고 밀도 영역을 이용한 향상된 2차원 히스토그램 기법

(An Enhanced Two Dimensional Histogram Method Utilizing Dense Regions)

노 요 한 † 정 연 돈 †† 김 호 진 † 김 명 호 †††
(Yohan Roh) (Yon Dohn Chung) (Hojin Ghim) (Myoung Ho Kim)

요 약 히스토그램은 데이타베이스 시스템에서 질의 결과 크기를 추정하는 데 널리 이용되고 있다. 히스토그램 기법에서 질의 결과 크기에 대한 추정은 각 버킷 영역 내의 객체들이 균등하게 분포한다는 가정하에 이루어진다. 그러나, 주어진 질의 영역 내의 객체들은 균등하게 분포하지 않을 수 있다. 다시 말해서, 버킷 영역 내에 높은 밀도의 객체 군집 즉 클러스터가 존재할 수 있으며 이로 인하여 히스토그램의 정확도가 현저히 저하될 수 있다. 본 연구의 목적은 히스토그램의 정확도를 향상시키는 데 있다. 이를 위하여 본 연구는 클러스터를 고려한 새로운 히스토그램 기법을 제안한다. 제안하는 기법은 주어진 데이타 분포 내에 존재하는 고 밀도 영역을 탐색하고 이를 히스토그램 생성에 활용한다. 제안하는 기법은 클러스터에 의한 정확도 저하를 효과적으로 감소시킴으로써 데이타가 균등하게 분포하지 않은 상황에서 향상된 성능을 제공할 수 있다. 실험을 통해 본 연구는 제안하는 기법이 기존 기법의 성능을 최대 74% 향상시킴을 확인하였다.

키워드 : 히스토그램, 선택도 추정, 질의 최적화, 데이타베이스

Abstract Histograms are popularly used for selectivity estimation in database systems. In conventional histogram methods, buckets return the approximated results based on the assumption that all objects in a bucket are uniformly distributed. However, the objects within the region of a query are not likely to be uniformly distributed. That is, there can be some skews (i.e., clusters) in the buckets, which may significantly degrade the accuracy of the histogram. The aim of this work is to enhance the accuracy of histograms. For this purpose, we propose a new two-dimensional histogram method considering clusters. The proposed method detects dense regions and exploits them for organizing buckets. Since the proposed method effectively reduces accuracy degradation caused by clusters, it can provide improved, robust accuracy against skewed data distributions. Through experiments, we show that the proposed method provides up to 74% improved performance compared with the conventional histogram.

Key words : Histograms, Selectivity estimation, Query optimization, Databases

· 이 논문은 2007년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R0A-2007-000-10046-0)

† 학생회원 : 한국과학기술원 전산학과
yhroh@dbserver.kaist.ac.kr
hojin@nslab.kaist.ac.kr
†† 종신회원 : 고려대학교 컴퓨터통신공학부 교수
ydchung@korea.ac.kr
††† 종신회원 : 한국과학기술원 전산학과 교수
mhkim@dbserver.kaist.ac.kr

논문접수 : 2008년 1월 22일
심사완료 : 2008년 8월 20일

정보과학회논문지: 데이타베이스 제35권 제6호(2008.12)

1. 서 론

히스토그램은 데이타베이스 시스템에서 질의 결과의 크기를 추정하는 데 널리 이용되고 있다[1]. 이러한 추정치는 질의 최적화기가 효율적인 질의 수행 계획을 수립하는 데 유용하게 활용된다[2]. 본 연구는 히스토그램을 이용한 단일 선택도 $sel(p_1)$, 논리합 선택도 $sel(p_1 \vee p_2)$ 및 논리곱 선택도 $sel(p_1 \wedge p_2)$ 추정 문제를 살펴본다. 여기에서, 선택도는 주어진 술어를 만족하는 데이타의 수를 의미하며, p_i는 차원 i의 술어(predicate)이다. 각 차원의 술어로는 "$100 \le X \le 200$"와 같은 영역 술어(range predicate)가 고려된다. 본 연구는 공간(geographic) 정보를 표현하는 데 널리 이용되는 2차원 데이타 공간을 주로 고려한다. 단일 선택도 $sel(p_1)$ 및 논리합 선택도 $sel(p_1 \vee p_2)$는 논리곱 선택도 $sel(p_1 \wedge p_2)$를 이용하여 각각 $sel(p_1 \wedge p_{2(\text{full range})})$ 및 $sel(p_1 \wedge p_{2(\text{full range})}) + sel(p_{1(\text{full range})} \wedge p_2) - sel(p_1 \wedge p_2)$와 같이 표현되므로, 본 연구는 논리곱 선택도 추정 문제에 초점을 둔다. 기존의 많은 질의 최적화기는 독립 가정(independence assumption)에 기반하여 논리곱 선택도를 계산한다. 이 가정 하에서는 두 속성 값의 관련성(correlation)이 고려되지 않기 때문에 추정의 정확도가 매우 낮아질 수 있다. 그 결과 질의 최적화기가 효율적인 질의 수행 계획을 생성하지 못하게 됨으로써 질의 처리 성능이 크게 저하될 수 있다[3]. 독립 가정에 대한 한 가지 유용한 대안으로 여러 속성에 대한 히스토그램 즉 다차원(multidimensional) 히스토그램을 활용하는 방안이 있다[4-7]. 본 논문은 공간 정보에 널리 활용 가능한 2차원의 경우를 고려한다.

