# 불완전 데이터를 위한 효율적 top-k(g) 스카이라인 그룹 질의 처리 기법

박미라, 민준기
*한국기술교육대학교 정보미디어공학과
e-mail: happypmr@kut.ac.kr, jkmin@kut.ac.kr

# An Efficient Processing of Top-k(g) skyline group queries for Incomplete Data

Mi-Ra Park, Jun-Ki Min
Dept of Information Media Engineering, Korea University of Technology and Education

요 약

대부분의 스카이라인 질의에 대한 연구는 완전한 데이터에 관하여 이루어지고 있다. 하지만, 우리가 웹이나 기타 다른 도구로 데이터베이스에 자료를 입력할 때는 null을 허용하는 부분이 존재한다. 현재 이런 불완전한 데이터를 처리하기 위한 많은 연구가 이루어지고 있다.

본 논문에서는 이러한 문제를 해결하기 위하여 기존에 제안되었던 불완전한 데이터를 처리하는 기법과 차원의 저주를 해결하기 위한 기법을 고려하여 이를 바탕으로 완전한 데이터와 동등하거나 혹은 더 좋을지도 모르는 데이터를 우선순위가 높은 순서대로 k(g)개 검색해주는 스카이라인 그룹 질의를 도입하고 이를 처리하는 방법을 제안한다.

## 1. 서론

처리하는 데이터가 방대해짐에 따라 사용자가 원하는 모든 조건에서 관심이 적은 데이터를 제외한 나머지 데이터를 검색하는 스카이라인 질의(skyline query)의 필요성이 증대되고 있다. 스카이라인 질의는 전체 객체 집합에서 대상 객체의 여러 속성을 다른 객체가 지배하지 않는 관심 있는 객체 집합을 검색한다[1].

대부분 스카이라인 질의에 대한 연구는 완전한 데이터에 관하여 이루어지고 있다. 하지만 우리가 웹이나 기타 다른 도구로 데이터베이스에 자료를 입력할 때는 null을 허용하는 부분이 존재한다. 현재 이런 불완전한 데이터를 처리하기 위한 많은 연구가 이루어지고 있다.

아래는 본 연구에서 연구하고자 하는 동기가 된 스카이라인 그룹 질의가 필요한 예를 소개한다.

**예제 1.** 어떤 큰 회사에서 신입사원을 채용할 때 인터넷 전형을 사용하는 예를 고려해보자. 수천명의 지원자의 이력서가 데이터베이스로 전송되고, 회사의 인사 담당자가 모든 사람의 이력서를 검토하는 것은 불가능하다. 이런 경우에 회사에서 원하는 사원을 적절하게 뽑기 위해 스카이라인 질의가 필요하다.

회사에 이력서를 쓰는 사람들은 서류전형을 통과하기에 불리한 이력은 기입하지 않는다. 지원자가 기입하지 않은 공란은 데이터베이스에서 빈 공간으로 기록하게 되고, 공란 없이 기입한 사람과는 달리 데이터베이스에 null 값으로 남게 된다.

회사마다 사원을 채용할 때 우선순위로 생각하는 기준은 다르다. 예를 들어 어떤 회사는 영어를 잘하는 사람을 원하고, 어떤 회사는 학점이 높고 수상 경력이 많은 사람을 원할지도 모른다. 학점이 높고 수상 경력이 많은 사람을 원하는 회사가 토익 점수도 추가로 고려할 때를 생각해보자. 토익점수를 기록하지 않은 사람이 0점이라고 볼 수는 없을 것이다. 점수를 기재하지 않았더라도 토익점수를 기재한 사람의 점수가 기준보다 낮다면 토익점수를 기록한 다른 지원자보다 영어를 잘할지도 모른다.

본 논문에서는 완전한 데이터와 동등하거나 혹은 더 좋을지도 모르는 데이터를 함께 검색해주는 스카이라인 그룹 질의를 도입하고 스카이라인 그룹 질의를 처리하는 방법을 제안한다. 또한, 불완전한 데이터를 그룹 지을 때 우선순위를 고려하는 효율적인 방법을 제안한다. 제안한 기법은 차원이 커지면 모든 데이터가 스카이라인이 되는 '차원의 저주(curse of dimensionality)' 문제를 해결하기 위해 기존에 연구된 Telescope알고리즘[2]을 활용한다.

