KSIAM IT series Vol.10, No.1, 107-119, 2006

데이터 베이스 특성에 따른 효율적인 데이터 마이닝 알고리즘

(An Efficient Data Mining Algorithm based on the Database Characteristics)

박지현, 고 찬
(Ji Hyun Park, Chan Koh)

요 약

인터넷과 웹 기술 발전에 따라 데이터베이스에 축적되는 자료의 양이 급속히 늘어나고 있다. 데이터베이스의 응용 범위가 확대되고 대용량 데이터베이스로부터 유용한 지식을 발견하고자 하는 데이터 마이닝(Data Mining) 기술에 대한 연구가 활발하게 진행되고 있다. 기존의 알고리즘들은 대부분 후보 항목 집합들을 줄임과 동시에 데이터베이스의 크기를 줄이는 방법으로 발전해 오고 있다. 그러나 후보 항목집합들을 줄이는 노력이나 데이터베이스의 크기를 줄이는 방법들이 빈발 항목집합들을 생성하는 전 과정에서 필요로 하지는 않는다. 그러한 방법들이 어느 과정에서는 시간을 줄이는데 효과가 있지만 다른 과정에서는 오히려 그러한 방법들을 적용하는데 더 많은 시간이 소요되기 때문이다.

본 논문에서는 트랜잭션들의 길이가 짧거나 데이터베이스를 이루는 항목들의 수가 비교적 적은 트랜잭션 데이터베이스에서 해쉬 기법을 사용하여 데이터베이스를 한 번 스캔하고 동시에 각 트랜잭션에서 발생 가능한 모든 부분집합들을 해쉬 테이블에 저장함으로써 최소 지지도에 영향을 받지 않고 기존의 알고리즘보다 더 짧은 시간에 빈발항목집합을 발견할 수 있는 효과적인 연관 규칙 탐사 알고리즘을 제안하고 실험하였다.

Abstract

Recently with developments of an internet and web techniques, the amount of data that are stored in database is increasing rapidly. So the range of adaption in database has been expanded and a research of Data Mining techniques finding useful skills from the huge database has been progressed. Many original algorithms have been developed by cutting down the item set and the size of database isn't required in the entire course of creating frequent item sets. Although those skills could save time in some course, it requires too much time for adapting those techniques in other courses.

In this paper, an algorithm is proposed. In an Transaction Database that the length of it's transactions are short or the number of items are relatively small,

키워드: 데이터 마이닝, 데이터베이스 특성, 연관 규칙 탐사

this algorithm scans a database once by using a Hashing Technique and at the same time, stores all parts of the set, can be appeared at each transaction, in an Hash-table. So without an influence of n minimum percentage of support, it can discover a set of frequent items in more shorter time than the time what is used by an original algorithm.

1. 서 론

최근 인터넷과 웹 기술이 발전됨에 따라 데이터베이스에 축적되는 자료의 양이 급속히 늘어나고 있다. 따라서 데이터베이스의 응용 범위가 확대되고 대용량 데이터베이스로부터 유용한 지식을 발견하고자 하는 데이터 마이닝(Data Mining) 기술에 대한 연구가 활발하게 진행되고 있다. 데이터 마이닝 기술은 연관 규칙(Association Rules), 클러스터링 / 세그멘테이션(Clustering / Segmentation), 분류 규칙(Classification Rules), 일반화 요약 규칙(Generalization & Summarization Rule), 순차적 패턴(Sequential Patterns), 특성화(characterization), 경향 분석(trend analysis) 등으로 나눌 수 있다. 특히 연관규칙 탐사는 그 형태의 명료함과 잠재적인 응용분야의 다양함으로 인해 많은 연구가 수행되고 있다[1, 2].

연관 규칙 탐사는 트랜잭션에 나타나는 항목의 발생 빈도수가 최소 지지도(Minimum Support)보다 큰 빈발 항목집합(Large frequent item sets)을 찾는 방법이다. 지금까지 빈발 항목집합을 찾기 위한 다양한 연관 규칙 탐사 알고리즘들이 제시되었다[2, 3, 4]. 기존의 연관 규칙 탐사 알고리즘들은 먼저 빈발 항목 집합들을 생성하기 위한 후보 항목집합들을 만들고 그 후보 항목집합들 중에서 빈발 항목 집합들을 찾아낸다. 그리고 한 번의 반복 시행에서 발견된 빈발 항목 집합들은 다음 반복과정에서 후보 항목집합들을 생성하기 위해 사용되고, 이 과정은 더 이상의 후보 항목집합들을 만들 수 없을 때까지 반복된다. 기존의 알고리즘들은 대부분 후보 항목 집합들을 줄임과 동시에 데이터베이스의 크기를 줄이는 방법으로 발전해 오고 있다. 그러나 후보 항목집합들을 줄이는 노력이나 데이터베이스의 크기를 줄이는 방법들이 빈발 항목집합들을 생성하는 전 과정에서 필요로 하지는 않는다. 그러한 방법들이 어느 과정에서는 시간을 줄이는데 효과가 있지만 다른 과정에서는 오히려 그러한 방법들을 적용하는데 더 많은 시간이 소요되기 때문이다. 물론 기존의 알고리즘들은 트랜잭션들의 길이가 길고 후보 항목들의 수가 많은 대용량 데이터베이스에서 많은 후보 항목집합들을 대상으로 수행되기 때문에 공간적인 문제를 고려해야만 했다. 그러나 각 트랜잭션들은 다양한 형태로 이루어진다. 즉, 적용하고자 하는 응용분야에 따라 트랜잭션들의 길이에 많은 차이가 있다. 예를 들면, 하나의 트랜잭션이 대형마트에서 하루 동안 판매되는 품목들이라고 가정하면, 많은 품목들과 다양한 소비자에 의해서 후보 항목 집합들의 수가 많아짐에 따라 트랜잭션의 길이는 길어진다. 하지만, 웹에서 정보를 얻고자 하는 사용자들이 그 정보가 저장되어 있는 사이트에 한 번 접근했을 때 참조되어지는 페이지들로 하나의 트랜잭션을 이룬다면 많은 페이지들을 접근하는 사용자도 있겠지만 대부분 사용자가 원하는 정보만을 탐색하기 때문에 트랜잭션의 길이는 상대적으로 짧아지게 될 것

