제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

데이터집합 특성에 기반한 데이터 마이닝 전처리 대수 연산자

김효숙, 이원석
연세대학교 컴퓨터산업시스템공학과
e-mail : {hskimd, leewo}@amadeus.yonsei.ac.kr}

Dataset Property – based Algebraic Operators for Data Mining Preprocessing

Hyo-Sook kim, Won-Suk Lee
* Dept. of Computer Science, Yonsei University

요 약

지식 탐사 연구의 핵심이 되어온 데이터 마이닝은 축적 데이터로부터 쉽게 추출되지 않는 데이터 상호관계나 일정 패턴과 같은 유용한 내재 정보 추출을 주된 목적으로 수행된다. 그러나, 데이터 마이닝은 대용량의 데이터 처리로 인해 빈번한 메모리 공간 제약과 처리 속도 저하 등의 한계성을 드러낸다. 이를 극복하기 위해 많은 마이닝 알고리즘 개발과 기존 알고리즘 개선 방법이 제시되어 왔으나 여전히 궁극적인 해결방안은 대두되지 않고 있다. 따라서, 만약 데이터 전처리 과정을 통해 마이닝 목적에 적합한 부분 데이터집합 추출 및 가공이 선행된다면 보다 효율적인 데이터 마이닝 작업을 유도할 수 있을 것이다. 본 논문은 효과적 데이터 전처리를 위한 필수 기본 연산 기능들을 주어진 데이터집합의 트랜잭션 및 데이터 특성에 기초하여 관계형 대수 형태로 의미를 정립하고, 적용 사례에 의한 상세 설명 및 실제 구현된 *온라인 데이터 전처리 시스템*을 제안한다.

1. 서론

지식 탐사(KDD : Knowledge Discovery in Databases)란 다량의 축적된 데이터로부터 이전에 알려지지 않았던 유효하고 잠재적인 유용정보를 분석이 용이한 형태로 최종사용자에게 제공하는 일련의 과정으로써 각 단계는 상호보완적 반복이 가능하다[1]. 특히 데이터 마이닝은 데이터간의 관계 분석을 위한 연관 규칙[2], 시간 변화에 따른 데이터 관계 분석을 위한 순차 패턴[3], 유사도 기반의 데이터 군집화를 위한 클러스터링[4] 등과 같은 마이닝 기법을 적용하여 데이터의 일정 패턴이나 모델을 산출하기 위한 지식 탐사의 한 단계로써 기존 연구의 핵심이 되어왔다. 그러나, 분석하고자 하는 데이터의 양이 방대해짐에 따라, 데이터 마이닝은 메모리 공간과 처리시간의 한계성을 드러내게 되었다. 이를 해소하기 위해 지지도, 신뢰도, 유사도 등과 같은 마이닝 조건 설정의 최적화를 위한 많은 알고리즘 연구가 이루어져 왔으나 여전히 마이닝 작업을 본래의 축적 데이터로부터 시작하기 때문에 만족스러운 근본 해결 방안을 제시하지는 못하고 있다. 또한, 기존의 데이터 마이닝 접근방식은 최적화된 마이닝 조건 설정을 위한 데이터 참조 정보 제공이 결여되었을 뿐만 아니라 사용자에 의한 마이닝 과정의 자유로운 통제가 불가능하기 때문에 실제 축적 데이터 크기에 상응하는 결과물을 생성하거나 마이닝 작업의 목적과 무관한 다수의 불필요한 정보를 도출하기도 한다. 이에 반해, 본래의 데이터로부터 마이닝 목적에 부합되고 적합한 형태의 데이터 추출을 수행함으로써 마이닝 과정의 부하를 경감시키고 효율성을 증대시킬 수 있는 데이터 전처리 과정은 상대적으로 데이터 마이닝보다 기존 연구의 관심에서 많이 소외되어 왔다.

