저차원 집계 테이블들을 사용한 고차원 데이터의 온라인 분석

최혜정[U] 김 명
이화여자대학교컴퓨터학과 고성능인터넷지식공학연구실
{choihj, mkim}@ewha.ac.kr

Analysis of High Dimensional Data using Low Dimensional Summary Tables

HaeJung Choi[U] Myung Kim
Dept. of Computer Science & Engineering, Ewha Womans University

요 약

다차원 데이터를 온라인으로 분석하기 위해서는 사전에 집계 테이블들을 계산해 둔다. 대용량 고차원 데이터의 경우는 집계 테이블의 분량이 천문학적으로 방대하기 때문에 사전 집계 계산이 현실적으로 불가능한 경우가 많다. 고차원 데이터 처리에 관한 연구로는 데이터의 차원 수를 감소시키거나 인덱스를 압축하여 질의처리 시간을 단축하려는 연구를 들 수 있는데, 이러한 방법들은 고차원 데이터의 온라인 분석시에 발생하는 데이터 폭발 현상을 근본적으로 해결하지는 못한다. 본 연구에서는 고차원 데이터가 분석될 때 실제로 저차원 집계 테이블들이 주로 사용된다는 점에 착안하여 데이터 폭발 현상을 감소시키면서 데이터를 분석하는 방안을 제시한다. 이 방법은 사전 집계 연산을 할 때 크기가 방대한 고차원 집계 테이블들의 생성을 생략하고, 3~6차원 또는 그 이하 차원의 집계 테이블들만을 고속으로 동시에 생성하는 방법이다.

1. 서론

기업 전략을 세우기 위한 비즈니스 데이터나 고객의 성향을 파악하기 위한 전자상거래 시스템의 데이터는 고차원으로 구성된 대용량 데이터인 경우가 많다. 이러한 고차원 데이터를 온라인 분석(OLAP, Online Analytical Processing)하는 것은 3~6차원 저차원 데이터를 분석할 때와는 달리 큰 어려움을 야기시킨다[1]. 그 이유는 OLAP 질의 처리가 시작되기 전에 분석 결과의 일부인 집계 테이블들을 미리 계산하여 저장해 두는데, 이 때 생성할 집계 테이블의 수와 데이터 분량이 방대하기 때문이다. n차원 데이터의 경우에 생성할 집계 테이블의 개수는 2^n이고, 각 차원의 크기를 d라고 했을 때 생성되는 데이터의 최대 셀 개수는 $(d+1)^n$ 이 된다. 예를 들어, 각 차원 크기가 100인 20차원 데이터의 경우를 살펴보면, 약 1백만 개에 이르는 집계 테이블이 생성되고, 차원이 계층 구조가 없는 단순 구조라고 하더라도 최대 10^{40} 셀이 생성되어 데이터 저장 자체도 쉽지 않은 상황이 된다.

고차원 데이터의 처리에 대한 기존의 연구는 다음을 들 수 있다. 고차원 데이터를 저장하고 검색하기 위해 R-트리 인덱스 구조[2]를 사용하기도 하는데, 이는 대상 데이터의 차원 수가 높아질수록 질의 처리 시간이 급증하는 문제점을 안고 있다. 이를 해결하는 방법으로 인덱스를 압축[3]하거나, 데이터 변환을 통해 데이터의 차원 수를 미리 줄여서 데이터의 저장 공간과 질의 시간을 단축시키는 연구 결과[4]가 있다. 또한 이와 같은 인덱스의 성능을 실제 환경에서 평가하기 위해 데이터와 질의를 생성하는 연구[5]가 있으며 이 연구는 데이터의 분포를 사용자가 다양하게 제어할 수 있도록 한다. 그러나 지금까지 살펴 본 방법들은 고차원 데이터를 저장하고 검색하는데는 유용하지만 데이터의 온라인 분석에 쓰일 때 데이터의 폭발 현상을 근본적으로 해결하는데는 한계를 보인다.

