INCORPORATION OF DATA MINING AND DOMAIN KNOWLEDGE FOR DISCOVERING QUASI-EQUIVALENCE DATABASE RELATIONSHIPS

Mei-Ling Shyu

Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124
U.S.A.
shyu@miami.edu

Shu-Ching Chen

School of Computer Science
Florida International University
Miami, FL 33199, U.S.A.
chens@cs.fiu.edu

Sheng-Tun Li

Department of Information Management
National Kaohsiung First University of Science and Technology
1 University Road, Yenchao,
Kaohsiung, Taiwan, R.O.C.
stli@ccms.nkfust.edu.tw

ABSTRACT

In this paper, we propose to incorporate domain knowledge in a generalized affinity-based association rule mining algorithm to reduce the size of the data so that only the potentially interesting and relevant portion of the data will be used in the computation procedure. The association rule mining algorithm is an approach to discover the quasi-equivalence relationships of the media objects for databases. After the incorporation of domain knowledge, the computational performance of the mining algorithm can be improved. Experimental results are provided and analyzed.

1. Introduction

The advanced data storage technology and database management systems have increased our capabilities to collect and store data of all kinds. It is very common for a database to have hundreds of fields and tables, millions of records, and multi-gigabyte size [9]. In fact, we can see lots of terabyte databases are becoming to appear recently [1, 3, 5, 7, 9]. Though the inexpensive multi-gigabyte disks and other storage devices allow us to keep all the data, our ability to interpret and analyze the data is still limited. Therefore, *knowledge discovery in databases (KDD)* or *data mining* becomes the only hope for us to elucidate the patterns from the data. As pointed out by [11], there is a need and an opportunity for at least a partially-automated form of KDD or data mining to handle the huge size of real-world database systems.

Data mining or knowledge discovery is defined as a process of extracting implicit, previously unknown, and potentially useful information from data [3, 4, 5, 7, 14]. The objective of this process is to sort through large quantities of data and discover new information [6]. Knowledge discovery process relies on databases to supply the raw data for input. However, when the knowledge discovery process has to deal with large databases, the high volume of data makes the discovery process computationally expensive since data in large databases may contain billions of patterns. In most cases, exhaustive analysis of all the data is infeasible because of the high computational complexity and poor performance. It is desirable to perform the knowledge discovery process on a relatively constrained subset of data to reduce the computational complexity. Thus, how to provide methods that reduce the size of the data to limit the search for patterns becomes important.

Toward this ends, *domain knowledge* can be utilized to reduce the size of the data being considered [9, 15]. It is a common fact that the human user has some previous concepts or knowledge about the domain represented by the database. This kind of information, known as the domain knowledge, can be defined as any information that is not explicitly presented in the data [1, 5, 7]. In our previous study, we explored a *generalized affinity-based association mining algorithm* that discovers quasi-equivalent media objects among databases, and demonstrated that the proposed association mining algorithm discovers the set of quasi-equivalent media objects correctly [13]. In this paper, we propose to incorporate domain knowledge in the generalized affinity-based association mining algorithm to reduce the size of the data in the computation procedure. We suggest some strategies to control the size of the data in the mining process by using domain knowl-

edge, and show that domain knowledge can be used in association with the algorithm to reduce the size of data by eliminating the irrelevant data. An experiment is conducted to compare the numbers of computational operations required in the mining algorithm with and without the inclusion of domain knowledge. The result shows that the incorporation of domain knowledge can effectively reduce the size of the data being processed and furthermore, significantly improve the computational performance. In addition, more domain knowledge rules can be derived from the discovered information and become domain knowledge themselves, which demonstrates the power of using domain knowledge in the knowledge discovery procedure.

This paper is organized as follows. In next section, we briefly give an overview of the generalized affinity-based association mining algorithm without domain knowledge. In Section 3, how to incorporate domain knowledge in the generalized affinity-based association mining algorithm is introduced. The evaluation of using domain knowledge in the mining algorithm by conducting an experiment is presented in Section 4. The comparison of the computational performance of the mining algorithm with and without the inclusion of domain knowledge is also discussed. Section 5 concludes the paper.

