1-1-2002

An Expectation-Maximization Algorithm Working on Data Summary

Huidong JIN
Chinese University of Hong Kong

Kwong Sak LEUNG
Chinese University of Hong Kong

Man Leung WONG

An Expectation-Maximization Algorithm Working on Data Summary

Huidong Jin[+], Kwong-Sak Leung[+] and Man-Leung Wong [*]

Abstract

Scalable cluster analysis addresses the problem of processing large data sets with limited resources, e.g., memory and computation time. A data summarization or sampling procedure is an essential step of most scalable algorithms. It forms a compact representation of the data. Based on it, traditional clustering algorithms can process large data sets efficiently. However, there is little work on how to effectively make cluster analysis on data summaries. From the principle of the general expectation-maximization algorithm, we propose a model-based clustering algorithm to make better use of these data summaries in this paper. The proposed EMACF (Expectation-Maximization Algorithm on Clustering Features) algorithm employs such data summary features as weight, mean, and variance explicitly. It is proved that EMACF converges to a local maximum likelihood value. EMACF is linear with the number of data summaries instead of data items, and thus can be integrated with any efficient data summarization procedure to construct a scalable clustering algorithm.

I. Introduction

Clustering is the unsupervised classification of data items into meaningful clusters based on similarity or density. It can reveal some intrinsic structures and hence has been widely used in exploratory data analysis [4] and data mining [1], [3]. Among a rich assortment of clustering algorithms, the model-based clustering algorithms have been attracting much research [4]. They can accommodate complicated data sets with both numerical and categorical attributes by assuming a model structure on data [2], which are hard for other clustering algorithms [3]. The model-based clustering techniques have strong theoretical support from the statistics community [4]. The model-based clustering techniques have been successfully applied on, such as, database query optimization [6] and data mining [1], [3].

The traditional clustering algorithms normally require numerous scans of whole data sets to get better results. This becomes prohibitive over modern large databases, for example, the data set with 100,000 data items as shown in Fig.1(a). Scalable cluster analysis addresses the problem of processing large data sets with limited resources, e.g., memory and computation time. A data summarization or sampling procedure is often a preprocessing step to scale up a clustering algorithm. Uniform random sampling is easy to scale-up clustering algorithms. The biggest disadvantage of this strategy is the inaccuracy introduced by sampling variance. Data summarization is to construct summaries of the large data set on which to base the desired cluster analysis. BIRCH, the seminal work on scalable clustering algorithms, incrementally assimilates data into a CF(clustering feature)-tree, which is adaptively reconstructed once the given memory is exhausted [7]. In the final cluster generation procedure, it employs a hierarchical agglomerative clustering algorithm on subclusters in the CF-tree. Its performance degrade as the data distribution is very skewed [5]. To our knowledge, there is little work on how to make effective use of these data summaries.

We propose a model-based clustering algorithm to handle data summaries more effectively. Our EMACF (Expectation-Maximization Algorithm on Clustering Features) takes each clustering feature as a data object. It explicitly processes such features as cardinality, mean, and the second-order statistics of a subcluster. It describes a subcluster of data items more accurate, and so less sensitive to the data summarization procedure. For example, for the data set in Fig. 1(a), the EMACF algorithm takes about 5 minutes to generate clusters from the data summaries as shown in Fig. 1(b). The generated clusters are very close to the original ones. But the canonical Expectation Maximization (EM) takes about 60 minutes to generate similar clusters. EMACF complements some other clustering algorithms by virtue of the fact that their clustering results can be enhanced by EMACF. After introducing the model-based techniques in Section 2, we derive EMACF and present a convergence theorem in Section 3. The last section concludes the paper.

This research was partially supported by RGC Earmarked Grant for Research CUHK 4212/01E of Hong Kong. Please direct all correspondence to Mr. Huidong Jin with email address: hdjin@cse.cuhk.edu.hk or the snail mail address: Department of Computer Science and Engineering, the Chinese University of Hong Kong, Shatin, N.T., Hong Kong; Tel: +852 2609 8412; Fax: +852 2603 5024.

(+) H. D. Jin and K. S. Leung are with the Department of Computer Science and Engineering, the Chinese University of Hong Kong, Shatin, N.T., Hong Kong. Email: {hdjin, ksleung}@cse.cuhk.edu.hk.