데이타 불균등 문제. 히스토그램 기법에서는 버킷 영역 내의 객체들이 균등하게 분포한다는 가정 하에 선택도가 추정된다. 이러한 경우 추정의 정확도는 높은 밀도를 형성하는 객체 군집 즉 클러스터(cluster)에 의해 크게 저하될 수 있다. 그림 1의 버킷 B와 질의 영역을 고려하자. 그림에서 버킷 영역과 질의 영역은 서로 겹쳐 있다. 버킷 B의 영역에는 21개의 객체가 위치하고, 이 중 16개 객체가 클러스터를 형성하고 있다. 만약 버킷 영역과 질의 영역이 서로 겹친 부분(회색 영역)의 크기가 버킷영역크기/2이라면 회색 영역에 대한 추정치는 $(1/2)*21 \approx 11$과 같이 계산된다. 이 수치는 객체의 실제 빈도수 3과 현저한 차이를 보임을 주목하자. 본 연구는 클러스터에 의하여 추정의 정확도가 저하되는 이러한 현상을 "데이타 불균등 문제"라 정의한다.

기존의 다차원 히스토그램은 데이타 불균등 문제의 원인인 클러스터를 적절히 처리하지 않는다. 그 결과 데이타 분포가 균등하지 않은 상황에서 추정 정확도가 현저히 저하될 수 있다. 본 연구는 클러스터를 고려한 새로운 히스토그램 기법을 제안한다. 제안하는 기법은 주어진 데이타 분포 내에 존재하는 클러스터를 탐색하고 그 정보를 히스토그램 생성에 활용함으로써 데이타 불균등 문제로 인한 추정 정확도의 저하를 효과적으로 감소시킨다. 본 연구는 여러 성능 평가 실험을 통하여 제안하는 기법이 데이타가 균등하게 분포하지 않은 여러 상황 즉 다양한 비대칭 데이타 분포(skew data distribution)에서 기존 기법보다 향상된 성능을 제공함을 보인다.

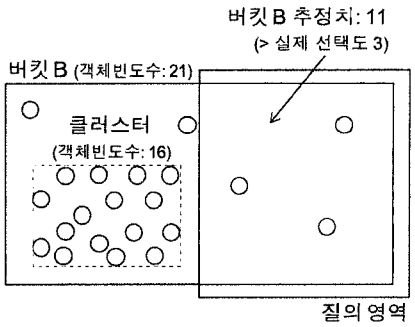


그림 1 클러스터를 포함한 버킷의 예

본 논문의 구성은 다음과 같다. 2장에서는 예비 지식 및 관련 연구를 소개한다. 3장에서는 클러스터 정보를 활용한 새로운 히스토그램 기법을 제안한다. 그리고, 4장에서는 성능 평가 결과를 제시한다. 마지막으로, 5장에서는 결론 및 추후 연구 방향을 제시한다.

2. 예비 지식 및 관련 연구

본 장에서는 히스토그램에 대해 간략히 알아보고, 본 연구의 관련 연구를 기술한다.

2.1 히스토그램

데이타 도메인 D에 대한 히스토그램 H는 버킷 B_i (i = 1, …, n)의 집합이다. 각 버킷 B_i는 데이타 영역 S_i와 영역 내에 위치하는 객체의 빈도수 F_i를 통계정보로 가진다. 각 데이타 영역마다 객체들이 균등하게 분포한다고 가정된다. 이러한 가정을 균등 분포 가정(uniformly distributed assumption)이라 한다. 일반적으로 버킷 수는 데이타베이스 관리자에 의해 설정된다. 질의 영역 I가 주어진 경우 질의 결과 크기의 추정치 즉 선택도 추정치는 다음과 같이 계산된다.

$$\sum_{i=1,\dots,n} \left(\frac{|S_i \wedge I|}{|S_i|} * F_i \right)$$

여기에서 $|S_i|$는 버킷 영역 S_i의 크기를 의미하고, '$\wedge$'는 교집합(intersection) 연산자를 의미한다. □

예 1. 데이타 도메인 [0, 100]에 대한 다음의 히스토

그램 H를 고려하자.

H = { B_1 (S: 0 ~ 50, F: 20),
B_2 (S: 50 ~ 60, F: 50),
B_3 (S: 60 ~ 100, F: 10) }

질의 영역 [20, 60]가 주어진 경우 선택도 추정치는 다음과 같이 계산된다.

$$\frac{30}{50}*20+\frac{10}{10}*50+\frac{0}{40}*10=62 \quad \square$$

2차원의 경우 버킷 영역으로 직사각 영역이 주로 이용된다. 주어진 질의 영역에 대한 버킷의 추정치는 균등 분포 가정 하에 버킷 영역과 질의 영역이 겹친 부분의 크기에 비례하여 계산되고, 선택도 추정치는 예 1과 같이 버킷 추정치의 총합으로 계산된다. 앞으로 모호함이 없는 경우 "버킷"이라는 용어를 버킷 영역의 의미로도 사용함으로써 서술의 편의를 도모하고자 한다.

2.2 관련 연구

선택도 추정 문제는 많은 관심을 받아 왔으며, 데이타베이스와 관련된 다양한 응용 분야에 활용되어 왔다[8-12]. 특히 다차원 히스토그램의 성능을 향상시키고자 하는 여러 연구가 있었다[4-7]. 다차원 히스토그램은 버킷 영역 내의 객체들이 균등하게 분포할 때 정확하게 동작한다. 그러나, 한정된 수의 버킷을 이용하여 그러한 히스토그램을 구성하는 것은 NP-hard임이 증명된 바 있다[13]. 따라서, 기존 연구는 히스토그램을 구성하는 휴리스틱을 제안한다.

기존 다차원 히스토그램 기법(버킷의 구성, 관리, 선택도 추정 방법)은 다음의 두 가지 유형으로 분류될 수 있다.