2. Overview of the Proposed Mining Algorithm

2.1. Affinity-Based Association Rules

An association rule is defined as an expression $X \rightarrow Y$ for a given set of transactions, where X and Y are sets of items with $X \cap Y = \emptyset$, and each transaction contains a set of items [2]. The *support*, *confidence* and *interest* are three measures for the association rules. Their original definitions are given as follows.

$$support(X) = P(X) = \frac{|X|}{N}$$

$$support(X \rightarrow Y) = P(X \cup Y) = \frac{|X \cup Y|}{N}$$

$$confidence(X \rightarrow Y) = \frac{P(X \cup Y)}{P(X)} = \frac{|X \cup Y|}{|X|}$$

$$interest(X \rightarrow Y) = \frac{P(X \cup Y)}{P(X)P(Y)}$$

where $P(X \cup Y)$ is the probability that all items in $X \cup Y$ are present in the transaction, N is the total number of tuples, and $\mid A \mid$ is the number of tuples containing all items in the set A.

To allow us to apply and extend the functionality of the association rules, we have defined the affinity-based association rules by considering the set of queries as the set of transactions [12]. Accordingly, a set of affinity-based *support*, *confidence* and *interest* measures is defined. Then, the generalized affinity-based association mining algorithm uses these affinity-based values [13].

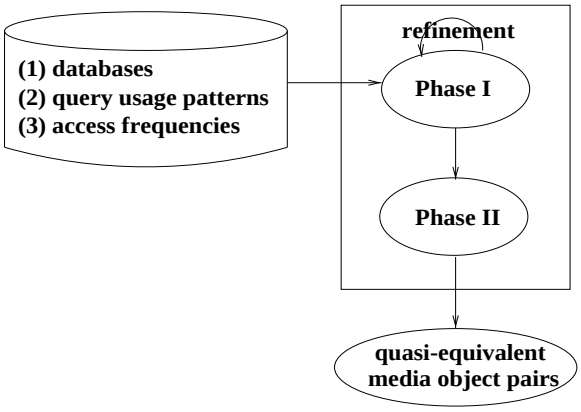

Figure 1: Architecture for the Generalized Affinity-Based Association Mining Without Domain Knowledge.

$$use_{m,k} = \begin{cases} 1 & \text{if media object } m \text{ is} \\ & \text{accessed by query k} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$aff_{m,n} = \sum_{k=1}^{q} use_{m,k} \times use_{n,k} \times access_k \quad (2)$$

$$support(m) = \frac{\sum_{k=1}^{q} use_{m,k} \times access_k}{\sum_{k=1}^{q} access_k} \quad (3)$$

$$support(m \rightarrow n) = \frac{aff_{m,n}}{\sum_{k=1}^{q} access_k} \quad (4)$$

$$confidence(m \rightarrow n) = \frac{support(m \rightarrow n)}{support(m)} \quad (5)$$

$$interest(m \rightarrow n) = \frac{support(m \rightarrow n)}{support(m)support(n)} \quad (6)$$

where m and n are media objects, q is the total number of queries, $access_k$ is the access frequency of query k per time period, $use_{m,k}$ is the usage pattern, and $aff_{m,n}$ is the relative affinity measure between m and n.

2.2. The Generalized Affinity-Based Association Mining Algorithm Without Incorporating Domain Knowledge

Figure 1 shows the architecture of the original generalized affinity-based association mining algorithm, i.e., without the incorporation of domain knowledge. The

databases, query usage patterns, and query access frequencies are the inputs for the knowledge discovery process (the association mining process). The association mining process consists of two phases. Phase I starts with the minimal interest threshold and the interestingness constraints, and is executed iteratively based on the refinement constraint. Phase II first checks the confidence threshold constraint, and then checks whether any further conditions need to be imposed to remove some unreasonable situations. The output (i.e., the discovered knowledge) is a set of quasi-equivalent media object pairs. The original generalized affinity-based association mining algorithm is briefly listed as follows [13].

⋆ **Steps** for Phase I:

