Conceptual Clustering of Heterogeneous Distributed Databases

Sally McClean, Bryan Scotney, Kieran Greer and Rónán Páircéir

School of Information and Software Engineering,
University of Ulster, Coleraine BT52 1SA, Northern Ireland

{SI.McClean, BW.Scotney, KRC.Greer, R.Pairceir}@ulst.ac.uk

Abstract. With increasingly more databases becoming available on the Internet, there is a growing opportunity to globalise knowledge discovery and learn general patterns, rather than restricting learning to specific databases from which the rules may not be generalisable. Clustering of distributed databases facilitates learning of new concepts that characterise common features of, and differences between, datasets. We are here concerned with clustering databases that hold aggregate count data on a set of attributes that have been classified according to heterogeneous classification schemes. Such aggregates are commonly used for summarising very large databases such as those encountered in data warehousing, large-scale transaction management, and statistical databases. For measuring difference between aggregates we utilise two distance metrics: the Euclidean distance and the Kullback-Leibler information divergence. A hybrid between Kullback-Leibler and the Euclidean distance, which uses the former to learn the class probabilities and the latter as the corresponding distance measure, looks particularly promising both in terms of accuracy and scalability. These metrics are evaluated using synthetic data. Important applications of the work include the clustering of heterogeneous customer databases for the discovery of new marketing concepts and the clustering of medical databases for the discovery of new epidemiological concepts.

1. Introduction

Clustering of distributed databases facilitates the learning of new concepts that characterise important common features of, and differences between, datasets. For example, we may have a number of supermarkets belonging to a multinational chain, and each supermarket maintains a database describing its customers. Then we may cluster the databases to learn new high level concepts that characterise groups of supermarkets. More generally, with increasingly more databases becoming available on the Internet, such an approach affords an opportunity to globalise knowledge discovery and learn general patterns, rather than restricting learning to specific databases from which the rules may not be generalisable.

In this paper we are concerned with clustering databases that hold aggregate count data on a set of attributes that have been classified according to heterogeneous classification schemes. Such data are often stored in an OLAP-style database but may also be obtained by pre-processing native databases. Aggregates are commonly used for summarising information in very large databases, typically those found in data warehouses, large-scale transaction management, and statistical databases. An important special case of native databases that may be summarised in this way is provided by itemset data which store, for example, binary data on whether, or not, a customer bought each of a given set of possible items in a transaction. Databases of this type may have been previously extracted from, or may be created on-the-fly from, on-line transactional databases as materialised views. The aggregates then represent sufficient statistics for subsequent clustering. This approach to distributed database clustering, which is based on aggregates that are sufficient statistics, not only results in big improvements in efficiency [1], but also allows us to preserve anonymisation of individual tuples.

Heterogeneity is common in distributed databases, which typically have developed independently. We consider situations where, for a common concept, there are heterogeneous classification schemes. Local ontologies, in the form of such classification schemes, may be mapped onto a global ontology; these mappings are described by a correspondence graph. Such heterogeneity may arise due to differences in the granularity of data stored in different distributed databases, may be due to differences in the underlying concepts, or indeed may be due to missing attributes. The resolution of such conflicts remains an important problem for database integration in general [2], [3], [4] and for database clustering, in particular. There are also obvious applications to Knowledge Discovery in Databases [5].

An important aspect of clustering is the identification of an appropriate distance (similarity) metric. For measuring the difference between aggregates we utilise two distance metrics: the Euclidean distance and the Kullback-Leibler information divergence. In our methodology we may need to use the correspondence graphs to first construct a *dynamic shared ontology* for the candidate clusters before we are able to calculate the distance between heterogeneous datasets.

We report on extensive performance experiments to assess the various clustering approaches, thus providing a rigorous evaluation of the various clustering methods. The results thus obtained are very encouraging with regard to accuracy and scalability.

While there has been preliminary work on clustering homogeneous databases [6], there has been little previous research on clustering heterogeneous databases. The novelty of our work resides both in the development of a methodology for clustering heterogeneous databases, and in the development and evaluation of scalable algorithms.

