이학석사 학위논문

다중데이터베이스 마이닝의 전처리를 위한 가중치 거리 기반 클러스터링

지도교수 유 선 대

이 논문을 이학석사 학위논문으로 제출함.

2004년 2월

부경대학교 대학원

전 자 계 산 학 과

김 진 현

김진현의 이학석사 학위논문을 인준함

2003년 12월 26일

주 심 공학박사 김 창 수 

위 원 이학박사 이 경 현 

위 원 이학박사 윤 성 대 

< 차 례 >

< 표 목 차 >

< 그 림 목 차 >

A Weight Distance-based Clustering for Preprocessing of MultiDatabase Mining

Jin-Hyun Kim

Dept. of Computer Science Graduate School of Pukyong National University

Abstract

Data mining is developed and available for real world applications. Most data mining algorithms assume a single data set for two or more real world applications which are made up of sales transactions. In particular, practitioners have to face the problem of discovering knowledge from multidatabase. In order to do so, we consider the problem of analyzing between items in multidatabase of sales transactions. Multidatabase mining is typically dependence of application, referred to databases selection when users require to mine their multidatabase without reference to any specific application.

In this paper, we propose a new algorithm for clustering databases which are based on support and items and we present a measure of Weight Distance, which is proposed for mining tasks with an objective of database clustering for preprocessing of multidatabase mining. Multidatabase is classified by constructing a distance measure between items called as Distance of between itemsets. An efficient

algorithm for identifying similarity databases is described and measure between items called as Weight Distance of between databases. Proposed algorithm improves the approach in RelevantDB algorithm and Ideal&Goodness algorithm for general applications. We have shown that the proposed approach is effective and promising.

1. 서론

대규모 데이터베이스로부터 유용한 정보나 의미 있는 사실을 추출하기 위한 전과정을 지식 탐사(KDD, Knowledge Discovery in Databases) 과정이라고 한다[1]. 특히 지식 탐사 과정 중에서 관찰된 데이터로부터 패턴이나 모델의 추출을 데이터 마이닝(Data Mining)이라고 한다[2].

지금까지 연구된 데이터 마이닝의 기법으로는 연관 규칙(Association Rule), 통계적 기법(Statistical Technique), 의사결정 트리(Decision Tree), 신경망 회로(Neural Network), 유전자 알고리즘(Genetic Algorithm), 가시화 기법(Visualization), 온라인 분석 처리(On-Line Analysis Processing), 사례-기반학습(Case-base learning, k-nearliest neighbor)등이 있다[3-10].

이러한 데이터 마이닝 기법은 여러 곳에서 분산 운영되는 다중데이터베이스 시스템(Multidatabase System)[11]으로부터 정보를 추출한 후, 중앙 집중적으로 대형 데이터 집합(Large Data Set)을 형성한 데이터 웨어하우스(Data Warehouse)[12]에 적용되고 있다. 데이터 웨어하우스를 구성하는 과정에서 발생하는 overhead는 유용한 마이닝 알고리즘[2-10, 22-31]의 실행 시간을 지연시킬 뿐 만 아니라, 알고리즘 동작에도 영향을 미치게 된다. 그러므로, 다중데이터베이스로부터 데이터 웨어하우스를 구성하는 작업에 앞서 데이터베이스간의 유사성(Similarity)을 고려하여 데이터베이스를 클러스터링(Clustering) 하고 필요에 따라 주어진 문제에 대해 적절한 마이닝 기법을 적용하는 방법이 제시되었다. 제시된 두 가지 기법은 질의어(Query)의 술어(Predicate)에 속한 테이블명과 연산자를

통한 관련성(Relevant) 측정 방법(RelevantDB algorithm)[13], 트랜잭션 데이터베이스(Transaction Databases)에 속한 모든 항목(Item)들에 대해서 비교 분석하는 유사성 측정 방법(Ideal&Goodness algorithm)[14]이 있다.

심의어의 술어를 이용한 데이터 마이닝 기법은 다중데이터베이스가 갖는 특징 중 하나인 지역 자치성(Local Autonomy) 때문에 응용분야에 독립적으로 수행되지 못하고 일반화(Generalization)되어 있지 못하다는 단점이 있다[13].

Ideal&Goodness algorithm에서는 클러스터링 방법에 대한 일반화는 고려되었으나, 항목집합 발생률이 유사한 경우에 대해서는 항목간의 식별이 불가능하다는 단점이 있다[14].

본 연구는 RelevantDB algorithm과 Ideal&Goodness algorithm의 단점을 보완하고, 일반성을 최대한 고려하여 다중데이터베이스 마이닝을 위한 전처리 작업 단계로써 관련성이 높은 데이터베이스를 식별하기 위하여 거리에 대한 가중치(Weight of Distance)를 이용한 클러스터링 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 데이터 집합을 형성하는 방법에 사용되는 연관규칙 탐사기법에 관한 기존의 연구와 RelevantDB algorithm, Ideal&Goodness algorithm에 대해 고찰한다. 3장에서는 빈발 1 항목집합 데이터베이스간의 거리 및 가중치에 대한 소개와 최종적으로 얻어진 측정값을 통한 클러스터링 결과를 보여준다. 4장에서는 시뮬레이션 및 결과를 통한 제안된 모델의 타당성을 검증한다. 마지막 5장에서는 결론을 내리고, 향후 과제를 제시한다.

2. 관련 연구

최근 기업들은 유통 환경의 대형화와 전자 상거래의 영향으로 고객들이 어떤 제품을 구입하는지에 대한 많은 세일즈 정보들을 생성, 수집하여 다중데이터베이스에 저장하고, 이러한 정보들을 마케팅에 활용하기 위해 노력하고 있다. 세일즈 정보를 활용하기 위해 필요한 것이 숨겨진 지식 발견이며 이를 위한 많은 연구 중에서 특히, 고객이 구입하는 항목들 간에 존재하는 연관규칙(Association Rules)을 찾아내는 연구를 '장바구니 분석(Market Basket Analysis)' 이라고 한다[3]. 장바구니 분석에 사용되는 장바구니 데이터(Market Basket Data)는 트랜잭션(Transaction) 단위로 구성된다. 트랜잭션내의 투플(Tuple)은 전형적으로 고객 ID, 트랜잭션 날짜, 고객이 구입한 항목(Item)으로 구성되어 있다[15]. 이러한 항목들은 항목집합(Itemset)을 형성하며 트랜잭션의 요소가 된다. 고객이 구입한 항목집합들에 대한 발생 빈도를 분석하면 고객의 항목집합 구입 성향에 대한 가치 있는 정보를 판단 할 수 있다.

2.1 연관 규칙

Agrawal *et al.*[3]에 의해 처음으로 소개되었고, 항목집합의 연관규칙 발견을 위해 사용되는 빈발 항목집합(Frequent, Large Itemset)에 대해 살펴보면 다음과 같다.

$I = \{a_1, a_2, a_3, \ldots, a_\ell\}$를 항목(Item)들의 전체 집합이라 한다. 전체 집

합에 속한 항목들로 이루어진 집합을 항목집합(Itemset)이라 하며, 항목으로 이루어진 전체 집합의 부분집합이 된다. 집합 I 에 속한 항목들로 구성된 집합 X가 트랜잭션 T에 포함되면, 즉 $X \subseteq T$ 관계이면 T는 X를 '지지한다(support)'라고 한다. X를 지지하는 데이터베이스 D에 있는 모든 트랜잭션들의 개수를 지지도(support)라고 하며, *supp(X)*로 나타낸다.

최소지지도(minimum support)는 사용자가 미리 지정한 값이며 최소지지도를 사용하는 이유는 항목에 대한 발생이 관심 수준 이상으로 빈발한지를 고려하기 위해서이다. 만일 사용자가 지정한 최소지지도(*minsup*)에 대하여 $minsup \leq supp(X)$ 이면, 항목집합 X는 '빈발하다(Frequent, Large)'라고 하고, X를 빈발 항목집합이라고 정의하며, k개의 항목으로 이루어진 빈발 항목집합을 빈발 k-항목집합(Frequent, Large k-itemset)이라고 한다[16]. 가령, 최소지지도가 0.5이고 하나의 데이터베이스에서 모든 트랜잭션의 발생횟수가 4번이라면, 동일한 항목이 2번 이상 발생한 경우에 대해 '빈발하다'라고 할 수 있다.

