韓國컴퓨터情報學會 論文誌
第14卷 第5號, 2009. 5.

2009-14-5-5-1

OLAP 데이터 큐브에서의 추론통제 프로세스 설계

이 덕 성*, 최 인 수**

Design of an Inference Control Process in OLAP Data Cubes

Duck-Sung Lee *, In-Soo Choi **

요 약

OLAP 데이터 큐브와 SDB(통계 데이터베이스) 모두 다차원 데이터 무리를 대상으로 하고, 이 데이터 무리의 모든 차원 별로 통계적인 요약처리를 한다는 데에는 공통점이 있으나 그 형성과정은 아주 다르다. SDB는 여러 베이스 데이터를 이용하여 자신이 쓸 베이스 데이터를 만들고 있으나 OLAP 데이터 큐브에서는 베이스 데이터 자체가 직접적으로 사용된다. 다시 말하면 SDB의 베이스 데이터는 머크로 데이터인데 반해 OLAP 데이터 큐브에서의 핵심 큐보이드 데이터는 마이크로 데이터라는 뜻이다.

OLAP 데이터 큐브에 측정값을 입주시키는 데에 베이스 테이블을 사용한다. 구체적으로 핵심 큐보이드의 각 셀에 마이크로 데이터를 입주시키는 데에 베이스 테이블의 각 레코드를 사용한다. 그런데 OLAP 데이터 큐브에서는 마이크로 데이터가 사용되는 경우가 태반이기 때문에 베이스 테이블에서의 어떤 레코드는 존재하지 않게 되는 상황이 생길 수도 있게 된다. 그리고 이렇게 되면 핵심 큐보이드의 어떤 셀은 공백으로 남게 되는 것이다.

Wang 등은 OLAP 데이터 큐브로부터 기밀 누설을 막을 수 있는 방법을 제안하였는데, 이 방법은 집계함수의 종류에 관계없이 적용시킬 수 있다고 주장하고 있다. 그러나 큐보이드의 어떤 셀 하나라도 공백으로 되어있는 경우는 집계함수의 종류에 관계없이 적용시킬 수 있다는 Wang의 주장이 틀리게 된다는 것을 본 연구에서는 밝히고 있다. 본 연구에서는 Wang의 오류를 없앤 OLAP 데이터 큐브에서의 새로운 추론통제 프로세스를 설계하는 데에 목적을 두고 있다.

Abstract

Both On-Line Analytical Processing (OLAP) data cubes and Statistical Databases (SDBs) deal with multidimensional data sets, and both are concerned with statistical summarizations over the dimensions of the data sets. However, there is a distinction between the two that can be made. While SDBs are usually derived from other base data, OLAP data cubes often represent directly the base data. In other word, the base data of SDBs are the macro-data, whereas the core cubiod data in OLAP data cubes are the micro-data.

The base table in OLAP is used to populate the data cube with values of the measure attribute, and each record in the base tables is used to populate a cell of the core cuboid. The fact that OLAP

• 제1저자 : 이덕성 교신저자 : 최인수
• 투고일 : 2009. 05. 14, 심사일 : 2009. 05. 19, 게재확정일 : 2009. 05. 26.
* 숭실대학교 대학원 산업·정보시스템공학과 재학 ** 숭실대학교 산업·정보시스템공학과 교수
※ 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌음.

data cubes mostly represent the micro-data may make some records be absent in the base table. Some cells of the core cuboid remain empty, if corresponding records are absent in the base table.

Wang and others proposed a method for securing OLAP data cubes against privacy breaches. They assert that the proposed method does not depend on specific types of aggregation functions. In this paper, however, it is found that their assertion on aggregate functions is wrong whenever any cell of the core cuboid remains empty. The objective of this study is to design an inference control process in OLAP data cubes which rectifying Wang's error.

▸ Keyword : 추론통제(Inference Control), 기밀누설(Privacy Breaches), 집계함수(Aggregation Function), 데이터 큐브(Data Cube)

Ⅰ. 서 론

1980년대부터 크게 주목받기 시작한 통계 데이터베이스(Statistical Database: 이하 SDB로 기재)의 목적은 회사와 같은 조직의 그룹이나 사람 그룹을 대상으로 그 빈도, 평균 같은 여러 통계 값을 공표함과 동시에 이 데이터베이스에 들어있는 개별 데이터의 기밀을 보호하는 데에 있는데, 사실 이 목적을 달성하기는 쉽지 않다. 왜냐하면 그룹별로 공표된 여러 통계 값을 잘 연계시켜 보면 개별 데이터와 같은 중요 데이터(sensitive data)를 유추할 수 있게 되기 때문이다. SDB에서는 그룹별 통계가 공표되기 때문에, 간단한 예로 여성 교수 1명을 포함해서 총 10명의 교수가 근무하고 있는 정보시스템학과의 경우, 과 전체교수그룹의 연봉합계에서 남성교수그룹의 연봉합계를 빼면 여성교수 1인의 개인연봉이라는 중요 데이터를 유추할 수 있게 되는 것이다. 이와 같이 어떤 데이터베이스에서 중요 데이터가 누설되게 된다면 이 데이터베이스를 위태롭다(compromised)라고 부르는데[1], 데이터베이스가 위태롭게 되지 않도록, 즉 중요 데이터가 누설되지 않도록 통제하는 것을 추론통제(inference control)라 한다.

SDB의 추론통제에는 크게 결과물 제약통제(Output Restriction Controls)와 혼란통제(Perturbation Controls)의 2가지가 있다. 결과물 제약통제란 비 중요데이터(nonsensitive data)의 공표를 억제시킴으로써 중요 데이터가 새어나가지 않도록 하는 방법이고, 혼란통제란 올바른 통계 값을 고의로 틀리게 조작하여 공표함으로써 중요 데이터를 추론하지 못하도록 하는 방법이다. 어떤 추론통제 방법이 좋은 가는 보안성, 정보손실 그리고 비용이라는 3가지 인자를 평가함으로써 결정하게 된다. 여기서 보안성이란 해당 추론통제 방법을 실행시켰을 경우 유추되는 중요 데이터의 숫자를 갖고 그 높고 낮음이 평가되는 것을 말하고, 정보손실이란 비 중요데이터를 얼마나 많이 공표할 수 없도록 억제시키느냐에 따라 그 많고 적음이 평가되는 것을 말하며, 비용이란 쿼리 처리에 관련된 비용을 말하는 것이다. 보안성을 높이면 정보손실이 많아지고 고 비용이 되기 때문에 이 3 가지 인자 간에 적정 균형을 맞춰주는 추론통제 방법을 선택할 필요가 있다고 본다.[2,3]

데이터에 내재되어 있는 특이 패턴을 찾고자 데이터 분석을 할 때에 보통 다차원적인 데이터 집계를 하는데, 이때에 표준 SQL(Structured Query Language) 쿼리를 사용해도 좋지만 쿼리가 아주 복잡해진다는 단점이 생기게 된다. 달리 말하면 쿼리가 복잡해지면 SDB의 근본 구성 데이터 즉 베이스 데이터(base data)를 여러 번 참조해야 되고 결과적으로 쿼리의 성능이 저하되게 된다는 뜻이다. 오늘날의 비즈니스 의사결정 용 쿼리는 복잡한 게 대다수이기 때문에 이상 언급한 SQL을 그대로 사용할 수 없는 실정이다. 따라서 이를 대신할 새로운 집계용 연산자인 OLAP(OnLine Analytical Processing) 데이터 큐브가 1990년대에 도입되었으며, 상술한 SDB에서의 추론통제가 이 OLAP 데이터 큐브에도 적용되기 시작한 것이다.