2. Terminology and Data Models

We are concerned with a conceptual clustering problem where we cluster data on the Cartesian product of attributes, which are classified according to heterogeneous classification schemes. Thus, in Table 1, we illustrate a clustering problem where each table represents the profile of customers in a different supermarket. The class {Non-Professional} in Supermarket 3 maps onto {Manual} $\cup$ {Other} in Supermarket 1, while {Young} in Supermarket 2 maps onto {Young $\cap$ Professional} $\cup$ {Young $\cap$ Manual} $\cup$ {Young $\cap$ Other} in Supermarket 1.

Supermarket 1	Young	Middle aged	Old
Professional	85	3	1
Manual	5	2	0
Other	3	1	0

Supermarket 3	Young	Middle aged	Old
Professional	1	0	0
Non-Professional	3	68	12

Supermarket 2	Young	Middle aged	Old
	5	70	25

Table 1. Heterogeneous clustering data

In such tables the underlying data are symbolic, e.g. the attribute Age might have domain values: Young, Middle aged and Old, and the aggregate values are derived by counting the number of tuples possessing a particular combination of domain values.

Definition 2.1: We define a *datacube* D as comprised of a set of attributes $A_1,\ldots, A_n$ with corresponding domains $D_1,\ldots, D_n$. Their Cartesian product is $D_1\times\ldots\times D_n$. Let the classes of domain D_j be given by $\{c_1^{(j)},\ldots,c_{k_j}^{(j)}\}$. Then the classes of the Cartesian product are of the form $\{c_{a_1}^{(1)}\times\ldots\times c_{a_n}^{(n)}\}$ where $a_i \in \{1,\ldots,g_i\}$, i =1,...n, and g_i is the number of classes in D_i; we code these value labels for the Cartesian product classes using integers, as in Table 2. Then the cardinality of class v_j in the Cartesian product is given by n_j for j=1,…k and $k=\prod_{j=1}^{n} g_j$. We note that, in Statistics, objects such as the datacubes we have defined are known as contingency tables.

Malvestuto [7] has discussed classification schemes that partition the values of an attribute into a finite number of classes. A classification P is defined to be finer than a classification Q if each class of P is a subset of a class of Q. Q is then said to be coarser than P. Such classification schemes may be specified by the database schema or may be identified by appropriate algorithms. The product (or join) of two classification schemes P and Q is the coarsest partition which is finer than both P and Q. The sum (meet) of two classification schemes P and Q is the finest partition which is coarser than both P and Q. The relationship between two classification schemes may be described by a correspondence graph [7], in which nodes represent classes and arcs indicate that the associated classes overlap.

Definition 2.2: A *marginal datacube* is obtained by projecting out one or more attributes of the Cartesian product in the datacube, e.g. Table 1 contains a marginal datacube for the Supermarket 2 data. In our framework, such marginal datacubes are an important source of heterogeneity.

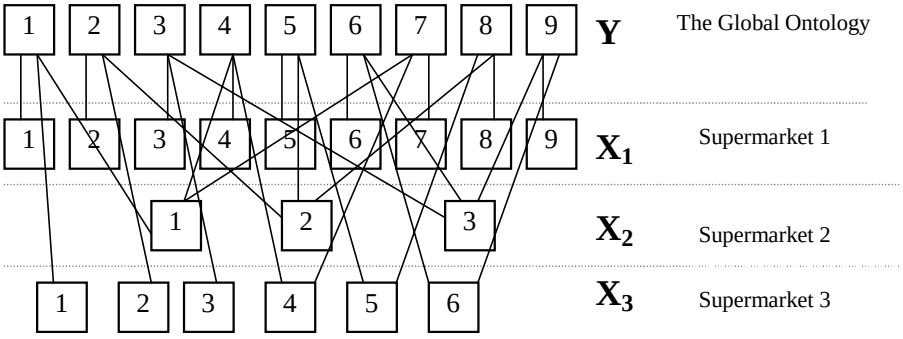
It is usually the case in a distributed database that there is a shared ontology that specifies how the local semantics correspond to the global meaning of the data; these ontologies are encapsulated in the classification schemes. The relationship between the heterogeneous local and global schema is then described by a correspondence graph held in a correspondence table. Thus, for example, for the data in Table 1, the global ontology is as presented in Table 2.