(*) Dr. M. L. Wong is affiliated with the Department of Computing and Decision Sciences, Lingnan University, Tuen Mun, Hong Kong. Email: mlwong@ln.edu.hk.

II. Model-based clustering techniques

Given a data set $X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$, the model-based clustering algorithms assume that each data item $\mathbf{x}_i = [x_{1i}, x_{2i}, \cdots, x_{1i}]^T$ is drawn from a finite mixture model Φ of K distributions:

$$p(\mathbf{x}_i|\Phi) = \sum_{k=1}^{K} p_k \phi(\mathbf{x}_i|\theta_k). \tag{1}$$

Here N is the total number of the data items, K is the number of clusters, p_k is the mixing proportion for the k^{th} cluster ($0 < p_k < 1$ for all $k = 1$, ..., K and $\sum_{k=1}^{K} p_k = 1$), $\phi(\mathbf{x}_i|\theta_k)$ is a *component density* function where the parameters are indicated by a vector θ_k.

This paper concentrates on the case where $\phi(\mathbf{x}_i|\theta_k)$ is a multivariate Gaussian distribution, even though the framework we used is able to be used for mixture models on complicated data sets. This Gaussian mixture model has been used with considerable success [1], [4]. In this case, the parameter θ_k consists of a mean vector μ_k and a covariance matrix Σ_k. The density function is of the form

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

where D is the dimension of data items.

Thus, a mixture model Φ includes the the mixing proportion, the component density function ϕ and parameters involved. Given Φ, we get a crisp classification by assigning the data items $\mathbf{x}_i$ to cluster k if $k = \arg\max_l \{p_l \phi(\mathbf{x}_i|\theta_l)\}$. Thus, a model Φ can be viewed as a solution for the clustering problem. So the clustering problem is transformed into solving parameters of model Φ. One way to get Φ is to maximize the *likelihood*, whose logarithm form is

$$L(\Phi) = \log\left[\prod_{i=1}^{N} p(\mathbf{x}_i|\Phi)\right] = \sum_{i=1}^{N} [\log p(\mathbf{x}_i|\Phi)]. \tag{3}$$

In general, it is impossible to solve it explicitly and iterative schemes must be employed.

The general Expectation-Maximization (EM) algorithm is a common iterative scheme to maximize the likelihood $L(\Phi)$. Thus a maximization likelihood estimate can be obtained [2], [4]. The general EM algorithm is profitably applied on *incomplete-data problems*, one typical example of which is cluster analysis if class indicator are regarded as 'missing' values. Its basic idea is to associate with the given incomplete-data problem, a *complete-data problem* for which the maximum likelihood estimate is computationally tractable. Suppose that we have a set of 'incomplete' data vectors $\{\mathbf{x}\}$ and wish to maximize the likelihood $L(\Phi) = p(\{\mathbf{x}\}|\Phi)$. Let $\{\mathbf{y}\}$ denote a typical 'complete' version of $\{\mathbf{x}\}$, that is, each vector $\mathbf{x}_i$ is augmented by the 'missing' values so that $\mathbf{y}_i^T = (\mathbf{x}_i^T, \mathbf{z}_i^T)$. There may be many possible vectors $\mathbf{y}_i$ in which $\mathbf{x}_i$ is embedded. For the finite mixture case, $\mathbf{z}_i$ is naturally a class indicator $\mathbf{z}_i = (z_{1i}, z_{2i}, ..., z_{Ki})^T$, where $z_{ki} = 1$ if $\mathbf{x}_i$ belongs to the k^{th} component and zero otherwise. Let the likelihood of $\{\mathbf{y}\}$ be $g(\{\mathbf{y}\}|\Phi)$ whose form we know explicitly so that the likelihood $p(\{\mathbf{x}\}|\Phi)$ is obtained from $g(\{\mathbf{y}\}|\Phi)$ by integrating over all possible $\{\mathbf{y}\}$ in which the set $\{\mathbf{x}\}$ is embedded:

$$L(\Phi) = p(\{\mathbf{x}\}|\Phi) = \int \prod_{i=1}^{N} g(\mathbf{x}_i, \mathbf{z}|\Phi) d\mathbf{z}. \tag{4}$$