본 연구에서는 분석가의 실제 분석 패턴을 고려하여 고차원 데이터를 시간, 공간적 측면에서 효율적으로 분석하는 방안을 제시한다. 고차원 데이터가 주어지는 경우라고 해도 분석가는 한꺼번에 3~6개 정도의 저차원들을 다룬다는 점에 착안하여 저차원 집계 테이블들만을 고속으로 동시에 생성하는 방법이다. 이를 통해 데이터의 폭발 현상을 줄일 수 있고, 고차원 데이터의 분석이 신속하게 이루어질 수 있도록 하였다.

2절에서 본 연구에서 제안하는 저차원 집계 테이블을 사용한 OLAP의 유용성을 설명하고, 3절에서 저차원 집계 테이블들을 생성하는 방법과 간단한 분석 결과를 제시하고 4절에서 결론을 맺는다.

* 본 연구는 한국과학기술기획평가원 연구기반확충사업 (과제번호: M10022040004-01G0509-00210)의 지원으로 수행되었음

2. 저차원 집계 테이블들을 사용한 OLAP의 유용성

저차원 집계 테이블들을 사용하여 고차원 데이터를 분석하는 방법의 유용성에 관해 예를 들어 설명하고자 한다. 마이크로소프트사의 OLAP 툴인 Analysis Service가 샘플로 제공하는 데이터 웨어하우스인 FoodMart를 사용하기로 한다. FoodMart의 데이터(사실 테이블)는 그림 1과 같은 'Sales' 테이블이다. 이는 미국, 멕시코, 캐나다 등지에 판매망을 가지고 있는 대형 식품 유통 체인으로 15차원 데이터이다. 각 차원의 멤버의 수는 그림 1과 같다.

예제 데이터로부터 15차원 큐브를 생성하는 경우, 집계 테이블 개수와 큐브 크기는 다음과 같다. 밀집 데이터의 경우에 큐브는 최대 10^{23} 셀이 되며, 집계 테이블의 개수는 $2^{15} = 32,768$ 개가 된다. 그림 2는 집계 테이블 개수를 각 차원별로 나타낸다. 1~5차원 집계 테이블 개수는 전체 집계 테이블의 수에 비해 상대적으로 작다. 그림 3은 집계 테이블이 차지하는 공간을 차원별로 표시한다. 10차원 이상의 집계 테이블이 공간의 거의 대부분을 차지한다.

번호	차원	멤버 수
1	Customer	100,000
2	Product	1,000
3	Time	1,000
4	Promotion	47
5	Store	25
6	Promotion Media	14
7	Product Stock	10
8	Occupation Type	10
9	Ad. Budget	10
10	Yearly Income	8
11	Store Size	6
12	Store Type	5
13	Education Level	5
14	Marital Status	2
15	Gender	2

그림 1. 예제 데이터: 15차원 큐브의 각 차원 멤버의 수.

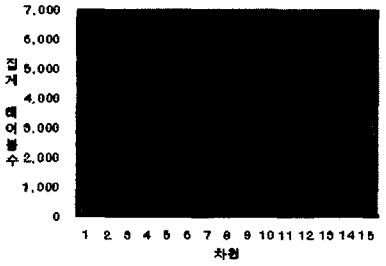

그림 2. 15차원 큐브의 각 차원별 집계 테이블 수.

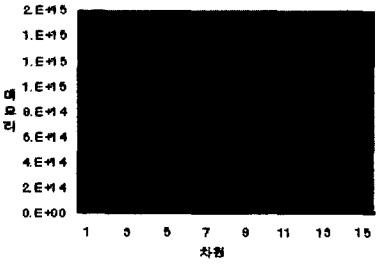

그림 3. 15차원 큐브의 각 차원별 필요한 메모리.

이제, FoodMart 데이터 상에서 흔히 일어날 수 있는 질의들을 살펴보자.