이다. 또, 판매 트랜잭션에서도 고가의 품목들에 대해서는 하루에 판매되는 종류나 양에 있어서 대형마트와는 많은 차이를 보이므로 트랜잭션의 길이 또한 대형마트에 비해서는 더 짧아지게 될 것이고 적은 수의 후보 항목집합들이 발생할 것이다. 기존의 연관 규칙 탐사 알고리즘들의 성능평가에서 보듯이 트랜잭션의 길이, 최소 지지도 등에 따라 알고리즘들은 수행시간에 많은 차이를 보인다. 이와 같은 트랜잭션의 데이터베이스 집합에서는 시간을 줄이기 위해 후보 항목집합의 개수를 줄이거나 데이터베이스의 크기를 작게 하는 노력이 오히려 연관 규칙을 도출하는 과정에서 더 많은 시간을 소비하게 된다. 따라서 탐사하고자 하는 응용분야와 트랜잭션의 특징을 고려하여 연관 규칙 탐사를 할 필요가 있다.

현재 전자상거래나 증권거래와 같이 웹을 통해 사업을 하는 경영자들은 경영하는 웹사이트의 디자인이 고객의 만족도를 충족시켜주는 것도 중요하지만 더 중요한 것은 웹사이트를 방문하는 고객에게 양질의 정보를 소개하는 것과 고객에 대한 정보를 빠른 시간에 찾아내는 것이 더욱 더 중요하게 여겨지고 있다. 기존의 연관 규칙을 응용하는 예를 보면 "빵 → 우유"와 같은 규칙이 도출되었다고 했을 때 마트와 같은 대형 판매점에서는 상품 진열을 할 때 빵 옆에 우유를 진열하는 마케팅을 세울 수도 있고, 빵과 우유를 끝과 끝에 진열하는 마케팅을 세울 수도 있다. 전자는 빵과 우유를 옆에 진열하면서 우유를 세일한다면 빵과 우유의 매출 진작을 가져 올 수도 있으며, 후자는 빵과 우유를 끝과 끝에 진열해서 중간 진열대에 시리얼을 진열해 놓아 빵을 산 고객이 우유를 사러가면서 시리얼을 덤으로 사게 하는 그런 마케팅 전략을 세울 수 있다. 마찬가지로 전자상거래에서도 고객이 상품을 구매하는 경향을 빠른 시간에 찾아내어 고객의 눈에 빠르게 보여 질수 있는 것도 좋은 방법일 것이다. 하지만 이는 고객이 지루해하지 않는 시간 내에 찾아내야 하는 제약 조건이 있기 때문에 기존의 연관 규칙 탐사 알고리즘을 웹사이트에 적용하기는 어렵다.

본 논문에서는 트랜잭션들의 길이가 짧거나 데이터베이스를 이루는 항목들의 수가 비교적 적은 트랜잭션 데이터베이스에서 해쉬 기법을 사용하여 데이터베이스를 한 번 스캔하고 동시에 각 트랜잭션에서 발생 가능한 모든 부분집합들을 해쉬 테이블에 저장함으로써 최소 지지도에 영향을 받지 않고 기존의 알고리즘보다 더 짧은 시간에 빈발항목집합을 발견할 수 있는 효과적인 연관 규칙 탐사 알고리즘을 제안한다. 1장에 서론, 2장에 데이터 마이닝 알고리즘 검토, 3장에 효과적인 제안 알고리즘 설명, 4장에 실험의 내용, 5장에 실험 결과 분석을 설명하였고, 6장에 결론을 실었다.