The general EM procedure generates a sequence of estimate of Φ, $\{\Phi^{(j)}\}$, from a initial estimate $\Phi^{(0)}$ and consists of two steps:

1. **E-step**: Evaluate $Q(\Phi; \Phi^{(j)}) \stackrel{\triangle}{=} E\left[\log(g(\{\mathbf{y}\}|\Phi))|\{\mathbf{x}\}, \Phi^{(j)}\right]$, that is, the expectation of the complete data log-likelihood, conditional on the observed data, $\{\mathbf{x}\}$, and the current value of the parameters, $\Phi^{(j)}$.
2. **M-step**: Find $\Phi = \Phi^{(j+1)}$ that maximizes $Q(\Phi; \Phi^{(j)})$.

The likelihoods of interest satisfy $L(\Phi^{(j+1)}) \geqslant L(\Phi^{(j)})$. Thus for a bounded sequence of likelihood values, $\{L(\Phi^{(j)})\}$ converges monotonically to some L^*.

III. An EM Algorithm on Clustering Features

There are many methods to partition a data set into subclusters which contain only similar data items. For example, we can use certain grid to partition the data space and all data items in a cell is regarded as a

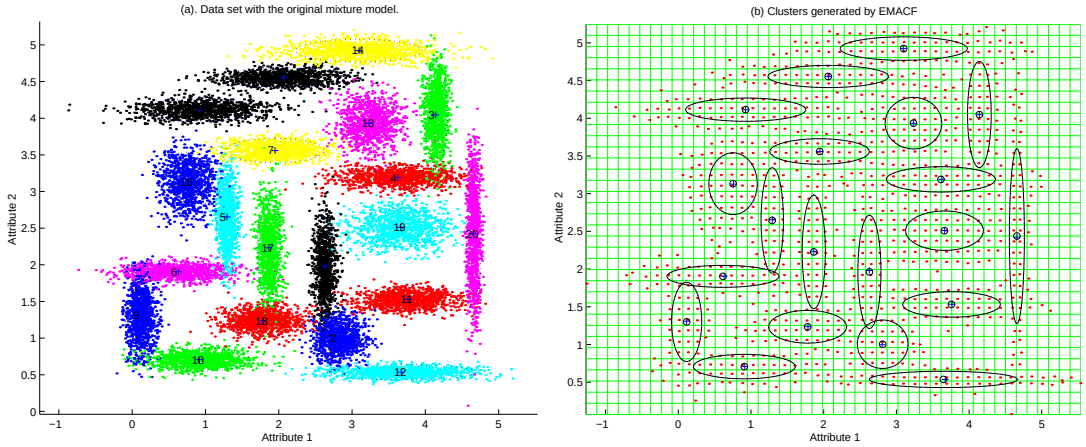

Fig. 1. (a). A data set with 100,000 data items and 20 clusters (10% data items are plotted); (b). the clusters generated by EMACF. A dot indicates a data item in (a) and a clustering feature in (b), respectively. The center 'o' (or '+') and its corresponding solid (or dotted) ellipse indicate the mean and a contour of a component Gaussian distribution.

subcluster. In this way, we can generate 1025 non-empty subclusters for the data set, as shown in Fig. 1(b). Even different forms of data summaries are used to describe a subcluster of data items [1], they basically contain the cardinality, mean, and a second-order statistics of the subcluster. They may be derived from clustering features [7]. The variance indicates the distribution within a subcluster, but it is rarely used in clustering algorithms directly. Integrated with specific feature of Gaussian mixture, we tactfully establish a new model-based clustering algorithms. It works directly on the three items and thus makes better use of the clustering features.

We use clustering feature $\mathbf{s}_m = \{n_m, \nu_m, \gamma_m\}$ $(m = 1, ..., M)$ as a data summary for a subcluster. Here M is the number of subclusters. n_m is the cardinality of the m^{th} subcluster, so $N = \sum_{m=1}^{M} n_m$ is the total number of data items. $\nu_m = \frac{1}{n_m} \sum_{\text{the } m^{th} \text{subcluster}} \mathbf{x}_i$ is the mean of the m^{th} subcluster of the data items; $\gamma_m = (\gamma_{1m}, \gamma_{2m}, ..., \gamma_{Dm})^T$ is the diagonal vector of the matrix $\Gamma_m = \frac{1}{n_m} \sum_{\text{the } m^{th} \text{ subcluster}} \mathbf{x}_i^T \mathbf{x}_i$, which indicates the average second order moments of the m^{th} subcluster. Our derivation below assumes that the component Gaussian distribution is independent, even the principle is applicable to any distribution.