연관규칙은 두 단계를 거쳐 정의된다[3]. 첫 번째 단계는 최소 지지도 이상의 지지도를 가지는 조합을 찾아 빈발 항목을 구성한다. 두 번째 단계는 데이터베이스로부터 연관 규칙을 생성하기 위하여 앞 단계에서 구성한 빈발 항목집합을 사용한다. 규칙의 생성 방법은 모든 빈발 항목집합(L)에 대해서 빈발 항목집합의 모든 공집합이 아닌 부분집합들을 찾는다. 각각의 부분집합(A)에 대하여, 만약 *supp*(A)에 대한 *supp*(L)의 비율이 적어도 최소 신뢰도(Minimum Confidence) 이상이면, $A \Rightarrow (L-A)$ 의 형태의 규칙을 출력한다. 이 규칙의 지지도는 *supp*(L)이고, 신뢰도는 *supp*(L)/*supp*(A)이다. 알고리즘 1은 Agrawal *et al.*[3]가 제시한 Apriori

알고리즘을 나타내며 후보집합의 생성은 빈발 항목집합을 생성하기 위하여 선행되는 작업이다. 후보 항목집합으로부터 항목수가 하나씩 증가한 빈발 항목집합을 결정하는 것이 알고리즘 1의 기본적인 흐름이다. 새로운 후보 항목집합을 만들고 각 항목집합의 지지도를 계산하여 최소지지도와 비교한 후, '빈발하다'라고 판단된 항목을 선택하여 빈발 항목집합을 구성하며 이를 다시 후보 항목집합으로 데이터집합을 재구성한다.

본 논문에서는 Apriori 알고리즘의 빈발 항목집합 생성 과정을 기초로 하여 빈발 1-항목집합만을 고려한다. 빈발 1-항목집합은 데이터베이스를 대표하는 특징적인 요소가 되며 비교 연산 수행시에 사용된다.

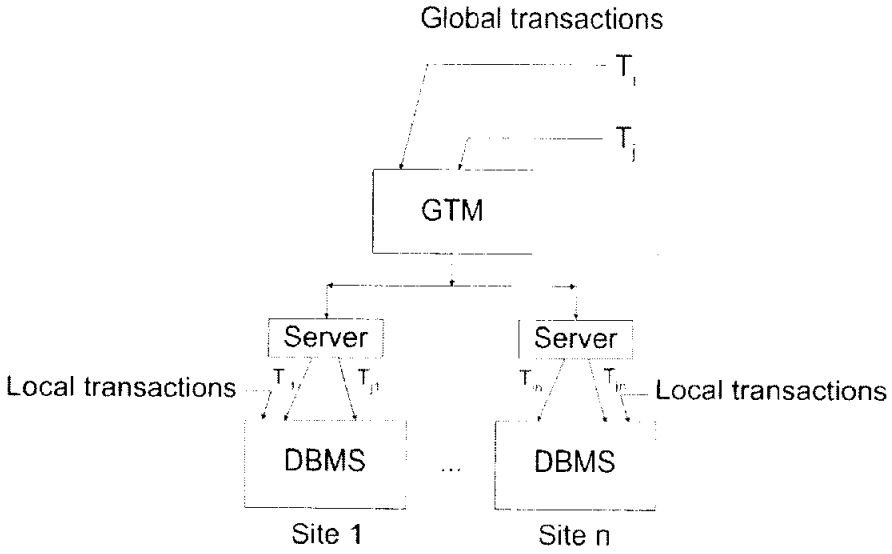
[알고리즘 1] Apriori 알고리즘

[Algorithm 1] Apriori Algorithm

```
Input : a database transaction t
Output : the large k-itemsets
L1={large 1-itemsets};                    // 빈발 항목을 구성
 For (k=2; Lk-1 <> 0; k++) do begin
 Ck=Apriori-gen(Lk-1);                    // 새로운 후보 항목
     Forall transactions t∈D do begin
         Ct=subset(Ck,t);                 // t에 포함된 후보항목
             Forall candidates c∈Ct do
                 c.count++;
             end
         Lk={c∈Ck|c.count >= min_support}
         // 최소지지도 이상의 항목의 조합을 추출
     End
  Answer=UkLk;
```

2.2 다중데이터베이스 시스템(MDBS)

다중데이터베이스 시스템(MDBS : Multidatabase system)은 데이터와 소프트웨어가 여러 형태의 통신망(Network)으로 연결된 다수의 사이트(Site)에 분산되어 있다. 분산된 사이트는 동질성(Homogeneous)을 고려하여 크게 두 가지로 구분한다. 모든 서버(Server) 또는 각 지역 DBMS(Database Management System)가 동일한 소프트웨어를 사용하고 모든 사용자가 동일한 소프트웨어를 사용한다면 이런 경우 동질 다중데이터베이스(Homogeneous MDB)라고 말하며, 그렇지 않은 경우는 이질 다중데이터베이스(Non-Homogeneous MDB)라고 말한다[11].



(그림 1) 다중데이터베이스 시스템 모델

(Figure 1) The MDBS Model

동질성의 정도와 관련된 또 다른 요소는 지역 자치성의 정도(Degree of local autonomy)이다. 만일 지역 사이트가 독립적인 DBMS로 동작할 수 없다면 그 시스템은 지역 자치성(Local Autonomy)을 갖지 못한다. 반면에 지역 트랜잭션(Local Transactions)들이 직접 서버에 접근할 수 있다면 그 시스템은 어느 정도의 지역 자치성을 갖는다고 말할 수 있다.

그림 1은 지역 자치성을 갖는 동질 다중데이터베이스를 나타내며 특징은 대용량 데이터베이스가 여러 사이트에 분산되어 각 사이트에는 보다 작은 데이터베이스가 존재하게 되고, 중앙 집중식 데이터베이스의 모든 트랜잭션이 수행되는 경우보다 더 적은 수의 트랜잭션들이 각 사이트에서 수행된다는 점이다. GTM(Global Transaction Management)은 트랜잭션들이 두 개 이상의 사이트에 접근할 때 실행에 대한 동기화와 전체 데이터베이스의 무결성(Integrity)을 유지하는 역할을 담당한다[15].

본 논문에서는 GTM의 트랜잭션 데이터베이스를 사용하여 데이터베이스간의 일반화를 최대한 고려하였다.

2.3 다중데이터베이스의 마이닝 기법

이 절에서는 기존의 논문에서 제안한 질의어(Query)의 술어(Predicate)에 속한 테이블명과 연산자를 통한 관련성 측정 기법(RelevantDB algorithm)[13]과 트랜잭션 데이터베이스에 속한 모든 항목(Item)들에 대해서 비교 분석하는 유사성 측정 방법(Ideal&Goodness algorithm)[14]에 대해 언급하고 있다.

(1) RelevantDB Algorithm

다중데이터베이스내의 관련성이 높은 데이터베이스를 식별하기 위한 데이터 마이닝 task로써 각 데이터베이스의 relation과 table을 이용하여, 질의어(Query)의 술어(Predicate)에 속한 table명과 operator를 통한 관련성(Relevance)을 측정한다. 마이닝 task는 몇 개의 attributes의 특성과 관련이 있으며 질의어(Query)의 술어(Predicate)를 이용하여 일치하는 attribute의 value에 대한 query에 포함된 operators를 *relevance factor*로 정의하고 있다. x개의 attributes로 구성되는 n개의 data table(relations)이 주어지고 속성 A_i에 대해 'A_i *relop* C'의 형태로 query predicate를 표현하며, *relop*은 relational operates 중 하나이다. 관련된 데이터베이스를 식별하는 것은 관련된 tables(relations)을 선택하는 것이고, 'A_i *relop* C'을 *selector*라고 말하며 query predicate를 Q로 표기한다. Q에 대하여 *selector* s의 *relevance factor*를 식 (1)과 같이 정의하며 Pr(s|Q)와 Pr(s)의 *different relationships*를 ratio로 하는 5가지의 경우로 분류하고 있다.

$$RF(s,Q) = \Pr(S|Q)\Pr(Q)\log\frac{\Pr(s|Q)}{\Pr(s)} \quad (1)$$

RF(s, Q)가 임계값(threshold)을 만족하는 경우를 *selector* s가 Q와 관련이 있다고 말한다. 그리고 Q와 관련이 있는 s_j가 한 개라도 존재한다면 table이 Q와 관련이 있다고 정의한다[13].

(2) Ideal and Goodness Algorithm

Ideal and Goodness Algorithm[14]은 다중데이터베이스를 구성하는 데이터베이스간의 유사성을 측정하기 위해 식 (2)를 제안하였다. 데이터베이스 D_i의 항목만을 고려한 방법이며 측정된 유사성의 임계치로써 $\alpha(>0)$를 적용한다. α는 데이터베이스로 이루어진 클래스의 수와 식 (2)의 종류에 따라 범위를 다르게 가진다. 가장 일반적인 경우로써 α가 0이면 모든 데이터베이스는 하나의 클래스로 구성되며 데이터베이스간의 모든 유사성 측정값은 언제나 α를 만족하게 된다. 이런 경우를 *Ideal Cluster*로 정의하며 이를 *Trivial Cluster*라고 부른다.

$$sim(D_i, D_j) = \frac{|Item(D_i, D_j) \cap Item(D_j, D_i)|}{|Item(D_i, D_j) \cup Item(D_j, D_i)|} \qquad (2)$$

식 (2)의 유사성 측정값이 0과 1사이의 값이므로 α가 1을 갖게 되면 유사성 측정값의 범위를 넘어서는 임계치가 되므로 1은 고려하지 않는다. 유사성 측정값 중에서 같은 범위에 속한 데이터베이스를 하나의 클래스로 묶는다. 모든 데이터베이스가 클래스에 속하게 되면 클래스의 수로써 0과 1사이를 분할하여 category로 구분한다. 그리고 category에 포함된 유사성 측정값만을 고려하여 모두 더한다. 따라서 발생하는 측정값의 종류가 많을수록 α값으로 분류되는 클러스터링 방법은 증가하며 α는 1보다 같거나 작은 경우에 대해서만 고려되어진다. 주어진 α값의 범위를 만족하는 유사성 측정값을 모두 더하여 가장 높은 값을 갖는 클러스터링 기법을 *Goodness*(class,α)로 정의한다. *Goodness*(class,α)에서 *Trivial*

*Cluster*를 제외한 결과 중에서 가장 높은 값을 갖는 클래스를 *GreedyClass*라고 정의한다[14].