- 질의 결과 기반 히스토그램 기법 (STHoles[4] 및 ISOMER[5])
- 데이타 스캔(scan) 기반 히스토그램 기법 (Min-Skew [6] 및 GenHist[7])

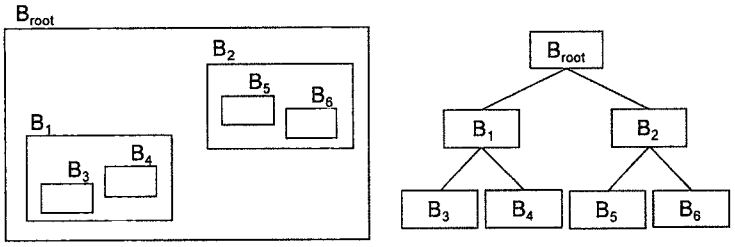
첫 번째 유형은 질의 결과 피드백(feedback)을 활용하여 히스토그램을 구성한다. 이 유형의 히스토그램은 시스템 내에서 수행된 질의의 실제 결과 크기에 기반하여 (재)구성된다. 이러한 질의 결과 크기에는 데이타 변경(update)이 반영될 수 있으므로 이 유형의 히스토그램은 데이타 변경에 대한 적응이 가능하다. 그러나, 질의 영역 혹은 그 주변 영역의 결과만이 고려되기 때문에 적응은 단지 질의 관련 버킷으로 한정될 수 밖에 없다. 즉, 질의와 무관한 영역에서의 데이타 변경은 히스토그램에 반영되지 않는다. 또한 이 유형은 질의 처리 과정 동안 오버헤드(overhead)를 유발한다.

두 번째 유형은 데이타베이스를 주기적으로 스캔하여 히스토그램을 (재)구성한다. Min-Skew는 데이타 스캔 히스토그램으로 가장 널리 활용되고 있는 기법 중 하나이다[14-19]. 이 기법은 한 버킷을 두 개의 하위 버킷(sub-bucket)으로 반복적으로 분할한다. 이와 같은 간단한 분할 정책은 히스토그램의 신속한 생성을 가능하게 한다. 1장에서 언급한 클러스터는 추정의 정확도를 저하시킬 수 있기 때문에 적절한 처리가 요구된다. 그러나, 이 휴리스틱은 한 시점에 단 하나의 차원 상에 나타난 데이타 불균등(skew)만을 고려하고 여러 차원을 동시에 고려하지 않기 때문에 2차원 상의 클러스터를 거의 식별하지 못한다는 단점이 있다. GenHist 히스토그램도 데이타 스캔 기반의 히스토그램이다. 이 히스토그램 기법은 다차원 격자(grid)를 이용하여 객체빈도수가 높은 격자 셀(cell)을 찾고 이를 버킷으로 구성한다. 한 히스토그램의 생성에는 다양한 크기의 여러 격자가 이용될 수 있으며, 각 격자마다 특정 수의 버킷이 구성된다. 이 기법에서 (초기) 격자의 크기 및 각 격자마다 생성되는 버킷의 수는 사용자에 의해 결정되는 매개변수이다. 이 기법은 결합 데이타 분포(joint data distribution)를 직접적으로 근사화하지만, 추정 성능이 매개변수에 크게 의존한다. 또한, 사용자가 이러한 매개변수를 적절히 설정하는 것이 어려운 것으로 알려져 있다[4].

3. 고 밀도 영역을 이용한 히스토그램 기법

본 연구는 데이타 스캔 방식의 새로운 2차원 히스토그램 기법을 제안한다. 제안하는 기법은 데이타에 존재하는 고 밀도 직사각 영역 즉 핫스팟(hotspot)을 탐색하고 이를 히스토그램 버킷으로 구성한다. 본 연구는 이와 같은 버킷 구성을 통해 1장에서 살펴본 클러스터에 의한 정확도 저하를 효과적으로 감소시킬 수 있다.

그림 2는 제안하는 기법에 의해 생성된 히스토그램의 예를 보여준다. 제안하는 히스토그램은 핫스팟에 기반하여 생성된 버킷의 집합으로 구성되며 그림과 같이 버킷 계층구조를 형성한다. 계층구조의 생성 과정은 다음과 같다. 우선, 주어진 데이타셋의 최소 경계 영역 즉 MBR(Minimum Bounding Rectangle)을 버킷 B_{root}로 결정한다. 히스토그램을 구성하는 버킷의 수가 7로 주어짐을 가정하자. 다음으로, 우리는 버킷 B_{root} 내의 핫스팟을 탐색하고 버킷으로 구성한다. 그림의 버킷 B_1과 B_2가 이에 해당된다. 이 과정 동안 구성되는 버킷의 수는 주어진 데이타 분포 및 핫스팟의 정의에 의존한다. 핫스팟의 정의는 3.1절에서 살펴볼 것이다. 만약, 현재까지 생성한 버킷 수가 주어진 버킷 수 즉 7보다 작다면, 남은 수의 버킷을 활용하여 새롭게 구성된 버킷 내의 데이타 불균등을 감소시킨다. 즉, 본 연구는 그림에서 남은 버킷 수 4를 버킷 B_1과 B_2에 분배하고, 버킷 B_1과 B_2 내부에서 각각 핫스팟을 탐색하고 버킷으로 구성한다. 그림에서 버킷 B_3, B_4, B_5, B_6이 이에 해당된다. 이



(a) 버킷 영역 (b) 버킷 계층구조

그림 2 제안하는 기법에 의한 히스토그램의 예

러한 과정을 주어진 수의 버킷이 생성될 때까지 진행한다. 그림에서 주어진 수의 버킷이 모두 생성되었으므로 우리는 버킷 계층구조 생성을 완료한다.

3.1 핫스팟의 개념

높은 밀도를 형성하는 객체 군집을 클러스터라 하자. 본 연구는 이러한 클러스터를 포함한 직사각 영역을 핫스팟으로 정의한다. 밀도를 객체빈도수/크기로 정의하자.

정의 1. (상대-밀도) 버킷 B의 하위 영역(subregion) R에 대한 상대-밀도(R)은 다음과 같이 정의된다.

$$\text{상대-밀도(R)} = \frac{\text{밀도(R)}}{\text{밀도(B)}}$$ □

핫스팟은 자신을 포함하는 버킷보다 밀도가 k배 이상 높은 영역이다(단, $k \geq 1$). 여기에서 k는 핫스팟이 되기 위한 최소 상대-밀도이다.