오늘날 OLAP 데이터 큐브에서의 추론통제기법이 여럿 개발되고 있으나 SDB에 적용한 기법을 그대로 OLAP 데이터 큐브에 적용시키는 게 대부분이다. 그러나 OLAP 데이터 큐브의 특성을 고려하지 않고 SDB에서의 추론통제방식을 데이터 큐브에 직접 적용시키면 데이터 큐브는 올바른 정보를 창출시키지 못한 채 추론통제를 하게 되는 모순에 빠질 수도 있게 된다. 올바른 정보를 갖고 있는 데이터 큐브를 만드는 것이 먼저이고, 이 큐브로부터 정보가 누설되지 않도록 통제하는 것은 다음이다. 다시 말하면, SDB에는 올바른 정보가 입주해 있기에 SDB로부터 정보가 누설되는 것을 막는 추론통제를 논하는 것이 정당하지만, OLAP 데이터 큐브에는 애초부터 올바른 정보가 입주할 수 없는 경우가 생길 수도 있기 때문에 이 데이터 큐브로부터의 추론통제를 논하는 것은 아무 의미가 없다는 뜻이다. 따라서 본 연구에서는 추론통제에 앞서 먼저 올바른 정보를 지니게 되는 데이터 큐브의 작성기법

을 논하고, 다음으로 이 큐브에서 정보가 누설되는 것을 막는 보안성이 좋고, 정보손실을 적게 하며, 비용이 적게 드는 추론통제기법을 개발하고자 한다.

II. 관련 연구

데이터 큐브의 추론통제에는 SDB의 추론통제에서 언급한 바 있는 2 종류의 방법 중 결과물 제약통제법이 주로 적용되고 있는데, 이 통제법은 먼저 검색한 다음 제한하는(detecting-then-removing) 기법을 쓰고 있다. 즉 먼저 쿼리를 실행시켜 얻은 결과물을 봄으로써 노출시키고 싶지 않은 중요 데이터가 이 쿼리를 통해서 추론되는 지의 여부를 확인한 다음 중요 데이터의 추론을 유발한 쿼리의 실행결과는 답하지 않고, 즉 제거시키고 중요 데이터의 추론을 유발하지 않는 쿼리의 실행결과만 답한다는 뜻이다. 그러나 이러한 결과물 제약통제법을 적용시키자면 쿼리를 실행시킨 후 바로 중요 데이터의 노출여부를 가리는 온라인 계산을 하여야 하고, 쿼리 실행으로 얻은 모든 결과물을 기록해야만 하는 등의 고비용 문제에 봉착하게 된다. 더구나 이러한 고비용 문제를 감수한다 하드라도 결과물 제약통제법에는 비현실적인 가정을 해야만 적용시킬 수 있게 된다는 문제점이 또 생기게 된다. 예를 들어, 집계함수 중 합계함수 SUM만 통용되는 데이터 큐브에서만 적용가능하다던가 실제로는 예를 들어 80% 정도 데이터가 누설되어도 프라이버시가 침해되었다고 볼 수 있는데 비해 100% 정확한 데이터가 누설되어야만 프라이버시가 침해당한다고 규정하는 등 능률적이지 못하면서도 비현실적인 전제를 하고 주로 적용시켜 왔다는 뜻이다.[4]

최근에는 이러한 결점이 많은 결과물 제약통제법을 대체할 새로운 기법을 Wang 등이 보고하고 있다. 결과물 제약통제법은 사용할 수 있는 집계함수가 몇 가지로 제한되어 있어서 현실적이지 못했는데 비해 이 기법은 분배적 집계함수를 사용한다면 집계함수 자체를 전혀 고려하지 않아도 되는 현실적인 기법이며, 온라인 계산을 할 필요가 없는 경제적인 기법이라고 주장하고 있다.[5]

Palpanas[6]와 Gray[7] 그리고 Wang은 셈 집계함수 COUNT를 분배적 함수로 보고 있다. 특히 Wang은 데이터 큐브를 작성하는데 관련된 집계함수 중 대수적 함수인 평균 집계함수 AVG는 매개함수 개념을 활용하면 데이터 큐브 작성에 직접 사용할 수 있다고 하였다. 즉, AVG 자체는 분배적 함수가 아니지만, (SUM, COUNT) 매개함수는 분배적 함수가 되기 때문에 AVG는 (SUM, COUNT)를 통해서 구하면 된다고 하였다.

본 연구에서는 사용하는 집계함수의 종류에 구애받지 않고 추론통제를 할 수 있다는 Wang의 추론통제법에 오류가 있음을 확인하고, 이러한 Wang의 추론통제법이 올바로 시행될 수 있는 새로운 체제를 구축하고자 하고 있다. 구체적으로, 본 연구에서는 현 OLAP 체제에서 셈 집계함수 COUNT의 기능이 올바로 발휘되지 못한다는 점과 COUNT를 활용해야만 계산할 수 있는 평균 집계함수 AVG의 기능도 올바로 발휘되지 못한다는 점을 확인하고, 이를 해결할 체제를 구축하고자 하는 데에 목적을 두고 있다.

III. OLAP 데이터 큐브

3.1 사례

OLAP 데이터 큐브를 설계하는 데에는 스타 스키마(star schema)를 채택하는 것이 가장 보편적이다. 스타 스키마는 중앙의 사실 테이블을 여러 개의 차원 테이블이 둘러싼 형태를 취한다. 사실 테이블에서의 각 행은 여러 차원의 기본 키들의 조합 즉, 외래 키들의 조합으로 구성된 하나의 복합키와 이것과 관련된 여러 측정값으로 구성된다. 본 사례[8]에서 채택한 차원 테이블과 사실 테이블의 가상 데이터를 살펴보면 다음과 같다.

차원 테이블로는 시간(TIME) 테이블, 상점(STORE) 테이블, 제품(PRODUCT) 테이블의 3개가 있다. TIME 차원 테이블은 all(2008년 1년), 분기(Quarter), 월(Month), 주(Week)의 4개 속성으로 구성된다. 각 속성의 구성원은 주 48개, Month 12개, Quarter 4개, all 2008년 1개의 구성원이 존재하며, 1개월은 4주로 구성된다. 또한 기본키를 구성하는 Time_ID 48개를 부여하여 각 행을 식별할 수 있도록 하였다. 이렇게 구성된 TIME 차원 테이블은 그림 1과 같다.

Time_ID	Week	Month	Quarter	all
1	1	1	1	2008
2	2	1	1	2008
3	3	1	1	2008
4	4	1	1	2008
5	1	2	1	2008
⋮	⋮	⋮	⋮	⋮
48	4	12	4	2008

그림 1. TIME 차원 테이블
Fig. 1. TIME Dimension Table

STORE 차원 테이블은 상점이 위치하는 지역을 대상으로 all(대한민국 1개국), 도(Province), 시(City)의 3개 속성으로

구성하였다. all 속성에는 대한민국 1개국의 구성원이 있으며, 도 속성에는 Kyunggi, Chungcheong, Kyeongsang, Jeolla 라는 4개의 구성원이 있다. 또한 시의 구성원으로서 Kyunggi의 구성원은 Ansan, Sihwa가 있으며, Chungcheong의 구성원은 Jaechon, Gongju가, Kyeongsang의 구성원은 Changwon이 있고, Jeolla의 구성원은 Yusu, Iksan이 있어 총 7개가 있으며, 기본키로 사용할 Store_ID 7개를 부여하여 각 행을 식별하도록 하였다. STORE 차원 테이블은 그림 2와 같다.