2. 데이터 마이닝 알고리즘

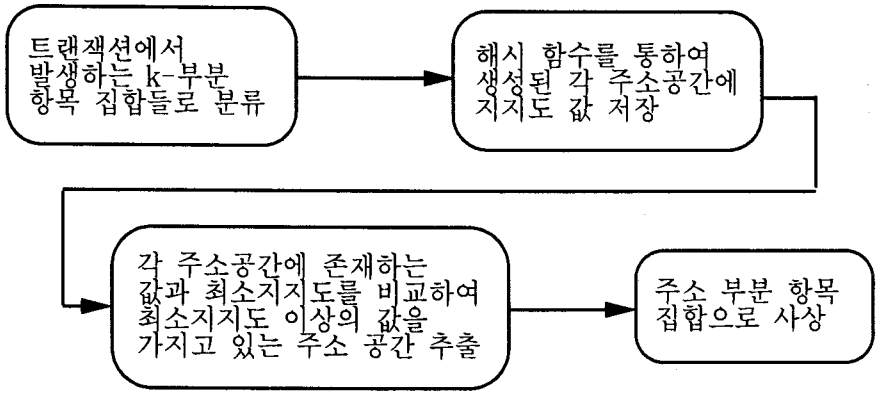
데이터 마이닝은 대량의 데이터로부터 쉽게 드러나지 않는 유용한 정보들을 추출하는 과정을 말한다. 여기서 정보는 묵시적이고 잘 알려져 있지 않지만 잠재적으로 활용가치가 있는 정보를 말한다. 다시 말해 데이터마이닝이란 기업이 보유하고 있는 일일 거래자료, 고객자료, 상품자료, 마케팅 활동의 피드백 자료와 기타 외부자료를 포함하여 사용가능한 데이터를 기반으로 숨겨진 지식, 기대하지 못했던 패턴, 새로운 법칙과 관계를 발견하고 이를 실제 경영의 의사결정 등을 위한 정보로 활용하고자 하는 것이다. 정보를 찾아내는 방법은 어떤 특정 기법과 그 기술 자체만을 의미하는 것이 아니고, 비즈니스 문제를 이해하고 이러한 문제를 해결하기 위하여 정보기술을 적용하는 포괄적인 과정을 의미한다. 즉 유

용한 정보의 추출을 위한 방법론이라고 할 수 있다. 따라서 데이터마이닝을 효율적으로 수행하기 위하여 시계열분석 등 각종 통계기법과 데이터베이스 기술뿐만 아니라 산업공학, 신경망, 인공지능, 전문가시스템, 퍼지논리, 패턴인식, 기계적 학습(Machine Learning), 불확실성 추론(Reasoning with Uncertainty), 정보검색에 이르기까지 각종 정보기술과 기법들을 사용하게 된다. 또한 경영전략, 마케팅 기법 등의 최신 경영기법들의 이용도 필요하다. 이러한 데이터마이닝을 통하여 거대한 데이터베이스에 숨어있는 전략적인 정보를 발견할 수 있으며 이러한 정보는 주요 비즈니스 프로세스 개선의 가장 원초적인 단계에서 사용될 수 있다. 데이터마이닝은 흔히 정보발견(Knowledge Discovery in Database)라고도 불리 우며 그 외에 지식추출(Knowledge Extraction), 정보추수(Information Harvesting), 정보고고학(Data Archeology), 자료패턴처리(Data Pattern Processing) 등으로도 불리 운다[5].

많이 사용되는 데이터 마이닝 알고리즘은 데이터 항목들 사이의 관련성을 규정하는 규칙을 개발하는 연관 규칙(Association rules), 비슷한 항목들을 통계적인 유사성에 의하여 묶는 과정인 클러스터링/세그멘테이션(Clustering/Segmentation), 새로운 레코드를 이미 편성된 집단에 할당하는 분류 규칙(Classification rules), 데이터베이스에서 많은 관련된 데이터를 낮은 개념 레벨에서 높은 개념 레벨로 추상화시키는 작업인 일반화/요약 규칙(Generalization & Summarization rules), 트렌드를 식별해 내기 위해 일정시간 동안의 레코드를 분석하여 순서 패턴을 찾아내는 순차적 패턴(Sequential Patterns), 이러한 지식추출 작업에 자주 사용되는 기법에는 신경망, 결정트리 등을 포함한 기법들이 많이 쓰이고 있다. 예를 들면, 노트북 컴퓨터 구매자의 80%는 3개월 이내에 배터리를 구입한다, 지난 3 개월 동안 25% 이상 셀룰러 통화량이 감소한 고객이 이 서비스의 구매를 중단할 가능성은 60% 정도임, 등이다.

3. 제안 데이터 마이닝 알고리즘 "EDHPA"

해쉉 기법을 이용하여 빈발 항목 집합을 생성하는 DHP 알고리즘을 리모델링 하여 한 번의 데이터베이스 스캔으로 각 트랜잭션에서 발생 가능한 모든 k-부분집합들의 지지도를 계산함으로써 빈발 항목집합들을 생성하는 EDHPA(Efficient DHP Algorithm) 알고리즘을 제안한다. EDHPA 알고리즘은 비교적 길이가 짧은 트랜잭션 데이터베이스를 대상으로 전체 트랜잭션을 이루는 모든 부분 항목집합들을 버켓이라는 기억 공간 속에 저장하는 알고리즘이다. EDHPA 알고리즘은 트랜잭션들의 길이가 짧거나 데이터베이스를 이루는 항목들의 수가 비교적 적은 트랜잭션 데이터베이스에서 해쉉 기법을 사용하여 기존의 알고리즘보다 더 짧은 시간에 빈발항목집합을 발견할 수 있는 효과적인 연관 규칙 탐사 알고리즘이다. EDHPA 알고리즘에서의 빈발항목집합을 찾아내는 과정은 [그림 3-1]과 같다.



