HIERARCHICAL MODEL-BASED CLUSTERING FOR RELATIONAL DATA

Jianzhong Chen, Mary Shapcott, Sally McClean, Kenny Adamson
School of Computing and Mathematics, Faculty of Engineering, University of Ulster
Shore Road, Newtownabbey, Co. Antrim, BT37 0QB, Northern Ireland, UK

Keywords: Hierarchical model-based clustering, relational data, frequency aggregates, EM algorithm.

Abstract: Relational data mining deals with datasets containing multiple types of objects and relationships that are presented in relational formats, e.g. relational databases that have multiple tables. This paper proposes a propositional hierarchical model-based method for clustering relational data. We first define an object-relational star schema to model composite objects, and present a method of flattening composite objects into aggregate objects by introducing a new type of aggregates – frequency aggregate, which can be used to record not only the observed values but also the distribution of the values of an attribute. A hierarchical agglomerative clustering algorithm with log-likelihood distance is then applied to cluster the aggregated data tentatively. After stopping at a coarse estimate of the number of clusters, a mixture model-based method with the EM algorithm is developed to perform a further relocation clustering, in which Bayes Information Criterion is used to determine the optimal number of clusters. Finally we evaluate our approach on a real-world dataset.

1 INTRODUCTION

Clustering aims at determining the intrinsic structure of clustered data when no information other than the observed values is available. Three types of clustering methods have been widely used – *hierarchical clustering* (Meilă and Heckerman, 1998), *partition-based clustering* and *model-based approach using mixture models* (Fraley and Raftery, 1998).

Most traditional clustering methods handle datasets that have single relation in flat formats. Recently, there has been a growing interest in *relational data mining* (RDM) (Džeroski and Raedt, 2003; Džeroski and Lavrač, 2001), which is tackling the problem of mining relational datasets that contain multiple types of objects and richer relationships and are presented in relational formats that have more than one table. RDM provides techniques for discovering useful or unknown patterns and dependencies embedded in relational databases. A common solution to RDM is developing *propositional* methods that integrate traditional data mining techniques into relational data by converting or "flattening" multiple tables into a single table on which standard algorithms can be run. One of the shortcomings of this approach is that it may cause loss of meaning or information. Another solution leads to *relational* approaches that are capable of dealing with data stored in multiple tables directly in the areas of *inductive logic programming* (ILP) (Džeroski and Raedt, 2003; Džeroski and Lavrač, 2001) and *probabilistic relational models* (PRMs) (Friedman et al., 1999). Some initial work of relational data classification and clustering based on ILP and PRMs have been developed in (Džeroski and Raedt, 2003; Džeroski and Lavrač, 2001; Emde and Wettschereck, 1996; Taskar et al., 2001).

In this paper, we present a propositional method which integrates traditional hierarchical model-based clustering algorithms with relational data that is composed of a set of composite objects. We use aggregation to efficiently flatten composite objects into flat aggregate objects, to which model-based hierarchical agglomerative clustering with log-likelihood distance and the EM algorithm are then applied. In order to discover rich aggregate knowledge from relational data, we define *frequency aggregates* for composite objects, which have vector data type and can be used to record not only the observed values but also the distribution of the values of an attribute. Frequency aggregates provide extended semantics in reducing the information loss during aggregation and are helpful to the computation of log-likelihood distance.

Chen J., Shapcott M., McClean S. and Adamson K. (2004).
HIERARCHICAL MODEL-BASED CLUSTERING FOR RELATIONAL DATA.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 92-97
DOI: 10.5220/0002624300920097

2 OBJECT-RELATIONAL STAR SCHEMA

Multi-relational data mining focuses on *composite objects*. A general characterization defines a composite object consisting of several components (possibly from different types) with relationships in between. We assume the relationship between a composite object and its components is *aggregation* and the objects are stored in multiple database tables. A composite object is defined as composed of a *base* (sub-object) associated with a set of additional *parts* (sub-objects). Two types of composite objects are distinguished corresponding with the two kinds of aggregation defined in object-oriented modelling – *shared aggregation*, where the parts may be parts in any wholes, and *composition aggregation*, in which the particular parts are owned by one whole at a time and the existence of the parts is strongly dependent on the existence of the whole (Eriksson and Penker, 1998).

