1998 International Computer Symposium
Workshop on Software Engineering and Database Systems
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

DATABASE CLUSTERING AND DATA WAREHOUSING

Mei-Ling Shyu, Shu-Ching Chen, and R. L. Kashyap

School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285, U.S.A.
E-mail:{shyu, shuching, kashyap}@ecn.purdue.edu

ABSTRACT

Due to the complexity of real-world applications, the number of databases and the volume of data have increased tremendously. Discovering qualitative and quantitative patterns from databases in such a distributed information-providing environment has been recognized as a challenging task. In response to such a demand, data mining and data warehousing techniques are emerging to extract the previously unknown and potentially useful knowledge to provide better decision support. This paper presents a mechanism called *Markov Model Mediators (MMMs)* to facilitate the understanding of the data warehouse schemas/views and the improvement of the query processing performance by analyzing and discovering the summarized knowledge at the database level. Simulation results show that the data mining process leads to a better federation of data warehouses and reduces the cost of query processing. To illustrate these benefits, our approach has been implemented and a simple example and several experiments on real databases are presented.

1. INTRODUCTION

With the increasing complexity of real world applications, the need for discovering useful information and knowledge in a distributed information-providing environment has posed a great challenge to the database research community. Online databases, consisting of millions of media objects and objects, have been used in business management, government administration, scientific and engineering data management, and many other applications owing to the recent advances in high-speed communication networks and large-capacity storage devices. In addition, the number of such databases keeps growing rapidly and this explosive growth in data and databases has resulted in the research areas of *data warehousing* and *data mining* [3] [5] [18].

Data warehousing deploys database technologies for storing and maintaining data. A data warehouse is a subject-oriented, integrated, time-varying, non-volatile collection of data that is used primarily in organizational decision making [9]. Data warehousing aims at enabling better and faster decision making for knowledge worker (executive, manager, analyst) [3]. Decision support usually requires the data from many heterogeneous databases that have data of varying quality or of different representations. The data inconsistency among the databases have to be reconciled. However, our focus in this paper is to design the data warehouse schema and views, and therefore the details of the resolution of conflicts among databases are not discussed. Data warehousing technologies have been beneficial to many industries such as the manufacturing, retail, transportation, healthcare, telecommunications, etc. Since data warehousing is targeted for decision support, the summarized and consolidated information from data is more important than the detailed and individual records. The summarized and consolidated information may be missing from the data in the database, but can be obtained from the data mining techniques.

Data mining is a process to extract nontrivial, implicit, previously unknown and potentially useful information from data in databases. Many other terms such as *knowledge discovery in databases*, *knowledge mining from databases*, *knowledge extraction*, *data archaeology*, *data dredging*, *data analysis*, etc. carry a similar or slightly different meaning in the existing articles and documents [5]. Data mining involves data analysis techniques that are used in statistical and machine learning and related algorithmic areas. Three of the most common methods to mine data are association rules [14] [15], data classification [4] [11] and data clustering [7] [20]. Association rules discover the co-occurrence associations among data. Data classification is the process that classifies a set of data into different classes according to some common properties and classification models. Finally, data clustering groups physical or abstract objects into disjoint sets that are similar in some respect.

In databases, data clustering places related or similar valued records or objects in the same page on disks for performance reasons. A good clustering method ensures that the intra-cluster similarity is high and the inter-cluster similarity is low. Many data clustering

strategies have been proposed in the literature. Methods that rely on the designers to give hints on what objects are related require the domain knowledge of the designers [2] [12]. Syntactic methods such as depth first and breadth first, determine a clustering strategy based solely on the static structure of the database [10]. The disadvantages of this strategy are that it ignores the actual access patterns and the queries might not traverse the database according to the static structure. The third type of methods gather the statistics of the access patterns and partition the objects based on the statistics [16] [17]. Other strategies such as the placement tree clustering method in [1] and the decomposition-based simulated annealing clustering method [8] combine two or all of the above strategies.

However, in a distributed information-providing environment, the number of databases has increased tremendously and much of the data in each database is structural in nature. Moreover, the workloads are query intensive with mostly complex queries that tend to access millions of records from a set of databases in such an environment. Hence, instead of data clustering, there is a need to analyze and discover summarized knowledge at the database level, i.e. database clustering. Similar to data clustering, database clustering is to group related databases in the same cluster (data warehouse) such that the intra-cluster similarity is high and the inter-cluster similarity is low. Here, two databases are said to be related in the sense that they either are accessed together frequently or have similar records or objects. Those member databases that are conceptually placed in the same cluster are the data in a data warehouse. A federation of data warehouses is constructed, each with its own decentralized administration.