3. 알고리즘의 제안

이번 절에서는 항목들에 대한 지지도를 이용하여 데이터베이스간의 거리(Distance)를 계산하고 유사한 항목으로 구성된 데이터베이스간의 식별 능력을 향상시키기 위해 빈발 1-항목집합을 이용한 데이터베이스간의 가중치 거리(Weight Distance) 기법을 제안한다.

3.1 빈발 1-항목집합과 항목 데이터집합

다중데이터베이스 집합을 MDB = {D_1, D_2, D_3, ... ,D_m}이라 하면, 데이터베이스 D_i, (i ∈ [1..m])는 MDB의 부분집합이고, 트랜잭션 T는 D_i 의 부분집합이다. 그리고 트랜잭션에 속한 항목과 각 항목에 대한 지지도로 구성된 집합을 본 논문에서는 항목 데이터집합(IS_i, i∈[1..m])이라고 한다.

본 논문에서는 앞서 살펴본 다중데이터베이스의 특징에서 각 사이트로 분산되는 전역 트랜잭션(Global Transaction)을 관리하는 GTM(Global Transaction Management)의 트랜잭션 로그(Transaction Log)와 각 사이트의 End-User로부터 입력되는 지역 트랜잭션을 관리하는 DBMS의 트랜잭션 로그를 저장하는 트랜잭션 데이터베이스를 이용하게 한다[11].

트랜잭션 데이터베이스의 속한 항목을 이용하여 다중데이터베이스 마이닝을 위한 전처리 단계를 수행한다. 일반적으로 다중데이터베이스의 트랜잭션 데이터베이스는 트랜잭션 ID(TID)와 항목집합으로 구성되며

표 1은 6개의 데이터베이스로 구성된 다중데이터베이스를 예시한 것으로서 각 데이터베이스는 항목집합과 트랜잭션 ID를 갖는 트랜잭션들로 구성된다.

<표 1> 6개의 데이터베이스로 구성된 다중데이터베이스 시스템

<Table 1> The MDBS consists of 6 Databases

D_1

TID	항목집합
110	a, b, c
120	a, b, d
130	a, b, c, d, e
140	a, e

D_2

TID	항목집합
110	a, c
120	a, b, c
130	b, c
140	a, c, e

D_3

TID	항목집합
110	a, b, d, e
120	b, c
130	a, b, c, d
140	a, b, c, d, e
150	a, b, c

D_4

TID	항목집합
110	d, g, h, i
120	d, j
130	g, i, j
140	d, h, i
150	h, i

D_5

TID	항목집합
110	e, g, h, i
120	e, h, i
130	e, h, j
140	h, j
150	e, g

D_6

TID	항목집합
110	g, i
120	g, h, i, j
130	g, h
140	g, h, j

위에서 예시한 6개의 데이터베이스에 대해 앞서 소개한 연관 규칙 기

법의 Apriori 알고리즘[3]의 k 값을 1로 수정하고 후보 1-항목집합을 생성하여 항목과 지지도를 갖는 새로운 데이터집합을 구성한다. 그리고 지지도에 대해 내림차순으로 정렬하면 표 2와 같다. 만일 지지도가 같은 항목은 항목에 대해 사전적 순서(Lexicographic Order)[18]에 따라 정렬한다.

<표 2> 6개의 데이터베이스에 대한 항목과 지지도

<Table 2> The Support Degree of Each Itemset in 6 Databases

D_1

TID	항목집합
110	a, b, c
120	a, b, d
130	a, b, c, d, e
140	a, e

⇨

IS_1

항목	지지도
a	4
b	3
c	2
d	2
e	2

D_2

TID	항목집합
110	a, c
120	a, b, c
130	b, c
140	a, c, e

⇨

IS_2

항목	지지도
c	4
a	3
b	2
e	1

D_3

TID	항목집합
110	a, b, d, e
120	b, c
130	a, b, c, d
140	a, b, c, d, e
150	a, b, c

⇨

IS_3

항목	지지도
b	5
a	4
c	4
d	3
e	2

D_4

TID	항목집합
110	d, g, h, i
120	d, j
130	g, i, j
140	d, h, i
150	h, i

⇨

IS_4

항목	지지도
d	3
h	3
i	3
g	2
j	2

D_5

TID	항목집합
110	c, g, h, i
120	c, h, i
130	c, h, j
140	h, j
150	c, g

⇨

IS_5

항목	지지도
c	4
h	4
g	2
i	2
j	2

D_6

TID	항목집합
110	g, i
120	g, h, i, j
130	g, h
140	g, h, j

⇨

IS_6

항목	지지도
g	4
h	3
i	2
j	2

Apriori 알고리즘의 수행과정을 살펴보면, 첫 단계는 트랜잭션 데이터베이스의 모든 항목들을 후보 1-항목집합으로 정의하고 항목의 지지도를 카운터하기 위해 데이터베이스를 scan한다. scan이 완료되면 카운팅된 항목의 지지도와 최소지지도를 비교하여 빈발 1-항목집합을 산출한다.

다음은 빈발 2-항목집합을 생성하기 위해 앞서 산출한 빈발 1-항목집합을 다시 후보 2-항목집합으로 재정의한다. 2번째 scan을 통해 2-항목집합에 대한 지지도를 누적하고 다시 최소지지도보다 크거나 같은 후보 2-항목집합만을 뽑아내어 빈발 2-항목집합으로 재구성한다. k의 값은 데이터베이스를 scan한 횟수와 데이터베이스에 포함된 항목들의 종류와 일치한다.

빈발 k-항목집합을 구성하기 위해서는 반드시 후보 k-항목집합이 필요하며 후보 k-항목집합을 구성하기 위해서는 빈발 (k-1)-항목집합이 필요하다. 그러므로 알고리즘의 k의 초기화가 2로 되어 있음을 알 수 있다. 본 논문에서는 Apriori 알고리즘의 빈발 k-항목집합을 구성하는 기법을 수정하여 사용한다. 데이터베이스의 scan 횟수와 일치하는 k를 1로

변경하여 빈발 1-항목집합만을 고려한다. 따라서 k를 조건으로 하는 loop 연산은 제외되며 높은 수행시간을 갖는 데이터베이스 scan 횟수를 한 번으로 줄인다.

본 논문에서 제안하는 데이터베이스간의 거리 연산을 위하여 항목만으로 구성된 항목집합을 정의한다. 위의 표 2의 결과로부터 얻어지는 항목만으로 구성된 집합은 다음과 같다.

$I(D_1)$={a, b, c, d, e}, $I(D_2)$={a, b, c, e}, $I(D_3)$={a, b, c, d, e},
$I(D_4)$={d, g, h, i, j}, $I(D_5)$={c, g, h, i, j}, $I(D_6)$={g, h, i, j}

표 3은 본 논문에서 제안한 수식 및 수행 절차의 사용하는 파라미터들을 요약한 것이다.

<표 3> 다중데이터베이스 클러스터링을 위한 파라미터

<Table 3> Parameters for MDB Clustering

기 호	의 미
$I(D_i)$	항목으로 구성된 데이터 집합
IS_i	항목 데이터 집합
$IS_i(a_n)$	IS_i에 속한 항목(a_n)
a_n[Supp]	항목 a_n의 지지도
$dist_{wgt}(D_i, D_j)$	항목 데이터 집합의 가중치 거리
C_k	클러스터 C_k

3.2 데이터베이스간의 거리

계층적 클러스터링(Hierarchical Clustering)의 Average Linkage Method[17]와 Symmetric Difference[18]의 성질을 고려하여 식 (3)과 데이터베이스간의 거리를 측정하는 기법을 제안한다.