Two types of relational models can be used to represent the two kinds of aggregation relationships between the tables. One is *relational star schema* (which can be generalized to the *relational snowflake schema*), where the base table is in the middle and the part tables radiate from the base[1]. A relational star schema represents a shared aggregation in the way that many-to-one relationships are specified from base to parts. In comparison, a so called *relational aggregate schema*, where in the middle is the base table and the parts converge on the base, is defined to represent a composition aggregation in the way that many-to-one relationships are denoted from parts to base. Figures 1 and 2 illustrate the schema graphs and object diagrams of the two schemas respectively. For example, it is natural to design a product sales database using a relational star schema (Figure 1), in which each sale is a composite object with a base object of *SaleRecord* class (X_0) and several sharable part objects of classes ($\{X_k\}$) such as *Product*, *Time*, *Geography*, etc. An example of a relational aggregate schema (Figure 2) used in the paper is to model a housing condition survey database as composed of a base table of *Dwelling* and two part tables of *Occupants* and *Rooms*. In this way, each house is represented as a composite object that has a dwelling description record, a set of occupants who are living in and a set of rooms of different living conditions.

The two schemas can be unified in the object-relational (OR) context by introducing object identifiers (OIDs), reference (REF) and collection data types (nested tables or collection of REF types) (Connolly and Begg, 2002), which allow us to convert

[1]The star schema is widely used in data warehousing and OLAP, where the base table and part tables are called the fact table and dimension tables respectively.

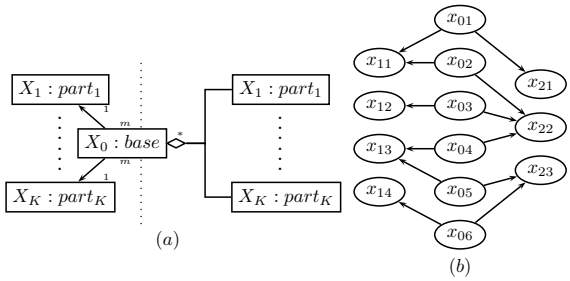

Figure 1: (a)The ***schema graph*** of ***relational star schema***. (b)An ***instance graph*** of ***relational star schema***, including 6 composite objects, that contain 6 objects of base class X_0, 4 objects of part class X_1 and 3 objects of part class X_2.

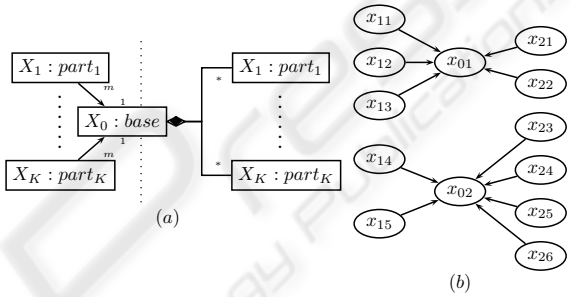

Figure 2: (a)The ***schema graph*** of ***relational aggregate schema***. (b)An ***instance graph*** of ***relational aggregate schema***, including 2 composite objects, that contain 2 objects of base class X_0, 5 objects of part class X_1 and 6 objects of part class X_2.

the many-to-one relationship in a relational aggregate schema from parts-to-base into base-to-parts. In this way, a relational aggregate schema can be replaced by a star schema. We call the resulting schema as *object-relational star schema* which is treated as a unified representation of the relational star schema and relational aggregate schema. For the sake of simplicity, we assume that only one composite class exists in the dataset without recursive structures.