This paper considers conceptual database clustering rather than physical database clustering. Conceptual modeling allows an abstract representation of the participating databases and describes the databases with a set of conceptual schemas at different abstract levels. The objective of conceptual database clustering is to facilitate the understanding of the data warehouse schemas/views and the improvement of the query processing performance. An efficient database clustering approach can enhance the performance by placing on the same data warehouse the related set of databases. Query processing, in general, involves the closely interrelated *communication cost* and *processing cost*. Data warehouses may contain large volumes of data. To answer query efficiently, it requires a good database clustering strategy, a good data warehouse schema, and a good query processing technique. Essentially, since a set of databases belonging to a certain application domain is put in the same data warehouse and is required consecutively on some query access path, the number of platter switches (communication cost) and the number of node traversed (processing cost) for data retrieval with respect to queries can be reduced. On the other hand, physical database clustering aims at improving the performance of databases by actually moving around the databases that is not realistic given the autonomy of the databases.

In this paper, we propose the use of Markov model mediators (MMMs) as the schemas and views for data warehouses. First, the network of databases is represented as a browsing graph with each database represented as a node in the browsing graph. Then, a set of historical data, i.e. the usage patterns and access frequencies of the queries issued to the databases, together the data in the databases are used to generate training traces for the data mining process to mine the useful and summarized knowledge. It is not hard to record the usage patterns and the access frequencies of the queries issued to the databases since programs can be developed for this purpose. Hence, the access patterns of the browsing graph are modeled as a Markov Chain and each database is modeled as a local MMM with model parameters $\lambda=(\mathcal{S}, \mathcal{F}, \mathcal{A}, \mathcal{B}, \Pi, \Psi)$. Finally, the large browsing graph is then decomposed into a federation of data warehouses via the proposed stochastic clustering strategy which uses similarity measures calculated based on the sets of model parameters of the local MMMs. A larger similarity value between two local MMMs implies that they should belong to the same data warehouse. The model parameters of the local MMMs and the information for stochastic clustering are extracted from the summarized knowledge in the data mining process. The stochastic strategy is adopted since it provides better performance in object clustering [17]. After the federation of data warehouses is constructed, one integrated MMM serves as the schema and the view for one data warehouse. Moreover, the data mining process is executed based on the query loads over a certain period of time so that changing workloads implies changing the construction of data warehouses. Hence, we might need to reconstruct the set of data warehouses periodically, say annually or bi-annually.

A simple example is first used to illustrate how the historical data is used to construct a set of data warehouses. Then, we conduct several experiments with the use of real database management systems at Purdue University. Our experimental results demonstrate that our approach can dramatically reduce the cost of query processing in comparison with the random clustering method.

The rest of the paper is organized as follows. Problem formulation is given in Section 2. The information mining process for constructing a federation of data warehouses based on the proposed MMM mechanism is introduced in Section 3. Performance results are presented in Section 4. Section 5 contains the summary.

2. PROBLEM FORMULATION

The essence of a distributed information-providing environment is a large number of databases which are navigated by queries. Many queries in such a distributed information-providing environment require not only the detailed and individual records but also the summary and consolidated information in the databases. In addition, the cost of query processing is very expensive especially in such a large-scaled environment because the workloads are query intensive with mostly complex queries that tend to access millions of records from a set of databases. For example, in order to identify possible fraudulent claims, an insurance company needs to access information from several databases such as its own databases, the databases of other insurance companies, the databases of the police, etc. All these needed databases might be distributed at different places. The cost of query processing is pretty high when accessing these databases. However, if the related databases are conceptually grouped together, the cost of query processing can be expected to be reduced since these databases usually belong to a certain application domain and are required consecutively on some query access path. Hence, the need to perform database clustering by discovering the summarized knowledge in the databases to accelerate query processing has become inevitable. In response to such a demand, data mining and data warehousing techniques are emerging to extract the previously unknown and potentially useful knowledge to assist in conceptually organizing the databases to reduce the cost of query processing.