3.2.1 항목으로 구성된 집합간의 거리

데이터베이스의 트랜잭션내의 존재하는 항목으로 구성된 항목집합을 본 논문에서 제안하는 식 (3)에 대입하여 데이터베이스간의 거리를 계산한다. 본 논문에서는 데이터베이스간의 거리 측정을 위해 앞에서 언급한 항목과 지지도로 구성된 데이터집합과 항목만으로 구성된 데이터집합을 이용하여 항목의 소속정도에 따른 항목집합의 차이를 이용한다. 제안하는 다음의 식 (3)에 지지도와 항목으로 구성된 표 3의 $I(D_i)$를 대입하여 데이터베이스간의 거리를 계산한다.

$$dist(I(D_i),\ I(D_j)) = \frac{|I(D_i)\Delta I(D_j)|}{|I(D_i)\cup I(D_j)|} = \frac{|(I(D_i)-I(D_j))\cup(I(D_j)-I(D_i))|}{|I(D_i)\cup I(D_j)|}$$

$$(1\le i,\ j\le m) \quad (3)$$

식 (3)을 살펴보면 Symmetric Difference의 부분식인 '$I(D_i) - I(D_j)$'는 $I(D_i)$에 속한 항목 중 $I(D_j)$에 속하지 않는 항목을 찾아내는 것이다. 같은 방법으로 $I(D_j)$에는 속하지만 $I(D_i)$에는 속하지 않는 항목을 찾아낸다. 이

렇게 얻어진 두 집합간의 차이를 union하고 그 결과에 $I(D_i)$, $I(D_j)$의 union의 cardinality로 나누어 거리를 연산한다. 다시 말하면, 두 개의 데이터 집합의 모든 항목들은 유한한 N개의 원소로 이루어지는 유한 개의 표본 공간이 되며 두 개의 데이터 집합의 모든 항목들로 이루어진 집합에서 공통으로 존재하는 데이터 집합을 빼면 상호 간의 항목들에 대한 차이를 구할 수 있다.

앞서 예시한 6개의 데이터베이스의 항목집합에 식 (3)을 적용한 결과를 나열하면 다음과 같다.

$$dist(I(D_1),\ I(D_2))=\frac{|I(D_1)\Delta I(D_2)|}{|I(D_1)\cup I(D_2)|}=\frac{|\{d\}\cup\varnothing|}{|\{a,b,c,d,e\}|}=\frac{|\{d\}|}{|\{a,b,c,d,e\}|}=\frac{1}{5}$$

$$dist(I(D_1),\ I(D_3))=\frac{|I(D_1)\Delta I(D_3)|}{|I(D_1)\cup I(D_3)|}=\frac{|\varnothing|}{|\{a,b,c,d,e\}|}=0$$

$$dist(I(D_1),\ I(D_4))=\frac{|I(D_1)\Delta I(D_4)|}{|I(D_1)\cup I(D_4)|}=\frac{|\{a,b,c,e\}\cup\{g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=\frac{|\{a,b,c,e,g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=\frac{8}{9}$$

$$dist(I(D_1),\ I(D_5))=\frac{|I(D_1)\Delta I(D_5)|}{|I(D_1)\cup I(D_5)|}=\frac{|\{a,b,c,d\}\cup\{g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=\frac{|\{a,b,c,d,g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}\frac{8}{9}$$

$$dist(I(D_1),\ I(D_6))=\frac{|I(D_1)\Delta I(D_6)|}{|I(D_1)\cup I(D_6)|}=\frac{|\{a,b,c,d,e\}\cup\{g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=\frac{|\{a,b,c,d,e,g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=1$$

$$dist(I(D_2),\ I(D_3))=\frac{|I(D_2)\Delta I(D_3)|}{|I(D_2)\cup I(D_3)|}=\frac{|\varnothing\cup\{d\}|}{|\{a,b,c,d,e\}|}=\frac{|\{d\}|}{|\{a,b,c,d,e\}|}=\frac{1}{5}$$

$$dist(I(D_2),\ I(D_4))=\frac{|I(D_2)\Delta I(D_4)|}{|I(D_2)\cup I(D_4)|}=\frac{|\{a,b,c,e\}\cup\{d,g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=\frac{|\{a,b,c,e,d,g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=1$$

$$dist(I(D_2),\ I(D_5))=\frac{|I(D_2)\Delta I(D_5)|}{|I(D_2)\cup I(D_5)|}=\frac{|\{a,b,c\}\cup\{g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=\frac{|\{a,b,c,g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=\frac{7}{8}$$

$$dist(I(D_2),\ I(D_6))=\frac{|I(D_2)\Delta I(D_6)|}{|I(D_2)\cup I(D_6)|}=\frac{|\{a,b,c,e\}\cup\{g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=\frac{|\{a,b,c,e,g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=1$$

$$dist(I(D_3),\ I(D_4))=\frac{|I(D_3)\Delta I(D_4)|}{|I(D_3)\cup I(D_4)|}=\frac{|\{a,b,c,e\}\cup\{g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=\frac{|\{a,b,c,e,g,h,i,j\}|}{|\{a,b,c,d,e,g,h,i,j\}|}=\frac{8}{9}$$

$$dist(I(D_3),\ I(D_5))=\frac{|I(D_3)\Delta I(D_5)|}{|I(D_3)\cup I(D_5)|}=\frac{|\{a,b,c\}\cup\{g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=\frac{|\{a,b,c,g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=\frac{7}{8}$$

$$dist(I(D_3),\ I(D_6))=\frac{|I(D_3)\Delta I(D_6)|}{|I(D_3)\cup I(D_6)|}=\frac{|\{a,b,c,e\}\cup\{g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=\frac{|\{a,b,c,e,g,h,i,j\}|}{|\{a,b,c,e,g,h,i,j\}|}=1$$

$$dist(I(D_4),\ I(D_5))=\frac{|I(D_4)\Delta I(D_5)|}{|I(D_4)\cup I(D_5)|}=\frac{|\{d\}\cup\{e\}|}{|\{d,e,g,h,i,j\}|}=\frac{|\{d,e\}|}{|\{d,e,g,h,i,j\}|}=\frac{1}{3}$$

$$dist(I(D_4),\ I(D_6))=\frac{|I(D_4)\Delta I(D_6)|}{|I(D_4)\cup I(D_6)|}=\frac{|\{d\}\cup\varnothing|}{|\{d,g,h,i,j\}|}=\frac{|\{d\}|}{|\{d,g,h,i,j\}|}=\frac{1}{5}$$

$$dist(I(D_5),\ I(D_6))=\frac{|I(D_5)\Delta I(D_6)|}{|I(D_5)\cup I(D_6)|}=\frac{|\{e\}\cup\varnothing|}{|\{e,g,h,i,j\}|}=\frac{|\{e\}|}{|\{e,g,h,i,j\}|}=\frac{1}{5}$$

위와 같이 연산한 결과를 matrix로 표현하면 표 4와 같다. matrix는 symmetric relation을 포함하며 reflexive relation에서는 거리가 0으로 나타난다.

3.2.2 데이터베이스간의 유사성

표 4에서 $dist(I(D_i), I(D_j))$는 0에서 1사이의 값을 가진다. 반면, $dist(I(D_i), I(D_j))$이 1이면 두 개의 데이터베이스간에는 일치하는 항목이 존재하지 않는다. 반면에 $dist(I(D_i), I(D_j))$이 0이면 동일한 항목으로 구성된 데이터베이스이다. 0과 1사이의 값은 유사한 항목들로 구성되어 있다는 것을 알 수 있다. 결과적으로 0에 근접할수록 두 데이터베이스간의 유사성은 증가한다.

<표 4> $dist(I(D_i), I(D_j))$의 매트릭스

<Table 4> Matrix of $dist(I(D_i), I(D_j))$

dist	D_1	D_2	D_3	D_4	D_5	D_6
D_1	0	$\frac{1}{5}$	0	$\frac{8}{9}$	$\frac{8}{9}$	1
D_2	$\frac{1}{5}$	0	$\frac{1}{5}$	1	$\frac{7}{8}$	1
D_3	0	$\frac{1}{5}$	0	$\frac{8}{9}$	$\frac{7}{8}$	1
D_4	$\frac{8}{9}$	1	$\frac{8}{9}$	0	$\frac{1}{3}$	$\frac{1}{5}$
D_5	$\frac{8}{9}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{1}{3}$	0	$\frac{1}{5}$
D_6	1	1	1	$\frac{1}{5}$	$\frac{1}{5}$	0

다중데이터베이스간의 거리로부터 확률의 성질을 이용하여 유사성을 정의할 수 있다. 확률의 기본 성질로부터 사상(Event) A에 포함된 한 원소가 관측되는 경우, 사상 A가 발생하였다고 말하고, A^c는 A의 여사상(Complement)이며 A가 발생하지 않는 사상이다. 식 (3)은 확률의 기본

성질로부터 유도되는 식 (4)와 같은 성질이 존재한다.

$$P[A^c] = 1 - P[A] \tag{4}$$

P[A]가 데이터베이스간의 거리이며 확률의 역을 나타내는 $P[A^c]$은 공통된 항목들을 포함한 두 데이터베이스간의 유사성을 나타낸다. 따라서 유사성에 대한 연산은 식 (3)을 이용하여 연산한 거리의 역을 취한다.

예시한 6개의 데이터베이스간의 유사성을 표 5에 나타내었다.

