

범주형 데이터에 대한 스카이라인 질의 알고리즘

(Skyline Query Algorithm in the Categorical Data)

이 우 기 [†] 최 중 호 ^{**}
(Wookey Lee) (Jungho Choi)

송 중 수 ^{**}
(Jongsu Song)

요약 스카이라인 질의는 다차원, 대량의 데이터 검색에서 효율적인 방법이다. '지배한다'의 개념을 활용하여 약 95% 이상으로 알려진 불필요한 데이터 집합을 검색 대상에서 제외하고 필요한 데이터에 집중하게 만들기 때문이다. 지금까지의 스카이라인 질의 알고리즘들은 데이터 집합이 모두 수치형 데이터일 경우에만 한정하여 개발되었다. 따라서 데이터베이스 등에 저장된 대부분의 텍스트 데이터들은 기존 스카이라인 질의 알고리즘을 사용하여 결과를 얻을 수 없었다. 본 연구는 스카이라인 질의의 대상을 범주형 데이터라는 전혀 새로운 영역을 개척한 점에서 의미가 있다. 우선 범주형 데이터 거리를 2종류를 개발하고 이를 스카이라인 질의에 적용하였고, 실험에서는 ACM의 실제 논문데이터를 사용하여 처리시간 및 정확도 비율 등에서 그 효과성을 입증하였다.

키워드 : 스카이라인, 범주형 데이터, 지배

Abstract The skyline query is one of the effective methods to deal with the large amounts and multi-dimensional data set. By utilizing the concept of 'dominate' the skyline query can pinpoint the target data so that the dominated ones, about 95% of them, can efficiently be excluded as an unnecessary data. Most of the

· 이 논문은 제36회 추계학술발표회에서 '범주형 데이터에 대한 스카이라인 검색 알고리즘'의 제목으로 발표된 논문을 확장한 것이다

[†] 통신회원 : 인하대학교 산업공학과 교수
wookeylee@gmail.com

^{**} 학생회원 : 인하대학교 산업공학과
cjoongho@gmail.com
jaegal83@gmail.com

논문접수 : 2010년 1월 20일

심사완료 : 2010년 3월 29일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제7호(2010.7)

skyline query algorithms, however, have been developed in terms of the numerical data set. This paper pioneers an entirely new domain, the categorical data, on which the corresponding ranking measures for the skyline queries are suggested. In the experiment, the ACM Computing Classification System has been exploited to which our methods are significantly represented with respect to performance thresholds such as the processing time and precision ratio, etc.

Key words : skyline, categorical data, domination

1. 서론

스카이라인 질의는 데이터 사이의 지배관계(dominate)를 활용하여 질의결과를 효과적으로 추출하는 기법이다[1-3]. 이 기법은 대용량 데이터 처리에 특히 효과적이라고 알려지면서 의사결정 지원과 시각화 등과 관련하여 최근 중요한 연구주제로 부각되고 있다[4-10]. 기존의 스카이라인 질의 알고리즘들은 대상의 다양성과 아울러 시간 및 효율성은 개선되어 가고 있지만 데이터 집합이 대부분 수치형 데이터에 한정되어 있어서 텍스트 기반 데이터에 대한 연구는 거의 전무하다.

본 논문에서는 텍스트 데이터들 중 트리로 구성된 범주형 데이터에 대한 스카이라인 질의 방법을 제안한다. 이러한 범주형 데이터에 대한 스카이라인 질의 방법은 데이터 집합이 수치형일 때와는 대상의 차이가 확연하다[11]. 범주형 메타 정보는 XML의 구조정보나 관계형 데이터베이스의 스키마, ACM 분류체계[12] 등으로 인식된다. 범주형 데이터에 대하여 다루는 본 연구에서는 특정 범주(Category)가 트리형 메타 데이터로 주어진 경우로 한정한다. 트리가 아닌 메타 데이터에 대해서는 그래프 거리 함수[13]를 도입할 수도 있지만, 본 연구의 논점을 넘어가므로 지면 제약상 여기서는 언급하지 않는다. 특히 범주형 데이터 간의 거리 계산법[13]을 도입하여 데이터 집합이 가진 범주형 데이터들과 질의와의 상대적 의미거리를 산출하여 이를 통해 스카이라인 질의에 적용하였다.