As a result, the network of databases is modeled as a browsing graph, called $\mathcal{G}$. Each database is associated with a node in the graph, and the directed arcs represent the relationships among the nodes in terms of traversing through databases. A query is processed by traversing the graph and retrieving the information as the required node (database) is visited. Since query processing in such a browsing graph is restricted directly by the topology of $\mathcal{G}$, given a node in $\mathcal{G}$ to be the current node, a query can access only one of the nodes incident with the current node and the selection of the next node is dependent only on the current node. An arc between two databases indicates that these two databases have some structurally equivalent media objects. Without loss of generality, it is assumed that any two databases are connected by two opposite directed arcs since the equivalence relationship is bi-directional. We can transform the browsing graph $\mathcal{G}$ into a Markov Chain in the following manner so that the browsing graph is equivalent to the transition diagram of the Markov Chain and the access patterns for queries on $\mathcal{G}$ can be modeled as a finite-state time-homogeneous Markov Chain.

(1) Each node in $\mathcal{G}$ is represented by a state in the Markov Chain.

(2) The branch probabilities in $\mathcal{G}$ are represented by the one-step transition probabilities in the Markov Chain.

Therefore, a new mechanism called *Markov Model Mediators (MMMs)* which adopt the *Markov Model* framework and the *mediator* concept is proposed in this paper. A Markov model is a well-researched mathematical construct which consists of a number of states connected by transitions. The states represent the alternatives of the stochastic process and the transitions contain probabilistic and other data used to determine which state should be selected next. All transitions $S_i \rightarrow S_j$ such that $Pr(S_j \mid S_i) > 0$ are said to be allowed, the rest are prohibited. A discrete-parameter Markov process or Markov sequence is characterized by the fact that each member of the sequence is conditioned by the value of the previous member of the sequence. A Markov Chain is a dynamic system, evolving in time. Since the current member, x_{k+1}, is conditionally independent of x_0, x_1, ..., x_{k-1} given x_k, the branch probabilities are independent of the time index k. Therefore, the Markov Chain is said to be homogeneous. The stochastic behavior of a homogeneous chain is determined completely by its model parameters. Since the access patterns of the databases can be modeled as a Markov Chain and the databases/media objects pertain to the characteristics of a discrete system, the compact notion $\lambda=(\mathcal{S}, \mathcal{F}, \mathcal{A}, \mathcal{B}, \Pi, \Psi)$, where $\mathcal{S}$ is a set of media objects called states, $\mathcal{F}$ is a set of attributes/features, $\mathcal{A}$ denotes the state transition probability distribution, $\mathcal{B}$ is the observation symbol probability distribution, Π is the initial state probability distribution, and Ψ is a set of augmented transition networks (ATNs), is adopted. The augmented transition network (ATN) is a semantic model to model multimedia presentations, multimedia database searching, and multimedia browsing. For the details of ATNs and how MMMs are used for database searching, please see [6] [13]. [19] defines a mediator to be a program that collects information from one or more sources, processes and combines it, and exports the resulting information. In other words, mediators can be said to be a program or a device which expresses how to integrate different databases.

Each database is modeled by a local MMM. The primary objectives for constructing local MMMs are to achieve data model homogeneity by transforming each local schema expressed in different data models into a single model, to achieve uniformity in the modeling constructs, and to store the semantic knowledge gathered about the media objects inside a database. The structure of the member media objects in a database is modeled by the sequence of the MMM states connected by transitions. The model parameters and the affinity relation between two databases are mined from

the state sequence, the individual databases, and a set of historical data such as the usage patterns and access frequencies of the queries issued to the databases. According to the discovered affinity relations, the large browsing graph can be clustered into a federation of data warehouses. Conceptually, a data warehouse has a set of related databases and an integrated MMM serves as the schema and the view of the data warehouse. Conceptual data warehouse is an abstract representation of the participating databases rather than actually moving the databases around. Two databases are said to be related in the sense that they either are accessed together frequently or have similar records or objects. Then the constructed data warehouses play the roles to reduce the cost of query processing.

Table 1: The usage patterns – the entity with value 1 indicates the query accessed the corresponding media object. For example, the media object 2 ($C_{1,2}$) has been accessed by queries q_1, q_2, q_5, and q_7.

usage	query							
	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
1 ($C_{1,1}$)	1	1	0	1	1	1	0	0
2 ($C_{1,2}$)	1	1	0	0	1	0	1	0
3 ($C_{2,1}$)	1	0	1	0	1	0	0	1
4 ($C_{2,2}$)	1	0	1	1	1	0	1	1
5 ($C_{2,3}$)	0	0	1	1	0	0	1	1
6 ($C_{3,1}$)	0	1	1	1	1	1	0	0
7 ($C_{3,2}$)	0	1	1	1	0	0	1	1
8 ($C_{3,3}$)	0	1	1	0	1	0	1	1
9 ($C_{3,4}$)	0	1	1	0	0	1	0	1
10 ($C_{4,1}$)	0	0	0	1	1	0	0	0
11 ($C_{4,2}$)	1	0	0	1	0	1	0	0
12 ($C_{4,3}$)	0	1	0	0	1	0	1	1
13 ($C_{4,4}$)	1	1	0	0	1	1	1	1

3. INFORMATION MINING PROCESS

A simple example is used to illustrate the information mining process. Assume there are four participating databases, each database has a set of *media objects* and each media object has a set of *attributes/features*.

