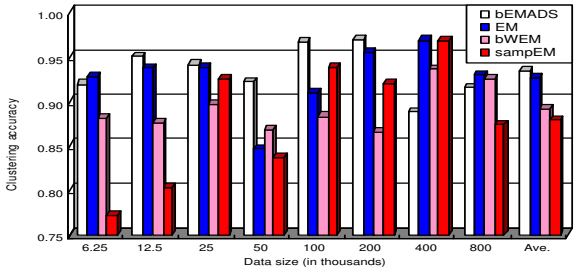
It can be observed from Figure 5(a) that the execution time of bEMADS increases very slowly with the number of data items. It takes 673.8 seconds for the data set with 6,250 data items, and takes 1,106.2 seconds for the data set with 800,000 data items. The execution time of bEMADS mainly spends on the mixture model generation. For example, the data summarization procedure takes about 255.2 seconds and EMADS takes 851.0 seconds for the largest data set.

The execution time of EM increases from 2,344.4 seconds for the data set with 6,250 data items to 359,966.7 seconds for the largest data set. It increases almost linearly with the data size, as plotted in Figure 5(a). This is because that the amount of main memory used is not restricted during the execution. For the 8 data sets, the speedup factors of bEMADS to EM range from 3.5 to 339.0. Thus, bEMADS can run several orders of magnitude faster than EM. In addition, as indicated in Figure 5(b), bEMADS can generate slightly more accurate clustering results than EM on four data sets. The average clustering accuracy of bEMADS is 93.5%. This is a little bit higher than the value of 92.7% for EM. The execution time of sampEM ranges from 125.2 seconds to 12,071.0 seconds. The ratio of the execution time of sampEM to bEMADS is 10.9:1 for the largest data sets. Although bEMADS does not run as fast as sampEM for those small data sets, it generates much better results than sampEM. As plotted in Figure 5(b), the average clustering accuracy of bEMADS is 5.5% higher than the value of 88.0% for sampEM, which is statistically significant at the 0.05 level.

The bWEM system, similar to bEMADS, is scalable too.



(a) Execution time.



(b) Clustering accuracy.

Figure 5. Performance of four clustering systems for eight 4-dimensional data sets.

As plotted in Figure 5(b), bEMADS generates more accurate results than bWEM for almost all eight data sets. The average clustering accuracy of bWEM is 89.2%, which is significantly lower than that of bEMADS at the 0.05 level.

Similar comparison results can be found in the third set of experiments for different mixture models. Figure 6 illustrates the clustering accuracy of bEMADS, EM, bWEM, and sampEM for the seven data sets in Table 1. For the seven data sets, bEMADS generates the most accurate results on the second and the third data sets. On average, the clustering accuracy values of bEMADS, EM, bWEM, and sampEM are 89.6%, 90.6%, 84.7%, and 86.3%, respectively. Though EM generates slightly more accurate clustering results than bEMADS does, the one-tailed paired t-Test does not indicate that it is significantly different at the 0.05 level. However, the average clustering accuracy of bEMADS is significantly better than that of bWEM and sampEM at the 0.05 level.

5.4 Application on Two Real-life Data Sets

For the two real-life data sets, the average log-likelihood serves as the accuracy metric of the generated mixture models. The larger the average log-likelihood is, the better a mixture model matches a data set. Table 2 summarizes the experimental results, including the standard deviations of log-likelihood and execution time.

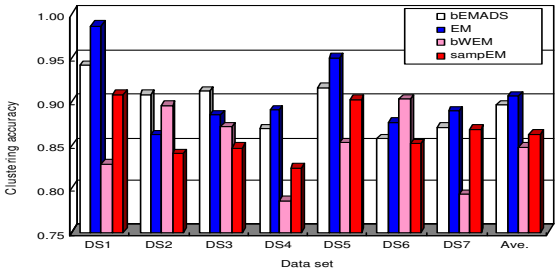


Figure 6. Clustering accuracy for the seven synthetic data sets.

The first real-life data set is the California housing data, downloaded from www.spatial-statistics.com. It has 20,640 data items. We use a 7-component Gaussian mixture model to describe the data set. The data summarization procedure of BIRCH generates 2,907 data summaries. Figure 1(a) illustrates the data set in the scaled Latitude-Longitude space. For this 8-dimensional data set, bEMADS takes about 3,232.4 seconds to generate the mixture models with the average log-likelihood of 7.517. EM spends about 5.1 times longer. Though the accuracy of bEMADS is slightly lower than the value of 7.682 for EM, the one-tailed t-Test indicate that the difference is not statistically significant at the 0.05 level. For this moderate data set, sampEM runs faster than bEMADS. However, the average log-likelihood of sampEM is as low as 6.776, significantly lower than that of bEMADS. The log-likelihood of bWEM is also significantly lower than that of bEMADS, though both spend similar execution time.

The second real-life data set, the Forest CoverType Data, is from the UCI KDD Archive (kdd.ics.uci.edu). The data set has 581,012 data items and five attributes are used in our experiments. We use a Gaussian mixture model with 15 components to describe the data set. The BIRCH's data summarization procedure generates 3,186 sub-clusters.