논문의 구성은 다음과 같다. 제2장에서는 범주형 데이터의 상대적 의미거리에 대해 알아보고, 제3장에서는 범주형 데이터 검색 알고리즘에 대해 소개하고, 제4장은 ACM 분류체계에 따른 검색실험을 진행하며, 제5장은 기존연구와의 차이점을 비교하면서 제6장에서 결론 및 추후 연구 방향을 제시하였다.

2. 범주형 데이터에 대한 거리

2.1 의미거리 함수

특정 범주형 데이터가 트리형태로 주어진 경우, 그 형식적 정의는 다음과 같다.

정의 1. 범주형 데이터 T(N, E)는 노드 N과 이들을 연결하는 에지 E로 구성되어 있으며, 개별 노드 $x, y \in N$ 은 키워드이며 이들 사이에 존재하는 개별 에지 (x,y)는 범주형 트리(T)로 표현된다.

예컨대, 그림 1은 범주형 데이터로서, ACM Computing Classification System[12]의 한 부분을 나타낸 트리이다. 여기서 개별노드 Logical Design과 Physical Design은 Database Management에 포함되고, 이는 다시 Information Systems에 포함되는 것을 알 수 있다.

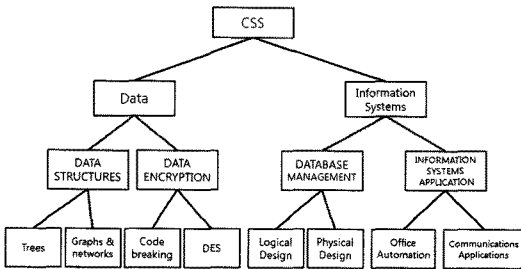


그림 1 ACM Computing Classification System 예

또한, 본 연구의 초점인 범주형 데이터에 대한 거리함수는 다음 정의와 같다.

정의 2. 범주형 트리(T)가 주어진 경우, 노드간의 거리 (WP)는 다음과 같은 거리함수에 의해 수치화되는 값이다.

$$D(x,y) = 1 - \frac{2 \times N_p}{N_q + N_t + (2 \times N_p)} \quad (1)$$

- Z : x와 y에 대한 common ancestor node in T
- N_q : x부터 z까지 경로상의 모든 노드 개수
- N_t : y부터 z까지 경로상의 모든 노드 개수
- N_p : z부터 루트까지 경로상의 모든 노드 개수

정리 1. 정의 2에서의 거리 D(x,y)는 다음 성질을 만족한다.

- 1) $D(x,z) < D(x,y) + D(y,z)$ for all $x,y,z \in T$
- 2) $D(x,x) = 0$ for all $x \in T$
- 3) $0 \leq D(x,y) \leq 1$ for all $x,y \in T$

그림 1에서 예컨대, $x=Trees, y=DES$ 이면, $z=Data$ 이다. 그리고 x, y, z에 대해 N_q, N_t, N_p 을 구해 식 (1)에 적용해 보면 0.5가 된다.

$$D(Trees, DES) = 1 - \frac{2 \times 2}{2 + 2 + 2 \times 2} = 0.5$$

식 (1)을 통해 두 노드 사이의 거리를 산출하였을 경우 0~1사이의 값으로 얻어진다. 0은 두 노드가 동일한 경우이며, 값이 커질수록 두 노드의 거리가 멀어짐을 알 수 있다. 이를 통해 범주형 데이터의 노드 간 관계 즉, 거리가 작아지면 두 노드가 가깝다는 의미에서 연관성도 커진다