Example 1: The media objects of four databases and part of the attributes/features for the databases are shown as follows.

$d_1 = \{center, department\} = \{C_{1,1}, C_{1,2}\}$;

$C_{1,1} \Leftrightarrow \{name, location, president, dname\}$

$d_2 = \{dept, emp, project\} = \{C_{2,1}, C_{2,2}, C_{2,3}\}$;

$d_3 = \{employee, secretary, engineer, manager\}$
$= \{C_{3,1}, C_{3,2}, C_{3,3}, C_{3,4}\}$;

$d_4 = \{InletValve, NeedleSeat, InletNeedle,$
$Manufacturer\} = \{C_{4,1}, C_{4,2}, C_{4,3}, C_{4,4}\}$.

A set of historical data is used to generate the training traces which are the central part of the information mining process. Assume there are eight sample queries with the access frequencies: 25, 100, 30, 70, 45, 35, 40, and 60. Table 1 shows the usage patterns of media objects versus a set of sample queries.

3.1. Formulation of the Model Parameters

There are three probability distributions for each MMM – $\mathcal{A}, \mathcal{B}$, and Π where $\mathcal{A}$ is the state transition probability distribution, $\mathcal{B}$ is the observation symbol probability distribution, and Π is the initial state probability distribution.

- *State Transition Probability Distribution*

We use the relative affinity measurements to indicate how frequently two media objects are accessed together. When two databases whose media objects are accessed together more frequently, they are said to have a higher relative affinity relationship. Accordingly, in terms of the state transition probability in a Markov Chain, if two databases have a higher relative affinity relationship, the probability that a traversal choice to node j given the current node is in i (or vice versa) should be higher. Realistically, the applications cannot be expected to specify these affinity values. Therefore, formulas to calculate these relative affinity values need to be defined.

Let Q = $\{q_1, q_2, \ldots, q_q\}$ be the set of sample queries that ran on the databases $d_1, d_2, \ldots, d_d$ with media object set OC = {1, 2, ..., g} in the distributed information-providing environment. Define the variables:

n_i = number of media objects in database d_i

$use_{m,k}$ = usage pattern of media object m with respect to query q_k per time period (available from the historical data)

$$use_{m,k} \begin{cases} 1 & \text{if media object } m \text{ is accessed by } q_k \\ 0 & \text{otherwise} \end{cases}$$

$access_k$ = access frequency of query q_k per time period (available from the historical data)

$aff_{m,n}$ = affinity measure of media objects m and n

$f_{m,n}$ = the joint probability which refers to the fraction of the relative affinity of media objects m and n in a database (or a warehouse) with respect to the total relative affinity for all the media objects in a database (or a warehouse)

f_m = the marginal probability
$a_{m,n}$ = the conditional probability which refers to the state transition probability for an MMM

$$aff_{m,n} = \sum_{k=1}^{q} use_{m,k} \times use_{n,k} \times access_k \quad (1)$$

$$f_{m,n} = \frac{aff_{m,n}}{\sum_{m \in d_i} \sum_{n \in d_i} aff_{m,n}} \quad (2)$$

$$f_m = \sum_n f_{m,n} \quad (3)$$

$$a_{m,n} = \frac{f_{m,n}}{f_m} \quad (4)$$

We denote $\mathcal{A}$ the state transition probability distribution whose elements are $a_{m,n}$. The $access_k$ and $use_{m,k}$ values required for calculating $f_{m,n}$ are assumed to be available from the historical data.

Example 2: Tables 2 to 5 give the calculated state transition probability distributions for d_1 to d_4.