Store_ID	City	Province	all
1	Ansan	Kyunggi	Korea
2	Sihwa	Kyunggi	Korea
3	Jaechon	Chungcheong	Korea
4	Gongju	Chungcheong	Korea
5	Changwon	Kyeongsang	Korea
6	Yusu	Jeolla	Korea
7	Iksan	Jeolla	Korea

그림 2. STORE 차원 테이블
Fig. 2. STORE Dimension Table

PRODUCT 차원 테이블은 각 상점에서 판매되는 제품을 나타내며, 4개의 속성으로 구성되며, all 속성에는 1개, Class 속성에는 2개, Category 속성에는 5개, Name 속성에는 20개의 구성원이 존재한다. all 속성은 모든 Class를 통합한 것이고, Class 속성에는 Drink와 Food 구성원이 있으며, Category 속성에는 Drink에 속하는 Soda와 Juice 구성원과 Food에 속하는 FastFood, Noodle, PackingFood 구성원이 있다. 또한 Name 속성에는 Soda에 속하는 Coke, Diet Coke, Fanta, Cider 구성원이, Juice에 속하는 Apple juice, Grape juice, Orange juice, Peach juice 구성원이, FastFood에 속하는 Burger, Corn Cheese, Fried Potato, Pizza, Salad 구성원이, Noodle에 속하는 Bibim Myun, Cup Myun, Saeng Myun, Spaghetti, Tang Myun 구성원이, PackingFood에 속하는 Peach, Tuna 구성원이 있다. 여기에 Product_ID 20개를 부여하여 기본키로 사용한다. PODUCT 차원 테이블은 그림 3과 같다.

Product_ID	Name	Category	Class	all
1	Coke	Soda	Drink	ALL
2	Diet Coke	Soda	Drink	ALL
3	Cider	Soda	Drink	ALL
4	Fanta	Soda	Drink	ALL
5	Orange Juice	Juice	Drink	ALL
⋮	⋮	⋮	⋮	⋮

그림 3. PRODUCT 차원 테이블
Fig. 3. PRODUCT Dimension Table

사실 테이블은 각 차원의 기본키로 구성된 복합키 부분과 측정값으로 나타나는데, 판매량을 나타내는 측정값은 Quantity이다. 이러한 사실 테이블 중 1월달 일부를 나타낸 것이 그림 4이다.

3.2 데이터 큐브

3.1의 사례를 데이터 큐브로 그려보면 그림 5와 같은데, 먼저 이 데이터 큐브에 대해서 설명하고자 한다. 이 데이터 큐브에는 TIME, STORE, PRODUCT이라는 3개의 차원(dimension)이 있다. TIME 차원은 Week, Month, Quarter, all이라는 4개의 속성(attribute)으로 구성되어 있으며, 이들 속성 간에는 종속관계가 맺어지게 된다. 즉 Week가 모여서 Month가 되고, Month가 모여서 Quarter로 되며, Quarter가 모여서 all이 되는 종속관계가 맺어져 있다는 뜻이다. 여기서 all은 최상위 속성의 이름이 되며, 이들 속성 간의 이와 같은 종속관계를 Week ≤ Month ≤ Quarter ≤ all 과 같이 나타내기도 한다. STORE 차원은 City, Province, all 이라는 3개의 속성으로 구성되며, City ≤ Province ≤ all 의 종속관계를 갖고 있으며, PRODUCT 차원은 Name, Category, Class, all이라는 4개의 속성으로 구성되며, Name ≤ Category ≤ Class ≤ all 의 종속관계를 갖고 있다.

이상 언급한 3개 차원에서의 각 속성 간의 상호 교차점(그림 6 참조)을 구해보면 48개(4x3x4)가 되는데, 이들 48개 각각의 교차점에서 하나씩의 큐보이드(cuboid) 〈TIME, STORE, PRODUCT〉가 생성된다.

Time_ID	Store_ID	Product_ID	Quantity
1	2	2	123
1	3	3	32
1	4	9	209
1	4	11	197
1	5	11	219
1	6	14	18
1	6	17	210
1	6	18	163
2	2	5	193
2	2	8	255
2	3	9	107
2	3	13	222
2	5	17	162
2	7	19	85
3	1	3	209
3	1	6	131
3	2	9	126
3	3	11	3
3	4	18	236

3	7	19	283
4	2	6	88
4	3	8	228
4	3	9	179
4	3	11	297
4	6	14	51
4	6	15	132
4	7	19	242
5	1	1	157
⋮	⋮	⋮	⋮

그림 4. ORDER 사실 테이블
Fig. 4. ORDER Fact Table

예를 들면, TIME 차원의 Month 속성, STORE 차원의 Province 속성, 그리고 PRODUCT 차원의 Category 속성이 교차하는 점에서 큐보이드 〈Month, Province, Category〉가 생성된다는 것이다. 또한, 이 큐보이드에는 240(Month 개수 12 x Province 개수 4 x Category 개수 5) 종류의 판매량 합계를 계산해 넣을 수 있는 240개의 셀(cell)이 마련되게 된다. 각 차원의 속성 이름 대신에 그림 6에서 표시한 바와 같은 숫자를 사용하고, 동시에 각 차원 속성의 종속관계를 지켜가면서 총 48개의 큐보이드로 구성되는 본 사례 데이터 큐브를 나타낸 것이 그림 5인 것이다. 큐보이드 〈Month, Province, Category〉는 그림 5에서는 큐보이드 〈2,2,2〉로 표시된다. 큐보이드 〈1,2,2〉의 주별 집계를 모아서 월별 집계를 하거나, 큐보이드 〈2,1,2〉의 City별 집계를 모아서 Province별 집계를 하던지, 아니면 큐보이드 〈2,2,1〉의 Name별 집계를 모아서 Category별 집계를 하면 큐보이드 〈2,2,2〉를 만들 수 있다는 것을 그림 5에서의 점선 연결부분을 살펴보면 쉽게 이해할 수 있으리라 본다.[9]

본 사례에서의 베이스 테이블(base table) 스키마(schema)는 (Week, City, Name, Quantity)와 같다. 그림 5에서의 큐보이드 〈1,1,1〉 즉 큐보이드 〈Week, City, Name〉을 핵심 큐보이드(core cuboid)라 부르는데, 이 핵심 큐보이드의 각 셀에 측정값 속성인 Quantity를 입주시킬 때에 이 베이스 테이블을 사용하게 된다. 그림 5에서 보다시피 큐보이드 〈1,1,1〉의 각 셀에 입주해 있는 Quantity 값만 활용하면 나머지 47개 큐보이드의 각 셀에 입주할 모든 Quantity 값을 집계할 수 있다. 따라서 데이터 큐브의 구성에 있어서 아주 유일하게 있어야만 하는 최소단위의 큐보이드가 바로 큐보이드 〈1,1,1〉이기 때문에 큐보이드 〈1,1,1〉을 핵심 큐보이드라고 부르는 것이며, 또한 이러한 핵심 큐보이드에 Quantity 값을 입주시키는 역할을 하는 테이블을 베이스 테이블이라고 부르는 것이다. 이상에서 보다시피 OLAP에서 사실 테이블이라고 부르는 베이스 테이블과 데이터 큐브는 서로 떼어놓고 볼 수 없는 아주 밀접한 관계를 맺고 있다고 보겠다.[10]