Code	Value label	Code	Value label
1.	Young Professional	6.	Old Manual
2.	Middle-aged Professional	7.	Young Other
3.	Old Professional	8.	Middle-aged Other
4.	Young Manual	9.	Old Other
5.	Middle-aged Manual		

Table 2. The global ontology for Table 1

Definition 2.3: We define a *correspondence graph* for a set of datacubes to be a multipartite graph $G=(X_1 \cup X_2 ... \cup ... X_m \cup Y, E)$; X_i is the set of nodes corresponding to the classification scheme (local ontology) of the ith datacube; Y is the set of nodes corresponding to the classification scheme of the global ontology; E is the set of edges which join the nodes in X_i to the nodes in Y. Each sub-graph $G=(X_i \cup Y, E)$ is bipartite for i=1,...,m. Thus each node in Y which represents a value contained in a value in X_i provides an edge. The graph thus describes the schema mappings between the local and global ontologies. The correspondence graph for Table 1 is presented in Figure 1 and the equivalent correspondence table in Table 3.

Figure 1. The Correspondence Graph for Table 1



Global Ontology	Local Ontology 1	Local Ontology 2	Local Ontology 3
1	1	1	1
2	2	2	2
3	3	3	3
4	4	1	4
5	5	2	5
6	6	3	6
7	7	1	4
8	8	2	5
9	9	3	6

Table 3. The Correspondence Table for Table 1

Such distributed datacubes are therefore heterogeneous with respect to classifications schemes, in the sense that some may be finer or coarser than others, as for the Supermarket 3 data in Table 1, or may be heterogeneous with respect to attributes in the sense that attributes may be missing in some of the marginal datacubes, as for the Supermarket 2 data in Table 1.

3. The Distance Metrics

In order to cluster datacubes such as we have defined, it is first necessary to develop appropriate distance (similarity) metrics. For measuring differences between such aggregates we here utilise two main distance metrics: the Euclidean distance and the Kullback-Leibler information divergence. In our approach we may need to use the correspondence graphs to first construct a *dynamic shared ontology* for the candidate clusters before we are able to calculate the distance between heterogeneous datasets. Essentially, in such situations, we need to homogenise the classification schemes before we can compare the corresponding datasets.

3.1 The Euclidean Distance Metric

For the Euclidean distance metric we must first homogenise by aggregating the datacubes to the partition level which is the sum classification scheme for the contributing schemes for the datacubes. The sum classification scheme is here the *dynamic shared ontology*. In Section 2 we have defined a datacube in terms of a set of cardinalities $\{n_j\}$ for j=1,..k. We now require to determine distance metrics for the distance between two datacubes $D_1 = \{n_{1j}\}$, $D_2 = \{n_{2j}\}$. We define the respective probabilities as π_{ij} of value v_j in datacube D_i, for j=1,…k. Here, k is the number of classes in the dynamic shared ontology and we are clustering the probability distributions of the respective datacubes.

Definition 3.1: The *Euclidean distance* between two heterogeneous datacubes D_1 and D_2 is then defined as

$$d_{12} = \sum_{j=1}^{k} (\pi_{1j} - \pi_{2j})^2 ,$$

where the π_i's are calculated for the dynamic shared ontology as $\pi_{ij} = n_{ij}/n_{i.}$, where $n_{i.} = \sum_{j=1}^{k} n_{ij}$.

3.2. The Kullback-Leibler Distance Metric

This approach allows us to measure the distance between datacubes by minimisation of the Kullback-Leibler information divergence, using the EM algorithm. The EM (Expectation-Maximisation) algorithm is a widely used general class of iterative procedures used for learning in the presence of missing information. In our case, the problem may be shown to belong to a general class of such problems termed Linear Inverse Problems [8]. The advantage in using such an approach is that, in this case, there is no need to compute the sum classification scheme; the datacubes may be compared directly. Our approach thus allows us to avoid the relatively computationally expensive stage of computing the shared ontology.

Notation 3.1: We consider a Cartesian product of attributes with corresponding global ontology domain $G = \{v_1,...,v_k\}$. Then for datacube D_r the local classification scheme is partitioned into sets $S_{r1},...,S_{rg_r}$, and n_{rs} is the cardinality of set S_{rs}. Here g_r is the number of sets in the local partition of datacube D_r. We further define:

$$q_{irs} = \begin{cases} 1 & \text{if } v_i \in S_{rs} \\ 0 & \text{otherwise} \end{cases}$$

and the correspondence table $R_{ij} = \{i : v_j \in S_{ri}\}$.
The proportion of each of the datacube values within the integrated datacube is $f_{rs} = n_{rs}/N$, where

$N = \sum_{r=1}^{2} \sum_{j=1}^{g_r} n_{rj}$

(the total cardinality).