Table 2: The state transition probability distribution $\mathcal{A}$ for d_1. For example, the transition goes from state 1 (media object $C_{1,1}$) to state 2 (media object $C_{1,2}$) is 0.3820.

state	1	2
1	0.6180	0.3820
2	0.4474	0.5526

Table 3: The state transition probability distribution $\mathcal{A}$ for d_2. For example, the transition goes from state 1 (media object $C_{2,1}$) to state 2 (media object $C_{2,2}$) is 0.3902.

state	1	2	3
1	0.3902	0.3902	0.2195
2	0.2540	0.4286	0.3175
3	0.1837	0.4082	0.4082

Table 4: The state transition probability distribution $\mathcal{A}$ for d_3. For example, the transition goes from state 1 (media object $C_{3,1}$) to state 2 (media object $C_{3,2}$) is 0.2439.

state	1	2	3	4
1	0.3415	0.2439	0.2134	0.2012
2	0.2174	0.3261	0.2500	0.2065
3	0.2011	0.2644	0.3161	0.2184
4	0.2143	0.2468	0.2468	0.2922

- *Observation Symbol Probability Distribution*

The observation symbol probability denotes the probability of observing an output symbol from a state.

Table 5: The state transition probability distribution $\mathcal{A}$ for d_4. For example, the transition goes from state 1 (media object $C_{4,1}$) to state 2 (media object $C_{4,2}$) is 0.2545.

state	1	2	3	4
1	0.4182	0.2545	0.1636	0.1636
2	0.2692	0.5000	0	0.2308
3	0.0841	0	0.4579	0.4579
4	0.0687	0.0916	0.3740	0.4656

Here, the observed output symbols represent the attributes and the states represent the media objects. Since a media object has one or more attributes and an attribute can appear in multiple media objects, the observation symbol probabilities shows the probabilities an attribute is observed from a set of media objects.

A temporary matrix BB whose rows are all the distinct media objects and columns are all the distinct attributes in the environment is defined as follows.

$$BB_{p,q} = \begin{cases} 1 & \text{if attribute } p \text{ appears in media object } q \\ 0 & \text{otherwise} \end{cases}$$

Each entity of BB is assigned a value 1 or 0 to indicate whether an attribute appears in a media object of the database. After BB is constructed, the observation symbol probability distribution $\mathcal{B}$ can be obtained via normalizing BB per column. In other words, the sum of the probabilities which the attributes are observed from a given media object should be 1.

Example 3: Similarly, Tables 6 to 9 are the observation symbol probability distributions for d_1 to d_4, respectively.

- *Initial State Probability Distribution*

Since the information from the training traces is available, the preference of the initial states for queries can be obtained. For any media object $m \in d_i$ (the ith database), the initial state probability is defined as the fraction of the number of occurrences of media object m with respect to the total number of occurrences for all the member media objects in d_i from the training traces.

$$\Pi_i = \{\pi_m\} = \frac{\sum_{k=1}^{q} use_{m,k}}{\sum_{l=1}^{n_i} \sum_{k=1}^{q} use_{l,k}} \quad (5)$$

Example 4: In this example, using Equation 5, the four initial state probability distributions for d_1 to d_4 can be determined.

Π_1 = [5/9 4/9] for database d_1
Π_2 = [4/14 6/14 4/14] for database d_2
Π_3 = [5/19 5/19 5/19 4/19] for database d_3
Π_4 = [2/15 3/15 4/15 6/15] for database d_4

Table 6: $\mathcal{B}$ for d_1.

	1	2
1	1/4	0
2	1/4	0
3	1/4	0
4	1/4	1/4
5	0	1/4
6	0	1/4
7	0	1/4
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0

Table 7: $\mathcal{B}$ for d_2.

	1	2	3
1	1/3	0	0
2	1/3	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	1/3	0	0
9	0	1/5	0
10	0	1/5	0
11	0	1/5	0
12	0	1/5	0
13	0	1/5	1/4
14	0	0	1/4
15	0	0	1/4
16	0	0	1/4
17	0	0	0
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0
25	0	0	0
26	0	0	0
27	0	0	0
28	0	0	0
29	0	0	0
30	0	0	0

Table 8: $\mathcal{B}$ for d_3.

	1	2	3	4
1	1/4	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	1/4	1/2	1/2	1/2
18	1/4	0	0	0
19	1/4	0	0	0
20	0	1/2	0	0
21	0	0	1/2	0
22	0	0	0	1/2
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0
26	0	0	0	0
27	0	0	0	0
28	0	0	0	0
29	0	0	0	0
30	0	0	0	0

Table 9: $\mathcal{B}$ for d_4.