For each data item $\mathbf{x}_i$ in the m^{th} subcluster, we give a 'density function' to approximate normal distribution, specified as follows:

$$\psi(\mathbf{x}_i \in the\ m^{th}\ subcluster|\theta_k) \triangleq \psi(\mathbf{s}_m|\theta_k) = \prod_{d=1}^{D} \frac{\exp\left\{-\frac{1}{2\sigma_{dk}}(\gamma_{dm} - 2\mu_{dk}\nu_{dm} + \mu_{dk}^2)\right\}}{(2\pi)^{\frac{1}{2}}\sigma_{dk}^{\frac{1}{2}}}. \quad (5)$$

where $\theta_k = \{\mu_k, \sigma_k\}$ refers to the parameters for the k^{th} Gaussian component. We get it through substituting γ_{dm} with item ν_{dm}^2 in $\phi(\nu_m|\theta_k)$, which is the probability of the m^{th} subcluster mean. It is a normal density function when $\gamma_{dm} = \nu_{dm}^2$, especially, when there is only one data item in the subcluster. However, this is not a density function in general. Under this function, this probability for a data item $\mathbf{x}_i$ explicitly depends on which subcluster it belongs to. It is only implicitly relevant with its values. This enable us to treat data items in a subcluster in a same way and we don't need to store $\mathbf{x}_i$, so this helps us save lots of computation time and memory. In addition, the second order moment γ_m is taken into consideration, so the data distribution within a subcluster may influence the final mixture model and then the final clustering. Intuitively, if a subcluster variance (second order moment) is small, that is, data are quite dense in this subcluster, the subcluster has larger ϕ and then more influence on clustering. This accords with the model-based clustering principle to locate cluster on dense areas.

With this 'density function', the probability for a data item $\mathbf{x}_i$ in the m^{th} subcluster under the mixture model is:

$$p(\mathbf{x}_i \in the\ m^{th}\ subcluster|\Psi) \triangleq p(\mathbf{s}_m|\Psi) = \sum_{k=1}^{K} p_k \psi(\mathbf{s}_m|\mu_k, \sigma_k). \quad (6)$$

Then the log-likelihood is given as

$$L(\Psi) = \log\left[\prod_{i=1}^{N} p(\mathbf{x}_i|\Psi)\right] = \sum_{m=1}^{M} n_m \log p(\mathbf{s}_m|\Psi). \tag{7}$$

By interpreting the class labels as missing values, we now derive an EM algorithm to get the maximum likelihood estimate efficiently. If $\mathbf{x}_i$ is in cluster k, we denote $\mathbf{z}_i$ an indicator vector of length K with a 1 in the k^{th} position and zeros elsewhere. Then the complete data vector $\mathbf{y}_i = (\mathbf{x}_i^T, \mathbf{z}_i^T)^T$, which is augmented by a class label. The likelihood of the complete data vector, $\mathbf{y}_i$, is $g(\mathbf{y}_i|\Psi) = p(\mathbf{x}_i|\mathbf{z}_i, \Psi)p(\mathbf{z}_i|\Psi) = \psi(\mathbf{x}_i|\theta_k)p_k = \prod_{k=1}^{K}[\psi(\mathbf{x}_i|\theta_k)p_k]^{z_{ki}}$. The last equation holds since z_{ki} is either zero or one. For all N data items, we have

$$g(\mathbf{y}_1, ..., \mathbf{y}_N|\Psi) = \prod_{i=1}^{N}\prod_{k=1}^{K}[\psi(\mathbf{x}_i|\theta_k)p_k]^{z_{ki}} = \prod_{m=1}^{M}\prod_{k=1}^{K}[\psi(\mathbf{s}_m|\theta_k)p_k]^{z_{km}n_m} \tag{8}$$

The last equation is valid because that data items within a subcluster have the same probability, and then have the same class indicator. That is, $\psi(\mathbf{x}_i|\theta_k) = \psi(\mathbf{x}_j|\theta_k)$ and $\mathbf{z}_i = \mathbf{z}_j$ if both $\mathbf{x}_i$ and $\mathbf{x}_j$ fall into one subcluster.