More formally, an object-relational star schema defines a composite class $\mathcal{X} = \{X_0, X_1, \ldots, X_K\}$, which consists of a base class X_0 and a set of part classes $\{X_1, \ldots, X_K\}$. Each class $X_k, 0 \leq k \leq K$, is an abstract type of an entity in the domain, and is associated with a set of *attributes*. For a star schema, the base class is denoted as $X_0(o, A_1, \ldots, A_{M_0}, R_1, \ldots, R_K)$ and the k-th part class as $X_k(o, A_1, \ldots, A_{M_k})$. Three types of attributes are distinguished. $X_k.o$ is used to specify an unique system-generated *object identifier* for each object of class X_k. A *descriptive attribute* $X_k.A_m, 1 \leq m \leq M_k$, represents an attribute of X_k and takes value from its *domain* $\mathrm{Dom}(X_k.A_m)$. A *reference attribute* $X_0.R_k, 1 \leq k \leq K$, has domain of REF

type or collection type (e.g., a set of REFs). When all the reference attributes are REF typed, an object-relational star schema is identical with a relational star schema; otherwise, we restrict it to stand for a relational aggregate schema with composition aggregation. In addition, an *instantiation* $\mathcal{I}$ of an object-relational star schema $\mathcal{X}$ is composed of a set of N composite objects, $\mathcal{I}_{\mathcal{X}} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$, where $\mathcal{I}_n = \{x_0(n), x_{n1}, \ldots, x_{nK}\}$, $1 \leq n \leq N$; $x_0(n)$ stands for the n-th object (or case) of the base class X_0 in the database; $x_{nk} = \{x_{nk}(1), \ldots, x_{nk}(T_{nk})\}$, $1 \leq k \leq K, T_{nk} \geq 1$, represents a subset of T_{nk} objects of a part class X_k involved in the n-th composite object; $x_{nk}(t), 1 \leq t \leq T_{nk}$, is the t-th object of class X_k in the n-th composite object. Each object is assigned an OID and a list of value mappings from descriptive attributes to their domains and, for the base class, an interpretation for all the reference attributes. Moreover, in the n-th composite object, we use $x_0(n).A_m$, $x_{nk}(t).A_m$ and $x_{nk}.A_m$ to denote the observed value of attribute $X_0.A_m$ in the base object, the observed value of attribute $X_k.A_m$ in any part object, and the subset of observed values of attributes $X_k.A_m$, respectively.

3 FREQUENCY AGGREGATES

In clustering, once the objects of analysis have been determined, we are faced with the problem of finding proper measures to decide how far, or how close the data objects are from each other. The measures can be either similarity or dissimilarity (Jain and Dubes, 1988). Dissimilarity, which is widely used in practice, can be measured in many ways and one of them is *distance*. Distance measures depend on the type, scale and domain of attributes we are analyzing. In order to measure likelihood distance between composite objects, we present a "relational-to-propositional" method of making composite objects comparable by defining an aggregate object. The basic idea is to convert each composite object into a single aggregate object by preserving aggregate information of part objects. The notion of aggregate is borrowed from relational algebra and set theory, where a multi-set of values can be converted into a single aggregation or summary value by applying with aggregate functions or operations, such as COUNT, AVG in SQL and MODE, MEDIAN in set theory.

More precisely, given a composite object $\mathcal{I}_n = \{x_0(n), x_{n1}, \ldots, x_{nK}\}$, we define its *aggregate object* as $\text{AGG}(\mathcal{I}_n) = (\text{AGG}(x_0(n)), \text{AGG}(x_{n1}), \ldots, \text{AGG}(x_{nK}))$, where $\text{AGG}(x_0(n)) = (x_0(n).o, x_0(n).A_1, \ldots, x_0(n).A_{M_0})$; $\text{AGG}(x_{nk}) = (\text{COUNT}(x_{nk}), \text{AGG}(x_{nk}.A_1), \ldots, \text{AGG}(x_{nk}.A_{M_k}))$; $\text{COUNT}(x_{nk}) = |x_{nk}| = T_{nk}$; if $T_{nk} = 1$, then $\text{AGG}(x_{nk}.A_m) = x_{nk}(1).A_m$, otherwise, $\text{AGG}(x_{nk}.A_m)$ is equal to a single value after applying an aggregate function to the multi-set of values $x_{nk}.A_m$.