〔그림 3-1〕 빈발 항목을 찾아내는 과정

기존의 알고리즘에서는 1-항목 부분집합들에 대한 지지도를 먼저 확인하고, 1-빈발 항목집합들이 될 수 있는지의 여부를 확인한다. 다시, 1-빈발 항목 집합들은 기존의 후보 항목집합들을 생성하기 위한 알고리즘에 의해서 2-후보 항목집합들이 생성되고 이는 2-빈발 항목집합을 찾기 위해 사용된다. 이러한 순차적인 과정이 기존의 대부분의 알고리즘들에서는 사용되지만 EDHPA 알고리즘에서는 이러한 반복과정을 거치지 않고 데이터베이스를 한 번 스캔한 후에 모든 부분 항목집합들의 지지도가 계산되기 때문에 매 단계마다 확인을 할 필요는 없다.

연관 규칙 탐사에서 1-부분집합은 기존의 알고리즘들에서 보듯이 다음단계의 후보를 생성하기 위해서는 필요하지만, 항목들의 연관성을 찾는 탐사에서 1-빈발 항목집합은 의미가 없다. 왜냐하면, 하나의 항목에 대한 연관성은 존재하지 않기 때문이다.

〔표 3-1〕에서 보는바와 같이 분류된 각 항목집합들은 해쉬 함수를 통하여 생성된 각 주소 공간, 즉 버켓에 부분 항목집합들이 아닌 지지도 값이 저장된다. 데이터베이스를 이루는 각 트랜잭션에서 발생 가능한 모든 부분 항목집합들은 해쉬 함수로 인해서 그들만의 유일한 주소를 가지게 되고 그 주소에는 같은 부분 항목집합들이 존재할 때마다 1씩 증가하게 된다. 결국 모든 부분 항목집합들에 대해서 수행이 완료되면 모든 트랜잭션에 대한 부분 항목집합들의 주소가 생성되고 이들 주소에는 결국 부분 항목집합들의 지지도가 저장되게 된다. 두 번째는 각 베켓에 존재하는 값과 최소 지지도를 비교하여 최소 지지도 이상의 값을 가지고 있는 버켓들을 추출하고 주소에 해당하는 부분 항목집합으로 사상 시켜주면 결국 우리가 원하는 빈발 항목집합들을 찾게 된다. EDHPA 알고리즘은 〔그림 3-2〕과 같다.

```
D : Database
H : Hash table
k : 1 < k ≤ length of transaction
sup = minimum support;
set all the buckets of Hash table(H) to zero;
forall transaction t ∈ D do begin
   scan t to D;
   forall k-subsets x of t do begin
      if (H ∋ x)  Hk.support++;
      else {
         Hk.support = 1;
         Hk.candidate = x;
      }
      endif
   end
end

forall Hk do begin
   if({x|Hk[x]≥sup}) then
      L = {x|x is a k-large itemset};
end
```

[그림 3-2] EDHPA 알고리즘

3.1. EDHPA 알고리즘의 실행 예

EDHPA 알고리즘의 실행 예를 설명한다. 예에 사용될 트랜잭션과 항목들은 앞에서 사용한 예제인 [표 2-1]과 같다. 이 예제에서 최소 지지도는 2로 설정하여 단 한 번의 데이터베이스 스캔으로 연관 규칙을 탐사하고자 한다.

〈표 3-1〉 트랜잭션 데이터베이스와 트랜잭션 분류

TID	1-Itemsets	2-Itemsets	3-Itemsets	4-Itemsets
100	{A}, {C}, {D}	{A, C}, {A, D}, {C, D}	{A, C, D}	
200	{B}, {C}, {E}	{B, C}, {B, E}, {C, E}	{B, C, E}	
300	{A}, {B}, {C}, {E}	{A, B}, {A, C}, {A, E}, {B, C}, {B, E}, {C, E}	{A, B, C}, {A, B, E}, {A, C, E}, {B, C, E}	{A, B, C, E}
400	{B}, {E}	{B, E}		

[표 3-1]은 트랜잭션을 k-부분집합별로 분류한 예이다. 먼저 TID가 100인 트랜잭션 A, C, D를 읽는다. 트랜잭션의 길이가 3이므로 1-항목 부분집합, 2-항목 부분집합, 3-항

목 부분집합이 생성된다. 각 항목 부분집합들을 생성하는 것은 기존의 알고리즘에서 사용하던 방법과 동일하다. 즉 A, C, D를 조인하여 2-항목 부분집합인 {A, C}, {A, D}, {C, D}가 생성되고 3-항목 부분집합인 {A, C, D}가 생성된다.