<표 5> 데이터베이스간의 유사성

<Table 5> Similarity of Between Databases

sim	D_1	D_2	D_3	D_4	D_5	D_6
D_1	1	$\frac{4}{5}$	1	$\frac{1}{9}$	$\frac{1}{9}$	0
D_2	$\frac{4}{5}$	1	$\frac{4}{5}$	0	$\frac{1}{8}$	0
D_3	1	$\frac{4}{5}$	1	$\frac{1}{9}$	$\frac{1}{8}$	0
D_4	$\frac{1}{9}$	0	$\frac{1}{9}$	1	$\frac{2}{3}$	$\frac{4}{5}$
D_5	$\frac{1}{9}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{2}{3}$	1	$\frac{4}{5}$
D_6	0	0	0	$\frac{4}{5}$	$\frac{4}{5}$	1

식 (1)을 적용하여 얻어지는 결과는 항목 발생만을 고려한 데이터베이스간의 거리를 나타낸다. 이러한 연산 결과는 데이터베이스의 트랜잭션들에 속한 항목이 여러 번 발생한 경우와 한 번 발생한 경우에 대해서는 두 데이터베이스간의 항목 차이, 즉 거리의 차이를 알 수 없다. 이것은

유사한 항목들로 구성된 데이터베이스간의 세밀한 거리 식별 능력을 갖지 못함을 의미한다. 이러한 단점을 보완하여 유사한 데이터베이스간의 거리를 명확하게 구분하는 방법으로써 아래와 같은 기법을 제안한다.

3.3 항목 데이터집합간의 가중치 거리

항목 데이터집합간의 가중치 거리는 데이터베이스간의 가중치 거리를 나타내며 각 항목의 지지도에 대한 확률로서 가중치를 계산한다.

3.3.1 항목 지지도에 대한 확률과 수행 절차

데이터베이스에 나타나는 항목의 발생률이 높을수록 확률은 증가하며 계산한 확률 중에서 최대 확률을 갖는 항목을 선택하여 그 차이를 두 데이터베이스간의 식별(Identity)을 위한 가중치로 결정한다.

$$P[a_i] = \frac{a_i[Supp]}{\sum_{i=1}^{n} a_i[Supp]} \quad (a_i \in IS_j,\ 1 \le j \le m) \qquad (5)$$

식 (5)는 전체 항목 지지도에 대한 각 항목의 지지도를 의미하며, 가중치 거리 연산을 위하여 아래에 기술한 수행 절차에 사용한다. 표 2의 항목 지지도를 이용하여 확률을 연산하면 표 6과 같다. $P[a_i]$에 대하여 내

림차순으로 정렬을 하고 최대 확률과 항목을 선택하여 수행절차에 적용한다. 최대 확률이면서 같은 확률이 2개 이상 발생하는 경우는 해당되는 항목과 확률을 모두 선택한다.

<표 6> 항목 지지도에 대한 확률

<Table 6> The Probability for the Support Degree of Each Item

IS_1

항목	지지도	$P[a_i]$
a	4	0.31
b	3	0.23
c	2	0.15
d	2	0.15
e	2	0.15

IS_2

항목	지지도	$P[a_i]$
c	4	0.4
a	3	0.3
b	2	0.2
e	1	0.1

IS_3

항목	지지도	$P[a_i]$
b	5	0.28
a	4	0.22
c	4	0.22
d	3	0.17
e	2	0.11

IS_4

항목	지지도	$P[a_i]$
i	4	0.29
d	3	0.21
h	3	0.21
g	2	0.14
j	2	0.14

IS_5

항목	지지도	$P[a_i]$
h	4	0.29
e	4	0.29
g	2	0.14
i	2	0.14
j	2	0.14

IS_6

항목	지지도	$P[a_i]$
g	4	0.36
h	3	0.27
i	2	0.18
j	2	0.18

항목 데이터 집합의 가중치 거리는 $dist_{wgt}(D_i, D_j)$, $(1 \le i, j \le m)$으로 표기하고 가중치 거리를 연산하기 위한 수행 절차는 다음과 같다.

단계 1. 각 항목 데이터 집합에서 최대지지도를 갖는 항목을 포함한 항목 데이터집합만을 선택하여 클래스(Class)를 구성한다.

단계 2. 클래스에 속한 항목 데이터집합 IS_i과 IS_j를 선택한다.

단계 3. 수식 (5)에 따라 IS_i의 각 항목 지지도의 확률을 구하고 확률이 가장 높은 것과 그에 대한 항목을 선택한다. 만일, 최대지지도를 갖는 항목이 하나의 항목 데이터집합에서 2개 이상 존재한다면, 다음 단계를 수행한다.

① 항목에 대해서 최대지지도가 같은 경우는 확률도 같으므로 하나의 확률만을 선택하고, 항목들에 대해서는 모두 선택한다.

② 선택한 항목들과 동일한 항목을 IS_j에서 선택하되, 확률 차이가 가장 큰 항목 하나만을 선택한다.

단계 4. 단계 3에서 선택한 항목과 동일한 항목, 그리고 그에 대한 확률을 IS_j에서 선택한다.

단계 5. 다음 식을 적용하여 연산한다. 여기서 *Weight*는 가중치를 나타낸다.

$$Weight(IS_i(a_n),IS_j(a_n))=|IS_i(a_n)\text{의 } P[a_n]-IS_j(a_n)\text{의 } P[a_n]| \quad (6)$$

단계 6. $Weight(IS_j(a_n), IS_i(a_n))$은 단계 2에서 6까지의 동일한 수행절차를 기쳐 연산한다.

단계 7. 다음 식을 적용하여 가중치 거리($dist_{wgt}(D_i, D_j)$)를 연산하고 종료한다.

$$dist_{wgt}(D_i, D_j)=Weight(IS_i(a_n),IS_j(a_n))+Weight(IS_j(a_n),IS_i(a_n)) \quad (7)$$

본 논문에서 제안한 수행 절차의 식 (7)은 데이터베이스간의 가중치 거리를 결정한다.

아래의 matrix는 위에서 예시한 6개의 데이터베이스간의 거리에 대해 단계 1을 거쳐 두 개의 클래스($Class_1$, $Class_2$)로 분류하고 각 클래스에 속한 항목 데이터집합간의 거리를 연산한 결과이다.

- 2개의 클래스로 구분 : $Class_1 = \{D_1, D_2, D_3\}$, $Class_2 = \{D_4, D_5, D_6\}$

		D_1	D_2	D_3
	D_1	0	-	-
$Class_1$:	D_2	0.26	0	-
	D_3	0.14	0.26	0

		D_4	D_5	D_6
	D_4	0	-	-
$Class_2$:	D_5	0.23	0	-
	D_6	0.33	0.24	0

단계 1에서 다중데이터베이스를 일차적으로 클래스로 구성한 이유는 최대지지도를 포함하지 않은 데이터베이스를 구분하고 데이터베이스간의 비교 연산 횟수를 줄이기 위해서이다. 위의 예시된 연산과정에서는 클래스로 구분하지 않은 경우보다 9번의 비교 연산 횟수가 줄어들었다.

3.3.2 다중데이터베이스의 클러스터링을 위한 가중치 거리의 적용

앞에서 제시한 식 (3)과 식 (7)을 이용하여 데이터베이스간의 가중치 거리에 대한 측정값을 연산하기 위해 식 (8)을 제안한다.

$$\text{Measure}(C_k) = |k - (\sum_{D_i, D_j \in C_k}^{1 \le x \le n} \text{dist}(D_i, D_j) + \text{dist}_{wgt}(D_i, D_j))| \quad (8)$$

식 (8)에서는 k가 나타내는 클러스터의 개수 증가에 따른 클러스터 내부의 중심점간의 거리의 합이 증가하게 되므로 가중치 거리와 거리를 합한 결과를 클러스터의 개수 k에서 빼주어 클러스터간의 추가적인 거리 비용을 적용한다.

Measure에서 우선, 첫 번째 $Class_1$에 속한 D_1, D_2, D_3 간의 Measure를 matrix로 표현하면,

$$\text{Measure}_{(Class_1)} = \begin{array}{c|ccc} & D_1 & D_2 & D_3 \\ \hline D_1 & 0 & - & - \\ D_2 & 0.46 & 0 & - \\ D_3 & 0.14 & 0.46 & 0 \end{array}$$

이고 $Class_2$에 속한 D_4, D_5, D_6 간의 Measure를 같은 방법으로 표현하면 다음과 같다.

$$\text{Measure}_{(Class_2)} = \begin{array}{c|ccc} & D_4 & D_5 & D_6 \\ \hline D_4 & 0 & - & - \\ D_5 & 0.56 & 0 & - \\ D_6 & 0.53 & 0.44 & 0 \end{array}$$

데이터베이스간의 거리에 대한 측정값에서 가장 작은 결과를 갖는 클러스터링을 선택하여 최종적으로 클러스터를 형성한다. 앞서 예시된 6개의 데이터베이스로 이루어진 다중데이터베이스에 대한 클러스터는 $Class_1$에 속한 3개의 데이터베이스를 본 논문에서 제안한 수행 절차에 따라 다음과 같이 연산한다.