그림 5에서의 핵심 큐보이드 〈Week, City, Name〉의 각 셀을 채워나갈 때에 OLAP에서는 SDB에서는 일어날 수 없는 다음과 같은 2가지 경우를 고려해야 한다. 첫째, 어느 셀에 해당되는 레코드(record)가 베이스 테이블에 2개 이상 있을 경우이다. 이때에 OLAP에서는 합 집계함수 SUM을 사용하여 이 2개 이상의 값을 더한 것을 하나의 값으로 간주하여 해당 셀에 기입하고 베이스 테이블에서의 레코드 개수도 하나로 셈하게 되는데, 이 때문에 OLAP에서 장바구니 분석을 할 때에 여러 문제가 생기게 되는 것이다.[11,12]

둘째로, 이 경우가 본 연구의 대상이 되는 경우인데, 어느 셀에 해당되는 레코드가 베이스 테이블에 없는 경우이다. 이 때에 OLAP에서는 셀을 공백(널 값)으로 남기게 되는데[13], 바로 이 사실 때문에 집계함수 COUNT와 AVG를 기존의 OLAP 환경에서 직접 사용할 수 없게 된다는 것을 확인한 것이 본 연구의 동기라 하겠다.

IV. Wang의 추론통제 문제점

Wang의 추론통제[14,15]가 기존의 추론통제와 다른 점은 크게 다음과 같은 3가지라고 볼 수 있다. 첫째, 기존의 추론통제에서는 핵심 큐보이드의 셀 값만 공표할 수 없는 중요정보로 간주하는데 비해, Wang의 추론통제에서는 핵심 큐보이드 외 다른 큐보이드의 셀 값까지도 공표할 수 없는 중요정보로 삼을 수 있고,

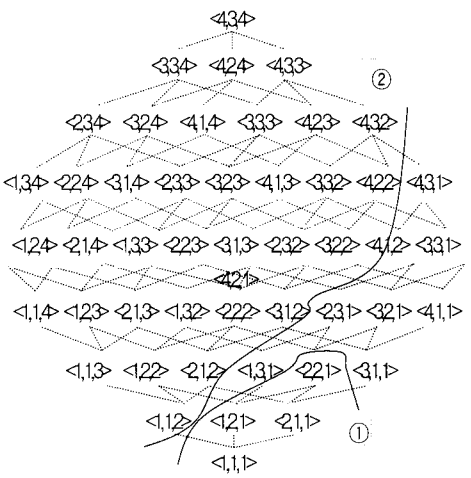

그림 5. 사례 데이터 큐브
Fig. 5. A Case Data Cube

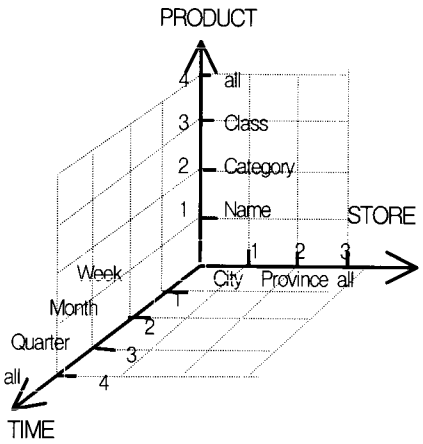

그림 6. 각 차원의 속성
Fig. 6. The Attributes of Each Dimension

둘째, 기존의 추론통제에서는 채택할 수 있는 집계함수가 몇 가지로 한정되어 있는데 비해, Wang의 추론통제에서는 어떠한 집계함수든 관계없이 다 채택할 수 있으며, 셋째, 기존의 추론통제는 먼저 검색한 다음 제한하는 (detecting-then-removing) 기법을 쓰고 있는데 비해, Wang의 추론통제는 검색을 하지 않고 애초부터 추론을 유발할 수 있는 부분을 봉쇄시키는 기법을 쓰고 있다.

4.1 중요 정보의 지정

그림 5의 핵심 큐보이드 〈1,1,1〉만 공표해서는 안 되는 중요 큐보이드로 삼아 추론통제를 해 온 기존 방법과는 달리 Wang은 여러 다른 큐보이드까지 그 영역을 넓히고 있다.

개별 상품의 월별, 도별 판매량 정보보다 더 개괄적인 정보는 공표해도 되지만 그렇지 않는 경우도 절대적으로 보호해야 한다고 하는 본 연구에서의 방침을 생각해 보자. 그림 5에서 볼 것 같으면 개별 상품의 월별, 도별 판매량 정보를 갖고 있는 것은 큐보이드 〈2,2,1〉이며, 이 큐보이드의 선조(ancestor) 큐보이드는 큐보이드 〈1,2,1〉, 〈2,1,1〉, 〈1,1,1〉이다. 이들 3개 선조 큐보이드는 큐보이드 〈2,2,1〉보다 더 상세한 정보를 갖고 있는 큐보이드들이다. 따라서 보호해야 할 큐보이드는 큐보이드 〈2,2,1〉, 〈1,2,1〉, 〈2,1,1〉, 〈1,1,1〉이 되는데 이를 Below(〈2,2,1〉)로 나타내며, 그림 5에서는 실선 ① 아래에 속하는 모든 큐보이드들이 여기에 해당된다.

Below(〈2,2,1〉) = { 〈2,2,1〉, 〈1,2,1〉, 〈2,1,1〉, 〈1,1,1〉 }

이와 같이 Wang의 추론통제에서는 Below(〈2,2,1〉) 개념을 채택함으로써 핵심 큐보이드 〈1,1,1〉을 포함해서 4개의 큐보이드를 누설시켜서는 안 되는 중요 큐보이드로 지정할 수 있게 되는 것이다.

4.2 데이터 큐브에서의 추론

데이터 큐브에서는 비보호 후손(unprotected descendants) 큐보이드의 셀 데이터를 활용하면 보호 큐보이드 셀 데이터를 추론할 수 있게 된다. 전술한 Below(〈2,2,1〉)이 보호 큐보이드 집합(이를 타깃(target)이라 부름)인데, 그림 5에서 실선 ① 위에 있는 큐보이드 집합(이를 소스(source)라 부르며 이하 S로 표기)에서의 셀 데이터를 잘 활용하면 보호 큐보이드 집합의 셀 데이터를 추론해낼 수 있다는 뜻이다. S에 속해 있는 큐보이드 〈2,2,2〉, 〈2,3,1〉, 〈3,2,1〉과 타깃에 속해있는 큐보이드 〈2,2,1〉을 예로 들어 이를 설명하고자 한다.

큐보이드 〈2,2,1〉의 직계 후손인 3개의 큐보이드 〈2,2,2〉, 〈2,3,1〉, 〈3,2,1〉 각자에는 서로 간의 선조-후손의 관계가 맺어져 있지 않음을 확인 할 수 있을 것이다. 이 때문에 이들 각자 큐보이드의 셀은 선조 큐보이드 〈2,2,1〉의 셀을 추론하는데 아무런 공헌을 할 수 없는 것이다. 그러나 이들 큐보이드 3개를 연합시킨다면 추론에 공헌할 수 있게 된다. 왜냐하면 후손 큐보이드 3개를 하나로 묶으면 선조 큐보이드의 완벽한 후손으로 정립되기 때문이다. 물론 큐보이드 〈2,2,1〉을 추론해낼 수 있는 S 큐보이드에는 이들 3개 큐보이드만 있는 것이 아니고 이들 3개 큐보이드의 모든 후손 큐보이드들도 있음을 확인할 필요가 있다. 그러나 그림 5에서와 같이 표현되는 격자 구조의 데이터 큐브에서는 선조의 추론에 필요한 최대 정보를 직계 후손에서 다 얻을 수 있게 되고, 또한 직계 후손의 모든 후손들로부터 얻을 수 있는 정보는 직계 후손들로부터 얻게 되는 정보와 중복(redundant)되기 때문에 큐보이드 〈2,2,1〉의 직계 후손 큐보이드 〈2,2,2〉, 〈2,3,1〉, 〈3,2,1〉만 취급하면 충분한 것이다. 여기서 이러한 직계 후손 큐보이드 〈2,2,2〉, 〈2,3,1〉, 〈3,2,1〉을 Basis(S, 〈2,2,1〉)라 부른다.