The basic aggregate functions to achieve the purpose could be any aggregate operations on a set: cardinality or count, maximum, minimum, mean or average, median, mode, sum, or even some composite aggregates, etc., depending on the type of attributes. However, the general aggregators are only good choices in some situations or under some conditions, they are unable to represent the complete distribution of values in a multi-set. We define a new type of aggregate that is able to represent both value and the distribution of values in a multi-set. A *partial frequency aggregate* $\text{PFA}(A, d)$ on a discrete attribute A with $\text{Dom}(A) = \{v_1, \ldots, v_k\}$ in an observed multi-set of objects d is defined to be a k-dimensional frequency vector $[f_1^d \ldots f_k^d]$, where $f_i^d, 1 \leq i \leq k$, is the frequency of value v_i within the set d. For example, assume the attribute $Gender$ has a domain of $\{$male,female$\}$. The partial frequency aggregates of two observations $\{2$ males and 1 female$\}$ and $\{2$ males and 3 females$\}$ are $[\frac{2}{3}\ \frac{1}{3}]$ and $[\frac{2}{5}\ \frac{3}{5}]$ respectively. Together with the count number, PFA provides a good description and statistics of a subset of part objects, so that they are sufficient in calculating the log-likelihood distances between (sets of) composite objects. An example of a composite object and its aggregate object is shown in Figure 3.

4 MODEL-BASED CLUSTERING

An integrated two-stage model-based clustering method is developed based on the model-based clustering strategy in (Fraley and Raftery, 1998), where a mixture model is dealt with by applying HAC to provide tentative and suboptimal partitions, and the EM algorithm to refine and relocate the partitions to reach the optimal result.

4.1 Clustering Models

Here we assume a *discrete multinomial mixture model* (Meilă and Heckerman, 1998) for a set of aggregate attributes $X = (AA_1, \ldots, AA_M)$ and a set of aggregate objects $D = \{x(1), \ldots, x(N)\}$. Let Θ stand for the set of parameters of the model, model-based HAC is associated with a *classification log-likelihood*

$$\ell_C(\Theta, C; D) = \sum_{n=1}^{N} \sum_{c=1}^{C} \sum_{m=1}^{M} \log P(x(n).AA_m|\theta_c), \quad (1)$$

where c is used to label the classification: $x(n)$ belongs to the c-th cluster only; and θ_c represents the

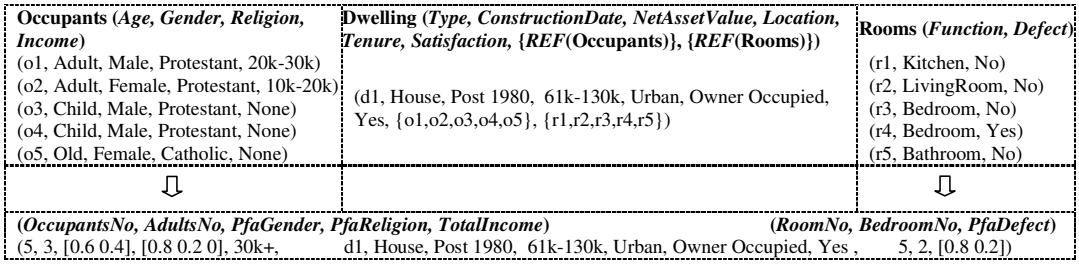

Figure 3: An example of a composite object and aggregate object. Each attribute is set to be categorical. Three PFA attributes are included.