## 2. 관련 연구

### 2.1 불완전한 데이터를 위한 스카이라인 질의 처리[3]

Khalefa 등은 불완전한 데이터를 위한 스카이라인 질의 처리 알고리즘인 ISkyline[3]을 제안하였다. ISkyline알고리즘은 Virtual Points와 Shadow Skylines라는 개념을

이용하여 버킷(Bucket) 알고리즘에 불완전한 데이터의 스카이라인질의를 적용할 때 발생하는 결점을 극복한다.

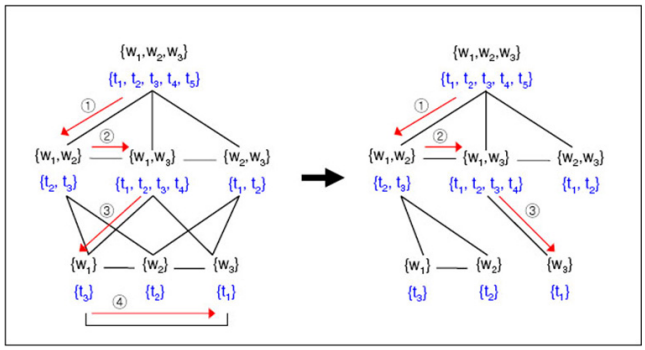
버킷 알고리즘은 모든 들어오는 데이터를 같은 비트 표현을 가지는 중복되지 않는 버킷들로 나눈다. 그리고 버킷별로 스카이라인을 구한다. 각 버킷을 위한 스카이라인들의 집합을 로컬 스카이라인이라고 부른다. 모든 로컬 스카이라인 집합들로부터 객체들을 모으고 그것을 하나의 리스트로 만든다. 이를 후보 스카이라인이라고 부른다. 후보 스카이라인들의 사이즈는 모든 버킷들에서 모든 지역 스카이라인들을 병합함으로 과도하게 클지도 모른다. 그리고 각 버킷에 로컬 스카이라인은 모든 다른 버킷들로부터 독립적으로 계산된다.

위와 같은 단점을 극복하기 위해, 즉 후보 스카이라인 리스트에서 객체들의 수를 줄이고자 virtual points의 개념을 사용한다. 어떤 버킷에 있는 스카이라인을 가져와서 현재 버킷의 가상 스카이라인으로 만들어 비교한 후, 스카이라인이 되는 것만 후보 스카이라인 리스트에 저장한다. shadow skylines를 사용하여 후보 스카이라인 리스트에서 모든 객체들의 쌍대 비교(pairwise comparison)를 수행하지 않고 질의 결과를 얻을 수 있다.

### 2.2 Telescope 알고리즘[2]

모든 차원에서 지배되지 않는 자료들을 스카이라인으로 선정하다보면, 차원이 많아질수록 모든 자료가 스카이라인이 되는 '차원의 저주' 문제가 발생하게 된다. 이 문제점을 해결하기 위해 차원에 우선순위를 두고 우선순위가 높은 k개만을 검색하고자 제안된 알고리즘이 Telescope 알고리즘이다.

이 알고리즘은 (그림 1)에서처럼 일반적인 lattice 그래프를 left-skewed 그래프로 변환하여 top-down 방식으로 검색을 하며 top-k 스카이라인을 찾는 알고리즘이다. 최상위 노드부터 검색하여 스카이라인 객체와 결과 집합에 들어있는 스카이라인 객체의 합집합이 k개보다 작으면 현재 노드의 스카이라인을 결과 집합에 넣은 후 그래프가 이어진 다음 노드로 이동하고, k개보다 크면 그래프에 자식노드로 이동한다. 결과 집합의 수가 k개가 되거나 그래프에서 더 이상 이동할 노드가 없으면 결과 집합을 반환하고 검색을 마친다.




(그림 1) lattice 그래프와 left-skewed 그래프.

## 3. 배경지식

본 연구를 설명하기 위하여, 표 1에 스카이라인의 정의에 사용된 기호들을 정의하였다.