Basis(S, 〈2,2,1〉) = { 〈2,2,2〉, 〈2,3,1〉, 〈3,2,1〉 }

따라서 타깃 큐보이드〈2,2,1〉의 추론에 관해서는 서로 간에는 선조-후손 관계를 논할 수 없는(non-comparable) 직계 후손 큐보이드 〈2,2,2〉, 〈2,3,1〉, 〈3,2,1〉만 다루면 된다.

선조-후손 관계를 논할 수 없는 큐보이드들 만으로 구성되는 것이 Basis(S, 〈2,2,1〉)이기 때문에 추론에 관해서 다음

과 같은 결론을 내릴 수 있다.

S로부터 타깃을 추론할 수 있으려면 Basis( )에는 큐보이드가 적어도 2개 이상 있어야 한다.

그러나, 이 결론을 데이터 큐브에서 중요 데이터가 누설되지 않도록 추론통제를 하는 본 연구의 목적에 맞게 고쳐 쓰면 다음과 같다.

Basis( )의 구성 큐보이드가 한 개 뿐이라면 S로부터 타깃을 추론할 수 없다.

4.3 공표가능 S의 최대 부분집합

그림 5에서 실선 ① 이하에 있는 큐보이드 집합이 타깃이 되고, 위에 있는 큐보이드 집합이 S가 되는데 S의 모든 큐보이드를 다 공표하면 이상에서 논한바 같이 타깃이 누설되게 된다. 따라서 공표를 하드라도 타깃을 위태롭게 만들지 않는 즉 누설시키지 않는 S의 최대 부분집합을 구하는 것이 중요하게 된다.

S의 최대 부분집합을 구하자면 Basis( )의 유일 큐보이드로부터 자라나가는 즉 이 유일 큐보이드의 모든 후손집합을 구하면 된다. 타깃 중에서 직계 후손을 하나만 갖는 큐보이드는 〈1,1,1〉 밖에 없다. 큐보이드 〈1,1,1〉의 유일 직계 큐보이드 〈1,1,2〉를 루트(root)라 부르는데, 이 루트의 모든 후손집합 즉 실선 ②의 윗부분에 있는 모든 큐보이드들이 S의 최대 부분집합이 된다.

S에 있는 비 중요데이터를 모두 공표할 수는 없지만 최대한 많이 공표해야만 정보손실을 줄일 수 있다. 이러한 의미에서 S의 최대 부분집합을 구하는 것이 의미가 있는데, 이러한 의미를 달성시킨 것이 데이터 큐브에서의 Wang의 추론통제인 것이다.

4.4 문제점

추론통제 기법으로서의 Wang의 추론통제는 중요 데이터의 폭을 핵심 큐보이드 외로 넓혔기에 보안성이 좋고, 공표할 수 있는 큐보이드를 최대한 넓혔기에 정보손실이 적으며, 검색을 먼저 해보고 그 결과를 갖고 공표여부를 결정하는 대신에 애초부터 추론을 유발할 수 있는 부분을 봉쇄시키기에 비용이 적게 들어가서 추론통제기법의 평가인자 면으로 볼 때엔 아주 우수한 통제기법이라고 판정내릴 수 있으나, 정확한 통계 값을 핵심 큐보이드에 입주시켜서 추론통제를 하느냐 하는 중요하면서도 원천적인 관점으로 볼 때에는 그렇지 못하다.

3.2에서 언급한 바 있지만 베이스 테이블과 데이터 큐브는 서로 밀접한 관계를 맺고 있다. 즉 핵심 큐보이드 셀에 측정값 속성을 입주시키는 것이 베이스테이블인 것이다. 그런데, Wang의 추론통제에서의 베이스 테이블은 셈(COUNT)과 평균(AVG)을 집계하는 데에는 올바른 역할을 할 수 없음을 본 연구에서는 밝히고 있다. 셈과 평균을 구하는데 있어서 베이스 테이블이 잘못되어 있으면 이에 직접 관련되어 있는 데이터 큐브도 잘못되게 된다. 따라서 잘못된 데이터 큐브에서 S의 최대 부분집합을 구하는 Wang의 추론통제도 잘못되었다고 생각한다.

V에서 셈과 평균을 구하는데 있어서 Wang의 베이스테이블이 왜 오류에 빠지는지를 설명하고, 올바른 베이스 테이블을 작성하는 방법을 논하고자 한다.

V. OLAP 데이터 큐브와 SDB

5.1 차이점

OLAP 데이터 큐브와 SDB는 양자 모두 다차원 데이터 세트(data set)를 대상으로 하며, 이 세트에 있는 모든 차원 별로 통계적인 집계를 한다는 면에는 공통점이 있으나, 그 적용분야에는 큰 차이점이 있기 때문에 서로 잘 어울리지 못한다고 본다. SDB는 인구통계, 국가생산/소비 패턴 같은 것을 다루는 사회경제(socio-economic) 데이터를 위주로 하여 구성 되며, 통계학자들이 주로 다루는 것임에 비해서, OLAP 데이터 큐브는 소매상 상품 거래상황과 같은 거래상황에 관련된 비즈니스 데이터를 위주로 하여 구성되며, 비즈니스 경영자의 의사결정을 돕는데 주로 사용된다. 의사결정자가 꼭 통계학자일 필요가 없음을, 더구나 비즈니스 경영자가 통계학자일 필요는 더더욱 없음을 감안해 볼 때에 이들 SDB와 OLAP 데이터 큐브는 그 적용영역에 큰 차이가 있음을 알게 되리라 본다.

이상, 개념적으로 본 차이점에 대하여 알아보았으나, 본 연구에서는 SDB와 OLAP 데이터큐브의 베이스 테이블 데이터(이하 베이스 데이터로 기록)의 출처의 차이점이라는 기술적 차이점을 규명하고, 이 기술적 차이점 때문에 발생할 수 있는 문제점을 해결하고자 한다. SDB는 여러 다른 종류의 데이터를 종합하여 만든 것을 자신의 베이스 데이터로 삼고 있으나, OLAP 데이터 큐브는 직접 얻은 데이터를 자신의 베이스 데이터로 삼고 있음에 주목할 필요가 있다고 본다. 예를 들면, SDB의 하나인 인구통계 데이터베이스의 베이스 데이터는 각 개인에 관한 데이터를 종합한 것이며, OLAP 데이터 큐브의 하나인 판매거래 데이터 큐브의 베이스 데이터는 판매거래 데이터 자체라는 것이다. 여기서 각 개인에 관한 데이터 즉 SDB에서 자신의 베이스 데이터를 구성하는데 참고하는

원래의 데이터를 마이크로 데이터(micro-data)라 부르고, 이들 마이크로 데이터를 종합하여 작성한 SDB에서의 베이스 데이터를 머크로 데이터(macro-data)라 부른다. 전술한 OLAP 데이터 큐브에서의 판매거래 데이터는 마이크로 데이터이다.[16]

SDB의 베이스 데이터 대부분이 머크로 데이터가 되는 이유는 개별 데이터의 프라이버시를 지키자면 마이크로 데이터를 사용할 수 없기 때문일 것이며, 또한 일례로 통계분석에서는 각 개별 데이터인 마이크로 데이터에는 별 의미가 없고, 대신 종합 데이터가 필요하게 되기 때문이라고 생각한다. 반면, 거래상황에 관련된 비즈니스 데이터를 주로 다루는 것이 OLAP 데이터 큐브이기 때문에 OLAP 데이터 큐브에서는 거래 데이터인 마이크로 데이터 자체가 사실 테이블(fact table)의 형태로 기입되어 베이스 데이터로 되는 것이다.