Definition 3.2: We define the *integrated probabilities* $\pi_1,\ldots \pi_k$ which pool a number of heterogeneous datacubes $D_1,..$ D_m by the iterative scheme:

$$\pi_i^{(n)} = \pi_i^{(n-1)} * (\sum_{r=1}^{m} \sum_{s=1}^{g_r} (f_{rs} q_{irs} / \sum_{u=1}^{k} \pi_u^{(n-1)} q_{urs})) \quad \text{for i = 1,...k.}$$

Here π_i is the probability of value v_i for i = 1,…,k for the integrated aggregate view in the global ontology.

Theorem 3.1: The integrated probabilities, as defined, minimise the Kullback-Leibler information divergence between the aggregated probabilities $\{\pi_i\}$ and the data $\{f_{rs}\}$. This is equivalent to maximising the likelihood of the model given the data.

Proof: In our case, minimising the Kullback-Leibler information divergence, or equivalently maximising the log-likelihood, becomes:

$$\text{Maximise } W = \sum_{r=1}^{m} \sum_{s=1}^{g_r} f_{rs} \log(\sum_{i=1}^{k} q_{irs} \pi_i) \text{ subject to } \sum_{i=1}^{k} \pi_i = 1.$$

Such missing information problems are also well known in Statistics [9], where they may be regarded as the maximum likelihood estimation of multinomial probabilities for grouped data. In our case, the EM algorithm can be shown to reduce to the iterative scheme in Definition 3.2. The EM algorithm has been shown to converge monotonically to the solution of the minimum information divergence equation for such problems [9].

Definition 3.3 The *Kullback-Leibler distance* between two heterogeneous datacubes D_1 and D_2 is defined as the gain in information divergence if we merge the two datacubes where distance = $L_1 + L_2 - L_{12}$.
Here, the log-likelihood L is a scaled version of W, the information divergence, given by:

$$L = \sum \sum n_{rs} \log(\sum q_{irs} \pi_i)$$

and the π_i's are the solution of the EM iterations in Definition 3.2. This metric is equivalent to the log-likelihood ratio, which is well known in Statistics. Here L_1 is the log-likelihood value for cluster 1, L_2 is the log-likelihood value for cluster 2 and L_{12} is the log-likelihood value for the clusters combined. When testing whether to combine two clusters, we calculate the log-likelihood values for the two clusters separately and then the two clusters combined. If the value of L_{12} is inside a threshold of $L_1 + L_2$ then the clusters are sufficiently close together to be combined. This threshold is derived from the chi-squared value for the Likelihood Ratio statistic. The corresponding degrees of freedom are calculated as:

$$\text{degrees of freedom} = df_1 + df_2 - df_{12},$$

where df_1 is the degrees of freedom for cluster 1, df_2 is the degrees of freedom for cluster 2 and df_{12} is the degrees of freedom for the combined clusters.

4. Clustering the Databases

Clustering is usually based on a distance metric that allows us to assess distances between objects and distances between clusters. Recent work in the database literature has described a number of systems for efficient clustering including CLARANS [10], BIRCH [11], DBSCAN [12], and CLIQUE [13]. Using summaries for clustering categorical data has been described in [14]. Some work has also been done on distributed clustering although this has tended to be subject to vertical partitioning [15], [16], [17]; we envisage data that is horizontally partitioned. Parallel clustering algorithms are discussed , for example in [18]. Methods which contain an inbuilt stopping rule, e.g. those based on a statistical test, such as we have described for the Kullback-Leibler distance metric, have an obvious advantage over their competitors.
Example: Heart Disease Databases

(Submitted by David Aha to the ML Repository)
There are four databases each containing 76 attributes, including the decision variable Coronary Heart Disease (CHD), with values 0 (no CHD) to 4 (severe CHD). The databases are respectively: 1. Cleveland Clinic Foundation, 2. Hungarian Institute of Cardiology, 3. University Hospital Zurich and Basel, 4. Long Beach Clinic Foundation, as illustrated in Table 4.