* 마케팅 부서에서 사용하는 질의의 예제
  - Q_1: 시간별 매장별 상품별 매출량 분석
  - Q_2: 가장 잘 팔리는 상품(상점)에 사용된 판촉 활동 분석
  - Q_3: 잘 팔리는 상품(상점)에 대한 고객의 연간 소득(직업, 결혼 여부, 성별, 교육 등)이 미친 영향 분석
* 매장 책임자나 제조업자가 하는 질의의 예제
  - Q_4: 제조업자한테 상품에 대한 주문을 하기 위해 시간별 상품별 재고량을 사전에 분석
  - Q_5: 상품이 잘 팔리는 매장에서의 광고 예산과 매장의 특징 분석.

이와 같은 질의는 그림 4와 같이 3차원 집계 테이블 상에서 이루어진다. 예를 들어 Q_1은 그림 3(a)의 Product, Time, Store의 차원으로 구성된 집계 테이블을 통해 처리된다. Q_2는 Q_1을 통해 얻은 특정 상품(매장)의 판촉 활동을 분석할 때 (b)의 집계 테이블을 통해 분석할 수 있고, Q_3도 같은 방식으로 (c)와 (f)를 통해, Q_4와 Q_5는 각각 (d)와 (e)를 통해 처리할 수 있다. 이와 같은 예제를 통해 저차원 집계 테이블을 사용한 온라인 분석의 유용함을 알 수 있다.

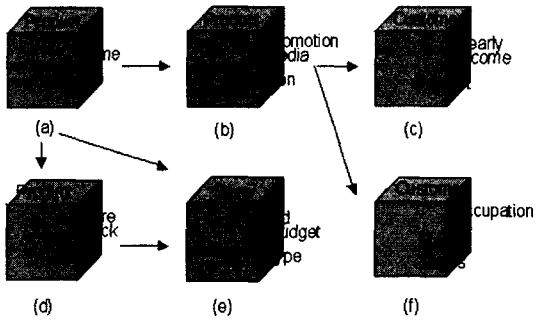

그림 4. 저차원 집계 테이블들을 사용한 OLAP 예제.

위의 예제와 같이 주어진 데이터가 비록 15차원 데이터라고 하더라도 분석가는 한 번에 3~5차원 정도의 차원만을 분석에 이용하는 경우가 많다. 따라서 고차원 데이터로부터 저차원 집계 테이블들만을 고속 생성하여 분석에 이용하는 것은 유용한 분석 방법이라 하겠다.

3. 저차원 집계 테이블들의 생성 방법과 분석

2절의 분석 결과를 바탕으로, 본 연구에서는 고차원 데이터로부터 특정 차원 이하의 저차원 집계 테이블들을 모두 생성하는 방법을 설계하고자 한다. 예를 들면, 그림 5는 $A,B,C,\cdots,Z$ 의 26개 차원으로 구성된 사실 테이블로부터 5차원 이하의 모든 집계 테이블을 생성하는 모습을 개념적으로 나타낸다. 그림에서 $ABCDE$로 표시된 트리는 $ABCDE$를 루트로 생성 가능한 5차원 이하의 집계 테이블들을 나타낸다.

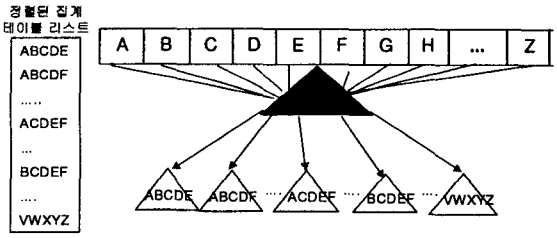

그림 5. 저차원 집계 테이블의 생성

n차원 데이터(사실테이블)로부터 k, $0 \le k \ll n$, 차원 이하의 저차원 집계 테이블들을 생성하려고 한다고 하자. k차원 집계 테이블들은 주어진 사실테이블로부터 계산되고, k차원 미만의 집계 테이블들인 i, $0 \le i < k$, 차원 집계 테이블들은 $i+1$차원 집계 테이블로부터 계산된다. 우선 k차원 집계 테이블들을 생성하는 방법을 소개한다. 집계 테이블의 크기가 대체로 작은 경우와 그렇지 않은 경우를 구별하여 설명하기로 한다.