본 논문에서는 관계형 대수 형태로 의미론적 기반을 정립한 필수 데이터 전처리 대수 연산자들을 기존 데이터 마이닝의 문제점에 대한 효과적 대안으로 제안하고, 적용 사례를 통한 구체적 기능 설명과 실제 구현 시스템을 함께 제시한다.

본 논문의 구성으로 2 절에서는 관련 연구에 대해 기술하고, 3 절에서는 각 데이터 전처리 대수 연산자를 정의한다. 4 절에서는 적용 사례의 설명과 구현된 *온라인 데이터 전처리 시스템* 모델을 소개하고 마지막으로 5 절에서는 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

본 절에서는 각 분야별로 해당 목적을 보다 효율적으로 수행하기 위해 요구되는 필수 기능들을 일정한 연산자 형태로 정의 하려는 기존 연구의 몇 가지 관련 사례를 소개한다.

비디오 대수 연산자들(Video Algebra Operators)[5]은 상위 레벨 의미 묘사를 가지는 *비디오 표현식들(Video Expressions)*의 구축을 위해 사용되고, 이들 비디오 표현식들의 계층적 결합은 다시 *대수적 비디오 데이터 모델*을 구성하게 된다. 비디오 대수 연산자들은 효율적 디지털 비디오 처리 및 접근을 위해 요구되는 기본 기능들을 비형식적 구문을 통해 *생성, 합성, 출력, 설명*의 4 가지 큰 카테고리로 분류하여 정의하고 있다.

***다차원 연산자들*(Multidimensional Operators)**[6]은 다차원 데이터베이스에 대한 의미적 기반을 제공하며 실시간 온라인 다차원 분석(OLAP)을 지원하기 위한 대수 연산자를 나타낸다. 관계형 데이터베이스나 다차원 데이터베이스에서의 구현과 SQL 로의 변환이 가능하도록 고안된 최소한의 다차원 연산자들은 *푸시, 풀, 제거, 제한, 조인, 결합* 등의 대수 연산 기능들을 제공한다.

DMQL(Data Mining Query Language) [7]은 데이터 마이닝의 4 가지 주된 기본적 요소를 선언적 명세 사항들로 기술하여 *연관 규칙* 이외에도 *데이터 일반화, 특성 규칙, 판별 규칙, 분류 규칙* 등과 같은 다양한 규칙들의 추출이 가능하도록 고안된 데이터 마이닝 질의 언어이다.

MINE RULE[8]은 연관규칙 탐사와 관련된 모든 문제를 SQL 구문과 유사한 일관된 형태로 묘사할 수 있도록 고안된 연산자이다. MINE RULE 연산자에 의한 연관규칙 탐사 과정은 추가적 기능이 포함된 확장 관계형 대수의 일정 형식에 따른 절차적 구문으로 그 의미가 묘사되고 있다.

MSQL[9]는 *규칙 질의 언어*로써 데이터로부터의 규칙을 생성하는 *데이터 마이닝 연산자*뿐만 아니라 이미 생성된 규칙들에 대한 질의를 위한 *후처리 연산자*도 소개하고 있다. 또한, 데이터베이스 내의 연속 속성값에 대한 이산화를 위한 *복호화 연산자* 및 데이터와 규칙간의 관계 검증을 위한 *교차 검증 연산자*도 함께 제시되고 있다.

지금까지 지식 탐사 과정을 지원하기 위해 소개된 DMQL, MINE RULE, MSQL 은 모두 관계형 데이터베이스를 기반으로 한 연산 언어들로써 기존의 SQL 구문과 유사한 형태로 정의되고 있다. 비록 DMQL 이 다양한 지식 유형의 탐사를 시도하고, MSQL 이 부분적인 전처리 기능과 후처리 기능을 제안하고 있지만 이들은 모두 연관 규칙 탐색을 위한 데이터 마이닝 연산 기능에만 주로 초점을 맞추고 있다.