	Male					Female					Total
Decision variable	0	1	2	3	4	0	1	2	3	4	
Cleveland	72	9	7	7	2	92	46	29	28	11	303
Hungarian	69	5	1	3	3	119	32	25	25	12	294
Swiss	0	6	3	1	0	8	42	29	29	5	123
Long Beach	3	3	0	0	0	48	53	41	42	10	200

Table 4. Clusters: {Cleveland, Hungarian}, {Swiss}, {Long Beach}

This is an example of what is known in Epidemiology as meta-analysis where we pool data from different epidemiological studies. This is often highly advantageous since such studies are often very expensive and cannot be frequently repeated or extended. We may thus use our approach to assess regional, cultural and temporal effects.

In general, clustering methods may be agglomerative (bottom up) or divisive (top down). Our approach is agglomerative. In Section 5, we evaluate datacube clustering using each of the Euclidean and Kullback-Leibler metrics. In addition we evaluate a hybrid algorithm which combines the advantages of these two approaches. In the hybrid approach we first obtain the probability distribution for each heterogeneous cluster using the Kullback-Leibler method; the Euclidean distance is then computed using these values. The advantage is that we can work at the global ontology level of the generalisation hierarchy and therefore save on computationally expensive homogenisation.

5. Performance Evaluation

5.1 An Overlap Metric

In order to generate synthetic distributed data with heterogeneous classification schemes, we must first provide a way of quantifying such heterogeneity. A metric has been defined to quantify the degree of overlap between the local and global classification schemes since we would expect the execution time to increase with increasing overlap. The metric is defined as follows

$$Overlap = \left(1 - \frac{\sum g_r}{m * k}\right) \Big/ \left(1 - \frac{1}{k}\right)$$

where k = the number of global classes, m = the number of local datacubes, and g_r = the number of classes in local classification scheme r. The *Overlap* metric then achieves a value of 0 if all the local schemes are identical to the global scheme (no overlap) and 1 (the maximum) if all the local schemes have only one category (complete overlap). Otherwise the *Overlap* is between 0 and 1.

5.2 Performance of the Clustering Algorithms

A set of synthetic data was generated where there were 5 distinct clusters and the number of global classes was 20. Overlap values of 0.1, 0.3, and 0.5, were tested. In all experiments a convergence tolerance of 10^{-4} for successive iterations was used as a stopping criterion for the EM algorithm. Also, in all experiments the initial probability values for the global categories were set to (1 / *the number of global probabilities*). To implement the clustering algorithms we have used *Java* classes called the banda classes, which were obtained from the WWW [19]. The graphs in Appendix A show the relative computation times for each test run; Appendix B presents the relative

accuracies. Here each measurement is averaged over 100 test runs. The tests were run on a Windows NT Workstation with a 700MHz processor.

From the Appendixes we see that for low overlap the Kullback-Leibler approach is faster than the Euclidean; this reverses for higher overlap. This is because for low overlap we would expect very few iterations of the EM algorithm while this reverses for high overlap, making the Kullback-Leibler approach inefficient in such circumstances. The hybrid method is faster in all cases, due to the fact that it requires less computation of candidate clusters. Accuracy of the Kullback-Leibler approach is always best, as we would expect, but the hybrid method improves in accuracy for higher overlap, making it a promising alternative to Kullback-Leibler for large-scale problems. This is probably explained by the fact that all algorithms become increasingly inefficient for high overlap, due to there being insufficient data at an appropriate granularity for accurate clustering. In fact, we would not expect clustering to be appropriate for databases with high overlap since the information quality is so low. Although these experiments are preliminary, the results presented here are nonetheless very encouraging.

6. Summary and Further Work

A methodology and algorithms have been provided for clustering datacubes. The datacubes (contingency tables) may be either homogeneous or heterogeneous with respect to the classification schemes in the contributing datacubes. Such an approach can lead to the discovery of new knowledge about similarities and dissimilarities between databases.