2008년 1년 동안 1개 상점 당 주별 각 상품의 판매량 평균을 알아보는 사례(3.1 참조)에 있어서 Product 차원에서는 Juice 류에 국한시키고, Time 차원에서는 1월에 국한시켜 OLAP 데이터 큐브의 베이스 데이터와 SDB에 해당되는 베이스 데이터, 그리고 ORDER 사실 테이블을 살펴보면 각각 그림 7, 8, 4(꼬딕 부분 참조)와 같다. 다음 절에서 논하겠지만 올바른 정보를 산출할 수 있는 베이스 데이터는 그림 8의 SDB 베이스 데이터이고, 그림 7의 OLAP 데이터 큐브 용 베이스 데이터는 그렇지 못한데, 이는 그림 4에서와 같은 OLAP에서의 사실 테이블 작성법을 따랐기 때문이라고 생각한다. 그림 4의 사실 테이블 데이터는 마이크로 데이터이고, 그림 8의 SDB 베이스 데이터는 머크로 데이터임을 주목하기 바란다.

	Month / Week / Product	January			
		Week 1	Week 2	Week 3	Week 4
Juice	Apple Juice	-	-	131	88
	Grape Juice	-	-	-	-
	Orange Juice	-	193	-	-
	Peach Juice	-	255	-	228

그림 7. OLAP 베이스 데이터
Fig. 7. OLAP Base Data

	Month / Week / Product	January			
		Week 1	Week 2	Week 3	Week 4
Juice	Apple Juice	0	0	18.7	12.6
	Grape Juice	0	0	0	0
	Orange Juice	0	27.6	0	0
	Peach Juice	0	36.4	0	32.6

그림 8. SDB 베이스 데이터
Fig. 8. SDB Base Data

5.2 결측값

결측 값(missing values) 또는 널(null) 값이라고 부르는 값들은 이제까지 퍼뜨려진 바 없는 값들을 말하는데 공백 값(blank values)과는 다르다. 공백 값에서는 아무 것도 없는 상태가 되는데 비해 결측 값에서는 알려지지는 않았지만 뭔가 존재하게 되는 상태가 된다.

결측 값에는 여러 가지가 있으나, 본 연구에서는 값 자체가 적합하고, 이미 알려져 있는 상태이지만 사정상 기록할 수 없게 되는 결측 값을 대상으로 하고자 한다. 예로 100종류의 상품을 비치하여 판매하는 어느 상점에서 매일 팔리는 상품별로 판매개수와 판매액을 기록하는 경우를 생각해 보자. 어느 날 20 종류의 상품이 팔렸다면 이 20 종류의 상품에 대해서만 그 판매개수와 판매액을 기록하지, 나머지 팔리지 않았던 80 종류의 상품에 대해서 각각 그 판매개수가 0이고 판매액이 0이라고 기록하지 않는 것이 일반적인 거래 데이터의 기입방법이고, 특히 OLAP의 사실 테이블 기입방법인 것이다. 이와 같이 기입하지 않는 정확히 말해 기입할 수 없는 0이라는 판매개수와 판매액은 합리적이며 알려져 있는 값이지만 현실적으로 경제적으로 기입하지 못하는 결측 값이 되며, 결코 공백 값은 아닌 것이다.

사실 테이블을 작성할 때에 어느 날 80 종류의 상품이 팔리지 않았으면 80 종류 상품 각각의 판매개수와 판매액을 각각 0으로 기입하고, 다음 날 70 종류의 상품이 팔리지 않았다면 이 70 종류 상품 각각의 판매개수와 판매액을 각각 0으로 또 기입한다는 식으로 나날이 한다는 것은 현실적으로도 그리고 경제적으로도 불합리하다. 따라서 그림 4에서와 같이 사실 테이블을 작성할 때에는 이상에서 논한 0이라는 결측 값을 기입하지 않는 것이며, 이 때문에 그림 7에서와 같이 공백 값으로 처리되는 부분이 생기게 된다.

그러나, 이상에서와 같이 OLAP에서 결측 값 처리를 하지 못하고, 대신 공백 값 처리를 하게 되면 집계함수 중의 하나인 평균함수 AVG를 적용시킬 때에 큰 오류가 발생하게 된다. 그림 4의 사실 테이블을 결측 값 처리를 해서 그림 9와 같이 만들고, 이를 통해 각 상품의 주간 평균 판매량을 구해야만 올바른 AVG 값을 구할 수 있는데 비해, 공백 값 처리를 한 그림 4의 사실 테이블을 통해 AVG 값을 구하는 것은 올바른 평균의 정의를 만족시키지 못하는, 즉 틀린 AVG 값을 구하게 되기 때문이다.[8] 또한 그림 9의 사실 테이블을 기초로 하여 각 상품의 주간 평균 판매량을 구해보면 그림 8의 SDB 베이스 데이터의 내용과 일치하게 됨을 볼 때에 이 SDB 베이스 데이터는 결측값 처리가 된 머크로 데이터임을 알게 되고, 결과적으로 이러한 머크로 데이터를 활용해야만 올바른 정보

를 얻게됨을 재차 확인할 수 있게 되는 것이다.

본 연구에서는 먼저 OLAP에 집계함수 AVG를 적용시킬 때에 마이크로 데이터를 활용해야만 하는 OLAP의 현실을 인정함과 동시에 정확한 정보를 산출하는데 필수적인 머크로 데이터를 활용할 수 있는 체계를 구축한 다음, 이 체계 하에서 Wang의 추론통제기법을 적용시키고자 하는 것이다.

VI. 새로운 추론통제 프로세스

머크로 데이터를 베이스 데이터로 정립시킨 후 추론통제를 해나가는 새로운 프로세스를 단계별로 기술하면 다음과 같다.

〔1 단계〕 공백 값 상태의 사실 테이블 활용(그림 7)

집계함수 중의 하나인 평균 집계함수 AVG를 사용하여 데이터 큐브를 작성할 때에는 그림 9와 같은 결측 값 처리를 한 베이스 데이터를 기초로 하여야만 정확한 정보를 산출할 수 있다고 5.2에서 논한 바 있다.

그러나 OLAP에서 그림 9와 같은 베이스 데이터를 입력시킨다는 것은 현실적으로 불가능하며, 경제적이지 못하다. 또한 5.1에서 논한바 같이 OLAP에서의 베이스 데이터는 마이크로 데이터일 수밖에 없다. 따라서 본 연구의 추론통제에서는 출발점은 현 OLAP에서의 방식인 그림 7에서와 같은 공백 값 상태의 베이스 데이터를 활용한다.