(1) k차원 집계 테이블의 크기가 대체로 작은 경우

예를 들어, 그림 1의 데이터로부터 5차원 집계 테이블들을 모두 생성한다고 할 때, 차원 4~15로만 구성된 집계 테이블들은 그 크기가 600셀 ~ 1.65M 셀이 된다. 이러한 크기의 집계 테이블들은 한꺼번에 수십~수백 개가 생성될 수 있다. 사실 테이블과 집계 테이블의 차원의 수가 큰 차이를 보이므로 사실 테이블은 희박해도 집계 테이블의 밀도는 높기 때문에 집계 테이블을 배열 형태로 메모리에 펴놓으면 사실 테이블을 임의 순서로 스캔하면서 임의의 집계 테이블들을 동시에 생성할 수 있다.

집계 테이블의 생성 순서와 집계 연산에 걸리는 시간을 살펴보자. 집계 테이블들이 사실 테이블로부터 직접 생성되므로, 집계 연산 시간이 사실 테이블 스캔 회수에 비례한다. 집계 테이블들은 테이블 이름의 사전식 순서 (lexicographic order)로 생성하며 메모리가 허용하는 한 많은 집계 테이블들을 한꺼번에 생성한다. 메모리 크기, 집계 테이블의 개수와 각 집계 테이블의 크기가 정해져 있는 경우에 최적의 순서를 찾는 것은 'minimum finish time non-preemptive job scheduling' 하는 문제와 같은 종류로써 NP-hard 문제에 속한다. 따라서 휴리스틱한 방법으로 본 연구에서는 사전식 순서를 택하기로 한다. 그림1의 예제 데이터에서 차원 4~15만으로 구성된 집계 테이블을 생성하는 경우에 100개의 집계 테이블 생성이 한꺼번에 이루어진다고 하면 평균적으로 30회 정도의 사실 테이블 스캔으로 5차원 집계 테이블들을 모두 생성할 수 있다.

(2) k차원의 집계 테이블의 크기가 큰 경우

그림 1의 예제 데이터를 보면 차원 1~3이 포함된 k 차원 집계 테이블들은 (예제에서 k= 5임.) 상당히 커서 메모리 상에서 전체 집계 테이블이 생성되기 힘들다. 이와 같이 한꺼번에 생성될 수 없는 집계 테이블들이 있는 경우는 다음 방법을 사용한다. 주어진 데이터가 n차원이고, 사실테이블은 D_i, $1 \le i \le n$차원들로 구성되어 있으며, 차원 간의 멤버의 수는 $|D_1| \ge |D_2| \ge |D_3| \ge \cdots \ge |D_n|$ 이고 멤버의 수가 특정 개수를 넘는 차원 개수를 j개라고 하자. $D_1, D_2, \cdots, D_j$가 이 경우에 속한다. 이 때는 먼저 사실 테이블을 D_1 , D_2 , . . . , D_j 기준으로 파티션 해놓는다. 즉 D_1 , D_2 , . . . , D_j 조합의 값마다 해당 파티션을 직접 엑세스할 수 있다. 즉, 그림 1의 경우 사실 테이블을 첫 3열을 기준으로 파티션이 된다.

이제 k차원 집계 테이블을 생성하는 방법을 살펴보자. 그림 1의 예제에서 차원 1, 2, 3, 12, 13차원으로 구성된 집계 테이블을 생성하고자 한다면 사실 테이블이 차원 1, 2, 3을 기준으로 파티션되어 있으므로 집계 테이블을 생성하기 위해서는 12차원과 13차원으로 구성된 2차원 테이블만 메모리에 상주해 있으면 된다. 차원 1, 2, 3의 값이 변화될 때마다 메모리의 집계 테이블 값을 디스크로 쓴 후에 메모리를 재사용하면서 집계 테이블 1, 2, 3, 12, 13을 완성할 수 있다. 이 때 필요한 메모리는 5× 5인 25셀이다. 차원 1, 2, 3이 포함된 5차원 집계 테이블들은 ${}_{12}C_2$ = 66개가 있으며 이들은 모두 2차원 집계 테이블 분량의 메모리 재사용을 통해 동시에 생성이 가능하다.