예 2. k=2라고 가정하자. 만약 영역 R의 크기가 자신을 포함하는 버킷 B 크기의 1/4이고 객체빈도수가 1/2이라면, 영역 R의 밀도는 B의 밀도보다 2배 높다. 따라서, R은 핫스팟이다. □

핫스팟의 결정은 기본적으로 특정 영역의 밀도에 기반한다. 그리고, 영역의 밀도는 크기와 객체빈도수에 의존한다. 다음의 정의 2는 (예 2와 같이) 간단히 밀도 매개변수 k만을 사용하는 대신 크기 매개변수 s와 객체빈도수 매개변수 f를 사용한다.

정의 2. (핫스팟) 크기가 S이고 객체빈도수가 F인 버킷 B를 고려하자. 두 개의 변수 s와 f(단, $s \geq f$)가 주어진 경우 다음 조건을 만족하는 버킷 B 내의 직사각 영역 HR이 핫스팟이다.

(1) 크기(HR) = $\frac{1}{s}$*S

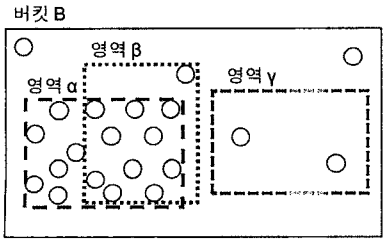
(2) 객체빈도수(HR) $\geq \frac{1}{f}$*F

(3) HR의 형태는 버킷 B와 동일하다. □

위 정의의 조건 (1)을 "크기-조건"이라 한다. 조건 (2)를 "객체빈도수-조건"이라 한다. 조건 (3)을 "형태-조건"이라 한다. 우리는 핫스팟 집합에 대한 다음 조건을 필요로 한다. 계층구조 상의 형제(sibling) 노드에 해당하는 어떤 두 핫스팟도 서로 겹치지 않아야 한다. 핫스팟의 밀도는 버킷 B보다 크거나 적어도 같아야 하므로 s 값은 f 이상이어야 한다. 조건 (1)의 s는 핫스팟의 크기(=S/s)를 결정한다. 조건 (2)의 f는 핫스팟의 최소 객체빈도수(=F/f)를 결정한다. 예를 들어, s=4와 f=2는 다음을 의미한다. 버킷 B 영역 1/4 크기의 영역 R을 고려하자. 핫스팟이 되기 위해서는 영역 R의 객체빈도수가 버킷 B의 1/2 이상이어야 한다. 만약, 모든 가능한 형태를 고려한다면, 핫스팟 탐색 문제의 복잡도가 상당히 높아진다. 조건 (3)은 이러한 이유에서 계산의 편의를 도모하고자 도입되었다.

관찰 1. 핫스팟의 상대-밀도는 적어도 1보다 크거나 같다.

예 3. 그림 3(a)의 버킷 B 내의 직사각 영역 α, β, γ를 고려하자. 버킷 B 및 영역에 관한 정보는 그림 3(b)



(a) 버킷 B 내 영역

영역	객체빈도수	크기
B	21	80
α	16	20
β	11	16
γ	2	20

(b) 버킷 B 및 영역 정보

그림 3 핫스팟의 예

와 같다. 만약 s=4 및 f=2가 사용된다면 영역 α만이 핫스팟이 된다. 영역 β는 형태-조건을 만족하지 않고, 영역 γ는 객체빈도수-조건을 위반한다(2<21/2).

관찰 2. 한 버킷 내의 핫스팟 수는 최대 f이다.

정의 3. (상대-에러) 버킷 B의 하위 영역 R에 대한 상대-에러(R)은 다음과 같이 정의된다.

$$\text{상대-에러(R)} = \frac{|\text{객체빈도수(R)} - \text{객체빈도수}_{\text{균등}}\text{(R)}|}{\text{객체빈도수(R)}}$$

여기에서, 객체빈도수(R)은 영역 R의 실제 객체빈도수이다. 객체빈도수$_{균등}$(R)은 균등 분포 가정 하에 버킷 B의 통계정보를 이용하여 추정된 영역 R의 객체빈도수이다. □

상대-에러는 객체가 균등하게 분포한다는 가정에 의해 발생하는 에러를 의미한다. 상대-에러가 높은 영역은 추정의 정확도를 저해할 수 있기 때문에 이에 대한 차별화된 처리가 요구된다. 본 연구는 이러한 영역을 핫스팟으로 검출하여 별도의 버킷으로 구성한다.

명제 1. 버킷 B 내의 핫스팟 HR에 대해서 다음 식이 성립한다.

$$\text{상대-에러(HR)} = 1 - \frac{1}{\text{상대-밀도(HR)}}$$

증명: 상대 - 에러(HR)

$$= \frac{|\text{객체빈도수(HR)} - \text{객체빈도수}_{\text{균등}}\text{(HR)}|}{\text{객체빈도수(HR)}}$$

$$= \frac{|\text{밀도(HR)} * \text{크기(HR)} - \text{밀도(B)} * \text{크기(HR)}|}{\text{밀도(HR)} * \text{크기(HR)}}$$

$$= \frac{|\text{상대-밀도(HR)} * \text{밀도(B)} * \text{크기(HR)} - \text{밀도(B)} * \text{크기(HR)}|}{\text{상대-밀도(HR)} * \text{밀도(B)} * \text{크기(HR)}}$$

$$= 1 - \frac{1}{\text{상대-밀도(HR)}} \qquad \text{(관찰 1 이용)}$$

3.2 핫스팟 기반 히스토그램 생성

이 절은 핫스팟을 이용한 히스토그램 구성법을 제시한다. 데이타셋 O와 버킷 수 NB가 주어짐을 가정하자.