1. For all the 1-itemsets, compute $support(m)$ (Equation 3).
2. For all the 2-itemsets,
   - Compute $aff_{m,n}$ (Equation 2).
   - Compute $support(m \rightarrow n)$ (Equation 4).
   - Compute $confidence(m \rightarrow n)$ (Equation 5).
   - Compute $interest(m \rightarrow n)$ (Equation 6).
3. Initialize candidate-pool $= \emptyset, cria1 = 20\%$, $cria2 = 50\%$, and $iter = 1$.
4. For $m = 1$ to g (where g is the total number of media objects),
   (a) If $iter = 1$ then find the maximal interest value I_m from $interest(m \rightarrow n)$ where a media object n is in a different database since the equivalence relationship can occur only when two media objects are from different databases.
   (b) Set the minimal interest threshold $IntTd = cria1 \times iter \times I_m$.
   (c) For those media objects n's, if $iter = 1$ then candidate-pool = candidate-pool $\bigcup \{(m, n)\}$ when $interest(m \rightarrow n) \geq IntTd$ else (m, n) is removed from candidate-pool when $interest(m \rightarrow n) < IntTd$.
5. Check the interestingness constraint: if $(m, n) \in$ candidate-pool and $(n, m) \notin$ candidate-pool, then (m, n) is removed from candidate-pool.
6. Check the refinement constraint: if the number of media object m (which has zero or one (m, n) in candidate-pool) $\geq cria2 \times g$, then goto Phase II else set $iter = iter + 1$ and goto step 4.

⋆ **Steps** for Phase II:

1. Set the confidence threshold $Conf = 99\%$.
2. For each pair (m, n) in candidate-pool, if $confidence(m \rightarrow n) < Conf$ and $confidence(n \rightarrow m) < Conf$, then (m, n) is removed from candidate-pool.
3. Check if further conditions need to be imposed to remove some unreasonable situations.

3. Incorporating Domain Knowledge into the Generalized Affinity-Based Association Mining Algorithm

3.1. Domain Knowledge

Although a database stores a large amount of data, in most cases, only a subset of data is relevant in the knowledge discovery process. As the volume of data increases, it is not realistic to include all the data in the database in the discovery task. Hence, it is often necessary to find the relevant portion of data to reduce the size of data so as to improve the efficiency of the knowledge discovery process. For this purpose, domain knowledge can be utilized since it can be used to reduce the size of data that is being considered for discovery by eliminating the irrelevant data in the discovery task.

Domain knowledge may originate from many sources including specifications and domain experts. It is often possible for the domain experts to provide valuable information that is significant in the discovery process. Also, newly discovered information can be added to the set of domain knowledge and used in the future as domain knowledge [15]. In this study, since our purpose is to discover the set of quasi-equivalent media object pairs, the domain knowledge used in the discovery task is the fact that the media object equivalent relationship cannot exist in a single database. It is known that a database schema represents a non-redundant view and therefore, only media objects across different databases can have an equivalence relationship. Two media objects are said to be equivalent if they are deemed to possess the same real world states *(RWS's)* [8, 10], i.e., if they represent the same sets of instances of the same real world entity. This domain knowledge is derived from the schema and meta-data in a database. With the incorporation of this domain knowledge, a huge amount of unnecessary computations can be avoided in a large database since only a subset of data needs to be included in the discovery process. How to incorporate

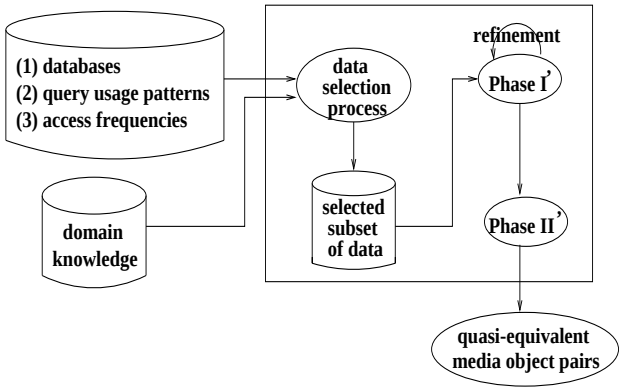

Figure 2: Architecture for Generalized Affinity-Based Association Mining With Domain Knowledge.

domain knowledge in the generalized affinity-based association mining algorithm will be discussed in details in the next subsection.

3.2. The Generalized Affinity-Based Association Mining Algorithm With Domain Knowledge

Figure 2 shows the architecture of the generalized affinity-based association mining algorithm when domain knowledge is incorporated. In comparison with Figure 1, it can be easily seen that in addition to the original inputs (databases, query usage patterns, and query access frequencies), domain knowledge is another input to the knowledge discovery process in Figure 2. Also, before going to the two phases of the association mining process, a data selection process is first executed. Therefore, only the proper subset of data needs to be included in the computation task in Phase $I^{'}$. Again, Phase $I^{'}$ is iterative based on the refinement constraint. Then, Phase $II^{'}$ is conducted to generate the output – the set of quasi-equivalent media object pairs. The idea is to use domain knowledge to eliminate the unnecessary computation efforts in Phase $I^{'}$ by reducing the size of the data in the computation tasks.