Clustering has been carried out using several distance functions, namely a Euclidean metric, a distance function based on the Kullback-Leibler information divergence and a hybrid of these two approaches. In all cases the methods derived a probability distribution for each cluster identified. This allows us to profile the clusters, with obvious advantages. The main advantages of the approach in our algorithms are:

- we can handle schema (classification) heterogeneity;
- we use aggregates and so only one scan of the database is required;
- clustering via the Kullback-Leibler distance metrics contains an in-built stopping criterion obtained from the likelihood ratio chi-squared test. Other common approaches require the number of clusters to be specified in advance;
- we can cluster a number of attributes simultaneously by clustering on their joint distribution; indeed attributes may be missing in some of the databases;
- we can use the probabilities found for each cluster to characterise the cluster.

We have illustrated our discussion with a number of potential applications from different domains. With burgeoning Internet developments, distributed databases are becoming increasingly commonplace; frequently these are heterogeneous. The potential for learning new knowledge from such data sources is enormous and open data systems are combining with Agent Technology to facilitate the uptake of such opportunities. There are a large number of domains that could benefit from such developments. In particular we have focussed on medicine, where pooling of epidemiological data can provide new insights into disease, and business, where database marketing can facilitate customer profiling and targeted selling.

Performance evaluations have been carried out and have provided promising indications of scalability, particularly for an approach that uses a distance metric that is a hybrid between the Euclidean and Kullback-Leibler distance. Further work will investigate other clustering approaches and consider the extension of the approach to clustering different types of homogeneous and heterogeneous objects.

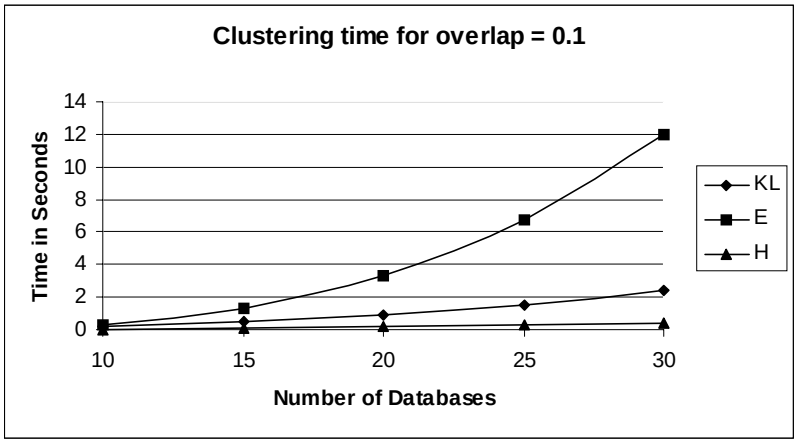
Acknowledgements

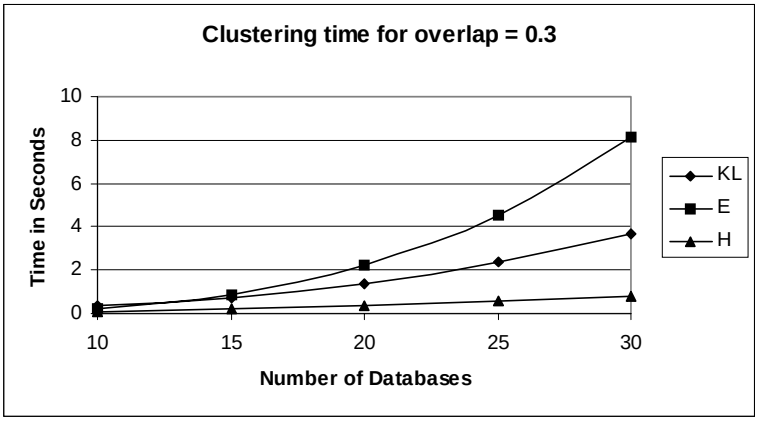
This work was partially funded by ADDSIA (ESPRIT project no. 22950) which is part of EUROSTAT's DOSIS (Development of Statistical Information Systems) initiative and partially funded by MISSION (IST project number 1999-10655) which is part of EUROSTAT's EPROS initiative.