핫스팟의 크기 및 최소 객체빈도수 결정. 본 연구는 정의 3의 상대-에러를 이용하여 버킷으로 구성될 핫스팟의 크기 및 최소 객체빈도수를 결정한다. 보다 구체적으로, 제안하는 기법은 상대-에러 0.1 이상의 고 밀도 영역을 버킷으로 구성한다. 주어진 데이타셋 O에 존재하는 고 밀도 영역에 비해 버킷으로 구성될 영역의 크기가 상대적으로 작게 설정된 경우를 고려하자. 이 경우 한정된 수의 영역으로 고 밀도 영역을 모두 커버하지 못할 수 있다. 이를 방지하기 위하여 본 연구는 버킷으로 구성되는 영역의 크기를 주어진 버킷 수와 상대-에러를 고려하여 가능한 큰 값으로 설정한다. 즉 우리는 핫스팟의 평균 크기의 최대값으로 설정한다. 단 핫스팟이 최대 NB개까지 생성될 수 있는 값이 설정되어야 한다. S와 F를 각각 핫스팟을 포함하는 버킷 B의 크기 및 객체빈도수라 하자. 이러한 설정에 의한 핫스팟의 크기는 속성 1에 의해 (0.9*S)/NB이고, 객체빈도수는 속성 2에 의해 F/NB 이상임을 알 수 있다. 따라서, 본 연구는 버킷으로 구성되는 영역의 크기를 (0.9*S)/NB로, 객체빈도수를 F/NB 이상으로 설정한다(즉, s=NB/0.9, f=NB).

속성 1. 버킷 수가 NB이고 핫스팟 HR의 최소 상대-에러가 0.1인 경우 핫스팟 HR의 평균 크기는 최대 $\frac{S}{(NB/0.9)}$이다.

증명: S와 F를 각각 핫스팟 HR을 포함하는 버킷 B의 크기 및 객체빈도수라 하자. 그리고, NB' (1≤NB'≤NB)를 버킷 B 내의 핫스팟 수라 하자.

$$F \geq \sum_{i=1...NB'} \text{객체빈도수}(HR_i)$$

$$= \sum_{i=1...NB'} \text{밀도}(HR_i) * \text{크기}(HR_i)$$

$$= \sum_{i=1...NB'} \text{상대-밀도}(HR_i) * \text{밀도(B)} * \text{크기}(HR_i)$$

$$\geq \sum_{i=1...NB'} (1/0.9) * \text{밀도(B)} * \text{크기}(HR_i) \quad \text{(명제 1 이용)}$$

$$= (1/0.9) * \text{밀도(B)} * \sum_{i=1...NB'} \text{크기}(HR_i)$$

$$\Leftrightarrow \frac{0.9*F}{NB' * \text{밀도(B)}} \geq \frac{\sum_{i=1...NB'} \text{크기}(HR_i)}{NB'}$$

$$\Leftrightarrow \frac{0.9*S}{NB'} \geq \text{평균크기}$$

$$\Leftrightarrow \frac{0.9*S}{NB} \geq \text{평균크기} \qquad NB' = NB \text{ 경우}$$

속성 2. 크기가 $\frac{0.9*S}{NB}$인 핫스팟 HR을 고려하자. 만약 핫스팟 HR의 최소 상대-에러가 0.1이면 객체빈도수는 F/NB 이상이다.

증명: S와 F를 각각 핫스팟 HR을 포함하는 버킷 B의 크기 및 객체빈도수라 하자.

$$\text{상대-에러(HR)} \geq 0.1$$

$$\Leftrightarrow \text{상대-밀도(HR)} \geq \frac{1}{0.9} \quad \text{(명제 1 이용)}$$

$$\Leftrightarrow \text{밀도(HR)} \geq \frac{1}{0.9} * \frac{F}{S}$$

$$\Leftrightarrow \text{객체빈도수(HR)} \geq \frac{1}{0.9} * \frac{F}{S} * \text{크기(HR)}$$

$$= \frac{F}{NB}$$

핫스팟 탐색 알고리즘. 주어진 데이타셋으로부터 핫스팟을 탐색하는 데 본 연구는 Distinct CR Extraction Algorithm[20]을 이용한다. 이 알고리즘은 원시적 알고리즘보다 향상된 $O(N^2)$의 시간 복잡도를 가진다. 여기

에서 N은 주어진 데이타셋의 크기이다. 이 알고리즘의 동작에는 정의 2의 s 및 f 값이 요구되며 본 연구는 속성 1 및 2에 기반하여 결정한 s=NB/0.9 와 f=NB를 활용한다. 관찰 2에 의해 이러한 설정은 검출되는 핫스팟의 수가 주어진 버킷 수 NB를 초과하지 않도록 한다. 이 알고리즘에 의해 검출된 핫스팟은 새로운 버킷으로 구성되며, 각 버킷마다 영역 및 객체빈도수 통계정보가 유지된다. 단, 각 핫스팟마다 객체의 최소 경계영역이 객체가 위치하고 있는 영역을 보다 정확하게 표현하기 때문에 본 연구는 이를 버킷 영역 통계정보로 활용한다.