고 볼 수 있으며, 이러한 성질은 거리함수들과 궤를 같이 한다[15]. 좀 더 상세한 예는 다음과 같다. $D(Trees, Des)=0.5, D(Trees,Trees)=0, D(Trees, Graph Networks)=0.25, D(Data Structure, Data Encryption) = 0.33$ 위의 거리(WP)는 Wu and Palmer[14]를 활용하였으므로 WP라 이름했다. 이 거리(WP)는 범주형 데이터에 대한 준거를 제공해줄 수 있다는 장점에도 불구하고, 몇가지 한계가 있다. 즉, 실제적용 과정에서 자식노드 수에 관계없이 같은 차원에 있는 노드들의 거리는 모두 동일하게 나온다. 예컨대 그림 2에서 {Arrays, Trees}, {Code Breaking, DES}이 속한 노드는 각각 4개 및 2개로 서로 차이가 있음에도 거리는 0.250으로 동일하다. {Composite structure, Hash-table}과 {Equipment, Groupware} 또한 0.200 이라는 동일한 값을 나타낸다. 심지어 형제노드 숫자가 매우 많거나 둘뿐이라도 거리는 모두 동일하게 되는 것이다. 결과적으로 대부분의 거리가 중복되는 모호성 때문에 척도자체의 가치가 희석되고, 또한 질의결과를 명확히 할 수 없다는 단점이 있다.

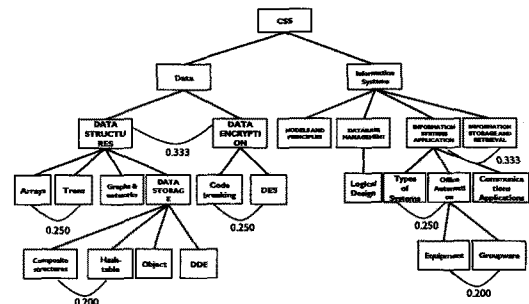


그림 2 WP 기반 거리계산 예

2.2 개선된 거리 함수

이를 개선하기 위해서 탐색 대상이 되는 노드의 수에 따른 가중치를 부과하여 형제노드 수에 따라서 거리가 세분화 되는 방법을 제안하였고, Extended-Wu and Palmer 척도(EWP)라 정의하며 다음 식 (2)와 같이 표현된다.

$$EWP: D_E(x,y) = 1 - \frac{2 * W_p * N_p}{N_q + N_t * W_t + (2 * W_p * N_p)} \quad (2)$$

$$\text{where } W_t = |N_t \text{ 부모의 자식노드수} + 1|$$

$$W_p = |N_p \text{ 부모의 자식노드수} + 1|$$

EWP를 이용하여 노드간의 의미거리를 계산하게 되면, 동일한 깊이의 노드 사이의 거리라도 그 부모노드의 자식 수의 가중치 영향으로 인해 결과값이 세분화 된다. 예컨대, 앞서 예에서 거리가 모두 0.250으로 동일한 값을 타나냈지만, EWP를 적용할 경우 {Code Breaking, DES}, {Types of Systems, Office Automation}의 거리가 각각 0.182, 0.172로 세분화된 것을 확인할 수 있다.

3. 스카이라인 질의 알고리즘

본 논문에서는 위와 같은 거리함수를 사용해서 범주형 데이터 스카이라인 질의 알고리즘을 제안하고자 한다. 우선 사용자 검색질의가 키워드일 때 범주형 데이터에 대한 스카이라인 질의는 다음과 같이 정의될 수 있다.

정의 3. 범주형 트리(T)에 대한 스카이라인 질의 $Q = (q_1, q_2, \dots, q_d)$ 의 각 키워드 q_i 에 대응하는 객체 $P = (p_1, p_2, \dots, p_d)$ 는 $p_k = \text{argmin}_i d(q_k, p_i)$ for $0 \leq i \leq d$ 에 대하여 다음과 같은 지배관계를 가진다. 즉, p_i 에 대하여 $p_i \leq p_j, \forall i=1, \dots, d$ 그리고 $\exists j, 1 \leq j \leq d$ 에 대하여 $p_i < p_j$ 이다.