The log-likelihood $L(\Psi)$ is obtained from $g(\{\mathbf{y}\}|\Psi)$ by integrating over all possible $\{\mathbf{y}\}$ in which the set $\{\mathbf{x}\}$ is embedded:

$$L(\Psi) = \log\left[p(\{\mathbf{x}\}|\Psi)\right] = \int \log\ \left[g(\{\mathbf{y}\}|\Psi)\right] d\mathbf{z} = \sum_{m=1}^{M} n_m \log p(\mathbf{s}_m|\Psi).$$

This agrees with the log-likelihood definition in Eq.(7).

Now let's calculate the expectation of the complete data log-likelihood, conditional on the observed data $\{\mathbf{x}\}$ (which is replaced by $\{\mathbf{s}\}$ literally) and the current value of the parameters, $\Psi^{(j)}$.

$$\begin{aligned} Q(\Psi; \Psi^{(j)}) &= E\left[\log(g(\{\mathbf{y}\}|\Psi))|\{\mathbf{x}\}, \Psi^{(j)}\right] = E\left[\log(g(\{\mathbf{y}\}|\Psi))|\{\mathbf{s}\}, \Psi^{(j)}\right] \\ &\triangleq \sum_{i=1}^{M} n_m \sum_{k=1}^{K} r_{mk}\left[\log p_k^{(j)} + \log(\psi(\mathbf{s}_m|\mu_k^{(j)}, \sigma_k^{(j)}))\right] \end{aligned} \tag{9}$$

where

$$r_{mk} = E\left[z_{km}|\{\mathbf{s}\}, \Psi^{(j)}\right] = \frac{p_k^{(j)}\psi(\mathbf{s}_m|\mu_k^{(j)}, \sigma_k^{(j)}))}{\sum_{l=1}^{K} p_l^{(j)}\psi(\mathbf{s}_m|\mu_l^{(j)}, \sigma_l^{(j)})}. \tag{10}$$

Now we turn to maximize $Q(\Psi; \Psi^{(j)})$ with respect to Ψ. Consider the parameters p_k, μ_k and σ_k in turn. We need to introduce a Lagrange multiplier λ to remove the constraint $\sum_{k=1}^{K} p_k = 1$. Differentiating $Q(\Psi; \Psi^{(j)}) - \lambda\left(\sum_{k=1}^{K} p_k - 1\right)$ with respect to p_k, we get $\sum_{m=1}^{M} n_m r_{mk}\frac{1}{p_k} - \lambda = 0$ for $k = 1, ..., K$. Sum up the K equations together, $\lambda\sum_{k=1}^{K} p_k = \sum_{k=1}^{K}\sum_{m=1}^{M} n_m r_{mk} = \sum_{m=1}^{M} n_m\left[\sum_{k=1}^{K} r_{mk}\right] = \sum_{m=1}^{M} n_m = N$. This leads to

$$\lambda = N, \tag{11}$$

$$\widehat{p}_k = \sum_{m=1}^{M} n_m r_{mk}/N. \tag{12}$$

For Gaussian distribution parameters μ_k and σ_k, we have partial derivative on the density function:

$$\begin{aligned} \frac{\partial \log\psi(\mathbf{s}_m|\mu_k, \sigma_k)}{\partial \mu_{dk}} &= \frac{1}{\sigma_{dk}}(\nu_{dm} - \mu_{dk}), \text{ and} \\ \frac{\partial \log\psi(\mathbf{s}_m|\mu_k, \sigma_k)}{\partial \sigma_{dk}} &= -\frac{1}{2\sigma_{dk}} + \frac{1}{2\sigma_{dk}^2}(\gamma_{dm} - 2\mu_{dk}\nu_{dm} + \mu_{dk}^2). \end{aligned}$$

Differentiating $Q(\Psi;\Psi^{(j)})$ with respect to μ_{dk} and equating to zero gives

$$\frac{\partial Q(\Psi;\Psi^{(j)})}{\partial \mu_{dk}} = \sum_{m=1}^{M} n_m r_{mk} \frac{1}{\sigma_{dk}}(\nu_{dm} - \mu_{dk}) = 0. \tag{13}$$