버킷 계층구조 생성. 버킷 B_{root}를 루트로 하는 버킷 계층구조의 생성 과정은 다음과 같다. 루트 버킷 B_{root}, 전체 데이타셋 O, 이후에 생성될 버킷 수 NB, 계층구조의 현재 레벨 1이 주어짐을 가정하자. 우리는 핫스팟 탐색 알고리즘을 이용하여 핫스팟을 탐색하고 이를 현재 버킷 노드 B_{root}의 자식 노드(child node)로 구성한다. 다음으로, 버킷의 생성 수 NB'를 조사한다. 관찰 2에 의해 NB'는 주어진 버킷 수 NB와 동일하거나 작다. 만약 NB'가 NB와 동일하다면 알고리즘을 성공적으로 종료한다. 만약 NB'가 NB보다 작다면 버킷 추가 생성 과정이 진행된다. 이 과정에서 우리는 추가 생성이 요구되는 버킷의 수 NB-NB'를 자식 노드 버킷에 할당하고, 자식 노드 버킷 내에 존재하는 핫스팟 탐색 및 버킷 생성을 진행한다. 이 과정은 정해진 수의 버킷이 생성될 때까지 재귀적으로 진행된다. 그림 4는 시간 복잡도가 $O(NB* N^2)$인 계층구조 생성 알고리즘을 보여준다. 여기에서 NB는 버킷 수이고 N은 데이타셋의 크기이다.

```
Procedure BuildBucketHierarchy(B, O, NB, l)
/* Input:  B (a bucket where hotspots are detected),
           O (a two-dimensional dataset in bucket B)
           NB (the number of buckets)
           l (the current level in a bucket hierarchy) */
/* initially, B=B_root, O=the whole dataset, l=1 */
/* Output: a bucket hierarchy rooted by bucket B_root */
01 :  BucketSet_l = Detect hotspot regions in B by using
02 :                the distinct CR extraction algorithm;
03 :  Make buckets in BucketSet_l be children of B;
04 :  NB' = | BucketSet_l |;    /* NB' denotes the number of
04 :                              detected hotspots */
06 :  IF (NB > NB') {
07 :     FOR EACH bucket B_i in BucketSet_l DO {
08 :        /* NB_i denotes the number of buckets
09 :           to be detected in B_i */
10 :        NB_i = (NB-NB')/NB';
11 :        IF (NB_i > 0) {
12 :              /* O_i denotes a dataset in B_i */
13 :              BuildBucketHierarchy(B_i, O_i, NB_i, l+1);
14 :        } // End_of_IF
15 :     } // End_of_FOR_EACH
16 :  } // End_of_IF
```

그림 4 버킷 계층구조 생성 알고리즘

주어진 데이타셋 O에서 핫스팟이 검출되지 않을 수 있다(1행). 이 경우 본 연구는 버킷으로 선택되는 조건을 완화한다. 즉 우리는 버킷 B 밀도 이상의 하위 영역이 자식 노드로 선택되도록 한다. 이를 위해 우리는 s, f 값을 동일한 값 NB로 설정하고 버킷 생성을 재시도한다. 이러한 버킷 생성 재시도는 하나 이상의 자식 노드가 생성될 때까지 s와 f 값을 두 배로 설정하며 진행된다. 이 과정에서 NB보다 많은 수의 영역이 검사될 수 있다. 이 경우 객체빈도수가 높은 NB개의 영역만을 버킷으로 구성함으로써 주어진 수의 버킷이 생성되도록 한다.

3.3 선택도 추정

이 절은 제안하는 히스토그램을 이용한 선택도 추정법을 제시한다. 루트 노드 버킷이 B_{root}인 히스토그램 H를 고려하자. 질의 영역 q가 주어진 경우 히스토그램 H에 대한 선택도 추정치 SelectivityEstimate(H, q)는 버킷 B_{root}에 대한 선택도 추정치 est(B_{root}, q)와 같이 계산된다. 버킷 B에 대한 선택도 추정치 est(B, q)는 잎 노드(leaf node)의 경우와 내부 노드(internal node)의 경우로 구분하여 다음과 같이 계산된다.

잎 노드 선택도 계산. 버킷이 잎 노드인 경우 균등분포 가정에 기반하여 선택도를 계산한다. 즉 우리는 버킷 영역과 질의 영역이 겹친 부분의 크기에 비례하여 선택도를 추정한다.

내부 노드 선택도 계산. 내부 노드의 경우 자식 노드 버킷의 통계정보를 활용함으로써 데이타 분포에 관한 보다 정확한 정보를 획득할 수 있다. 따라서, 본 연구는 자식 노드 버킷의 통계정보를 최대한 활용한다. 우선, 주어진 질의 영역이 자식 노드 버킷의 영역과 겹치는 경우 우리는 내부 노드 버킷의 통계정보보다 자식 노드 버킷의 통계정보를 우선적으로 활용한다. 또한, 우리는 자식 노드 버킷의 통계정보를 활용하여 자식 노드와 겹치지 않는 영역에 관한 정보를 획득한다. 보다 구체적으로, 먼저 자식 노드 버킷들의 객체빈도수 합을 구한 후 이 수치를 내부 노드 버킷의 객체빈도수에서 감소시킴으로써 우리는 자식 노드 버킷과 겹치지 않는 영역의 객체빈도수를 계산할 수 있다. 이 값은 내부 노드 버킷의 통계정보보다 데이타 분포를 더욱 정확하게 반영하므로 본 연구는 이를 선택도 추정에 활용한다.

위의 설명에 기반하여 버킷 B에 대한 질의 영역 q의 선택도 추정치 est(B, q)는 다음과 같이 계산된다. 여기에서 B_c는 내부 노드 버킷의 자식 노드를 의미하고, 크기겹침(B, q)는 버킷 영역과 질의 영역이 겹친 부분의 크기이다.

est(B, q)=

$$\begin{cases} \dfrac{\text{크기}_{\text{겹침}}(B,q)}{\text{크기}(B)} * \text{객체빈도수}(B), \text{ 잎 노드 경우} \\ \sum_{c=1\ldots n} \text{est}(B_c,q) + \dfrac{\text{크기}_{\text{겹침}}(B,q) - \sum_{c=1\ldots n} \text{크기}_{\text{겹침}}(B_c,q)}{\text{크기}(B) - \sum_{c=1\ldots n} \text{크기}(B_c)} * \\ (\text{객체빈도수}(B) - \sum_{c=1\ldots n} \text{객체빈도수}(B_c)), \text{ 내부 노드 경우} \end{cases}$$

예 4. 그림 5(a)의 히스토그램 H를 고려하자. 그림 5(b)는 버킷 계층구조를 보여주고 그림 5(c)는 버킷들의 객체빈도수, X-범위, Y-범위, 크기를 보여준다. 히스토그램 H에 대한 질의 영역 q (34≤X≤65, 20≤Y≤55)의 선택도는 다음과 같이 추정된다.