	1	2	3	4
1	0	0	0	0
2	0	0	0	1/2
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	1/3	0	0	0
24	1/3	1/4	0	0
25	1/3	0	1/5	0
26	0	1/4	0	0
27	0	1/4	1/5	0
28	0	1/4	1/5	1/2
29	0	0	1/5	0
30	0	0	1/5	0

3.2. Stochastic Clustering Strategy

The proposed stochastic clustering strategy uses a similarity measure between two databases to measure how well these two databases together match the observations generated by the sample queries. The similarity measure is formulated under the assumptions that the observation set O^k is conditionally independent given X and Y, and the sets $X \in d_i$ and $Y \in d_j$ are conditionally independent given d_i and d_j. Let $N_k = k1+k2$. The similarity measure is defined as follows.

$$S(d_i, d_j)$$

$$= \Big(\sum_{O_k \in \mathcal{OS}} P(O^k \mid X, Y; d_i, d_j) P(X, Y; d_i, d_j) \Big) F(N_k) \quad (6)$$

$$P(O^k \mid X, Y; d_i, d_j)$$

$$= P(o_1, \ldots, o_{k1} \mid X; d_i) P(o_{k1+1}, \ldots, o_{N_k} \mid Y; d_j) \quad (7)$$

$$P(X, Y; d_i, d_j) = P(X; d_i) P(Y; d_j) \quad (8)$$

$$F(X, Y) = 10^{N_k} \quad (9)$$

$$P(X; d_i) = P(x_1, \ldots, x_{k1}; d_i) = \prod_{u=2}^{k1} \underbrace{P(x_u \mid x_{u-1})}_{A_i(x_u \mid x_{u-1})} \underbrace{P(x_1)}_{\Pi_i(x_1)}$$

$$P(Y; d_j) = P(y_1, \ldots, y_{k2}; d_j)$$

$$= \prod_{v=k1+2}^{N^k} \underbrace{P(y_{v-k1} \mid y_{v-k1-1})}_{A_j(y_{v-k1} \mid y_{v-k1-1})} \underbrace{P(y_1)}_{\Pi_j(y_1)}$$

$$P(o_1, \ldots, o_{k1} \mid X; d_i) = P(o_1, \ldots, o_{k1} \mid x_1, \ldots, x_{k1}; d_i)$$

$$= \prod_{u=1}^{k1} \underbrace{P(o_u \mid x_u)}_{B_i(o_u \mid x_u)}$$

$$P(o_{k1+1}, \ldots, o_{N_k} \mid Y; d_j)$$

$$= P(o_{k1+1}, \ldots, o_{N^k} \mid y_1, \ldots, y_{k2}; d_j)$$

$$= \prod_{v=k1+1}^{N^k} \underbrace{P(o_v \mid y_{v-k1})}_{B_j(o_v \mid y_{v-k1})}$$

- $S(d_i, d_j)$ = similarity measure between databases d_i and d_j
- $\mathcal{OS}$ = set of all observation sets
- $O^k = \{o_1, \ldots, o_{N_k}\}$ is an observation set with the attributes belonging to d_i and d_j and generated by query q_k
- $X = \{x_1, \ldots, x_{k1}\}$ is a set of media objects belonging to d_i in O^k
- $Y = \{y_1, \ldots, y_{k2}\}$ is a set of media objects belonging to d_j in O^k
- $P(O^k \mid X, Y; d_i, d_j)$ = the probability of occurrence of O^k given $X \in d_i$ and $Y \in d_j$
- $F(N_k)$ = an adjusting factor which is used because the lengths of the observation sets are variable

The similarity values are computed for all pairs of databases and are transformed into the branch probabilities among the nodes (databases) in the browsing graph. Then the stationary probability ϕ_i for each node i of the browsing graph can be obtained from the branch probabilities. The

stationary probability ϕ_i denotes the relative frequency of accessing node i (the ith database) in the long run.

$$\sum_i \phi_i = 1 \qquad \phi_j = \sum_i \phi_i P_{i,j} \quad j = 1, 2, \cdots \qquad (10)$$

Our stochastic clustering strategy is traversal based and greedy. Databases are partitioned with the order of their stationary probabilities. The database which has the largest stationary probability is selected to start a data warehouse. While there is room in the current warehouse, all databases accessible in terms of the browsing graph from the current member databases of the warehouse are considered. The database which has the largest stationary probability is selected and the process continues until the warehouse fills up. At this point, the next un-partitioned database from the sorted list starts a new data warehouse, and the whole process is repeated until no un-partitioned databases remain. The time complexity for our stochastic clustering strategy is $O(d \log d)$, where d is the number of databases.

4. EXPERIMENTAL RESULTS

In the experiments, we generated the training traces from part of the historical data of the financial database management systems at Purdue University for the year 1997. The information in the training traces were applied to the information mining process to construct the federation of data warehouses.