3.2.1. Data selection process

This subsection discusses how to use domain knowledge to select a proper subset of data for the mining algorithm in the data selection process. Consider the quasi-equivalence relationship that we are interested in discovering, we can reduce the size of data involved in the computation by using the domain knowledge, i.e., the equivalence relationship occurs only for two media objects from different databases.

Table 1: Data Selection Process

INPUT:
- Let $\mathcal{D} = \{d_1, d_2, \ldots, d_p\}$ be the set of databases involving in the knowledge discovery (association mining) process, where p is the total number of databases.
- Let $\mathcal{OC} = \{1, 2, \ldots, g\}$ be the set of media objects in $\mathcal{D}$, where g is the total number of media objects.
- Let $\mathcal{Q} = \{1, 2, \ldots, q\}$ be the set of sample queries that run on $\mathcal{D}$, where q is the total number of queries.
- Let $\mathcal{DK} = \{DK_1, DK_2, \ldots, DK_d\}$ be the set of domain knowledge rules (defined or derived), where d is the total number of domain knowledge rules.

OUTPUT:
- $\mathcal{S}$ = the set of media object pairs that satisfy one of the domain knowledge rules in $\mathcal{DK}$.

SELECTION PROCESS :

```
S = ∅;
DK_1 = {Two media objects can be
        quasi-equivalent only when they are
        in different databases.};
for m = 1 to g-1 {
    for n = m+1 to g {
        if ((m,n) satisfies DK_1) {
            S = S ∪ {m,n};
        }
    }
}
```

Table 1 details the data selection process. The inputs for the data selection process include the set of databases along with their media objects, the set of sample queries with their usage patterns and access frequencies, and the set of domain knowledge. Initially, only one domain knowledge rule is defined and used. The output is the set of selected media object pairs that satisfy the defined domain knowledge rule. In this process, the media object pair (m, n) is selected when $m \in d_i$, $n \in d_j$, and $i < j$. The reason for considering $i < j$ is that if the media object pair (m, n) is equivalent, then the media object pair (n, m) is also equivalent. Under this constraint, the number of the selected media object pairs in $\mathcal{S}$ can be further reduced.

3.2.2. Algorithm

Most of the steps in the modified generalized affinity-based association mining algorithm (i.e., with domain knowledge) are the same as the steps in the original algorithm shown in Section 2.2. For Phase $I^{'}$, only step

2 is different since only the media object pairs in the selected data set $\mathcal{S}$ are required in the computation. Therefore, the number of media object pairs involved in the computation is reduced. As for Phase $II^{'}$, steps 1 and 2 are the same. Step 3 is different since there is no need to manually check the unreasonable situations for the candidate-pool. Based on the information obtained so far, more domain knowledge rules can be derived and put into the domain knowledge set $\mathcal{DK}$. Then, unreasonable media object pairs are removed according to the derived domain knwoledge rules.

$\star$ For Phase $I^{'}$:

step 1: Same as before.
step 2: For each (m, n) pair in $\mathcal{S}$,

- Compute $aff_{m,n}$ and $aff_{n,m} = aff_{m,n}$.
- Compute $support(m \rightarrow n)$ and $support(n \rightarrow m) = support(m \rightarrow n)$.
- Compute $interest(m \rightarrow n)$ and $interest(n \rightarrow m) = interest(m \rightarrow n)$.
- Compute both $confidence(m \rightarrow n)$ and $confidence(n \rightarrow m)$.

steps 3 to 6: Same as before.

$\star$ For Phase $II^{'}$:

steps 1 to 2: Same as before.
step 3: Derive more domain knowledge rules from the information obtained so far. Then, remove the media object pairs that satisfy the derived domain knowledge rules. Here, two domain knowledge rules can be derived.

- DK_2 = {If a media object has quasi-equivalence relationships with two or more media objects in the same database, then these media object pairs are removed from the candidate-pool.}
- DK_3 = {If a media object has quasi-equivalence relationship with one media object that is quasi-equivalent to any media object being removed using DK_2, then this media object pair is removed from the candidate-pool. }

4. Evaluation of Using Domain Knowledge

In this section, we evaluate the benefits of utilizing domain knowledge in the knowledge discovery process (i.e., the association mining algorithm) by comparing the computational performance of step 2 in the first phase with and without the incorporation of domain knowledge. We expect that the incorporation of domain knowledge can significantly reduce the number of computational operations involved in the modified mining algorithm.