Because EM cannot generate a mixture model after running for 200 hours, we use sampEM(15%) for comparison.

Table 2. The performance of bEMADS on the two real-life data sets.

Housing	bEMADS	bWEM	EM	sampEM
log-likelihood	7.517 ± 0.191	6.882 ± 0.153	**7.682** ± 0. 159	6.776 ± 0.239
time(Sec.)	3232.4 ± 525.8	3488.6 ± 317.7	16405.5 ± 2906.2	1433.9 ± 514.7
Forest	bEMADS	bWEM	sampEM(15%)	sampEM
log-likelihood	-3.083 ± 0.011	-3.278 ± 0.053	**-3.078** ± 0.017	-3.086 ± 0.018
time(Sec.)	7985.5 ± 3635.2	6039.7 ± 1313.5	173672.5 ± 80054.2	49745.8 ± 10328.9

In fact, even it takes about 173,672.5 seconds. On average, bEMADS takes about 7,985.5 seconds. It runs 21.7 times faster than sampEM(15%), and 6.2 times faster than sampEM. The average log-likelihood value of bEMADS is -3.083. It is slightly larger than the value of -3.086 for sampEM, and slightly smaller than the value of -3.078 for sampEM(15%). However, the one-tailed t-Test indicate that no significant difference exists among them. The bWEM system runs a bit faster than bEMADS. However, it generates the worst mixture models among the four systems with the average log-likelihood value of -3.278. The one-tailed t-Test indicates that the log-likelihood value of bWEM is significantly worse than its three counterparts at the 0.05 level.

6 Conclusion

Through working on summary statistics, we have given a two-phase scalable model-based clustering framework: First, a large data set is categorized into mutually exclusive sub-clusters; Second, a new model-based clustering algorithm is used to directly generate clusters from the summary statistics of the sub-clusters. The new algorithm is designed for a pseudo mixture model that approximates the aggregate behavior of each sub-cluster of data items under the original mixture model.

To exemplify this framework, we have established two model-based clustering systems for the Gaussian mixture model. The main novelties are the pseudo component density function for data summaries and its associated iterative algorithm — EMADS (Expectation-Maximization Algorithm for Data Summaries). EMADS, derived from the general EM algorithm, can embody the cardinality, mean, and covariance information of each sub-cluster into both E-step and M-step to generate accurate Gaussian mixtures. We have also shown that EMADS converges to local maxima, which renders it the first mathematically sound algorithm to generate mixture models directly from data summaries. We have illustrated the insensitivity of EMADS to different data summary granularities by combining EMADS with the grid-based data summarization procedure. By combining EMADS with the BIRCH's data summarization procedures, we have established the scalable clustering system, bEMADS. The comprehensive experimental results on both the synthetic and real-life data sets have shown that bEMADS can run several orders of magnitude faster than the classical EM algorithm with little or no loss of accuracy. It runs faster and generates higher quality results than the random sampling EM algorithm for large data sets. It, using comparable computational resources, has generated statistically significantly more accurate results than the density-biased-sampling clustering system.

For future work, we will apply the scalable model-based clustering framework to heterogeneous, or other more complicated, data sets. We will also develop some effective approaches to automatically determine the number of clusters for large databases.

References

[1] P. Bradley, U. Fayyad, and C. Reina. Clustering very large databases using EM mixture models. In *ICPR'00*, volume 2, pages 76–80, 2000.

[2] I. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals. In *KDD-2000*, pages 140–149, 2000.

[3] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In U. M. Fayyad and *et al.*, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180, 1996.

[4] T. Chiu, D. Fang, J. Chen, Y. Wang, and C. Jeris. A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *KDD-2001*, pages 263–268, 2001.

[5] W. DuMouchel, C. Volinsky, T. Johnson, C. Cortes, and D. Pregibon. Squashing flat files flatter. In *KDD-1999*, pages 6–15, 1999.

[6] C. Fraley. Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281, Jan. 1999.

[7] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[8] H.-D. Jin. *Scalable Model-based Clustering Algorithms for Large Databases and Their Applications*. Ph.D. thesis, the Chinese University of Hong Kong, Hong Kong, Aug. 2002.

[9] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., New York, 1997.

[10] M. Meilă and D. Heckerman. An experimental comparison of model-based clustering methods. *Machine Learning*, 42(1/2):9–29, 2001.

[11] A. Moore. Very fast EM-based mixture model clustering using multiresolution KD-trees. In *NIPS'99*, pages 543–549, 1999.

[12] R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *VLDB'94*, pages 144–155, 1994.

[13] C. R. Palmer and C. Faloutsos. Density biased sampling: An improved method for data mining and clustering. In *SIGMOD-2000*, pages 82–92, 2000.

[14] J. Shanmugasundaram, U. Fayyad, and P. S. Bradley. Compressed data cubes for OLAP aggregate query approximation on continuous dimensions. In *KDD-1999*, pages 223–232, 1999.

[15] B. Thiesson, C. Meek, and D. Heckerman. Accelerating EM for large databases. *Machine Learning*, 45:279–299, 2001.

[16] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.