〈표 3-2〉 TID가 100인 트랜잭션을 읽은 후

k-Itemsets	{A, C}	{A, D}	{C, D}	{A, C, D}
Support	1	1	1	1
Address	13	14	34	134

〈표 3-3〉 TID가 200인 트랜잭션을 읽은 후

{A, C}	{A, D}	{B, C}	{B, E}	{C, D}	{C, E}	{A, C, D}	{B, C, E}
1	1	1	1	1	1	1	1
13	14	23	25	34	35	134	235

〈표 3-4〉 TID가 300인 트랜잭션을 읽은 후

{A,B}	{A,C}	{A,D}	{A,E}	{B,C}	{B,E}	{C,D}	{C,E}	{A,B, C}	{A,C, D}	{A,C, E}	{B,C, E}	{A,B, C, E}
1	2	1	1	2	2	1	2	1	1	1	2	1
12	13	14	15	23	25	34	35	123	134	135	235	1235

〈표 3-5〉 TID가 400인 트랜잭션을 읽은 후

{A,B}	{A,C}	{A,D}	{A,E}	{B,C}	{B,E}	{C,D}	{C,E}	{A,B, C}	{A,C, D}	{A,C, E}	{B,C, E}	{A,B, C, E}
1	2	1	1	2	3	1	2	1	1	1	2	1
12	13	14	15	23	25	34	35	123	134	135	235	1235

〈표 3-6〉 모든 트랜잭션의 수행을 끝낸 후

{A, C}	{B, C}	{B, E}	{C, E}	{B, C, E}
2	2	3	2	2
13	23	25	35	235

[표 3-2]부터 [표 3-5]까지는 각 트랜잭션을 스캔하고 하나의 트랜잭션에서 발생 가능한 모든 부분 항목집합들이 해쉬 함수를 통하여 존재하지 않는 주소 공간은 새로이 1이라는 값을 가지면서 생성되고 만약 존재하면 그 주소 공간의 값을 1씩 증가시키는 과정을 보여준다. 표들에서 알 수 있듯이 먼저, TID가 100인 트랜잭션을 분류하고 분류된 {A, C}, {A, D}, {C, D}, {A, C, D}는 해당하는 주소 공간 13, 14, 34, 134번지의 값에 1이

기억된다. 그 다음 200인 트랜잭션을 분류한 후, 해당하는 주소 공간이 존재하면 그 주소 공간의 값에 1을 증가시키고 해당하는 주소 공간이 없으면 새로운 주소 공간을 생성하고 초기 값을 1로 설정한다. 같은 방법으로 300인 트랜잭션과 400인 트랜잭션을 수행하고 나면 위 [표 3-2]에서 [표 3-5]와 같은 결과를 얻을 수 있다.

기존의 알고리즘에서는 1-항목 부분집합들에 대한 지지도를 먼저 확인하고, 1-빈발 항목집합들이 될 수 있는지의 여부를 확인한다. 다시, 1-빈발 항목 집합들은 기존의 후보 항목집합들을 생성하기 위한 알고리즘에 의해서 2-후보 항목집합들이 생성되고 이는 2-빈발 항목집합을 찾기 위해 사용된다. 이러한 순차적인 과정이 기존의 대부분의 알고리즘들에서는 사용되지만 EDHPA 알고리즘에서는 이러한 반복과정을 거치지 않고 데이터베이스를 한 번 스캔한 후에 모든 부분 항목집합들의 지지도가 계산되기 때문에 매 단계마다 확인을 할 필요는 없다. 연관 규칙 탐사에서 1-부분집합은 기존의 알고리즘들에서 보듯이 다음단계의 후보를 생성하기 위해서는 필요하지만, 항목들의 연관성을 찾는 탐사에서 1-빈발 항목집합은 의미가 없다. 왜냐하면, 하나의 항목에 대한 연관성은 존재하지 않기 때문이다.

[표 3-1]에서 보는바와 같이 분류된 각 항목집합들은 해쉬 함수를 통하여 생성된 각 주소 공간, 즉 버켓,에 부분 항목집합들이 아닌 지지도 값이 저장된다. 데이터베이스를 이루는 각 트랜잭션에서 발생 가능한 모든 부분 항목집합들은 해쉬 함수로 인해서 그들만의 유일한 주소를 가지게 되고 그 주소에는 같은 부분 항목집합들이 존재할 때마다 1씩 증가하게 된다. 결국 모든 부분 항목집합들에 대해서 수행이 완료되면 모든 트랜잭션에 대한 부분 항목집합들의 주소가 생성되고 이들 주소에는 결국 부분 항목집합들의 지지도가 저장되게 된다. 두 번째는 각 베켓에 존재하는 값과 최소 지지도를 비교하여 최소 지지도 이상의 값을 가지고 있는 버켓들을 추출하고 주소에 해당하는 부분 항목집합으로 사상 시켜주면 결국 우리가 원하는 빈발 항목집합들을 찾게 된다.