4.1. Performance Analysis

The performance analysis considers the number of operations required in step 2 of the first phase in the mining algorithm. In the experiment, the performance metric used is the number of operations required in the computation. In the original mining algorithm, the computations of $aff_{m,n}$, $support(m \rightarrow n)$, $interest(m \rightarrow n)$, and $confidence(m \rightarrow n)$ are required for every media object pair in the databases. However, in the modified mining algorithm, only the media object pairs in the selected data set $\mathcal{S}$ need to do the computations.

In order to simplify the comparison procedure, it is assumed that the number of media objects is the same in each database. That is, we assume that each database has $\frac{g}{p}$ media objects and ignore whether $\frac{g}{p}$ is an integer value. Here, the same variable notations given in Table 1 are used, where p, g, and q represent the number of databases, media objects, and queries, respectively. The number of media object pairs involved in the computation in the modified mining algorithm is as follows.

- For database $d_1 \Rightarrow \frac{g}{p} \times (g - 1 \times \frac{g}{p})$.
- For database $d_2 \Rightarrow \frac{g}{p} \times (g - 2 \times \frac{g}{p})$.
- ...
- For database $d_{p-1} \Rightarrow \frac{g}{p} \times (g - (p-1) \times \frac{g}{p})$.

Hence, the total number of media object pairs considered in the computation is:

$$\begin{aligned} & \tfrac{g}{p} \times (\textstyle\sum_{i=1}^{p-1}(g - (i \times \tfrac{g}{p}))) \\ = & \tfrac{g^2}{p} \times (\textstyle\sum_{i=1}^{p-1}(1 - \tfrac{i}{p})) \\ = & g^2 \times \tfrac{(p-1)}{2 \times p}. \end{aligned}$$

If we compare the computational performance using the same financial database management systems used in the empirical study in [13], the computational performance improvement can be easily seen. The financial database management systems are real database management systems and the data were collected in the year 1997. In the financial database management systems, there are 5 databases that represent 22 media

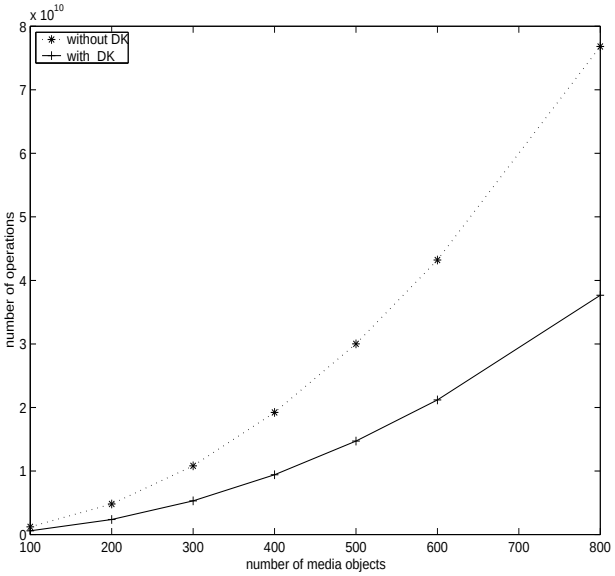

Figure 3: Comparison of the number of operations with domain knowledge (with DK) and without domain knowledge (without DK) when the number of media objects (with fixed numbers of databases and queries) varies.

objects accessed by 17222 queries. That is, the values of p, q, and g are 5, 17222 and 22, respectively. Then, the total number of operations reduced is more than *20,005,000* which is a significant saving. To make the performance analysis more general, we conduct an experiment by varying the number of media objects to compare the number of operations in step 2 of the first phase in the original and modified algorithms in the next subsection.

4.2. Experiment

We conduct an experiment for the computational performance based on the number of media objects (g). The number of operations is used as the performance metric in the experiment. The number of operations is used to compare the original and modified mining algorithms. The fewer the number of operations is, the better the performance is.

Figure 3 shows the comparison of the number of operations required for the original algorithm (i.e., without domain knowledge) and the modified mining algorithm (i.e., with domain knowledge) under fixed numbers of databases and queries, and varying number of media objects. From this figure, we observe that the number of operations reduces significantly in the experiment. The larger the number of media objects is, the larger the difference of the number of reduced operations between the two algorithms is. In addition, the number of reduced operations increases exponentially (approximately) since the value of g^2 is involved.