물론 각각의 스카이라인 질의는 다음 알고리즘 CDSS를 따라 구해진다. 이때 데이터 집합을 선택하기 위한 키워드 집합, 데이터 집합이 포함하고 있기를 원하는 다음과 같은 범주 키워드 집합을 가진다. 예를 들어, 논문을 검색할 때, retrieval이라는 단어를 제목에 포함하는 논문 중에서 {Trees, Graphs & networks}가 논문 분류로서 포함되기를 원한다면, retrieval은 키워드이고 {Trees, Graphs & networks}는 범주 키워드 집합이다.

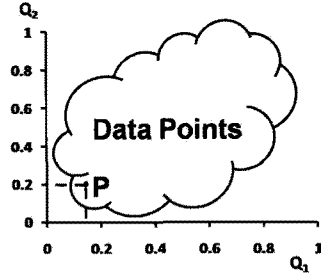


그림 4 거리를 부여한 상대적 데이터 위치배정

표 1 범주형 데이터 스카이라인 질의 알고리즘

Algorithm CDSS (Category Data Skyline Search)	
Input :	KeywordSet, 데이터 P={p1, p2, ..., pd}, 범주형 트리(T)
Output :	Skyline Set
1	D = ∅, S = ∅, T = Tern Tree, P = ∅
2	D := Find the Top-K DataSet about KeywordSet
3	while p ∈ P is not null
4	for all t ∈ T
5	D := MeanDistance(t, p, T)
6	P := P - {p}
7	return S := ComputeSkyline(D)

표 2 개선된 거리계산 알고리즘

Algorithm MeanDistance	
Input :	Tern, Data point, 범주형 트리(T)
Output :	Mean Distance
1	t = Tern, p = Data point, T = 범주형 트리, M = {}
2	while p ∈ T is not null
3	Z _i := Find the abstract node between t and p from T
4	N _t _i := calculate node count between t and Z
5	N _p _i := calculate node count between p and Z
6	N _r _i := calculate node count between Z and root
7	W _t _i := number of child node of parent node of N _t _i + 1
8	W _p _i := number of child node of parent node of N _p _i + 1
9	M += 1 - ((2 × N _p _i × W _t _i) / (N _t _i + N _t _i × W _t _i + (2 × N _p _i × W _p _i))
10	return minimum M

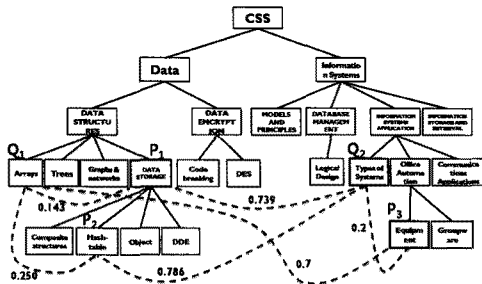


그림 3 EWP 기반 거리 및 스카이라인 질의과정

그림 3은 이러한 EWP를 이용하여 Query=(q1, q2), Paper=(p1,p2,p3)에 대한 D(qk)를 구하는 과정이다. q1 =Arrays에 대한 p1=Data Storage, p2=Hash-Table, p3 =Equipment의 의미거리는 각각 0.143, 0.250, 0.7이 되고, q2=Types of Systems에 대한 p1=Data Storage, p2=Hash-Table, p3=Equipment의 의미거리는 각각 0.739, 0.786, 0.2인데, 하나의 범주 키워드와 대비되는 거리는 하나로 표현해야 하므로, 거리 결과 집합에서 최소값인 0.143과 0.2를 선택한다. 가장 낮은 수치가 해당 데이터가 가진 범주 속성들과 범주 키워드간의 가장 가까운 거리를 나타내기 때문이다. 따라서 D(q1)=0.143, D(q2)=0.2라는 거리 값을 갖게 된다.

그림 4는 계산된 상대적 의미거리 값인 0.143, 0.2를 2차원으로 표현한 것이다. 각 차원은 하나의 범주 키워드를 나타내며, 각 데이터는 계산된 거리에 따라 그림과

같은 공간에 위치하게 된다. 이와 같이 모든 데이터에 대한 거리를 계산이 이루어지면 위의 표 1의 스카이라인 질의 알고리즘을 사용하여 스카이라인 검색을 수행할 수 있게 된다.