4. 실험 및 방법

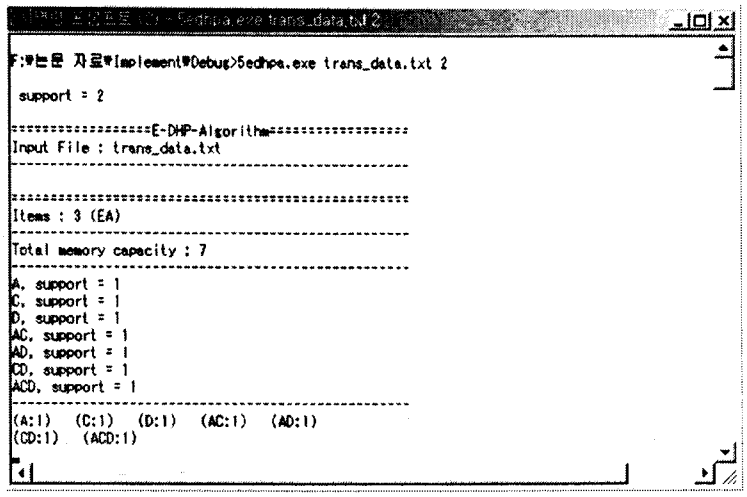
EDHPA 알고리즘을 기존의 연관 규칙 탐사 방법인 Apriori 알고리즘, DHP 알고리즘 등과 비교하여 평가한다. 제안된 알고리즘의 실행 결과를 보여주고 각 알고리즘과의 성능 평가를 하였다.

4.1. 실험 방법

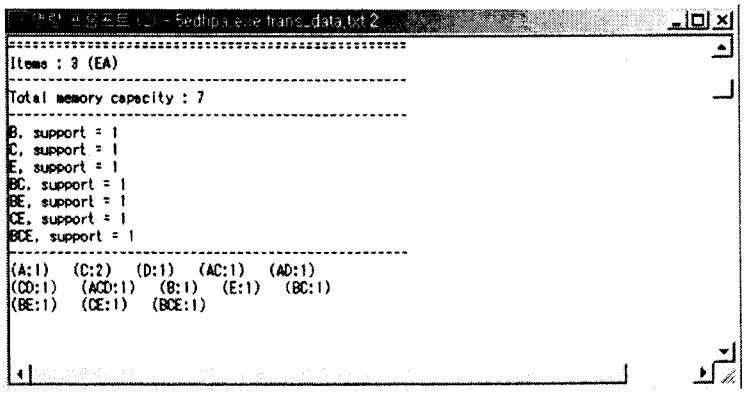
평가 환경은 Pentium III 700, 384MB, Windows 2000 Server, Visual C++ 환경에서 수행하였고 아이템의 수는 5개이고, 그 아이템을 랜덤으로 1,000개를 발생시켜 각 알고리즘의 성능평가를 수행하였다. 각 시행 결과는 2회 이상 반복 수행한 후의 결과이다. 각 알고리즘의 성능 평가를 위하여 트랜잭션 10라인당 수행 속도를 측정하여 기존의 알고리즘과 비교하였다.

4.2. EDHPA 알고리즘의 실행 결과

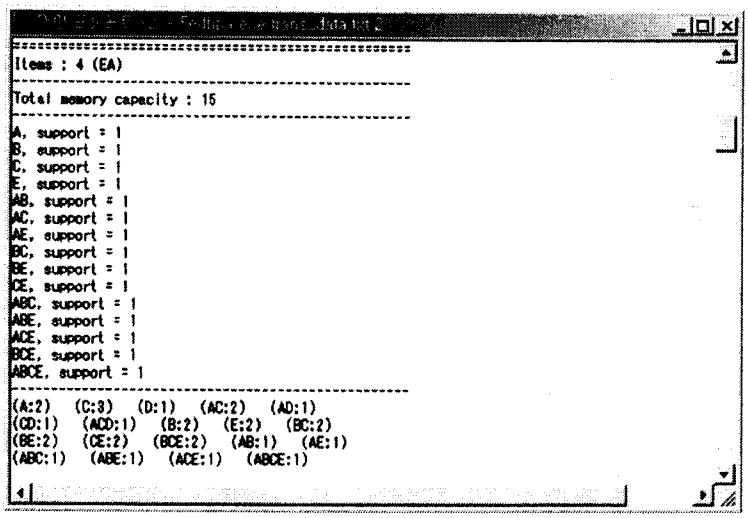
EDHPA 알고리즘에서의 실행은 트랜잭션과 지지도를 입력 변수로 받아 처리하였다. 다음 그림들은 EDHPA 알고리즘에서 각 트랜잭션을 읽은 후의 실행한 결과를 보여준다.