5. Conclusions

As the number of databases and the amount of data increase, the knowledge discovery procedure involves the processing of large volume of data which makes the procedure computationally expensively. In most cases, only a portion of data in the databases is relevant to some specific knowledge discovery procedure. Hence, how to develop methodologies to reduce the size of data and to improve the performance for the discovery procedure is important. For this purpose, domain knowledge can be used.

In this paper, we incorporated domain knowledge in the generalized affinity-based association mining algorithm to eliminate the irrelevant data so that the size of data involved in the computation task is reduced. The set of domain knowledge is an additional input to the mining algorithm. Also, a data selection process is proposed as the pre-processing step in the mining algorithm. This data selection process suggests the strategies to select the relevant media object pairs for the computation by utilizing domain knowledge rules. Then, the mining algorithm is executed on the selected media object pairs without the need to go through all the media object pairs in the databases. This reduces the number of operations involved in the computation tasks.

An experiment was conducted to compare the number of operations for the original and modified mining algorithms with and without the incorporation of domain knowledge. We also discussed the benefits of using domain knowledge to reduce the number of operations in the mining algorithm by analyzing the experimental result. The experimental result demonstrates that the number of operation reduction can be achieved significantly when domain knowledge is incorporated in the mining algorithm. Moreover, more domain knowledge rules are derived within the knowledge discovery procedure and used as domain knowledge to eliminate unreasonable situations in the modified mining algorithm. This shows the power of incorporating domain knowledge into the knowledge discovery procedure.

6. References

[1] P. Adriaans and D. Zantinge. *Data Mining*, Addison-Wesley, 1996.

[2] R. Agrawal, T. Imielinski, A. Swami, "Mining association rules between sets of items in large databases, " Proc. 1993 ACM SIGMOD Conference on Management of Data, pp. 207-216, 1993.

[3] J. Chattratichat, J. Darlington, and M. Ghahem, "Large scale data mining: Challenges and responses," *Proc. of the third International Conference on Knowledge Discovery and Data Mining*, pp. 143-146, 1997.

[4] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: An overview," in U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pp. 1-34, AAAI/MIT Press, 1996.

[5] U.M. Fayyad, "Data mining abd knowledge discovery: Making sense out of data," *IEEE Expert*, vol. 11, pp. 20-25, 1996.

[6] R. Groth. *Data mining: Building competitive advantage*. Prentice Hall, 2000.

[7] R. Groth. *Data mining: A hands-on apporach for business professionals*. Prentice Hall, 1998.

[8] J.A. Larson, S.B. Navathe, and R. Elmasri, "A theory of attribute equivalence in databases with application to schema integration," *IEEE Transaction on Software Engineering*, vol. 15, no. 4, Apr. 1989.

[9] M. Mehdi and O. Owrang, "Optimization of the knowledge discovery process in very large databases," *5th International Conference on Computer Science and Informatics*, pp. 490-495, 2000.

[10] S.B. Navathe, R. Elmasri, and J.A. Larson, "Integration user views in database design," *Comput.*, vol. 19, Jan. 1986.

[11] G. Piatetsky-Shapiro, "Knowledge discovery in real databases: A report on the IJCAI-89 Workshop," AI Magazine, vol. 11, no. 5, Special issue, pp. 69-70, Jan. 1991.

[12] M-L. Shyu, S-C. Chen, and R. L. Kashyap, "Database Clustering and Data Warehousing," 1998 ICS Workshop on Software Engineering and Database Systems, pp. 30-27, Dec. 17-19, 1998.

[13] M-L. Shyu, S-C. Chen, and R. L. Kashyap, "Discovering Quasi-Equivalence Relationships From Database Systems," ACM Eighth International Conference on Information and Knowledge Management (CIKM'99), pp. 102-108, November 2-6, 1999, Kansas City, MO, U.S.A.

[14] E. Simoudis, "Reality check for data mining," *IEEE Expert*, vol. 11, pp. 26-33, 1996.

[15] S. Yoon, L.J. Henschen, E.K. Park, S. Makki, "Using domain knowledge in knowledge discovery," ACM Eighth International Conference on Information and Knowledge Management (CIKM'99), pp. 243-250, November 2-6, 1999, Kansas City, MO, U.S.A.