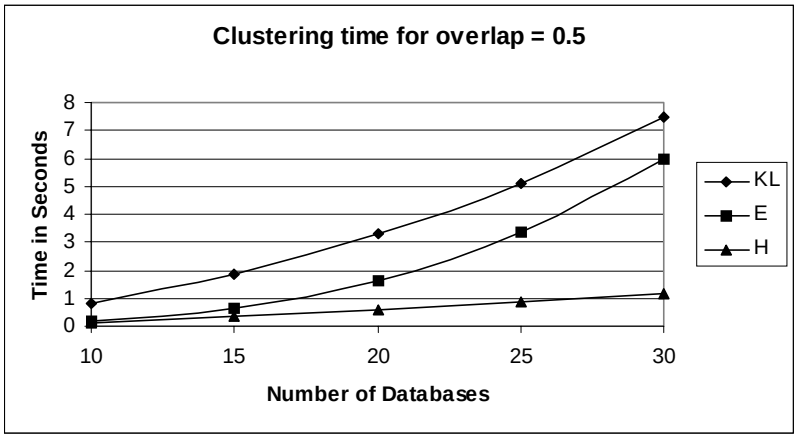
References

1. Forman, G., Zhang, B.: Distributed Data Clustering can be Efficient and Exact. SIGKDD Explorations, **2(2)** (2001) 34-38.
2. Lim, E.-P., Srivastava, J., Shekher, S.: An Evidential Reasoning Approach to Attribute Value Conflict Resolution in Database Management. IEEE Transactions on Knowledge and Data Engineering **8** (1996) 707-723.
3. Scotney, B.W., McClean S.I.: Efficient Knowledge Discovery through the Integration of Heterogeneous Data. Information and Software Technology (Special Issue on Knowledge Discovery and Data Mining) **41** (1999) 569-578.
4. Scotney, B.W., McClean, S.I., Rodgers, M.C.: Optimal and Efficient Integration of Heterogeneous Summary Tables in a Distributed Database. Data and Knowledge Engineering **29** (1999) 337-350.
5. Anand, S.S., Scotney, B.W., Tan, M.G., McClean, S.I., Bell, D.A., Hughes, J.G. Magill, I.C.: Designing a Kernel for Data Mining. IEEE Expert (1997) 65 - 74.
6. Parthasarathy, S., Ogihara, M.: Clustering Distributed Homogeneous Datasets. Proceedings of PKDD 2000, LNAI **1910**, (2000) 566-574.
7. Malvestuto F.M.: The Derivation Problem for Summary Data. Proc. ACM-SIGMOD Conference on Management of Data (1988) 87-96.
8. Vardi, Y., Lee, D.: From Image Deblurring to Optimal Investments: Maximum Likelihood Solutions for Positive Linear Inverse Problems (with discussion). J. Royal Statistical Society B (1993) 569-612.
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society **B39** (1977) 1-38.
10. Ng R.T., Han J.: Efficient and Effective Clustering Methods for Spatial Data Mining. Proc. International Conference on Very Large Databases (VLDB'94) (1994) 144-155.
11. Zhang, T., Ramakrishnan, R., Livny M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proc. ACM-SIGMOD International Conference on Management of Data (1996) 103-144.
12. Ester, M., Kriegel, H.-P., Sander, J., Wu, X.: A Density-based Algorithm for Discovering Clusters in large Spatial Databases. Proc. 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96) (1996) 226-231.
13. Agrawal R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proc. ACM SIGMOD International Conference on Management of Data, Seattle, Washington (1998).
14. Ganti, V., Gehrke, J., Ramakrishnam, R: CACTUS – Clustering Categorical Data using Summaries. Proc. 5th International Conference on Knowledge Discovery and Data Mining (KDD'99) (1999) 73-83.
15. Park, B.H., Ayyagari, R., Kargupta, H: A Fourier Analysis Based Approach to Learning Decision Trees in a Distributed Environment. Proc 1st SIAM International Conference on Data Mining (2001).
16. Johnson, K., Kargupta, H.: Collective, Hierarchical Clustering from Distributed, Heterogeneous Data. In Large-scale Parallel KDD Systems, eds. Zaki, M., Ho, C., LNCS (1999) 221-244.
17. Kargupta, H., Huang, W., Krishnamoorthy, S, Johnson, E.: Distributed Clustering Using Collective Principal Component Analysis. Knowledge and Information Systems (to appear) (2001).
18. Dhillon, I.S., Modha, D.S.: A Data-Clustering Algorithm on Distributed Memory Multiprocessors. Large-Scale Parallel Data Mining (1999) 245-260.
19. Bush, B.W.: BANDA Java Packages Version 7.6. http://www.sladen.com/Java/ (1998).

Appendix A: Comparison of clustering times







Appendix B: Comparison of clustering accuracy