set of parameters of the c-th model distribution, such that $\Theta = \{\theta_1, \ldots, \theta_C\}$. In contrast, a *mixture clustering model* is used in the model-based clustering with EM algorithm, and the relevant *mixture log-likelihood* is expressed as

$$\ell_M(\Theta, C; D) = \sum_{n=1}^{N} \log \big[\sum_{c=1}^{C} \pi_c \prod_{m=1}^{M} P(x(n).AA_m|\theta_c) \big], \quad (2)$$

where π_c is the mixing probability that an object belongs to the c-th cluster, $\pi_c \geq 0$, $\sum_{c=1}^{C} \pi_c = 1$; and $\Theta = \{\theta_1, \ldots, \theta_C; \pi_1, \ldots, \pi_C\}$.

Moreover, the *likelihood ratio* (LR) criterion (Everitt, 1981) and *Bayesian information criterion* (BIC) (Fraley and Raftery, 1998) are used to detect the stopping rules and to determine the optimal number of clusters in the course of clustering. Let k be an arbitrary number of clusters, q_m be the number of categories of attribute AA_m, r_m be the number of vector values if AA_m is a frequency aggregate attribute, and M_v be the total number of frequency aggregate attributes; we then define, for a given data set D, the log-likelihood ratio $LR(D, k)$, the BIC score for mixture classification model $BIC_C(D, k)$ and the BIC score for mixture clustering model $BIC_M(D, k)$, respectively, as

$$LR(D, k) = -2 \log \frac{\ell_C(\Theta, k; D)}{\ell_C(\Theta, k+1; D)}, \quad (3)$$

$$BIC_C(D, k) = -2\ell_C(\Theta, k; D) + \delta_k \log(N), \quad (4)$$

$$BIC_M(D, k) = -2\ell_M(\Theta, k; D) + (\delta_k + k - 1) \log(N), \quad (5)$$

where $\delta_k = k\big[\sum_{m=1}^{M}(q_m - 1) + \sum_{m=1}^{M_v}(r_m - 1)\big]$. Note that a frequency aggregate attribute has a domain of vector values with a dimension equal to the number of categories of the original attribute it aggregates from, so the total number of the independent parameters of both two types of attributes (δ_k) are considered in the model complexity penalized term of BIC scores. In addition, the number of mixing probabilities ($k - 1$ for each object) must be penalized in BIC for mixture models as well. The overall hierarchical model-based clustering algorithm can be expressed as follows.

1. **Detecting stopping rules**: Perform model-based HAC for the data set D to reach up to 2 clusters, while computing $LR(D, c)$ and $BIC_C(D, c)$ for each cluster number c in each step; let $C_l = \arg\min(LR(D, c))$ and $C_u = \lceil \frac{C_l + \arg\min(BIC_C(D,c))}{2} \rceil$ be the lower bound and upper bound of the stopping rules of further clustering.
2. **Clustering**: Perform the following two steps for each number of clusters $c = C_l, \ldots, C_u$
   2.1. **Tentative clustering**: Perform model-based HAC to reach up to c clusters.
   2.2. **Relocation partitions**: Perform the EM algorithm, starting with c clusters from HAC and compute $BIC_M(D, c)$.
3. **Determining the optimal number of clusters**: Choose the clustering with the first local minimum of all the $BIC_M(D, c)$ as the clustering result with the optimal number of clusters, $C = \arg\min(BIC_M(D, c))$.