(1) $Class_1$에 대한 측정값

1) $\{\{D_1\}, \{D_2\}, \{D_3\}\}$: $Measure(C_3) = |3 - (0 + 0 + 0)| = 3$

2) $\{\{D_1, D_2\}, D_3\}$: $Measure(C_2) = |2 - (0.46 + 0)| = 1.54$

3) $\{D_1, \{D_2, D_3\}\}$: $Measure(C_2) = |2 - (0 + 0.26)| = 1.74$

4) $\{\{D_1, D_3\}, D_2\}$: $Measure(C_2) = |2 - (0.14 + 0)| = 1.86$

5) $\{D_1, D_2, D_3\}$: $Measure(C_1) = |1 - (0.26 + 0.14 + 0.26)| = 0.34$

$Class_2$에 속한 3개의 데이터베이스를 클러스터링 할 수 있는 경우의 수는 5가지가 존재한다. 각 클러스터링에 대한 Measure는 다음과 같이 연산한다.

(2) $Class_2$에 대한 측정값

1) $\{\{D_4\}, \{D_5\}, \{D_6\}\}$: $Measure(C_3) = |3 - (0 + 0 + 0)| = 3$

2) $\{\{D_4, D_5\}, D_6\}$: $Measure(C_2) = |2 - (0.56 + 0)| = 1.44$

3) $\{D_4, \{D_5, D_6\}\}$: $Measure(C_2) = |2 - (0 + 0.44)| = 1.56$

4) $\{\{D_4, D_6\}, D_5\}$: $Measure(C_2) = |2 - (0.33 + 0)| = 1.67$

5) $\{D_4, D_5, D_6\}$: $Measure(C_1) = |1 - (0.23 + 0.33 + 0.24)| = 0.2$

연산한 결과를 보면 가장 낮은 Measure를 갖는 경우가 관련성이 가장 높은 데이터베이스를 클러스터링 한 경우로써 $Class_1$은 0.34로 계산되는 $\{D_1, D_2, D_3\}$로 클러스터를 구성하고, $Class_2$는 0.2에 해당되는 $\{D_4, D_5, D_6\}$로 클러스터링 한다.

4. 시뮬레이션 및 성능 평가

질의어(Query)를 이용한 RelevantDB algorithm[13]과 Ideal and Goodness algorithm[14]을 본 논문에서 제안하는 알고리즘에 시뮬레이션 데이터를 적용하여 비교 평가하였다.

4.1 Synthetic DataSets과 트랜잭션 데이터베이스

본 논문에서는 성능 평가를 위해 표 7과 같은 Synthetic DataSets[19]을 사용하였다. ARMiner[20, 21]를 사용하여 트랜잭션 데이터베이스를 생성하였고 다중데이터베이스를 구성하는 15개의 데이터베이스는 항목 값 범위와 트랜잭션 발생 순서를 random하게 선택하여 구성하였다.

<표 7> 종합적 데이터집합의 파라미터 설정

<Table 7> Parameter Settings of Synthetic Datasets

데이터베이스 개수	15	전체 항목 수	1000	패턴 수	5
데이터베이스 당 트랜잭션 수	2000	전체 트랜잭션 총량	30000	트랜잭션 당 평균 항목 수	10

실험에 사용한 파라미터의 입력 값 중에서 하나를 예시하면 다음과 같다.

· 데이터베이스 개수 : 15

· 전체 트랜잭션 총량 : 30000

· 데이터베이스 당 트랜잭션 수 : 2000

· 패턴 수 : 4

· 트랜잭션 당 평균 항목 수 : 10

아래에 나열된 것은 트랜잭션 30000개를 발생시킨 결과이다. Itemsets의 각 정수는 항목을 나타낸다.

TID Itemsets

1 : 157 303 399 411 575 608 725 737 790 795 878

2 : 72 90 248 284 691 742 751 815 822 897 941

3 : 12 46 92 157 213 234 259 267 323 358 411 645 698 779 920

4 : 133 168 199 221 409 480 531 718 727 782 808 822

5 : 31 35 59 124 263 406 445 483 596 718 724 754 761

6 : 74 171 186 226 271 389 459 691 705 718 771 779

7 : 72 90 99 199 246 342 467 507 578 738 742 895 987

8 : 79 137 186 201 213 326 382 481 607 636 717 741 826 901

9 : 123 133 234 332 361 471 494 607 618 636 796 993

10 : 34 88 124 152 206 251 296 367 411 419 481 502 528 741 815 867 879 888

. . . . . .

29991 : 18 64 140 233 387 654 687 736 767 774 804

29992 : 223

29993 : 133 186 532 645 695 737 774 779

29994 : 122 133 259 284 332 359 408 431 531 568 699 833

29995 : 133 171 172 180 238 325 378 506 687 717 788 958

29996 : 12 138 163 171 271 285 335 557 596 691 771 954

29997 : 34 60 141 213 309 335 694 779 786 795 822

29998 : 213 224 246 925

29999 : 72 210 285 538 578 585 699 827 885 887

30000 : 31 426 474 726 767 774 800 913

위의 30000개의 트랜잭션을 random하게 2000개씩 그룹핑하여 MDB에 속한 15개의 데이터베이스로 구성했다. 다음은 각 데이터베이스에서 발생하는 항목과 지지도를 나타낸 것이다.

DB	Item[Supp]
D_1	71[496], 175[330], 38[299], 263[266],…
D_2	71[484], 175[347], 38[284], 263[254],…
D_3	38[283], 263[270], 195[229], 178[228],…
D_4	175[318], 38[296], 263[248], 195[227],…
D_5	71[508], 175[333], 263[280], 38[264],…
D_6	175[315], 38[282], 263[275], 195[242],…
D_7	38[476], 175[327], 13[310], 175[237],…
D_8	263[502], 175[307], 38[279], 143[259],…
D_9	71[508], 175[339], 38[294], 263[267],…
D_{10}	71[486], 175[323], 38[274], 173[240],…
D_{11}	175[490], 263[382], 38[299], 263[266],…
D_{12}	263[480], 38[319], 175[288], 263[272],…

D_{13} : 38[396], 175[320], 195[269], 13[250],…

D_{14} : 175[422], 263[308], 38[291], 195[250],…

D_{15} : 71[388], 175[312], 263[284], 13[271],…

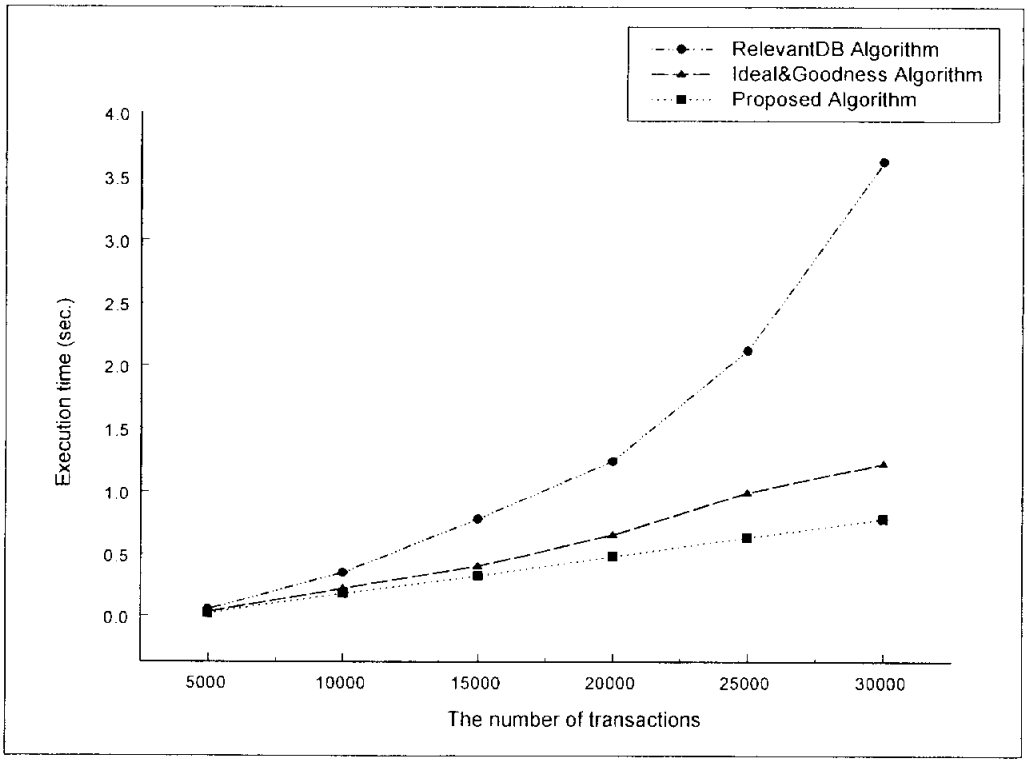
4.2 시뮬레이션 결과

다음의 그림 2는 트랜잭션 수의 증가에 따른 탐색 시간 및 알고리즘 수행 시간의 선형적인 증가를 나타낸다.

트랜잭션의 수가 증가함에 따라 추가적인 트랜잭션간의 비교 연산이 증가하여 RelevantDB algorithm[13], Ideal&Goodness algorithm[14], 과 제안한 기법 모두 실행시간이 선형적으로 상승하였다. RelevantDB algorithm의 실행시간이 상대적으로 길어지는 이유는 트랜잭션내의 항목과 항목을 포함한 테이블명을 탐색하는데 소요되는 비용이 더 추가되기 때문이다.