우선, 우리는 잎 노드 버킷에 대한 선택도 추정치를 계산한다. 그림 5(a)의 버킷 B_3, B_4, B_5, B_6이 잎 노드 버킷에 해당된다. 선택도 추정치 est(B_3, q), est(B_4, q), est(B_5, q), est(B_6, q)는 다음과 같이 계산된다.

$$\text{est}(B_3,q) = \frac{0}{112} * 6 = 0 \quad \text{est}(B_4,q) = \frac{28}{112} * 8 = 2$$

$$\text{est}(B_5,q) = \frac{56}{112} * 4 = 2 \quad \text{est}(B_6,q) = \frac{0}{112} * 5 = 0$$

다음으로, 선택도가 추정된 버킷의 부모 노드(parent node) 버킷을 검색한다. 그림 5(a)의 버킷 B_1, B_2가 이에 해당된다. 이제 부모 노드 버킷마다 모든 자식 노드 버킷의 선택도가 이미 계산되었는지 검사한다. 그리고, 이를 만족하는 버킷들의 선택도가 계산된다. 버킷 B_1 및 B_2 모두 해당되므로 다음과 같이 est(B_1, q), est(B_2, q)를 계산한다.

$$\text{est}(B_1,q)$$
$$= \text{est}(B_3,q) + \text{est}(B_4,q) + \left(\frac{110-(0+28)}{900-(112+112)} * (40-(6+8))\right)$$
$$= 0+2+3.15 = 5.15$$

$$\text{est}(B_2,q)$$
$$= \text{est}(B_5,q) + \text{est}(B_6,q) + \left(\frac{225-(56+0)}{900-(112+112)} * (24-(4+5))\right)$$
$$= 2+0+3.75 = 5.75$$

우리는 est(B_{root}, q)가 계산될 때까지 이전 단계를 반복한다. 이전 단계를 진행함으로써 버킷 B_{root}의 선택도가 계산될 수 있음을 알 수 있다. 따라서, est(B_{root}, q)를 다음과 같이 계산한다.

$$\text{est}(B_{root},q)$$
$$= \text{est}(B_1,q) + \text{est}(B_2,q) + \left(\frac{1085-(110+225)}{6000-(900+900)} * (100-(40+24))\right)$$
$$= 5.15+5.75+6.43 = 17.33$$

결국, 질의 영역 q의 선택도 추정치는 17.33이다.

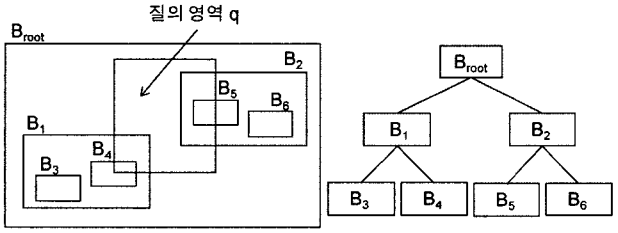
4. 성능 평가

4.1 실험 환경

GSTD 데이타셋[21]은 공간/시공간 데이타베이스에서 널리 활용되는 벤치마크 데이타셋이다. 우리는 본 실험에 GSTD 데이타셋 비대칭 데이타 분포(skew data distribution)를 이용하였다. 이 데이타셋은 데이타 불균등(skew)에 대한 척도를 제공하는데, 이를 데이타 불균등 정도(skew degree)라 한다. 이 척도의 최소값 0은 데이타가 균등하게 분포함을 의미하며 수치가 높아질수록 데이타가 더욱 불균등하게 분포한다. 데이타 불균등에 따른 히스토그램 성능을 측정하기 위해, 본 연구는 데이타 불균등 정도 0.1~0.9의 다양한 데이타셋을 이용하였다. 각 데이타 셋은 2차원 데이타 공간 $[0, 1000]^2$ 상의 10,000개 객체를 가지도록 설정되었다. 또한, 버킷 수에 따른 히스토그램 성능을 측정하기 위해, 우리는 200, 300, 400개 버킷으로 구성된 히스토그램을 생성하였다. 본 연구는 다차원 데이타에 가장 널리 활용되는 기법 중 하나인[14-19] Min-Skew 히스토그램과 제안하는 기법을 비교하였다. 성능 평가의 척도로는 널리 알려진 평균 상대 에러가 이용되었다. Θ를 질의 영역 q의 실제 객체빈도수라 하고, Θ'를 히스토그램을 이용하여 추정된 객체빈도수라 하자. 절대 에러 e^{abs}와 상대 에러는 e^{rel}은 각각 다음과 같이 정의된다.

$$e^{abs} = |\theta - \theta'| \quad e^{rel} = \frac{e^{abs}}{\theta} = \frac{|\theta - \theta'|}{\theta}$$

Q개의 질의 영역에 대해, 평균 상대 에러 E^{rel}은 다음



(a) 버킷 및 질의 영역 (b) 버킷 계층구조

버킷	객체빈도수	X-범위	Y-범위	크기
B_{root}	100	0~100	0~60	6000
B_1	40	5~45	7.5~30	900
B_2	24	55~95	27.5~50	900
B_3	6	8~22	9~17	112
B_4	8	27~41	16~24	112
B_5	4	58~72	34~42	112
B_6	5	77~91	29~37	112

(c) 버킷 정보

그림 5 선택도 추정의 예

과 같이 정의된다.

$$E^{rel} = \frac{1}{Q} * \sum_{i=1}^{Q} e^{rel}$$

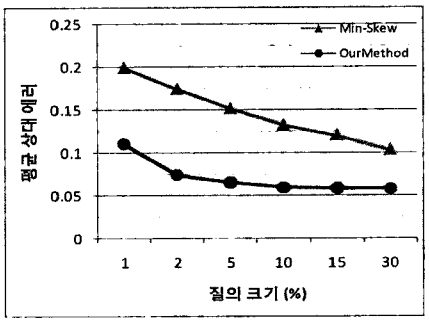
우리는 각 실험마다 10,000개 테스트 질의를 이용하였다. 각 질의는 데이타 공간에 임의로 위치하도록 설정되었다. 질의 크기에 따른 성능을 측정하기 위해, 본 연구는 다양한 크기(전체 데이타 영역의 1%, 2%, 5%, 10%, 15%, 30%)의 질의를 이용하였다.