4.1. Experimental Parameters

We compare our MMM mechanism with the Breadth First Search (BFS) algorithm, Depth First Search (DFS) algorithm, Maximum-Cost Spanning Tree (MCST) algorithm, and the random clustering method. In BFS, DFS, and MCST algorithms, the weights of the browsing graph are used to decide the traversal sequence according to the breadth first, depth first, and maximum-cost spanning tree algorithms, respectively to get a sequence of numbers representing the databases. The node (database) with a larger weight is traversed with a higher priority. The weights of the browsing graph can be obtained from the data mining process. In the random clustering method, we used random number generators to produce a sequence of numbers. For all the methods, we partitioned the sequence of databases according to the default data warehouse size three.

The performance metrics we used is the number of inter-warehouse accesses with respect to queries. Our objective is to minimize the cost of query processing or equivalently, minimize the query response time. Query processing, in general, involves the closely interrelated *communication cost* and *processing cost*. Essentially, since related databases belonging to a certain application domain is put in the same data warehouse and is required consecutively on some query access path, the number of platter switches (communication cost) and the information searching time (processing cost) for data retrieval with respect to queries can be reduced. The communication cost and the processing cost depend partially on the overhead and time required for information retrieval. Conceptually, communication overhead occurs when the warehouse which a query originates needs

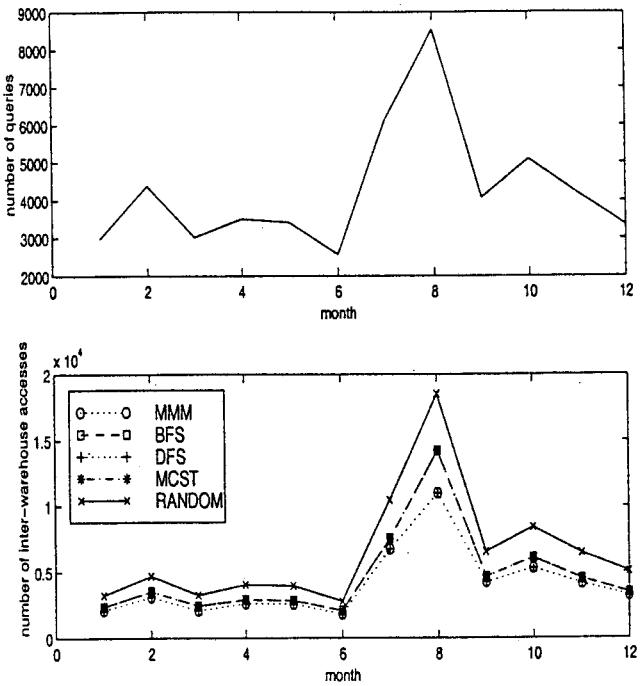

Figure 1: The upper figure lists the number of queries issued to the databases for the year 1997. The performance for the MMM, BFS, DFS, MCST, and the random clustering approaches is shown in the lower figure.

to access information from a database in a different warehouse. Since the information in the same warehouse can be accessed in one time, there is no need to consider the searching time within the same warehouse. The searching time that needs to be taken into account is the time required to retrieve information across multiple warehouses. Both of the communication overhead and the searching time across warehouses can be represented by the number of inter-warehouse accesses. Therefore, the number of inter-warehouse accesses is used to compare the performance.

4.2. Results

We measured the number of inter-warehouse accesses by traversing the databases and counting the number of inter-warehouse accesses at the end of the traversal query by query for all the queries. For each query, the data warehouse which the first database belongs to is regarded as the originated warehouse. Any access to the database in the warehouse different from the originated warehouse is then considered as an inter-warehouse access.

Figure 1 lists the number of queries issued in the year 1997 to the databases at Purdue University (the upper figure) and shows the number of inter-warehouse accesses for the MMM, BFS, DFS, MCST, and the random clustering approaches (the lower figure). The results tell us that the set of data warehouses constructed from our MMM mechanism and the proposed stochastic clustering strategy gives

the best performance among all the approaches excluding DFS. DFS has the same performance as the MMM mechanism since the number of participating databases in our experiments is not large so that two approaches could yield to the same sequence of databases. Another finding is that the savings of the number of inter-warehouse accesses increases as the number of queries increases. The reason for this trend is that the structurally equivalent relationships are captured within the same data warehouse. It also can be seen the random clustering method is the most inferior approach as compared to the rest of the approaches.