4.2 Clustering Algorithms

The model-based HAC provides a likelihood distance measure (Meilă and Heckerman, 1998), such that a maximum log-likelihood (ML) can be maintained for the joint probability density of all the data records. For the discrete multinomial mixture model, the ML of C_j, the j-th cluster, takes the form

$$\hat{l}_j(\hat{\theta}_j; D_j) = \sum_{m=1}^{M} \sum_{q=1}^{q_m} N_{jmq} \log \frac{N_{jmq}}{N_j}, \quad (6)$$

where $\hat{\theta}_j$ is the ML parameters of C_j; D_j is the set of data cases involved in C_j; N_j and N_{jmq} are the number of cases (sufficient statistics) in C_j and the number of cases in C_j whose m-th attribute takes the q-th category of values, respectively. By merging two clusters, e.g. C_j and C_s, and assigning all their data cases to the newly formed cluster $C_{<j,s>}$, the *log-likelihood distance* $d(j, s)$ is set to be the decrease in ML resulting by the merge

$$d(j, s) = \hat{l}_j(\hat{\theta}_j; D_j) + \hat{l}_s(\hat{\theta}_s; D_s) - \hat{l}_{<j,s>}(\hat{\theta}_{<j,s>}; D_{<j,s>}).$$

The algorithm is described as follows, assuming we maintain two linked lists of clusters and of aggregate objects, and the stopping number of clusters is set to be a pre-specified number $K < N$.

1. **Initialization**:
   1.1. For $n = 1, \ldots, N$, initialize C_n to contain $x(n)$;

1.2. For $n = 1, \ldots, N-1$, [for $j = n+1, \ldots, N$, compute $\alpha_n = \min(d(n,j))$ and $\beta_n = \arg\min_j \alpha_n$].

2. **Iteration**: For $k = N, N-1, \ldots, K$, do

2.1. **get cluster with minimum distance**: for $i = 1, \ldots, k$, search C_n with $\min(\alpha_i)$;

2.2. **merge clusters**: form $C_{<n,\beta_n>}$ by merging C_n and C_{β_n}, and set $C_n \leftarrow C_{<n,\beta_n>}$;

2.3. **update clusters preceding** C_n: for $n' = 1, \ldots, n-1$, [compute $d(n', n)$ and update $\alpha_{n'}$ and $\beta_{n'}$ if necessary; if $\beta_{n'} = \beta_n$ then recompute $\alpha_{n'}$ and $\beta_{n'}$];

2.4. **update the new formed cluster** C_n: for $n' = n+1, \ldots, k$, compute $d(n, n')$ and update α_n and β_n;

2.5. **update clusters following** C_n: for $n' = n+1, \ldots, \beta_n - 1$, if $\beta_{n'} = \beta_n$ then recompute $\alpha_{n'}, \beta_{n'}$;

2.6. erase cluster C_{β_n} from the cluster list.

3. **Finish**: For $k = 1, \ldots, K$, output θ_k, π_k and C_k.

The log-likelihood distance depends only on the objects of the clusters being merged, and all the other distances remain unchanged. However, the time complexity of the algorithm is between $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$ (Meilă and Heckerman, 1998).

Another issue is the computation of the distance between two clusters that contain each one object. For a nominal (unordered) attribute, the *Hamming distance* is used to calculate the differences between two observed values; for an ordinal attribute, the *normalized Manhattan distance* is applied; for a frequency aggregate attribute that takes a vector value, the *normalized Euclidean distance* between two vector values is calculated with a normalized constant $\frac{1}{\sqrt{2}}$.

In practice, HAC based on classification model often gives good, but suboptimal partitions. The EM algorithm can further refine and relocate partitions when started sufficiently close to the optimal value. The mixture clustering likelihood is used as the basis for the EM algorithm, because it models a conditional probability τ_{nk} that an object $x(n)$ belongs to a cluster C_k, in contrast, τ_{nk} is assumed to be either 1 or 0 in the classification model. The EM algorithm is a general approach for maximizing likelihood in the presence of hidden variables and missing data (Fraley and Raftery, 1998), i.e. the class label attribute, τ_{nk} and π_k.