3. 데이터 전처리 대수 연산자 정의

본 논문의 기본적인 입력 데이터집합은 각 데이터 항목들로 구성된 트랜잭션들의 집합을 의미하며 트랜잭션 시간과 같은 추가적 정보가 포함될 수 있다. 단, 연관 규칙 탐색을 위한 데이터집합인 경우는 각 트랜잭션 내에 중복 데이터 항목들이 존재하지 않으나 순차패턴의 경우는 중복 데이터 항목의 존재가 가능하다. 또한, 클러스터링을 위한 데이터집합의 각 트랜잭션들은 중복이 허용되지 않는 수치 데이터 항목으로 구성됨을 전제로 한다.

3.1 데이터 전처리 대수 연산의 전체 구문의 구성

데이터 전처리 대수 연산자는 확장된 BNF 구문 형식을 통해 다음 표기법에 의해 정의된다.

[표 1] 구문표기를 위한 메타기호의 의미

기 호	의 미	기 호	의 미
::=	좌측을 우측과 같이 정의	|	"또는" 의미
←	우측 결과를 좌측에 할당	[]	선택적 구문
< >	비종결형 카테고리 규칙	{}	반복 구문
		()	입출력 항목

<Data_Preprocess_Operation> ::=
<Transaction_Selection> | <Data_Projection>
| <Dataset_Item_Rename> | <Dataset_Union>
| <Dataset_Intersection > | <Dataset_Difference>
| <Aggregate_Function> | <Logical _Not>
| <Logical_Or> | <Logical_And >

3.2 데이터 전처리 단항 대수 연산

3.2.1 트랜잭션 선택 연산

*트랜잭션 선택 연산*은 한 데이터집합으로부터 *수평적 탐색*을 통해 주어진 선택 조건을 만족하는 트랜잭션들만을 선택하는 기능을 수행한다.

<Transaction_Selection> ::=
[(OTDB)←]σ <Selection_Condition>(ITDB)

<Selection_Condition> ::=
<Selection_Criteria><Comparison_Op> <Constant_Value>
|<Selection_Criteria><Comparison_Op><Aggregate_Operation>
|<Selection_Criteria><Equality_Op><Data_Item_List>
|[<Selection_Condition>]{[<Boolean_Op><Selection_Condition>]}
<Selection_Criteria> ::= any transaction properties.
<Comparison_Op> ::= <Relational_Op> | <Equality_Op>
<Relational_Op> ::= < | > | ≤ | ≥
<Equality_Op> ::= ≠ | =
<Boolean_Op> ::= ¬ | ∧ | ∨
<Constant_Value> ::= a numeric value from the domain of Selection_Criteria
<Data_Item_List>::= <Data_Item>| <Data_Item_List><Data_Item>
<Data_Item> ::= a string or a numeric value from the domain of Selection_Criteria

선택 기준으로는 *트랜잭션 길이, 트랜잭션 시간, 트랜잭션 필수 포함 데이터 항목, 트랜잭션 시간 카테고*

리, 데이터 항목 중복수 등과 같은 *임의의 모든 트랜잭션 선택 속성*이 그 대상이 될 수 있다. 또한, 선택 기준에 대한 평균, 합계, 최소값, 최대값 등과 같은 집합함수연산의 적용에 의한 더욱 정교한 조건 설정이 가능하다. 하나 이상의 선택 조건은 논리곱(∧)이나 논리합(∨)연산자에 의해 조합이 가능하고, 논리 부정 연산자(¬)에 의해 주어진 선택 조건과 상반된 트랜잭션 선출도 또한 가능하다.

3.2.2 데이터 프로젝션 연산

*데이터 프로젝션 연산*은 한 데이터집합으로부터 *수직적 검색*을 통해 주어진 프로젝션 조건을 만족하는 특정 데이터 항목들만을 추출하는 기능을 수행한다.