나머지 Juice의 주별 평균 판매량도 그림 9의 각 행을 조사함으로써 구할 수 있는데, 여기서 주별 판매회수의 합이 어떤 경우든 7이 됨을 확인할 수 있을 것이다.

주별 판매회수의 합과 같은 COUNT 값은 늘 일정하게 된다. 본 사례에서는 STORE 차원의 잎 멤버 구성원 수는 7로서 고정되어 있기 때문인 것이다. 이렇게 일정한 값을 갖는 COUNT 값을 다음과 같이 활용하면 그림 7에서와 같은 공백 값 상태의 베이스 데이터를 사용하드라도 그림 9에서와 같은 베이스 데이터의 입력효과를 얻을 수 있다.

〔2-1 단계〕 베이스 테이블의 셀이 공백으로 되어 있으면 여기에 0을 기입한다.

〔2-2 단계〕 나머지 셀의 값은 고정 COUNT 값으로 나누어 기입한다.

이상의 세부단계를 거치면 OLAP에서 직접 사용할 수 있는 베이스 데이터를 얻게 되는데, 이 베이스 데이터는 그림 8의 SDB 베이스 데이터와 동일하게 됨을 주목하기 바란다.

Time_ID	Store_ID	Product_ID	Quantity
1	1	5	0
1	1	6	0
1	1	7	0
1	1	8	0
1	2	5	0
1	2	6	0
1	2	7	0
1	2	8	0
1	3	5	0
1	3	6	0
1	3	7	0
1	3	8	0
1	4	5	0
1	4	6	0
1	4	7	0
1	4	8	0
1	5	5	0
1	5	6	0
1	5	7	0
1	5	8	0
1	6	5	0
1	6	6	0
1	6	7	0
1	6	8	0
1	7	5	0
1	7	6	0
1	7	7	0
1	7	8	0
2	1	5	0
2	1	6	0
2	1	7	0
2	1	8	0
2	2	5	193
2	2	6	0
2	2	7	0
2	2	8	255
2	3	5	0
2	3	6	0
2	3	7	0
2	3	8	0
2	4	5	0
2	4	6	0
2	4	7	0
2	4	8	0
2	5	5	0
2	5	6	0
2	5	7	0
2	5	8	0
2	6	5	0
2	6	6	0
2	6	7	0
2	6	8	0
2	7	5	0
2	7	6	0
2	7	7	0
2	7	8	0

Time_ID	Store_ID	Product_ID	Quantity
5	1	5	0
5	1	6	131
5	1	7	0
5	1	8	0
3	2	5	0
3	2	6	0
3	2	7	0
3	2	8	0
3	3	5	0
3	3	6	0
3	3	7	0
3	3	8	0
3	4	5	0
3	4	6	0
3	4	7	0
3	4	8	0
3	5	5	0
3	5	6	0
3	5	7	0
3	5	8	0
3	6	5	0
3	6	6	0
3	6	7	0
3	6	8	0
3	7	5	0
3	7	6	0
5	7	7	0
3	7	8	0
4	1	5	0
4	1	6	0
4	1	7	0
4	1	8	0
4	2	5	0
4	2	6	88
4	2	7	0
4	2	8	0
4	3	5	0
4	3	6	0
4	3	7	0
4	3	8	228
4	4	5	0
4	4	6	0
4	4	7	0
4	4	8	0
4	5	5	0
4	5	6	0
4	5	7	0
4	5	8	0
4	6	5	0
4	6	6	0
4	6	7	0
4	6	8	0
4	7	5	0
4	7	6	0
4	7	7	0
4	7	8	0

그림 9. 결측값 처리를 한 사실 테이블
Fig. 9. A Fact Table with Missing Values

〔3 단계〕 새로운 데이터 큐브 구축(그림 10)

3.1의 사례에서는 1개 상점 당 상품집합의 판매량 평균을 알아보기 때문에 이를 위해서는 STORE 차원의 잎 구성원 상점 7개의 판매량을 모두 더한 값을 7로 나누어야 하는데, 여기서 보다시피 이러한 작업에서는 개별 상점을 분석대상에 넣을 필요가 없다. 따라서 2 단계를 거쳐 생성된 베이스 테이블의 스키마는 (Week, Name, AvgQuantity)로 된다. 이 베이스 테이블을 사용하여 그림 10과 같은 데이터 큐브의 핵심 큐보이드 〈Week, Name〉의 셀에 그림 8에서와 같은 각 상품의 주별 평균 판매량 베이스 데이터를 입주시킴으로써 올바른 평균 판매량(AvgQuantity) 집계를 할 수 있는 OLAP 체제를 구축하게 되는 것이다.

〔4 단계〕 중요 정보의 재지정(Below(⟨2,2,1⟩) → Below(⟨Month, Name⟩))

절대적으로 보호해야할 중요 큐보이드 집합으로서 그림 5에서의 Below(⟨2,2,1⟩)에 해당되는 부분을 그림 10의 새로운 데이터 큐브에서 지정한다. 이에 해당되는 부분은 실선 ①의 아래 부분인 Below(⟨Month, Name⟩)이다.

〔5단계〕 공표가능 S의 최대 부분집합

그림 10에서 공표가능 S의 최대 부분집합은 실선 ②의 윗부분에 있는 큐보이드 집합이 된다. 이때의 루트는 큐보이드 ⟨Week, Category⟩이다.

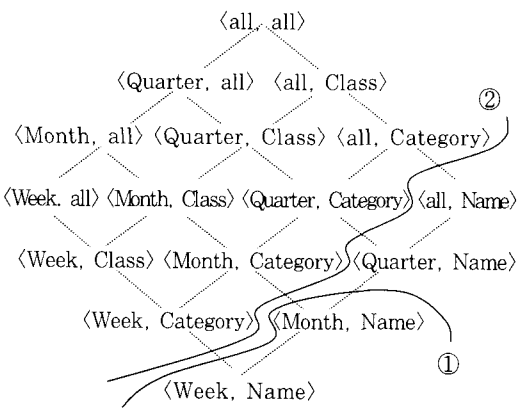

그림 10. 새로운 데이터 큐브
Fig. 10. The Modified Data Cube

Ⅶ. 결론

본 연구에서는 데이터 큐브에서의 추론통제에 있어서 COUNT, SUM, MIN, MAX, AVG 등 어떠한 집계함수든 관계없이 모두 적용 가능하다는 Wang의 이론에 오류가 있음을 발견하고, 이를 해결할 수 있는 새로운 추론통제 절차를 제안하고 있다. 즉 기존의 OLAP 환경에서 어느 셀에 해당되는 레코드가 베이스 테이블에 없는 경우, 즉 사건이 발생하지 않은 경우, 데이터 큐브의 셀은 공백으로 남게 된다. 바로 이 공백으로 남게 되는 베이스 테이블의 레코드 때문에 집계함수 COUNT와 AVG를 기존의 OLAP에서는 직접 사용할 수 없게 된다.