1. **E-step**: for $n = 1, \ldots, N$ and $k = 1, \ldots, K$, compute the conditional expectation of τ_{nk} by

$$\hat{\tau}_{nk} = \frac{\hat{\pi}_k P_k(x(n)|\hat{\theta}_k)}{P(x(n))} = \frac{\hat{\pi}_k \prod_{m=1}^{M} \prod_{q=1}^{q_m} \hat{\theta}_{kmq}^{x_{mq}(n)}}{\sum_{k=1}^{K} \hat{\pi}_k \prod_{m=1}^{M} \prod_{q=1}^{q_m} \hat{\theta}_{kmq}^{x_{mq}(n)}}$$

where $x_{mq}(n)$ stands for the value (1 or 0) of $x(n).AA_m$ in its q-th category.

2. **M-step**: for $k = 1, \ldots, K$, estimate the expectation of π_k and θ_k by

$$\hat{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} \hat{\tau}_{nk}, \quad \hat{\theta}_{kmq} = \frac{\sum_{n=1}^{N} \hat{\tau}_{nk} x_{mq}(n)}{\sum_{n=1}^{N} \hat{\tau}_{nk}}.$$

The iteration will converge to a local maximum of the likelihood under mild conditions, although the convergence rate may be slow in most cases.

The BIC provides a kind of score functions that not only measures the goodness of fit of the model to the data, but also penalizes the model complexity, e.g. the total number of model parameters or the storage space of model structure. We apply BIC to both the classification model (Equation (4)) and the mixture clustering model (Equation (5)). Accordingly, the smaller the value of BIC, the stronger the model. BIC_C, in model-based HAC, is used to compute the upper bound (stopping rule) of the EM; and BIC_M, in the EM algorithm, is applied to find the optimal number of clusters. A decisive first local minimum indicates strong evidence for a model with optimal parameters and number of clusters (see Figure 4 for example).

5 EXPERIMENTAL RESULTS

We apply the approach to a real world relational dataset, which contains about 10,000 records of the survey information of various types of dwellings. As mentioned in section 2, the data is modelled using a relational aggregate schema, where *Dwelling* table plays a role of base class, with *Occupants* table and *Rooms* table being two part classes. We chose some significant attributes from the three tables and dealt with their domains of values so that all the attributes are categorical. After aggregating the attributes of Occupants table and House table, we got a set of composite objects with aggregate attributes of (*OccupantsNo, AdultsNo, PfaGender, PfaReligion, TotalIncome, RoomNo, BedroomNo, PfaDefect*), in which *PfaGender, PfaReligion* and *PfaDefect* are three partial frequency aggregate attributes with vector values (see an example in Figure 3).

Table 1: Experimental Result

number of objects	1,000	3,000	5,000	9,530
(Lower,Upper) Bound	(2,7)	(2,11)	(3,13)	(2,15)
number of clusters	6	9	11	14
HAC running time (sec.)	50	427	1,159	4,024

After clearing the objects that have missing data, we got 9,530 aggregate objects left, from which four groups are selected for clustering, 1,000 objects, 3,000 objects, 5,000 objects and the whole dataset. The EM algorithm runs until either the difference between successive log-likelihood is less than 10^{-5} or 100 iterations are reached. The results for the four groups is listed in Table 1. Figure 4 show the two plots of the mixture BIC scores and -2log-likelihood values against the number of clusters for the last two