This gives the re-estimate of μ_{dk} as

$$\widehat{\mu}_{dk} = \frac{\sum\limits_{m=1}^{M} n_m r_{mk} \nu_{dm}}{\sum\limits_{m=1}^{M} n_m r_{mk}}. \tag{14}$$

Similarly, differentiating $Q(\Psi;\Psi^{(j)})$ with respect to σ_{dk} and equating to zero leads to

$$\widehat{\sigma}_{dk} = \frac{\sum\limits_{m=1}^{M} n_m r_{mk}(\gamma_{dm} - 2\mu_{dk}\nu_{dm} + \mu_{dk}^2)}{\sum\limits_{m=1}^{M} n_m r_{mk}} \tag{15}$$

Thus, EMACF is to alternate between the E-step of estimating r_{mk} (Eq. (10)) and the M-step of calculating $\widehat{p}_k$, $\widehat{\mu}_{dk}$, $\widehat{\sigma}_{dk}$ given the r_{mk} (Eqs. (12), (14) and (15)). We re-write the algorithm in terms of vector as follows.

1. **Initialization**: Fixing the number of clusters K, initialize the parameters in the mixture model: $p_k^{(j)}(> 0), \mu_k^{(j)}$ and $\Sigma_k^{(j)}(> 0)$ $(k = 1, ..., K)$, and set the current iteration $j = 0$.
2. **E-step**: Given the mixture model parameters $\Psi^{(j)}$, compute the membership $r_{mk}^{(j)}$:

$$r_{mk}^{(j)} = \frac{p_k^{(j)} \psi(\mathbf{s}_m | u_k^{(j)}, \sigma_k^{(j)})}{\sum\limits_{i=1}^{K} p_i^{(j)} \psi(\mathbf{s}_m | u_i^{(j)}, \sigma_i^{(j)})}. \tag{16}$$

3. **M-step**: given $r_{mk}^{(j)}$, update the mixture model parameters for $k = 1, ..., K$:

$$p_k^{(j+1)} = \frac{1}{N} \sum_{m=1}^{M} n_m r_{mk}^{(j)}, \tag{17}$$

$$\mu_k^{(j+1)} = \frac{\sum\limits_{i=1}^{M} n_m r_{mk}^{(j)} \nu_m^{(j)}}{\sum\limits_{m=1}^{M} n_m r_{mk}^{(j)}} = \frac{\sum\limits_{i=1}^{M} n_m r_{mk}^{(j)} \nu_m^{(j)}}{N \cdot p_k^{(j+1)}}, \tag{18}$$

$$\sigma_k^{(j+1)} = \frac{\sum\limits_{m=1}^{M} n_m r_{mk}^{(j)} \left[\gamma_m - 2\mu_k^{(j)} \circledast \nu_m + \mu_k^{(j)} \circledast \mu_k^{(j)}\right]}{N \cdot p_k^{(j+1)}} \tag{19}$$

   where $\circledast$ indicates the array multiplication. That is, the (i, j) element of $A \circledast B$ is $a_{ij} b_{ij}$.
4. **Termination**: If $L(\Psi^{(j+1)})$ is not close $L(\Psi^{(j)})$ enough, set $j = j + 1$ and go to step 2.

The algorithm ends when the log-likelihood values $L(\Psi^{(j)})$ changes very small. This criterion is certainly reachable as supported by the following convergence theorem.

Theorem 1: $L(\Psi)$ for EMACF converges monotonically to some value $L^* = L(\Psi^*)$ for some stationary record Ψ^*.

Proof: In the deduction procedure above, we follow the general EM algorithm to get EMACF, and thus EMACF is an instance of the EM algorithm. Then theorems of the general EM algorithm is also applicable to EMACF. We may have that $\{L(\Psi^{(j)})\}$ doesn't decrease, e.g., $L(\Psi^{(j+1)}) \geq L(\Psi^{(j)})$. Now we show $\{L(\Psi)\}$ has an upper bound. With Jensen's inequality, $\gamma_{dm} = \frac{1}{n_m} \sum\limits_{\mathbf{x}_i \in \text{the } m^{th} \text{ subcluster}} x_{di}^2 \geq$