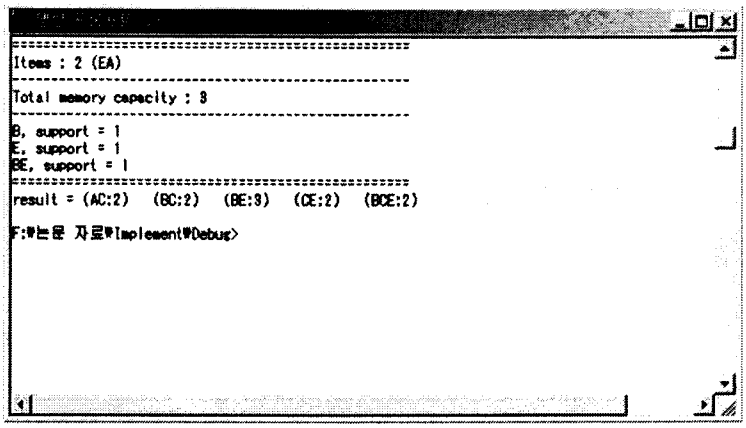
〔그림 4-1〕 첫 번째 트랜잭션을 읽은 후의 실행 결과



〔그림 4-2〕 두 번째 트랜잭션을 읽은 후



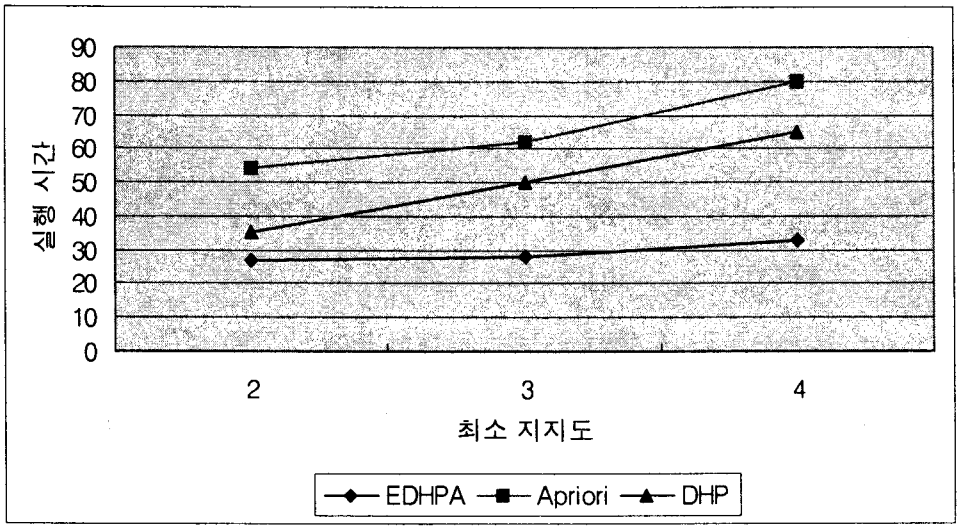
〔그림 4-3〕 세 번째 트랜잭션을 읽은 후



〔그림 4-4〕 네 번째 트랜잭션을 읽은 후

5. 실험 결과 분석

기존 알고리즘과 본 논문에서 제안하는 EDHPA 알고리즘과의 수행 성능을 비교하여 보여준다. 다음 〔그림 4-5〕는 주어진 데이터의 트랜잭션들의 개수가 1,000일 때 최소 지지도를 2, 3, 4로 변화시키면서 실험을 한 결과이다.



〔그림 4-5〕 실험결과

위의 그림에서 보듯이 Apriori나 DHP는 1-itemset이나 2-itemset을 찾을 때마다 데이터베이스를 스캔해야 하지만, EDHPA는 한번의 데이터베이스 스캔과 해싱 기법을 사용하기 때문에 실행 시간이 최소 지지도에 대한 영향을 받지 않는 것을 볼 수 있다.

6. 결 론

본 연구에서는 트랜잭션들의 길이가 짧거나 데이터베이스를 이루는 항목들의 수가 비교적 적은 트랜잭션 데이터베이스에 대해서 트랜잭션을 이루는 항목집합들 간의 연관 규칙 탐사문제를 다루었다. 빈발 항목 집합들을 발견하는 문제는 먼저 항목 집합들의 후보 항목집합들을 생성하고 이 후보 항목집합들 내에서 빈발 항목집합들의 요구 조건을 만족시키는 항목집합들을 규명함으로써 해결할 수 있다. 기존의 알고리즘들은 데이터베이스를 이루는 항목들의 수가 많고 트랜잭션들의 길이가 긴 경우에 적용되었기 때문에 최소 지지도의 변화에 따라 생성되는 후보 항목집합들의 개수가 많은 차이를 나타내고 이는 곧 알고리즘의 수행시간에 큰 영향을 미친다. 또한, 알고리즘의 성능을 개선하기 위해서 후보 항목집합들의 개수를 줄이거나 데이터베이스의 크기를 줄이는데 많은 노력들을 해 왔지만 그러한 노력들이 빈발 항목집합들을 생성하는 전 과정에서 효과적이지는 않는다. 후보 항목집합들의 개수가 적을 때에는 오히려 시간을 절약하기 위한 노력에 더 많은 시간을 소비하는 현상이 발생하기도 한다.

현재 인터넷이 다양한 사용자에게 전파됨에 따라 많은 응용영역들이 생겨나고 있다. 이에 편승해서 데이터베이스의 크기나 구조가 다양해지고 하드웨어의 기술 또한 빠른 속도로 발전하고 있다. 본 연구에서는 트랜잭션 데이터베이스를 이루는 각 트랜잭션의 길이가 비교적 짧은 경우와 전체 데이터베이스를 이루는 항목의 수가 적은 경우에 해쉬 기법을 사용하여 효과적으로 빈발 항목집합을 생성하는 알고리즘을 제안하였고, 모의실험을 통하여 기

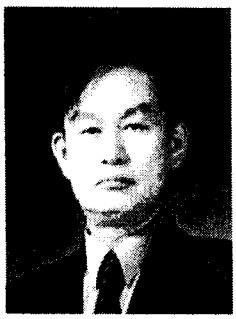
하여 효과적으로 빈발 항목집합을 생성하는 알고리즘을 제안하였고, 모의실험을 통하여 기존의 알고리즘을 적용했을 때의 비효율성을 증명하였다.