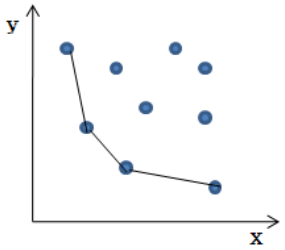
<표 1> 사용되는 표기법들

| 표기 | 정의 |
|---|---|
| $S$ | The dataset |
| $D$ | The dimension set |
| $t_i$ | A tuple in $S$ |
| $d_i$ | A data dimension($1 \le i \le n$) |
| $t_i(d_j)$ | The value of a tuple $t_i$ on $d_j$ |

스카이라인은 기존 연구들[1,2,4,5,6]에서 다음과 같이 정의된다.

**정의 1 (지배).** 만약 $\forall d_k \in D$, $t_i(d_k) \ge t_j(d_k)$ 그리고 $\exists d_s \in D$, $t_i(d_s) > t_j(d_s)$ 이면, 튜플 $t_i$는 튜플 $t_j$를 지배(dominate)한다.

**정의 2 (스카이라인).** 만약 어떤 다른 튜플들 $\forall t_j (\neq t_i) \in S$ 가 $D$에 있는 $t_i$를 지배하지 않으면, 튜플 $t_i$는 $D$에 스카이라인 객체이다.




(그림 2) 스카이라인

(그림 2)는 스카이라인 정의에 따라서 표현한 스카이라인 그림이다. 본 연구에서는 이전의 연구들에서 정의한 스카이라인 정의를 그대로 적용한다.

## 4. 스카이라인 그룹 질의

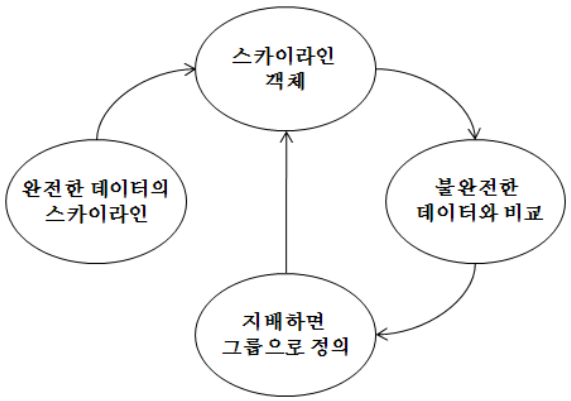
### 4.1 스카이라인 그룹

기존의 연구[3]는 불완전한 데이터를 완전한 데이터와 섞어서 스카이라인을 검색하는 것이 기본 생각이다. 이와는 달리 본 논문에서는 완전한 데이터의 스카이라인 질의에 추가로 불완전한 데이터를 그룹 질의하여 검색한다. 불완전한 데이터가 완전한 데이터의 그룹이 되어 검색할 때 추가적으로 보여주기 위함이 기본 생각이다. 이를 위해서 스카이라인 그룹이라는 새로운 개념을 사용하고, 아래와 같이 정의한다.

**정의3 (스카이라인 그룹).** 불완전한 객체가 값을 가지고 있는 차원에 한해서 스카이라인 객체를 지배하면 그 불완전한 객체는 완전한 데이터의 스카이라인 객체의 그룹이 된다.

불완전한 데이터의 값을 가지고 있는 차원의 값이 완전한 데이터의 객체를 지배한다는 것은 그 값이 0일지라도 스카이라인이 됨을 뜻한다. 이러한 불완전한 데이터는 완전한 데이터를 지배하거나 혹은 적어도 한 개 이상의 차원에서 완전한 데이터를 지배하는 스카이라인이 된다. 그러므로 우리는 불완전한 데이터 보다 더 좋을지도 모르는 데이터를 함께 보여주기 위해 스카이라인 그룹이라는 개념을 사용하여 스카이라인을 검색한다.

불완전한 데이터는 여러 개의 완전한 데이터의 스카이라인 그룹이 될 수 있고, 완전한 데이터가 불완전한 데이터의 스카이라인 그룹이 되지는 않는다.