4. 실험

실험은 ACM Digital Library[16]에서 제공하고 있는 논문을 대상으로 진행하였다. 질의 정보로서 키워드 집합과 범주 키워드 집합을 입력하면 결과로서 스카이라인에 포함된 논문을 제공한다. 본 논문의 실험에서는 초기 데이터 집합을 선택하기 위한 키워드 집합으로 {TODS, 2005, 2006, 2007, 2008}, {WWW conference, 2007, 2008}을 입력하여 2005~2008년 TODS, 2007~2008년

표 3 실험 대상 데이터

문서	키워드 집합	데이터 수
Doc1	{TODS-2005,2006,2007,2008}	165
Doc2	{WWW conference-2007,2008}	300

표 4 실험 범주 키워드

문서	Q	범주형 키워드 집합
Doc1	q1	{Relational databases, Abstracting methods}
	q2	{Fault-tolerance, Abstracting methods}
	q3	{Content Analysis and Indexing, Query languages}
Doc2	q4	{Distributed objects, Information hiding}
	q5	{Information filtering, Distributed data structures}
	q6	{Retrieval models, Web-based services}

WWW conference에 실린 논문으로 데이터를 한정하였다. 표 3~4는 각 키워드에 의해 얻어진 데이터 수로서 여기에 제시된 범주 키워드와 얻어진 데이터 집합에 포함된 분류들과 의미거리를 산출하였고 산출된 거리를 바탕으로 스카이라인 질의를 수행하였다.

또한 WP와 EWP의 차이를 Precision ratio라고 하고 이를 위한 새로운 지표로서 다음 식 (3)과 같이 정의하였다. 이 지표는 질의 결과의 정밀도를 확인해주는 기준이 될 수 있다.

$$\text{Precision ratio} = (\# \text{Skyline 결과노드}) / (\# \text{전체 대상노드}) \quad (3)$$

실험에 따른 결과는 데이터 포인트의 배치로써 그림 5~6과 같다. 각각의 질의마다 WP와 EWP를 가지고 거리를 측정 후 비교하였으며 이를 가지고 의미거리로서 배치된 데이터들을 BBS[6]를 사용하여 스카이라인을 검색하였다. 각 실험에서 스카이라인으로 검색된 데이터들은 선으로 연결하여 표시하였다. 전체적으로 본

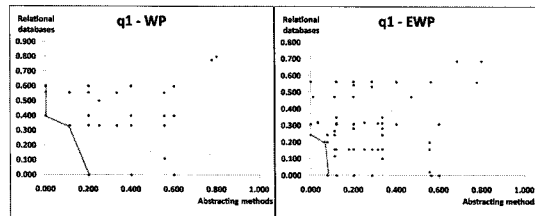


그림 5 Doc1-q1 실험결과

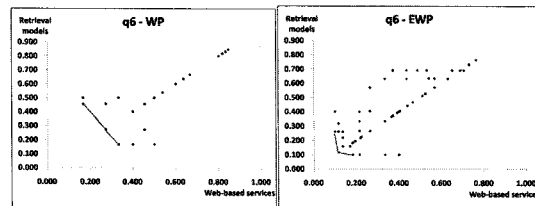


그림 6 Doc2-q6 실험결과

논문의 실험을 통하여서 지금까지 스카이라인 대상에 포함할 수 없었던 범주형 데이터에 대하여 스카이라인 질의가 가능함을 보였다.

그림 5~6에서 트리 깊이가 낮은 데이터에 대해 거리가 중복되는 사실을 각각 좌측 WP의 경우에서 확인 할 수 있다. ACM 분류체계의 경우 최대 깊이가 5이고 매우 횡적인 구조이며, 문서 집합 Doc1, Doc2의 평균 깊이가 4.62, 4.71이다. 이는 두 노드간의 공통노드를 탐색 시 깊이가 유사하므로 거리 값의 중복이 과다하여 스카이라인 질의의 효용이 약화될 정도로 중복되며, 또한 EWP를 통해 이점이 명확히 개선되는 것을 확인할 수 있다. 특히 q1~q3과 q4~q6의 데이터 수는 각각 165개, 300개로 전체적으로 q1~q3의 Precision ratio이 q4~q6에 비해서 높게 나타났다. 그 이유는 표본수가 적으므로 중복된 스카이라인 데이터의 영향을 특히 많이 받았기 때문이다. 특히 q1~q3에서 WP는 EWP 보다 데이터의 거리 중복이 심하므로, 그로인한 Precision ratio가 높게 나타났다.