<Data_Projection>::= [(OTDB)←]$\pi_{<Projection\_Condition>}$(ITDB)

<Projection_Condition> ::=
 <Projection_Criteria><Comparison_Op><Constant_Value>
| <Projection_Criteria><Comparison_Op><Aggregate_Operation>
| <Projection_Criteria><Equality_Op><Data_Item_List>
[[<Projection_Criteria>]{[<Boolean_Op><Projection_Condition>]}
<Projection_Criteria> ::= any data properties
<Constant_Value> ::= a numeric value from the domain of Projection_Criteria
<Data_Item_List> ::= <Data_Item>
| <Data_Item_List><Data_Item>
<Data_Item> ::= a string or a numeric value from the domain of Projection_Criteria

프로젝션 기준으로는 *특정 데이터 항목 목록, 데이터 항목 카테고리, 클러스터링 수치 데이터 값* 등과 같은 *임의의 모든 데이터 항목 추출 속성*이 대상이 될 수 있다. 또한, 논리곱 또는 논리합의 연산자에 의한 복합 프로젝션 조건설정과 논리 부정 연산자에 의해 주어진 프로젝션 조건을 만족하는 데이터 항목들을 제외한 나머지 데이터 항목 추출도 가능하다.

3.2.3 명칭 재설정 연산

*명칭 재설정 연산*은 한 데이터집합으로부터 재명명 조건에 설정된 데이터 항목 목록의 각 항목별 명칭 변경이나 주어진 데이터집합의 명칭을 변경하는 기능을 수행한다.

<Dataset_Item_Rename> ::=
[(OTDB)←]$\rho_{<Rename\_Condition>}$(ITDB)

<Rename_Condition> ::=
<Old_Data_Item_List> = <New_Data_Item_List>
<Old_Data_Item_List> ::= NULL
| <Data_Item>
| <Old_Data_Item_List ><Data_Item>
<New_Data_Item_List> ::= NULL
| <Data_Item>
| <New_Data_Item><Data_Item>
<Data_Item> ::= a string or a numeric value from data domain

데이터 항목의 명칭 변경은 주어진 재명명 조건에 설정된 데이터 항목의 개수와 순서를 준수하며 일대일 대칭적으로 변경해야 한다. 반면, 데이터집합의 명칭 변경은 이전 데이터 항목 목록과 이후 데이터 항목 목록이 모두 NULL 로 설정된 경우에만 수행된다.

3.3 데이터 전처리 이항 대수 연산

3.3.1 데이터집합의 합집합 연산

*데이터집합의 합집합 연산*은 주어진 두 데이터집합으로부터 어느 한쪽이나 양쪽 모두에 존재하는 트랜잭션들을 선택하는 기능을 수행하며, 연이은 합집합 연산에 대한 교환법칙과 결합법칙의 적용이 가능하다.

<Dataset_Union> ::= [(OTDB)←] $(ITDB_i) \cup (ITDB_j)$

3.3.2 데이터집합의 교집합 연산

*데이터집합의 교집합 연산*은 주어진 두 데이터집합으로부터 양쪽 모두에 존재하는 트랜잭션들만을 선택하는 기능을 수행하며, 연이은 교집합 연산에 대한 교환법칙과 결합법칙의 적용이 가능하다.

<Dataset_Intersection> ::= [(OTDB)←]$(ITDB_i) \cap (ITDB_j)$

3.3.3 데이터집합의 차집합 연산

*데이터집합의 차집합 연산*은 주어진 두 데이터집합 중에서 처음 데이터집합에는 존재하고 두 번째 데이터집합에는 존재하지 않는 트랜잭션만을 선택하는 기능을 수행한다.

<Dataset_Difference> ::= [(OTDB)←]$(ITDB_i) - (ITDB_j)$

3.4 기타 대수 연산

3.4.1 집합 함수 연산

*집합 함수 연산*은 한 데이터집합에 대한 특정 트랜잭션 속성이나 데이터 항목 속성 집합을 대상으로 지정 집합 함수를 적용하여 하나의 결과 값을 도출하는 기능을 수행한다.