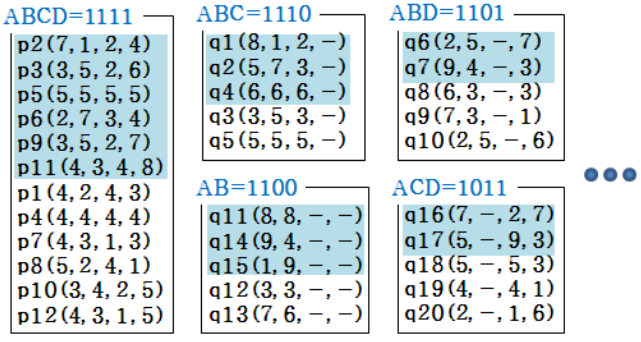

(그림 3) 스카이라인 그룹을 구하는 방법

(그림 3)은 스카이라인 구룹을 구하는 방법을 간단히 도식화하여 표현한 것이다.

### 4.2 완전한 데이터의 top-k 스카이라인 질의

관련연구 2.1에서 사용한 버킷 알고리즘의 기본적인 개념을 완전한 데이터의 스카이라인 질의에 응용한다. 데이터의 값이 있는 부분을 1, 없는 부분을 0과 같이 1과 0의 2비트로 표현할 때, 비트 표현이 같은 데이터를 같은 버킷으로 만든다. 예를 들어, q1(8, 1, 2, -), q2(5, 7, 3, -) 두 객체가 있다면 이것의 비트표현은 1110이 되고, 두 객체는 비트표현이 1110인 데이터들과 같은 버킷이 된다.

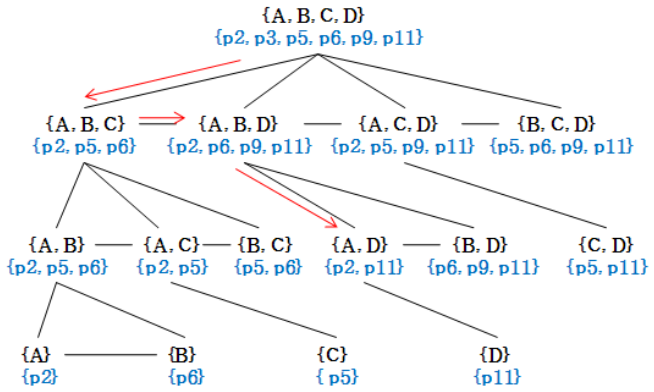
(그림 4)는 데이터의 버킷들을 표현한 것이고, 색칠된 부분은 각 버킷에서의 스카이라인 객체이다. 버킷의 개수는 최대 $2^n$-1개가 된다. 버킷별로 스카이라인을 미리 구하여 스카이라인 그룹 질의에 사용할 수 있도록 유지한다.




(그림 4) 데이터의 분류

완전한 데이터들의 스카이라인 질의를 수행하는 과정은 기존에 연구된 방법을 활용한다. 본 연구에서는 '차원의 저주' 문제를 피해 사용자가 선호하는 k개의 데이터를 검색하기 위해 (그림 4)에서 만든 버킷들 중 비트표현이 1로만 구성된 버킷에 Telescope알고리즘을 적용한다.

예를 들어, 4차원의 데이터를 고려하여 (그림 5)와 같이 left-skewed 그래프로 나타낼 수 있다. 왼쪽 자식 노드에서 검색한 스카이라인은 자손들의 스카이라인 객체들을 모두 포함하므로 하나의 부모 노드에 연결된 자식 노드는 형제노드의 자식 노드와는 연결하지 않는다.




(그림 5) 4차원 데이터에 Telescope 알고리즘 적용

완전한 데이터의 스카이라인은 p2, p3, p5, p6, p9, p11이다. 그래프에서 부모노드의 자식노드는 부모노드의 부분집합 중 하나가 된다. 부모노드에 있는 객체들 중 검은색 글자로 표시한 차원을 고려하여 스카이라인을 구하여 그래프를 완성한다. (그림 5)은 A, B, C, D 차원의 순서로 사용자가 선호한다고 가정하고 만든 그래프이다.