또한 그림 7은 EWP와 WP의 Precision ratio비교이며, 그림 8은 차원 증가에 따른 수행도 변화를 비교하면서 특히 데이터 분포변화 즉, uniform, correlated, anti-correlated형의 분포에 대하여 스카이라인 질의처리 결과를 살펴보았다. 결과적으로 본 연구의 방법은 데이터량의 증가에 대한 처리시간 증가비율을 보면 anti-correlated의 경우 0.1초 이내의 범위 내에서 약간 커지만 유의한 차이가 없었다.

한편 차원 증가에 대해서는 anti-correlated 및 uni-

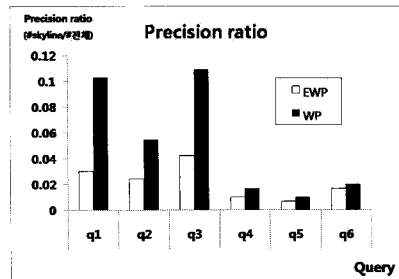


그림 7 Precision ratio

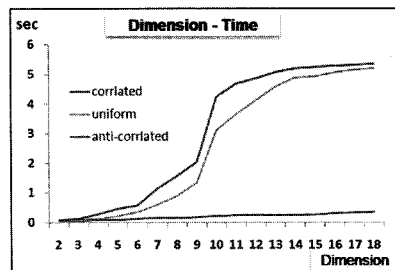


그림 8 차원 증가 관점

form데이터의 경우 9차원부터 11차원까지 급격한 증가를 보였으나 12차원 이상에서의 지속적 차원증가에 대해서는 처리시간의 증가폭이 미미하였다. 즉 다양한 데이터 특성을 가지는 차원증가에 대해서도 본 연구의 방법이 효과적임을 확인할 수 있다. 한편 correlated 데이터에 대해서는 시간의 변화가 거의 없어 결과적으로 스카이라인 질의는 데이터처리량에 따라 주로 성능이 결정됨을 알 수 있다.

그림 7의 경우 본 연구에서 제안하는 두 가지 랭킹메커니즘에서 WP는 전체 데이터의 약 10%를 스카이라인 질의 결과로 제시하는데 비해, EWP는 최대 약 4%에서 0.5%이므로 매우 정밀한 결과를 산출한다는 점을 알 수 있다.

결론적으로 본 연구에서 제안하는 방식 특히, EWP는 범주형 데이터에 대한 스카이라인 질의처리에 있어서 대용량 다차원 데이터의 효과적인 처리라는 스카이라인 질의처리의 근본적 효용성에 매우 부합되는 방법임을 확인할 수 있다.

5. 기존 연구

고전적인 알고리즘인 Block Nested Loop(BNL)[2], Divide and Conquer(D&C)[2], Nearest Neighbor Search(NN)[1], Branch and Bounded Search(BBS)[6], 또한 최근에 등장한 거리기반 스카이라인[8], Uncertain Data Skyline[10]등은 모두 스카이라인 질의를 수치 데이터에 효과적으로 적용한 연구들이다. 한편 본 연구와 같이 범주형 데이터를 대상으로 하는 경우에도 데이터의 분류에 초점을 맞추거나 혹은 랭킹메커니즘에 관한 연구들도 있지만, 범주형 데이터에 대한 스카이라인 질의를 직접 시도한 연구는 없다. 목표가 유사한 연구[17]가 있었으나 거리함수를 구체적으로 제시하지는 않았다.