<Aggregate_Operation> ::=
<Aggregate_Function>(<Function_Criteria>)

<Aggregate_Function> ::= Sum | Avg | Min | Max | Count
<Function_Criteria> ::= any transaction properties
| any data properties

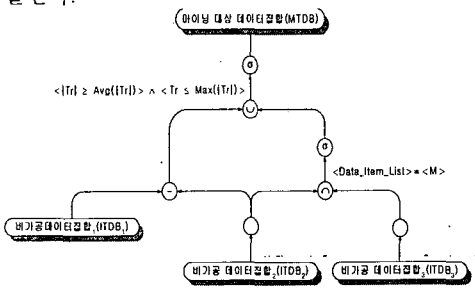
3.4.2 논리 연산

*논리 연산*은 트랜잭션 선택 연산과 데이터 프로젝션 연산에서 조건 반전이나 복합 조건 설정을 위해 활용된다. 즉, 단항 연산자인 논리 부정 연산은 주어진 조건을 부정하여 상반된 결과를 도출하는 기능을 수행하고, 이항 연산자인 논리곱 연산은 두 개의 주어진 조건이 모두 만족되는 결과만을 도출하는 기능을 수행하며, 이항 연산자인 논리합 연산은 각각의 두 조건을 만족하는 결과들의 합집합을 도출하는 기능을 수행한다.

<Logical_Not> ::= ㄱ <Condition>
<Logical_And > ::= <Condition$_i$> ∧ <Condition$_j$>
<Logical_Or> ::= <Condition$_i$> ∨ <Condition$_j$>

4. 데이터 전처리 연산의 적용 예제와 구현 모델

[그림 1]은 지금까지 소개된 데이터 전처리 대수 연산자들의 조합을 통해 마이닝 대상 데이터집합이 생성되는 과정을 보여주며, [그림 2]는 단계별 세부 내역을 하단에서 상단으로, 좌측에서 우측의 순으로 상술한다.



[그림 1] 데이터 전처리 대수 연산 트리

1> **(ITDB$_2$)←ρ** <<A', F'>=<A, F>>**(ITDB$_2$)**
*명칭 재설정 연산*을 통해 동일한 항목에 대한 명칭의 비호완성을 해결한다. 즉, **ITDB$_2$** 의 데이터 항목 A', F'를 **ITDB$_1$** 과 **ITDB$_3$**의 A, F 와 동일하게 변경한다.

2 **(OTDB$_1$)←π ㄱ** <<Data_Item_List>=<C, K>>**(ITDB$_3$)**
*데이터 프로젝션 연산*과 *논리 부정 연산*의 조합을 통해 **ITDB$_3$** 에서 관심 대상이 아닌 항목 C, K 를 제외한 나머지 데이터 항목들만을 추출한다.

3> **(OTDB$_2$)←(ITDB$_1$) – (ITDB$_2$)**
*데이터집합의 차집합 연산*을 통해 **ITDB$_1$** 이 **ITDB$_2$** 와 차별화되는 정보를 추출한다. 즉 **ITDB$_1$** 으로부터 **ITDB$_2$** 의 공통부분을 제외한 나머지 트랜잭션들만을 선출한다.

4> **(OTDB$_3$)←(ITDB$_2$) ∩ (OTDB$_1$)**
*데이터집합 교집합 연산*을 통해 **ITDB$_2$** 와 **OTDB$_1$** 의 공통되는 트래잭션들만을 선출한다.

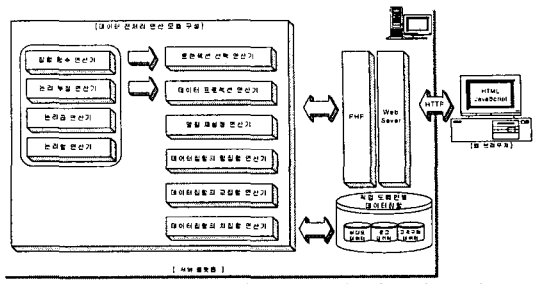
5> **(OTDB$_4$)←σ** <<Data_Item_List>=<M>>**(OTDB$_3$)**
*트랜재션 선택 연산*을 통해 **OTDB$_3$** 으로부터 특정 데이터항목, M,을 포함하는 트랜잭션들만을 선출한다.