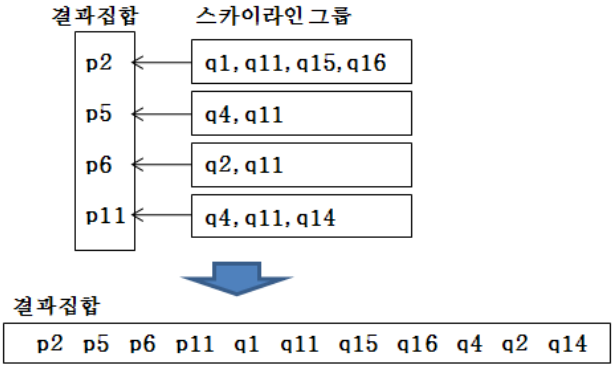
빨간색으로 표시한 화살표는 우선순위를 고려하여 상위 4개를 검색하고자 할 때 구해지는 과정이다 (즉 top-k에서 k를 4로 설정). 최상위 노드의 스카이라인 집합의 개수가 4개보다 크므로 그래프가 연결된 다음 노드로 이동한다. {A, B, C} 차원을 고려한 스카이라인 집합은 {p2, p5, p6} 이다. 이는 4개보다 작으므로 결과 집합에 넣은 후 다음 노드로 이동한다. {A, B, D} 차원의 스카이라인 집합인 {p2, p6, p9, p11}과 결과 집합을 합집합하면 {p2, p5, p6, p9, p11}이 되어 4보다 크므로 자식노드로 이동한다. {A, D} 차원의 스카이라인 집합인 {p2, p11}을 결과 집합과 합집합하면 {p2, p5, p6, p11}이 되어 검색하고자 했던 개수와 같아진다. 현재노드의 스카이라인 객체 중 결과 집합에 포함되어있지 않은 객체를 결과 집합에 삽입한다. 결과 집합에 포함된 스카이라인 객체는 완전한 데이터의 우선순위를 고려한 top-k 스카이라인이 된다.

### 4.3 불완전한 데이터의 top-k(g) 스카이라인 그룹 질의

스카이라인을 검색할 때 완전한 데이터와 함께 동등하거나 더 좋을지도 모르는 불안전한 데이터를 보여주기 위해 불완전한 데이터를 대응되는 완전한 데이터의 그룹으

로 만든다. 4.2에서 구한 k개의 완전한 데이터의 각 스카이라인 데이터에 대하여 g개의 불완전 스카이라인 데이터를 구한다. 즉, 각 스카이라인 객체는 최대 g개의 원소를 가지는 불완전 데이터 스카이라인 그룹을 갖는다.

스카이라인 그룹을 만드는 과정은 각 스카이라인 객체당 g개 혹은 더 이상 없을 때까지 버킷들의 스카이라인을 검색한다. ABCD라는 4개의 차원을 고려하여 스카이라인 그룹을 구한다면 'ABC->ABD->AB->ACD->AC->AD->A->BCD->BC->BD->B->CD->C->D'와 같은 순서로 선호하는 차원순서와 더 많은 정보가 채워진 순서로 그룹을 찾는다. 이 순서는 비트값이 큰 값을 가지는 버킷이 먼저 검색되는 것이고, 스카이라인 큐브[4]에서 스카이라인을 구할 때 ABC를 검색하기 위해서는 AB, AC, BC가 검색되어야 하고 AB를 검색하기 위해서는 A와 B를 먼저 검색하여야 하는 방식을 역으로 표현한 것과도 같다.