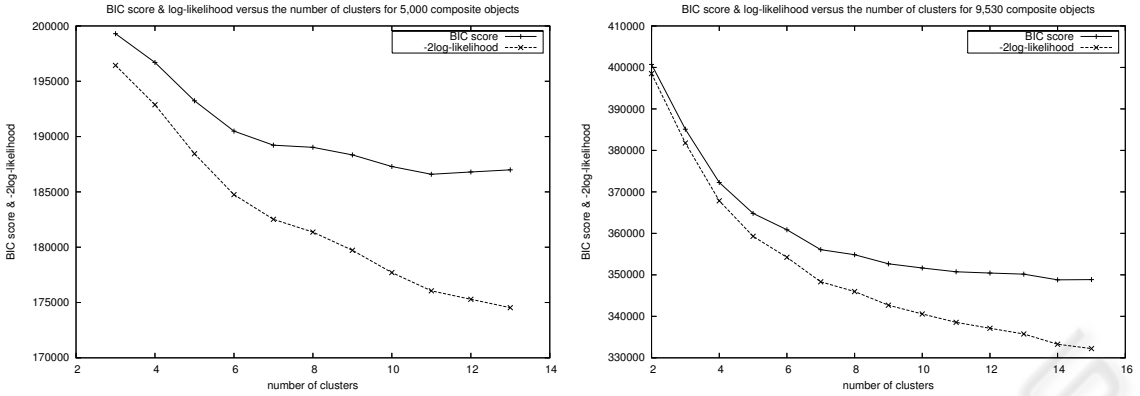

Figure 4: The BIC and log-likelihood versus the number of clusters for the datasets of 5,000 and 9,530 composite objects. The first local minimum shows the optimal numbers of clusters found by the EM algorithm are 11 and 14 clusters respectively.

groups respectively. An observation from the experiments is that the optimal number of clusters found by the algorithm is increasing as the total number of objects increases. This can be verified from Equation (5), where the likelihood term ($\mathcal{O}(N)$) dominates the penalty term ($\mathcal{O}(\log N)$) as N gets larger.

The clustering results are significant and convincing. The count aggregates and frequency aggregates play important roles in the clustering. The dataset tends to be partitioned into groups that have distinct number of part objects, e.g., dwellings with distinct number of occupants and rooms, together with properties of distinct aggregate frequencies, e.g., dwellings of protestant families and dwellings in which fewer room defects are reported. In addition, by analysing the BIC curves in Figure 4, it is reasonable for us to partition the whole dataset into 7 distinct clusters at last.

6 CONCLUSION

Compared with other work, our method is a propositional approach in relational data mining. We borrowed some ideas from (Fraley and Raftery, 1998; Meilă and Heckerman, 1998), and provide some extensions in dealing with aggregate attributes. We define frequency aggregates so that both the values and the distribution of values can be recorded for composite objects. Frequency aggregates are well applied in computing log-likelihood distance. We also present a method of determining the lower and upper bounds for the EM and get good results from the experiments.

Some future work are planned to do: handling continuous attributes as well as discrete attributes; dealing with missing data or data with noise; and applying relational distance measurements, e.g. (Emde and Wettschereck, 1996) to develop a relational model-based clustering method.

REFERENCES

Connolly, T. M. and Begg, C. E. (2002). *Database Systems: A Practical Approach to Design, Implementation, and Management*. Harlow: Addison-Wesley, third edition. International computer science series.

Džeroski, S. and Lavrač, N. (2001). *Relational Data Mining*. Springe-Verlag, Berlin.

Džeroski, S. and Raedt, L. D. (2003). Multi-relational data mining: a workshop report. *SIGKDD Explorations*, 4(2):122–124.

Emde, W. and Wettschereck, D. (1996). Relational instance-based learning. In *Proc. ICML-96*, pages 122–130, San Mateo, CA. Morgan Kaufmann.

Eriksson, H.-E. and Penker, M. (1998). *UML Toolkit*. John Wiley and Sons, New York.

Everitt, B. (1981). *Cluster Analysis*. Halsted Press: John Wiley and Sons, New York, second edition.

Fraley, C. and Raftery, A. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588.

Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. (1999). Learning probabilistic relational models. In *Proc. IJCAI-99*, pages 1300–1307, Stockholm, Sweden. Morgan Kaufmann.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall.

Meilă, M. and Heckerman, D. (1998). An experimental comparison of several clustering and initialization methods. In *Proc. UAI 98*, pages 386–395, San Francisco, CA. Morgan Kaufmann.

Taskar, B., Segal, E., and Koller, D. (2001). Probabilistic classification and clustering in relational data. In Nebel, B., editor, *Proc. IJCAI-01*, pages 870–878, Seattle, US.