Ideal&Goodness algorithm과 제안한 기법은 트랜잭션의 총량이 16500에서부터 실행시간의 차이를 보인다. 그 이유는 빈발 항목만을 고려한 제안한 기법의 비교 연산 횟수가 항목집합 전체를 대상으로 하는 Ideal&Goodness algorithm보다 작기 때문이다.

Ideal&Goodness algorithm은 데이터베이스에 포함된 모든 항목들에 대해서 유사성을 고려하고 α를 조건으로 하는 클러스터링 수행 과정과 class를 결정하기 위한 부하가 많기 때문에 실행시간의 상대적인 추가가 필요하다.



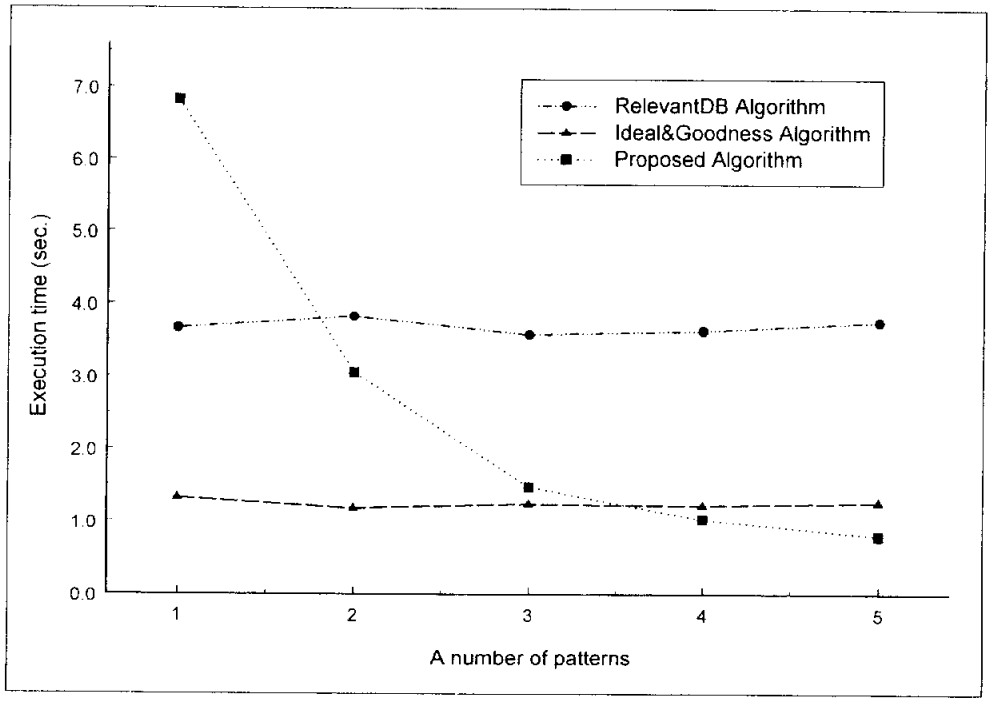
(그림 2) 트랜잭션 수의 증가에 따른 실행시간의 비교

(Figure 2) Execution Time as the Number of Transactions Scale-Up

그림 3의 시뮬레이션 환경은 지지도가 같은 항목의 발생을 나타내는 패턴의 변화를 적용한 것이다. 패턴의 수가 감소하면 데이터베이스간의 지지도에 대한 동일한 확률이 빈번하게 발생한다. 따라서 확률이 가장 높은 항목도 추가적으로 증가하게 되어, 가중치를 결정하기 위한 수행 절차의 비교 연산 횟수가 늘어나 패턴 발생이 높은 경우보다 실행시간이 높게 나타난다.

그림 4는 제안한 기법과 Ideal&Goodness algorithm[14]의 항목 수의

증가에 따른 클러스터 개수를 비교한 것이다. RelevantDB algorithm은 *relevance factor*를 통한 수치값의 해석으로 데이터베이스간의 관련성을 측정하므로 시뮬레이션에서 제외시켰다. 트랜잭션 수가 일정하고, 항목 수가 증가하게 되면 데이터베이스를 구성하는 항목의 종류는 많아지고, 최대지지도는 감소한다.

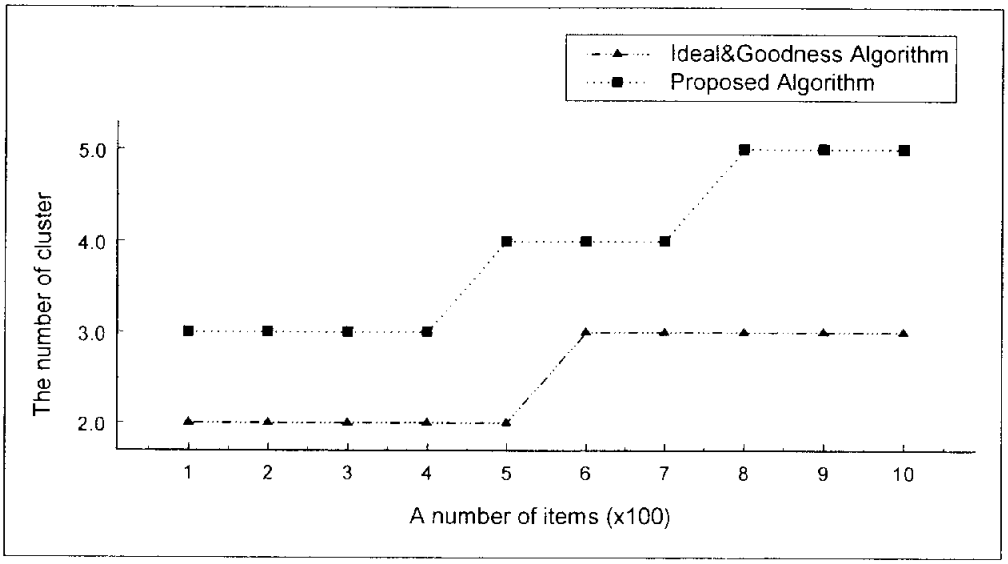


(그림 3) 패턴 수의 증가에 따른 실행 시간의 비교

(Figure 3) Execution Time as a Number of Patterns Scale-Up

따라서 항목 수가 작은 경우, 즉 데이터베이스간의 유사성이 높은 경우에 대한 Ideal&Goodness algorithm[14]의 결과는 항목 종류만을 고려

하였기에 식별이 제대로 이루어지지 않아 클러스터의 수가 작게 나타난다. 그러나, 제안한 기법은 동일한 항목이 존재하더라도 지지도에 따른 확률의 차이가 반영된 가중치를 이용하여 유사성 높은 데이터베이스들을 세밀하게 식별한다.



(그림 4) 항목 수의 증가에 따른 클러스터 개수의 비교

(Figure 4) The Number of Clusters as a Number of Items Scale-Up

5. 결론 및 향후과제

데이터 마이닝의 연구는 장바구니 데이터로 구성된 트랜잭션 데이터베이스들을 식별하는 방법으로 하나의 데이터 집합을 구축한 후, 여기에 적합한 알고리즘을 설계하는 것이었다. 이처럼 유용한 정보를 발견하기 위한 데이터 마이닝 알고리즘의 적용단계에 앞서 수행되는 다중데이터베이스내의 트랜잭션의 항목집합을 하나의 데이터 집합으로 형성하는 작업에 소요되는 비용이 많으므로 다중데이터베이스의 데이터베이스를 식별하는 방법이 제시되었다.

제안한 기법은 항목집합을 포함한 데이터베이스간의 거리를 연산하기 위해 빈발 1-항목집합을 구성한다. 빈발 1-항목집합은 항목과 지지도로 구성되며 항목집합간의 거리를 연산하기 위해 항목만을 선택한다. 항목집합간의 거리는 데이터베이스간의 거리를 나타내며 거리의 역은 데이터베이스간의 유사성을 반영한다. 항목만을 고려하여 측정된 데이터베이스간의 거리는 유사한 항목이 빈번하게 존재하는 경우에 대해서는 식별 능력이 떨어지는 단점이 있다. 데이터베이스간의 거리에 대한 식별성을 향상시키기 위해 빈발 1-항목집합의 지지도를 이용한 확률을 연산하여 가중치를 산출한다. 데이터베이스간의 거리와 가중치 거리를 연산하여 Measure를 측정하고, 측정된 결과 중에서 최소 Measure를 갖는 클러스터링을 선택하여 다중데이터베이스의 클러스터를 구성한다.

제안하는 알고리즘은 질의어의 술어를 비교하는 방법의 문제점과 모든 항목에 대한 유사성을 측정하는 방법의 단점을 보완한다.

제안한 기법의 타당성을 입증하기 위한 시뮬레이션은 유사한 항목 발

생 수의 증가에 따른 클러스터 개수를 비교하였고, 패턴 수의 증가와 트랜잭션의 수를 증가하며 실행시간을 비교하였다. 성능평가를 통하여 제안한 기법이 기존의 연구보다 데이터베이스간의 세밀한 식별 능력을 갖고 있음을 알 수 있었고 최대 빈발항목만을 고려한 비교 연산의 수행이므로 실행시간이 단축되었다.