6. 결론 및 향후 연구

기존의 스카이라인질의는 수치형 데이터에서만 수행되어 왔으며, 텍스트로 저장된 대부분의 데이터에 대한 질의를 수행할 수 없었다. 본 연구는 스카이라인의 적용 데이터 범위를 범주형 데이터까지 가능하게 하는 새로운 알고리즘을 제안한 것이다. 즉, 범주형 데이터에 포함된 두 노드간의 거리함수를 이용해 계산된 거리를 바탕으로 스카이라인을 질의하는 알고리즘을 제안하였다. 실험에서는 ACM Computing Classification System의 데이터를 사용하여 스카이라인 질의를 수행하였으며 본 연구의 두가지 거리함수 특히 EWP의 경우 WP에 비해 성능상의 다양한 이점과 Precision ratio에 있어서 30~60% 이상 개선된 결과를 얻어 효과적인 방법임을 입증했다.

향후 연구로는 시맨틱 웹을 대상으로 구축되는 다양한 온톨로지에 대하여 본 방법을 적용하는 연구가 필요

하며, 범주형 데이터 및 수치형 데이터의 통합 지표의 개발과 함께 트리가 아닌 일반적 그래프형의 메타 데이터에 대한 적용은 쉽게 이루어질 수 있다. 그리고 메타 데이터가 없는 대상에 대한 스카이라인 질의에 관한 연구가 필요하다. 아울러 일반적 텍스트에 대해서 WordNet, Wikipedia등을 활용하여 거리를 효과적으로 구하는 스카이라인의 연구가 후속될 필요가 있다.

참고 문헌

- [1] D. Kossmann, F. Ramsak, S. Rost, "Shooting stars in the sky: an online algorithm for skyline queries," *Proc. VLDB*, pp.275-286, 2002.
- [2] S. Börzsönyi, D. Kossmann, K. Stocker, "The Skyline Operator," *Proc. ICDE*, pp.421-430, 2001.
- [3] K. Tan, P. Eng, B. Ooi, "Efficient Progressive Skyline Computation," *Proc. VLDB*, pp.301-310, 2001.
- [4] K. Chakrabarti, S. Chaudhuri, S. Hwang, "Automatic categorization of query results," *Proc. SIGMOD*, pp.755-766, 2004.
- [5] N. Sarkas, G. Das, N. Koudas, A. Tung, "Categorical skylines for streaming data," *Proc. SIGMOD*, pp.239-250, 2008.
- [6] D. Papadias, Y. Tao, G. Fu, B. Seeger, "An optimal and progressive algorithm for skyline queries," *Proc. SIGMOD*, pp.467-478, 2003.
- [7] S. Cohen, M. Shiloach, "Flexible XML Querying Using Skyline Semantics," *Proc. ICDE*, pp.553-564, 2009.
- [8] Y. Tao, L. Ding, X. Lin, J. Pei, "Distance-Based Representative Skyline," *Proc. ICDE*, pp.892-903, 2009.
- [9] S. Zhang, N. Mamoulis, D. Cheung, "Scalable skyline computation using object-based space partitioning," *Proc. SIGMOD*, pp.483-494, 2009.
- [10] M. Atallah, Y. Qi, "Computing all skyline probabilities for uncertain data," *Proc. PODS*, pp.279-287, 2009.
- [11] S. Dimitris, P. Stavros, P. Dimitris, "Topologically Sorted Skylines for Partially Ordered Domains," *Proc. ICDE*, pp.1072-1083, 2009.
- [12] The 1998 ACM Computing Classification System - Association for Computing Machinery, <http://www.acm.org/about/class/1998/>
- [13] M. Shin, S. Huh, D. Park, W. Lee, "Relaxing Queries with Hierarchical Quantified Data Abstraction," *Journal of Database Management*, vol.19, no.4, pp.47-61, 2008.
- [14] Z. Wu, M. Palmer, "Verbs semantics and lexical selection," *Proc. ACL*, pp.133-138, 1994.
- [15] S. Hwang, H. Yu, "Mining and processing category ranking," *Proc. SAC*, pp.441-442, 2007.
- [16] ACM Digital Library, <http://portal.acm.org/>
- [17] C. Chan, P. Eng, K. Tan, "Stratified computation of skylines with partially-ordered domains," *Proc. SIGMOD*, pp.203-214, 2005.