본 연구에서 제안하는 알고리즘은 전자상거래와 같은 웹 응용에서 필요한 정보를 추출하고 분석하여 사용자들에게 빠르게 대처해야 하는 경우에 효과적으로 이용될 수 있다. 향후 후보 항목 생성 시 소요되는 많은 기억 공간을 절약하기 위한 알고리즘에 대한 연구가 필요하다.

참 고 문 헌

1. Peiter Adriaans and Dolf Zantinge, "Data Mining" Addition Wesley, 1996.
2. R. Agrawal, T. Imielinski, A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proc. of ACM SIGMOD Conf, Washington D.C., pp207-216, 1993.
3. R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules", In Proceedings of 20th VLDB Conference Santiago, Chile, Sept., 1994.
4. J.S. Park, M.-S. Chen, an d P.S. Yu, "An effective hash-based algorithm for mining association rules", In Proceedings of ACM SIGM OD Conference on Management of Data, pp. 175-186, San Jose, California, May, 1995.
5. 데이터 마이닝 http://honey.gwangju.ac.kr/%7Esoft/part/study/datamining.htm.
6. 네이버 백과사전 http://www.naver.com/.
7. 민창우, "데이터 마이닝을 위한 간결한 퍼지 규칙 생성 알고리즘", 숭실대학교 대학원 전자계산학과 석사학위논문, 1998.
8. 김신곤, "데이터 마이닝과 지식발견", 광운대학교 경영정보학과, 2000.
9. 이미희, "데이터 마이닝 기법의 소개", 창신대학 정보통신과 전임강사, 2000.
10. 송임영, " 빈발 항목과 의미있는 희소 항목을 포함한 이미지 데이터 마이닝", 서울산업대학교 대학원 컴퓨터공학과 석사학위논문, pp. 5-14, 2003.
11. 통계 데이터베이스 강의 자료 http://www.webcai.pe.kr/statdb.
12. Jiawei Han, Micheline Kamber, "Data Mining : Concepts and Techniques" Morgan Kaufmann Publishers, 2001.

박 지 현(Ji Hyun Park)
· ciesta75@hotmail.com
· 1994년 3월 광운대학교 수학과 입학
· 1995년 9월 광운대학교 수학과 자퇴
· 1996년 3월 서울산업대학교 전자계산학과 입학
· 2000년 2월 서울산업대학교 전자계산학과 졸업
· 2002년 8월 서울산업대학교 대학원 컴퓨터공학과 입학
· 2005년 2월 서울산업대학교 대학원 컴퓨터공학과 졸업(공학석사)
· 1999년 3월~12월 노원구청 컴퓨터 교실 강사
· 2000년 5월 카고피아 웹 및 네트워크 관리부 입사
· 2001년 5월 카고피아 퇴사
· 2004년 4월~12월 두원공과 대학 소프트웨어개발과 시간강사
· 2005년 1월~현재 (주)더모스트 시스템사업부 재직중
· 관심분야 : 데이터베이스, 데이터마이닝, 웹 마이닝, 정보검색



고 찬(Chan Koh)
· e-mail : chankoh@snut.ac.kr
· 2002. 2. 서울대학교 경제학 박사수료(기술경제 및 정책)
· 1991. 2. 경희대학교 공학박사(전자공학)
· 1983. 8. 연세대학교 공학석사(전산학)
· 1974. 2. 경희대학교 공학사(기계공학)
· 1992.~1993. 미국 North Carolina State University, Post Doc. (1년)
· 2002. 6.~2002. 8. 미국 Carnegie Mellon 대학, Advanced Software Eng. 과정이수.
· 2004. 12.~2005. 2. 미국 Carnegie Mellon 대학, Game Production 과정이수.
· 1974. 10.~1078. 2. 해군장교 복무(해군대학, 컴퓨터 전쟁게임 교관)
· 1978.12 정보처리기술사 (과학기술부), 1999. 8 공인정보시스템 감리인(한국전산원)
· 2006. 8.~2006. 8. 필란드 Evtek 대학교 IT학과 초빙강의교수(게임프로그래밍 강의).
· 2006. 6.~2006. 8. 연세대학교 국제교육교류원 국제교환학생 강의(경영정보학).
· 2005. 12.~2006. 1. 필란드 Evtek 대학교 IT학과 초빙강의교수(게임프로그래밍 강의).
· 2004. 6.~2004. 8. 미국 뉴욕시립대 컴퓨터과학과 초빙강의교수(경영정보학 강의).
· 1983. 10.~현재 : 서울산업대학교 컴퓨터공학과 교수, 어학원장
· 2006. 현재 : 한국디지털정책학회 부회장, 한국정보통신기술사협회 부회장, 행정자치부 지방행정혁신브랜드사업 지도자문위원, 여성부 정보화평가위원, 문화관광부 2010 게임산업 전략위원회 해외협력분과위원, 행정자치부 지방행정 혁신 평가위원, 행정자치부 지방행정브랜드 사업 지도위원, 한국산업응용수학회 이사
· 관심분야 : 경영정보학, 기술경제 및 정책학, 그래픽스/게임엔진, 소프트웨어공학