식별되어진 클러스터에 포함된 데이터베이스를 하나의 데이터 집합으로 구성한 후, 문제의 정의와 관심 수준에 맞는 적합한 데이터 마이닝 알고리즘을 선택하여 적용하면 효율적인 마이닝 작업과 정확한 결과를 얻을 수 있다.

향후 연구방향으로는 빈발 1-항목집합에서 최대 빈발항목만을 고려하여 생성한 데이터 집합을 실수 값 범위의 시간 데이터에 접목시켜 사건 발생 시간과 시간관계 유사성을 고려한 클러스터링 기법에 관한 연구로 확장하는 것이다.

참고 문헌

[1] J. Han, Y. Cai and N. Cercone, "Knowledge Discovery in Large Databases", In Proc. of *Computer World'92(2nd Intern'l Symp.)*, Kobe, Japan, pages 60-67, Nov. 1992.

[2] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. "Advances in Knowledge Discovery and Data Mining", MIT Press, 1996.

[3] R. Agrawal and T. Imielinski and A. Swani, "Mining association rules between sets of items in large databases", Proc. of the 1993. eds., *ACM SIGMOD Conference*, Washington DC, USA, pages 207-216, May 1993.

[4] K.C.C. Chan and A. K. C. Wong, K.C.C. Chan and A. K. C. Wong, "A statistical technique for extracting classificatory knowledge from databases," in G. Piatetsky-Shapirc and W. I. Frawley, eds., *Knowledge Discovery in Databases,* Menlo Park, CA: AAAI/MIT, pages 107-124, 1991.

[5] Quinlan, J.R. 1996(b). "Induction of decision trees", Machine Learning, 1: pages 81-106.

[6] Pedrycz, W. "Data mining and knowledge discovery through fuzzy neurocomputing", In: *Fuzzy-Neuro-Systems 97-Computational Intelligence*, pages 45-59, 1997.

[7] Ian W. Flockhart, Nicholas J. Radcliffe, "A Genetic

Algorithm-Based Approach to Data Mining", In Proc. of *the Second International Conference on Knowledge Discovery and Data Mining*, pages 299-302, 1996.

[8] Daniel A. Keim, Hans-Peter Kriege" Visualization Techniques for Mining Large Databases: A Comparison" *IEEE Transactions on Knowledge and Data Engineering*, 8(6): pages 923-938, 1996.

[9] Jiawei Han, "OLAP Mining: An Integration of OLAP with Data Mining", In Proc. *IFIP Conference on Data Semantics*(DS-7), Leysin, Switzerland, pages 1-11, Oct. 1997.

[10] K Hanney and M Keane, "Learning adaptation rules from a case-base", In *Advances in Case-Based Reasoning*, vol. 1168 of Lecture Notes in AI, pages 179-192, Jan. 1996.

[11] Y. Breitbart, H. Garcia-Molina, A. Silberschatz, "Overview of Multidatabase Transaction Management", *VLDB Journal on Very Large Data Bases*, Vol. 1, No. 2, pages 181-240, Oct. 1992.

[12] Bokun, M. and C. Taglienti, "Incremental data warehouse updates: approaches and strategies for capturing changed data", Data Mgmnt Rev. 8(5), 1998.

[13] H. Liu, H. Lu, and J. Yao, "Identifying Relevant Databases for Multidatabase Mining", In Proc. of *PAKDD*, pages 210-221, 1998.

[14] C. Zhang and S. Zhang, "Database Clustering for Mining Multi-Databases", In Proc. of *the 2002 IEEE Internatioanl*

Conference, Vol: 2, pages 974-979, 2002.

[15] Brachman, R., T. Khabaza, W. Klösgen, G. Piatetsky-Shapiro, and E. Simoudis, "Mining Business Databases", *Commun, ACM* 36(11): pages 42-48, 1996.

[16] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", In Proc. of *20th International Conference on VLDB*, Santiago, pages 487-499, Sep. 1994.

[17] Backer, E, "Computer-assisted Reasoning in Cluster Analysis", Prentice Hall, 1995.

[18] William B. Robinson Romualdas Skvarcius, "Discrete Mathematics with Computer Science Applications" The Benjamin/Cummings Publishing Company, 1986.

[19] Ashoka Savasere, Edward Omiecinski, Shamkant B. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases", In Proc. of *the 21st VLDB Conference*, Zurich, Swizerland, pages 432-444, 1995

[20] Jiawei Han, Jian Pei, Yiwen Yin, "Mining Frequent Patterns without Candidate Generation", In Proc. *ACM SIGMOD*, pages. 1-12, 2000.

[21] http://www.cs.umb.edu/laur/ARMiner

[22] Jean-Mark Adamo, "Data Mining for Association Rules and Sequential Patterns", Springer Verlag, New York, 2000.

[23] Yonaton Aumann and Yehuda Lindell, "A Statistical Theory for Quantitative Association Rules", In Proc. of *the 5th ACM SIGKDD Conference on Knowledge Discovery and Datamining*, pages 261-270, San Diego, CA, Aug. 1999.

[24] Robert Godin, Rokia Missaoui, and Hassan Alaoui, "Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices", *Computational Intelligence*, 11(2): pages 246-267, 1995.

[25] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining Frequent Patterns without Candidate Generation", In W. Chen, J. Naughton, and P. Bernstein, editors, Proc. of *2000 SIGMOD Conf. on Management of Data*, Vol. 29, pages 1-12, Dallas, TX, June 2000.

[26] X. Hu and N. Cercone, "Rough Set Similarity Based Learning from Databases", In Proc. of *the first international Conference on Knowledge Discovery and Data Mining*, pages 162-167, 1995.

[27] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lofti Lakhal, "Discovering Frequent Closed Itemsets for association Rules", In Proc. *7th international Conf. on Database Theory (ICDT)*, pages 398-416, Jan. 1999.

[28] Mohammed J. Zaki, "Generating Non-Redundant Association Rules", In *6th ACM SIGKDD Intern'l Conf. on Knowledge Discovery and Data Mining*, pages 34-43, Boston, MA, Aug. 2000.

[29] Zhang, T., R. Ramakrishnan, and M. Livny, "BIRCH: a new data

clustering algorithm and its applications" In Proc. of *Data Mining and Knowledge Discovery* 1: pages 141-182, 1997.

[30] R. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, no. 5, pages 1003-1016, 2002.

[31] Nürnberger, A., C. Borgelt, and A. Klose, "improving naive Bayes classifiers using neuro-fuzzy learning" In Proc. of *the Sixth International Conference on Neural Information Processing '99 (ICONIP'99)*, Perth, Australia, pages 154-159, Piscataway, NJ: IEEE.

[32] R. Agrawal, T. Imielinski, and A. Swami. "Mining association rules between sets of items in large databases", In Proc. of *the 20th 1993 ACM SIGMOD Int. Conf. on Management of Data*, pages 207-216, 1993.

감사의 글

이 논문이 완성되기까지 제게 언제나 아낌없는 조언과 저의 부족한 면을 채워 주지고, 학문의 길로 인도해주신 윤성대 교수님께 진심으로 감사드립니다. 그리고 부족한 이 논문을 심사해 주신 김창수 교수님, 이경현 교수님께도 깊은 감사를 드립니다. 또한 여러 가지 면에서 지도해주신 박홍복 교수님, 여정모 교수님, 김영봉 교수님, 박승섭 교수님, 박지환 교수님, 정순호 교수님, 박만곤 교수님께도 많은 감사를 드립니다.

늘 어려운 환경 속에서도 최선을 다하는 연구실 실장 황순환 선배님, 연구실의 흐름을 조절하는 박상일 선배님, 항상 변하지 않는 박현호 선배님, 조언으로 이끌어 주신 박월선 선배님, 다재한 김형욱 선배님, 그리고 용기를 주신 유은아 님, 마지막으로 연구실 막내 고봉현 님, 손문한 님께도 밝은 미래가 함께 하길 기원합니다. 이외에 연구실 교육대학원 송동영 님, 오윤주 님, 한동우 님, 박성련 님, 한지영 님, 정귀녀 님, 서정애 님, 이미나 님과 산업대학원 전홍태 님, 김양택 님, 이은화 님, 유종찬 님, 이경미 님, 천소영 님 등 저에게 여러 가지 어려운 상황에서도 용기를 주셨던 분들에게 많은 감사를 드립니다. 그리고 어려운 상황에서 항상 제 자리를 지켜주셨던 배은호 님, 김영만 님, 김남호 선배님들께도 특별한 감사를 전합니다. 그리고 항상 저와 함께 했던 백형구 선배님과 김대업 선배님, 김영하 선배님께 헤어짐의 아쉬움이 남으며, 저의 동기들인 이진호, 김정수, 이태훈, 김남진, 이영경, 김희준 학우가 항상 생각날 것입니다. 그리고 미처 적지 못한 많은 학우님들께도 고마운 마음을 전합니다.

저에게 항상 양보와 격려의 말씀을 아끼지 않았던 저의 가족들에게 부족한 저의 논문을 드립니다.

For the better tomorrow,

2004년 1월 1일

金 鎭 賢 올림