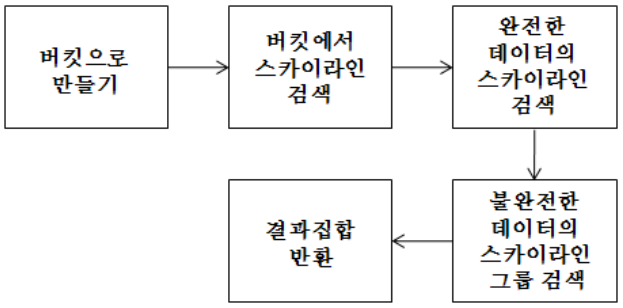



(그림 6) 스카이라인 그룹

(그림 6)는 (그림 5)에서 구한 완전한 데이터의 스카이라인 객체의 g가 4인 경우 top-g 스카이라인 그룹을 구하는 예를 소개한다. 결과 집합의 객체를 한 개씩 가져와서 (그림 4)의 각 버킷별로 구한 스카이라인과 비교한다. p2의 경우, 비트연산자 1110을 가지는 버킷의 스카이라인과 처음으로 비교된다. p2의 ABC차원인 {7, 1, 2}와 ABC 버킷의 첫 번째 스카이라인 객체인 q1{8, 1, 2}를 비교해보면 q1이 p2를 지배함을 볼 수 있다. 그러면 q1은 p2의 스카이라인 그룹이 된다. 다음으로 p2{7, 1, 2}는 ABC 버킷의 두 번째 스카이라인 객체인 q2{5, 7, 3}과 비교된다. q2는 p2를 지배하지 못하므로 다음 스카이라인 객체인 q4로 넘어간다. q4와 비교한 후에는 1110다음으로 비트표현이 큰 1101표현을 가지는 ABD 버킷의 스카이라인 객체로 넘어간다. 이런 방식으로 선호하는 차원 순서대로 스카이라인을 비교해보면 (그림 6)와 같이 스카이라인 그룹을 만들 수 있다.

검색이 끝나면 결과 집합에 포함된 객체들에 스카이라인 그룹으로 지정된 불완전한 데이터를 가져온다. 중복이 되는지 여부를 확인한 후, 결과 집합에 포함시킨다.

(그림 7)은 본 논문에서 제안하는 스카이라인 그룹 질의를 처리하는 방법을 흐름도로 나타낸다. 데이터들을 값이 채워진 비트 표현이 같은 것들끼리 버킷으로 만들고 각




(그림 7) 전체 흐름도

버킷에서 스카이라인을 구한다. 그 다음 telescope 알고리즘을 활용하여 완전한 데이터들의 우선순위를 고려해 k개의 스카이라인 객체를 구하고, 그 스카이라인 객체에 대응하는 각각의 스카이라인 그룹을 구한다. 중복된 스카이라인 그룹을 제외하고 완전한 데이터와 스카이라인 그룹으로 지정된 객체를 반환한다.

## 5. 결론

불완전한 데이터에서 스카이라인을 검색하는 이전의 연구에서는 불완전한 데이터를 완전한 데이터와 섞어서 스카이라인을 검색하였다. 본 논문에서는 완전한 데이터의 스카이라인 질의에 추가로 불완전한 데이터를 그룹 질의하여 검색한다.

불완전한 데이터를 완전한 데이터의 그룹으로 묶여서 검색함으로써 데이터의 손실 없이 사용자가 원하는 우선순위를 고려하여 검색을 가능하게 하였다. 완전한 데이터와 동등하거나 혹은 더 좋을지도 모르는 데이터를 함께 검색해주는 스카이라인 그룹 질의를 라는 개념을 만들었다. 향후 연구 내용으로는 저장 공간을 축소하여 성능을 향상시키고, 스트림 환경에의 적용을 고려한다.

## 참고문헌

[1] Borzsonyi. S., Kossmann, D., Stocker, K. "The Skyline Operator" ICDE, p.421-430, 2001

[2] Jongwuk Lee, Gae-won You, and Seung-won Hwang. "Telescope: Zooming to Interesting Skylines" LNCS 4443, pp.539-550, 2007

[3] Mohamed E. Khalefa, Mohamed F. Mokbel, Justin J. Levandoski, "Skyline Query Processing for incomplete Data" DISC

[4] Yidong Yuan, Xuemin Lin,, Qing Liu, Wei Wang, jeffery Xu Yu, Qing Zhang, "Efficient computation of the skyline cube", VLDB, 2005

[5] Jian Pei, Wen Jin, Martin Ester, Yufei Tao, "Catching the best views of skyline: A semantic approach based on decisive subspaces" VLDB, 2005

[6] Chee-Yong Chan, H.V. Jagadish, Kian-Lee Tan, Authony K.H. Tung, Zhenjie Zhang, "Finding k-Dominant Skylines in High Dimensional Space" SIGMOD, 2006