5. CONCLUSIONS

In this paper, we introduced a mathematically sound framework, Markov model mediators (MMMs), to facilitate the data mining process for database clustering. A stochastic clustering strategy based on the MMM mechanism is proposed to partition the databases into a federation of data warehouses. Since a warehouse consists of several related databases which are usually required for queries in the same application domain, the cost of query processing can be reduced.

Several experiments were performed to compare the performance of our MMM mechanism with the BFS, DFS, MCST, and the random clustering approaches. From the experimental results, we observe that the set of data warehouses constructed from the data mining process gives the fewest number of inter-warehouse accesses. Moreover, the results suggest that the MMM mechanism, when used in a large-scaled heterogeneous database environment, can be applied as the preceding process of the schema integration tasks.

6. REFERENCES

[1] V. Benzaken and C. Delobel, "Enhancing performance in a persistent object store: Clustering strategies in O_2," in A. Dearle, G.M. Shaw, and S.B. Zdonik, editors, Implementing Persistent Object Bases: Principles and Practice, pp. 403-412, Morgan Kaufmann, 1991.

[2] M.J. Carey, D.J. DeWitt, J.E. Richardson, and E.J. Shekita, "Object and file management in the EXODUS extensible database system," Proc. 12th Int'l Conf. on Very Large Data Bases, pp. 91-100, August 1986.

[3] S. Chaudhuri and U. Dayal, "An overview of data warehousing and OLAP technology," SIGMOD Record, pp. 65-74, Vol. 26, No. 1, March 1997.

[4] P. Cheeseman and J. Stutz, "Bayesian classification (AutoClass): theory and results," in U.M. Fayyad, G. Piatesky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pp. 153-180, AAAI/MIT Press, 1996.

[5] M.S. Chen, J. Han, and P.S. Yu, "Data mining: An overview from a database perspective," IEEE Transactions on Knowledge and Data Engineering, pp. 866-883, Vol. 8, No. 6, December 1996.

[6] Shu-Ching Chen and R.L. Kashyap, "Temporal and spatial semantic models for multimedia presentations," 1997 International Symposium on Multimedia Information Processing, pp. 441-446, Dec. 11-13, 1997.

[7] M. Ester, H.P. Kriegel, and X. Xu, "Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification," Proc. Fourth Int'l Symp. Large Spatial Databases (SSD'95), pp. 67-82, August 1995.

[8] K.A. Hua, S.D. Lang, and W.K. Lee, "A decomposition-based simulated annealing technique for data clustering," Proc. of the Thirteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), 1994.

[9] W.H. Inmon, Building the data warehouse. John Wiley, 1992.

[10] S.J. Kim, J. Baberjee, W. Kim, and J.F. Garza, "Clustering a DAG for CAD databases," IEEE Transactions on Software Engineering, 14(11), pp. 1684-1699, November 1988.

[11] H. Lu, R. Setiono, and H. Liu, "NeuroRule: A connectionist approach to data mining," Proc. 21th Int'l Conf. Very Large Data Bases, pp. 478-489, September 1995.

[12] K. Shannon and R. Snodgrass, "Semantic clustering," in A. Dearle, G.M. Shaw, and S.B. Zdonik, editors, Implementing Persistent Object Bases: Principles and Practice, pp. 389-402, Morgan Kaufmann, 1991.

[13] Mei-Ling Shyu, Shu-Ching Chen and R. L. Kashyap, "Information retrieval using Markov model mediators in multimedia database systems," to appear in 1998 International Symposium on Multimedia Information Processing, Dec. 14-16, 1998, Taiwan.

[14] R. Srikant and R. Agrawal, "Mining generalized association rules," Proc. 21th Int'l Conf. Very Large Data Bases, pp. 407-419, September 1995.

[15] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," Proc. 1996 ACM SIGMOD Int'l Conf. Management Data, pp. 1-12, June 1996. September 1995.

[16] M.M. Tsangaris and J.F. Naughton, "A stochastic approach for clustering in object bases," Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp. 12-21, May 1991.

[17] M.M. Tsangaris and J.F. Naughton, "On the performance of object clustering techniques," Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp. 144-153, June 1992.

[18] J. Widom, "Research problems in data warehousing," Proc. Fourth Int'l Conf. Information and Knowledge Management, pp. 25-30, November 1995.

[19] G. Wiederhold, "Mediators in the architecture of future information systems," IEEE Computer, pp. 38-49, March 1992.

[20] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," Proc. 1996 ACM SIGMOD Int'l Conf. Management Data, June 1996.