SUM의 경우는 문제가 없겠으나 COUNT와 AVG 집계함수를 기존의 OLAP 체제에서 직접 사용하면 틀린 정보를 얻게 된다. 올바른 정보를 대상으로 추론통제를 논하는 것이 의의가 있지, 틀린 정보를 대상으로 하여 논하는 것은 논리에 맞지 않는 것이다. 추론통제 기법의 효율성을 측정하는 지표인 보안성, 정보손실, 비용의 3가지 측면에서 Wang의 기법을 살펴보면, 중요 정보의 폭을 핵심 큐보이드 외에 더 추가함으로써 보안성을 확보하고 있고, 또한 소스에서 공표할 수 있는 큐보이드를 최대한 넓힘으로써 정보손실이 적으며, 비용측면에서는 검색후 그 결과로 공표여부를 결정하는 방법 대신에 사전에 추론할 수 있는 큐보이드를 통제하는 경제적인 기법이라 할 수 있다. 이러한 Wang의 추론통제 이론은 SUM 집계함수를 사용한 때에는 아주 훌륭하다고 할 수 있겠으나, COUNT와 AVG 집계함수를 사용할 때에는 틀린 정보를 대상으로 하여 추론통제를 논하는 것으로 되기 때문에 채택할 수 없는 추론통제 이론이 되는 것이다.

본 연구에서는 데이터 큐브에서 COUNT와 AVG 집계처리를 하는 경우 공백으로 남게 되는 레코드의 값을 결측값 처리를 함으로써 올바른 정보를 지니게 되는 새로운 베이스 테이블을 생성하여 이를 핵심 큐보이드로 삼고 있다. 이상의 언급사항을 본 연구에서 제안하고 있는 새로운 추론통제 프로세스의 각 단계(Ⅵ참조)와 매치시켜 보면 다음과 같다. '공백으로 남게 되는 레코드의 값'이라는 부분은 1단계이고, '결측값 처리를 함으로써'라는 부분은 2단계에 해당되며, '새로운 베이스 테이블을 생성'한다는 부분은 3단계에 해당된다. 본 연구에서 제안하고 있는 이상의 3단계를 거쳐야만 비로소 4, 5단계에서와 같이 Wang의 추론통제 이론을 적용시킬 수 있게 된다는 것이 본 연구의 취지이다.

본 연구의 활용방안에 대하여 살펴보면 다음과 같다.

첫째, 정부의 연구개발 예산에 대해 살펴보면, 공표데이터는 평균경쟁률, 평균지원액, 선정기관(또는 기업)수 등에 그치고 있다. 즉 정부에서는 최소의 정보만을 공표함으로써 보안성은 높일 수 있지만, 그 외의 모든 정보를 비공개함으로써 예산의 투명성, 효율성 및 공공성을 지향하는 목적은 달성할 수 없게 된다. 이 목적을 위해서는 현재 공표되는 정보보다 더욱 자세한 정보를 공개해야 하는데, 현실적으로는 기업경영과 영업상의 기밀 및 기업보안 또는 정부사업의 효율적 운영을 위하여 모든 정보를 공개할 수 없는 실정이다. 기업의 기밀을 유지하고, 예산의 투명성을 만족시키는 공표 가능한 정보의 수준을 결정하는 데에 본 연구에서 제안한 추론통제 방법을 활용할 수 있을 것으로 생각한다.

둘째, 정부의 각종 자격평가 및 심사의 경우를 살펴보면 신청경쟁률(인정경쟁률)과 선정·탈락 여부만을 공표하고 있는 실정이다. 특히 선정·탈락 여부 공표시 각 항목의 평가점수에 대한 구체적 정보는 공개하지 않고 있는데, 이는 사업의 불공정성과 의혹을 일으킬 소지가 된다. 본 연구에서 제안한 추론통제 방법을 활용하면, 신청측면에는 각 항목의 평가점수에 대한 평균값 등을 공개함으로써 기업의 부족한 면을 발굴

개발할 수 있는 동기부여를 할 수 있고, 정책입안 측면에는 사업효율성 개선 등에 활용 가능한 기초자료 등을 제공할 수 있으리라 본다.

본 연구에서 제안하고 있는 새로운 추론통제 방법의 적용 분야는 이상 언급한 정부의 연구개발 예산, 자격심사 등에 국한되지 않고 아주 폭 넓을 것으로 생각한다. 특히 정부의 연구개발 예산 및 동향 분석, 각종 평가 및 심사 등의 공표자료는 주로 평균값과 횟수로서 공개되고 있는 현실은 감안할 때, 본 연구에서 제안하고 있는 추론통제 방법은 특히 셈 집계함수 COUNT와 평균 집계함수 AVG가 많이 적용되는 분야에서 뛰어난 분석 능력을 발휘할 수 있으리라 생각한다.

참고문헌

[1] L. Brankovic, M. Miller, P. Horak, and G. wrightson, "Usability of Compromise-Free Statistical Databases for Range Sum Queries", Scientific and Statistical Database Management, pp. 144-154, 1997.

[2] D. Denning and J. Schloerer, "Inference Controls for Statistical Databases". IEEE Computer, Vol. 16, No. 7, pp. 69-82, 1983.

[3] L. Beck, "A Security Mechanism for Statistical Databases", ACM Transactions on Database Systems, Vol. 5, No. 3, pp. 316-338, Sept. 1980.

[4] L. Wang, S. Jajodia, and D. Wijesekera, "Preserving Privacy in On-Line Analytical Processing(OLAP)", Springer, pp. 37-51, 2007.

[5] L. Wang, S. Jajodia, and D. Wijesekera, [4], pp. 127-131.

[6] Incremental Maintenance for Non-Distributive Aggregate Functions, *http://seminars.di.uoa.gr/infosys/palpanas*

[7] J. Gray et al., "Data Cube: A Relational Algorithm Operator Generalizing Group-By, Cross-Tab, and Sub-Totals", Data Mining and Knowledge Discovery, Vol.1, pp. 29-53, 1997.

[8] 이승현, 이덕성, 최인수, "OLAP 큐브에서의 집계함수 AVG의 적용", 한국컴퓨터정보학회논문지, 제14권, 제1호, 217-228쪽, 2009년 1월.

[9] A. Casali, R. Cicchetti, and L. Lakhal, "Cube Lattices: A Framework for Multidimensional Data Mining", Proceedings of the 3rd SIAM International Conference on Data Mining, SDM, pp. 304-308, 2003.

[10] L. Lakshmanan, J. Pei, and J. Han, "Quotient Cube: How to Summarize the Semantics of a Data Cube", Proceedings of the 28th VLDB Conference, 2002.

[11] 유한주, 이덕성, 최인수, "비유일 외래키 조합 복합키 기반의 사실 테이블 모델링과 MDX 쿼리문 작성법", 한국컴퓨터정보학회논문지, 제12권, 제1호, 177-188쪽, 2007년 3월.

[12] 유한주, 최인수, "장바구니 분석용 OLAP 큐브 구조의 설계" 한국컴퓨터정보학회논문지, 제12권, 제4호, 179-189쪽, 2007년 9월.

[13] L. Wang, S. Jajodia, and D. Wijesekera[4], p. 121.

[14] L. Wang, S. Jajodia, and D. Wijesekera, "Securing OLAP Data Cubes Against Privacy Breaches", Proceedings of the 2004 IEEE Symposium in Security and Privacy, 2004.

[15] L. Wang, S. Jajodia, and D. Wijesekera, [4], pp. 131-136.

[16] A. Shoshani, "OLAP and Statistical Database: Similarities and Differences", Principles of Database Systems, pp. 185-196, 1997.

저 자 소 개

이 덕 성

1990: 전남대학교 산업공학과 공학사
1995: 전남대학교 산업공학과 공학석사
현재: 숭실대학교 대학원 산업·정보시스템공학과 박사과정
관심분야 : MIS, DW, OLAP, MDX, CRM

최 인 수

1985: 서울대학교 산업공학과 공학박사
현재: 숭실대학교 산업·정보시스템공학과 교수
관심분야 : MIS, DW, OLAP, MDX, CRM