j개의 차원을 기준으로 사실 테이블을 파티션을 해 놓은 경우에 사실 테이블을 2^j회 스캔하게 된다. 그러나 본 연구에서 생성하는 집계 테이블들은 4~5차원 정도의 작은 집계 테이블들이고, j 또한 2~4차원 정도가 될 것이므로, 2^j는 4~16회 정도가 된다. 사실 테이블을 재정렬하지 않고 수 백개의 k차원 집계 테이블들이 생성될 수 있다는 점에서 이는 큰 오버헤드라고 볼 수 없다.

k차원 집계 테이블에 파티션 기준이 되는 첫 j 차원 중에서 m개 만이 포함되어 있는 경우, 메모리 오버헤드를 살펴보자. 그러한 집계 테이블들은 ${}_{n-m}C_{k-j}$ 개 존재하며, 각 집계 테이블의 크기는 멤버의 수가 작은 차원들로만 구성된 $k-m$차원 배열이 된다. 즉 첫 3차원이 모두 포함된 경우는 2차원 배열이, 2차원만 포함된 경우는 3차원 배열이 된다. 멤버의 수가 작은 차원들로만 구성된 이러한 배열은 수십 개 또는 수백 개 정도가 메모리 상에서 한꺼번에 생성이 가능하다. 알고리즘의 특성 상, 한 번 생성된 집계 테이블 조각들은 완성된 후에 하드 디스크에 출력되고, 다시 메모리에 불려 들여서 계산에 사용되는 것이 아니므로 효율적이라고 볼 수 있다.

(3) k차원 미만의 집계 테이블들 생성

k차원 미만의 집계 테이블들은 이미 생성되어 저장된 k차원 집계 테이블들로부터 생성된다. 이들은 크기가 가장 작은 부모들로부터 생성이 되며, k차원 집계 테이블보다 현저히 작기 때문에 최소 부모 노드로부터 가능한 많은 집계 테이블을 계산함으로써 계산 시간을 줄일 수 있다.

4. 결론

기업의 고차원 데이터로부터 비즈니스 전략에 사용될 수 있는 분석 결과를 효율적으로 얻으려면 방대한 양의 데이터를 사전에 계산해 두어야 하며, 이로 인해 메모리와 사전 연산 시간에 큰 오버헤드가 발생하게 된다. 본 연구에서는 분석자의 입장에서 고차원 데이터가 어떻게 취급되는가를 살펴 보아, 그 결과 저차원 집계 테이블들을 사용하여 데이터를 효율적으로 OLAP을 할 수 있는 방안을 제시하였다. 사전 집계 연산을 시간적, 공간적 측면에서 효율성을 높였다. 본 연구에서 제안한 방법은 10~20차원 정도의 데이터에 적용가능한 방법이며, 100차원 이상의 데이터 분석에도 원하는 차원들을 추출하여 확장 적용 가능한 방법이다. 실제 비즈니스 환경에서는 여러 부서의 데이터로부터 전사적 데이터 웨어하우스를 구축하여 각 부서의 데이터를 위와 같은 방법으로 공유함으로 시간, 공간적인 자원의 낭비를 막을 수 있다.

5. 참고문헌

[1] MicroStrategy, Inc., "The Case for Relational-OLAP," White Paper, http://www.microstrategy.com/files/whitepapers/wp_rolap.pdf, 2000.

[2] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Int. Conf. ACM SIGMOD on Management of Data, 1984.

[3] S. Berchtold, C. Bohm, H.V. Jagadish, H.-P. Kriegel, J. Sander, "Independent Qualtization: An Index Compression Technique for High-Dimensional Data Space," 16th Int. Conf. on Data Engineering (ICDE), 2000.

[4] C. C. Aggarwal. "On the Effects of Dimensionality Reduction on High Dimensional Similarity Search," ACM PODS Conf, 2001.

[5] 김상욱, 이헌길, "고차원 데이터 공간내에서 데이터 및 질의의 생성," Journal of Telecommunications and Information, Vol. 5, pp. 98-105, 2001.