$\left[\frac{1}{n_m}\sum_{\mathbf{x}_i\in \text{ the } m^{th}\text{subcluster}} x_{di}\right]^2 = \nu_{dm}^2$, then

$$\begin{aligned}\psi(\mathbf{x}_i &\in \text{ the } m^{th} \text{ subcluster}|\theta_k) = \psi(\mathbf{s}_m|\theta_k)\\ &\leq \phi(\nu_m|\theta_k)\end{aligned}$$

So,

$$L(\Psi) \leq \sum_{m=1}^{M} n_m \log p(\mathbf{s}_m|\Psi) \leq \sum_{m=1}^{M} n_m \left[\sum_{k=1}^{K} p_k \psi(\mathbf{s}_m|\mu_k,\sigma_k)\right] \leq 0.$$

We have $L(\Psi^{(j)})$ converges monotonically to some value $L(\Psi^*)$.

Now we show the Q function $Q(\Phi;\Psi)$ is continuous in both Φ and Ψ. Once $\{p_k\}$ and $\{\sigma_{dk}\}$ are initialized larger than zero, from Eqs.(16-19), they will always be larger than zero. Bearing this in mind, the function $Q(\Phi;\Psi)$ in Eq.(9) is continuous in both Φ and Ψ, because it consists of only arithmetic and logarithm operations. Recall that the convergence theorem about the EM algorithm, e.g., Theorem 3.2 in [4, p.88], says that all the limit records of any instance $L(\Psi^{(j)})$ of the EM algorithm are stationary records of $L(\Psi)$ provided that the function $Q(\Phi;\Psi)$ is continuous in both Φ and Ψ. The proof of the theorem is completed. ■

Let's have a brief look at the complexity of EMACF. For E-step, EMACF needs to calculate $M * K$ membership probability $r_{mk}^{(j)}$. For each $r_{mk}^{(j)}$, we has to calculate the probability for each component distribution according to Eq.(5). Based on the independence assumption, it involves $O(D)$ arithmetic operations. Thus E-step takes $O(MKD)$ operations. Similarly, M-step in EMACF takes $O(MKD)$. Usually, the EM algorithm is terminated before a constant number, say, 500 in our implementation, of iterations are used up. In a word, the computation complexity of EMACF is $O(MKD)$.

EMACF requires $M(2D+1)$ floating points to store the clustering features, MK for the membership, $K(2D+1)$ for the mixture model. Thus, the total storage requirement of EMACF is $2MD+MK+2KD+K+M$. Consequently, the storage requirement of EMACF is independent to the number of data items N. In other words, we can choose an appropriate M to summarize a given data set into the given main-memory and then quickly generate the clusters from the data summaries in the main-memory.

IV. Conclusion

Data summarization or sampling has been becoming a standard preprocessing procedure for scalable clustering algorithms. Based on the generated data summaries, the traditional clustering algorithms are able to handle large data sets efficiently with a little modification. The EMACF (Expectation-Maximization Algorithm on Clustering Features) algorithm has been proposed to make use of these data summaries effectively. It takes into consideration the data summary features such as cardinality, mean and variance. Furthermore, we have proved that EMACF converges to a local maximum likelihood value monotonically. EMACF is linear with the number of data summaries instead of the number of data items.

EMACF can complement many data summarization procedures. It is subject to our future work on how to integrate EMACF with other data summarization procedures, e.g., BIRCH, and apply it to real-world problems. Another interesting direction of this work is to establish other model-based clustering algorithms on data summaries of complicated data sets.

References

[1] P. Bradley, U. Fayyad, and C.R. Reina. Clustering very large databases using EM mixture models. In *Proceedings of 15th International Conference on Pattern Recognition*, volume 2, pages 76–80, 2000.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 1977.

[3] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[4] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., New York, 1997.

[5] Christopher R. Palmer and Christos Faloutsos. Density biased sampling: An improved method for data mining and clustering. In *Proceedings of ACM SIGKOD International Conference on Management of data*, pages 82–92, 2000.

[6] Jayavel Shanmugasundaram, Usama Fayyad, and P. S. Bradley. Compressed data cubes for olap aggregate query approximation on continuous dimensions. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 223–232. ACM Press, 1999.

[7] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: An efficient data clustering method for very large databases. *SIGMOD Record: Proc. ACM SIGMOD Int. Conf. Management of Data*, 25(2):103–114, June 4 - June 6 1996.