6> **(OTDB$_5$)←(OTDB$_2$) ∪ (OTDB$_4$)**
*데이터집합의 합집합 연산*을 통해 **OTDB$_2$** 와 **OTDB$_4$**의 어느 한 쪽이나 양쪽 모두에 다 존재하는 트랜잭션들을 선출한다.

7> **(MTDB)←σ** <<|Tr| ≥ Avg(|Tr|)> ∧ <|Tr| ≤ Max(|Tr|)>>**(OTDB$_5$)**
마지막으로 *트랜잭션 선택 연산*과 *집합 함수 연산*, Avg,를 통해 **OTDB$_5$** 로부터 트랜재션 길이가 평균 길이 이상, 최대 길이 이하인 트랜잭션들만을 선출하여 최종 마이닝 대상 데이터집합, **MTDB,**를 생성한다. 단, |Tr|은 한 트랜잭션, Tr,을 구성하는 데이터 항목의 개수로서 *트랜잭션 길이*를 의미한다.

[그림 2] 데이터 전처리 대수 연산 적용 상세 설명

다음 [그림 3]은 실제로 *구현된 온라인 데이터 전처리 시스템* 모델의 전반적인 구성도를 나타낸다. 기본적 운용방법은 파일 형태로 저장된 트랜잭션 데이터를 대상으로 웹 브라우져 인터페이스를 통해 유닉스 서버상에서 실행되는 데이터 전처리 연산 모듈을 접근하게 된다.



[그림 3] 온라인 데이터 전처리 시스템 구성도

5. 결론 및 향후 연구

본 논문에서는 주어진 데이터집합의 특성을 기반으로 정의한 필수 데이터 전처리 대수 연산자들을 현재 데이터 마이닝이 직면하고 있는 메모리 공간 부족과 처리 속도 저하 등의 문제점에 대한 효과적인 대안으로 제안한다. 또한, 구체적 적용 사례에 의한 용도 설명과 실제 구현 시스템도 함께 제시하고 있다.

향후에는 데이터 마이닝 결과의 이해 및 분석과 추후 정보 활용의 도모를 위해 데이터 후처리 연산자에 대한 연구도 필요하다.

참고문헌

[1] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and R. Uthurusamy, R., (Eds.). "Advances in Knowledge Discovery and Data Mining", AAAI Press, The MIT Press, CA, USA ,1996

[2] R. Agrawal, T. Imielinski, and A. Swami. "Mining association rules between sets of items in large databases". In Proc. Of the ACM SIGMOD Conference on Management of Data, p207-216, Washington, D.C, 1993.

[3] Agrawal, R. and Srikant, R. "Mining sequential patterns"., In International Conference Data Engineering, Taipei, Taiwan, March 1995

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In E. Simoudis, J. Han, and U.M. Fayyad, editors, Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96), pages 226--231. AAAI Pres, 1996

[5] Weiss, R., Duda, A., Gifford, D.K., "Composition and search with a video Algebra". IEEE Multimedia, vol 1 num. 2. 2/1995

[6] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In Proc. of ICDE-97, 1997

[7] J. Han, Y. Fu, K. Koperski, W. Wang , and O. Zaiane "DMQL: A Data Mining Query Language for Relational Databases",. In SIGMOD'96 Workshop, DMKD'96, Montreal, Canada, June 1996

[8] R. Meo, G. Psaila, and S. Ceri. "A New SQL-like Operator for Mining Association Rules". In T. M. Vijayaraman, A. Buchmann, C. Mohan, and N. Sarda, editors, Proc. VLDB'96, Mumbai (Bombay), India, pages 122--133. Morgan Kaufmann, 1996

[9] Tomasz Imielinski and Aashu Virmani." MSQL: A Query Language for Database Mining". Data Mining and Knowledge Discovery, 3(4):373--408, Dec. 1999