4.2 결과 및 분석

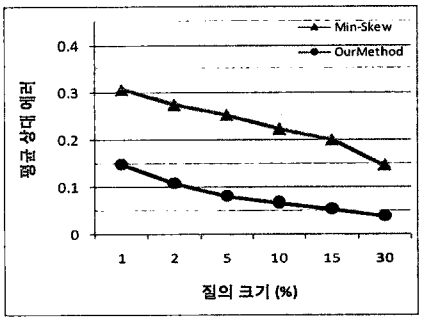
히스토그램의 성능을 평가하고자 본 연구는 여러 실험을 수행하였다. 이 절에서는 다음의 매개변수를 변경하며 수행한 히스토그램 성능 평가의 결과를 살펴본다.

- 질의 크기
- 데이타 불균등 정도 (skew degree)
- 버킷 수 즉 히스토그램 크기

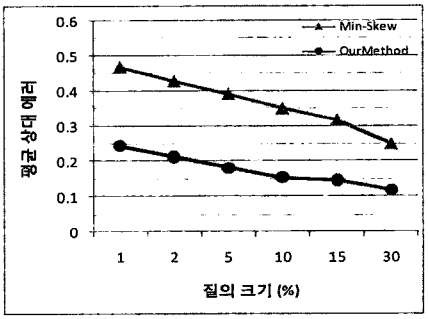
그림 6은 다양한 질의 크기에 대한 성능 평가 결 과를 보여준다. 이 성능 평가에는 전체 영역 1%~30%의 다양한 질의 크기가 이용되었고, 데이타 불균등 정도 0.3~0.7의 다양한 데이타셋이 이용되었다. 버킷 수는 400으로 설정되었다. 그림 6에서 히스토그램의 성능은 질의 크기가 커짐에 따라 향상되는 경향을 보인다. 이러한 경향의 원인은 다음과 같다. 주어진 질의 영역과 부



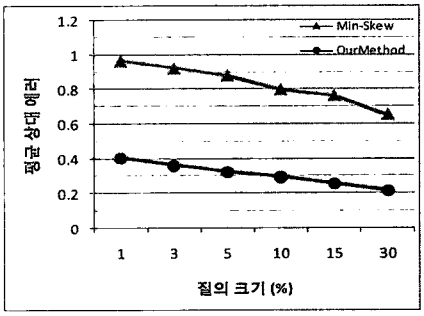
(a) skew degree 0.3



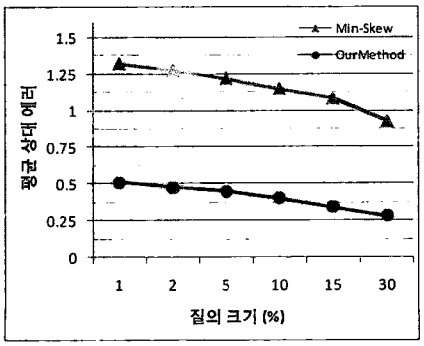
(b) skew degree 0.4



(c) skew degree 0.5



(d) skew degree 0.6



(e) skew degree 0.7

그림 6 질의 크기에 대한 성능 평가

분적으로 겹치는 버킷은 추정치를 반환하는 반면 질의 영역에 포함되는 버킷은 객체빈도수 즉 정확한 값을 반환한다. 질의 크기가 커짐에 따라 질의 영역과 부분적으로 겹치는 버킷에 비해 질의 영역에 포함되는 버킷의 수가 증가하기 때문에 이와 같은 경향이 나타난다. 그러나, 제안하는 히스토그램이 다양한 데이타셋에 대해 모든 질의 크기에서 Min-Skew 히스토그램보다 향상된 성능을 보임을 확인할 수 있다. 이 실험에서 우리는 최대 74%의 성능 향상을 확인하였다. 이러한 성능 향상의 요인은 제안하는 기법이 정확도 저하의 원인인 클러스터를 보다 효과적으로 처리하기 때문이다. 제안하는 기법은 핫스팟을 탐색하고 이를 히스토그램의 버킷 구성에 활용한다.

그림 7은 다양한 데이타 불균등에 대한 성능 평가 결과를 보여준다. 이 성능 평가에는 데이타 불균등 정도 0.1~0.9의 다양한 데이타셋이 이용되었다. 질의는 전체 영역 15%의 크기의 질의가 이용되었으며, 버킷 수는 400으로 설정되었다. 그림 7에서 히스토그램 성능은 데이타 불균등 정도가 높아질수록 낮아지는 즉 평균 상대 에러가 증가하는 경향을 보인다. 그러나, 제안하는 히스토그램은 데이타 불균등에 더욱 견고한 성능을 보임을

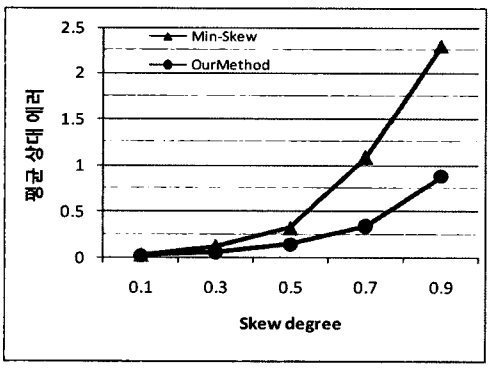


그림 7 데이타 불균등에 대한 성능 평가

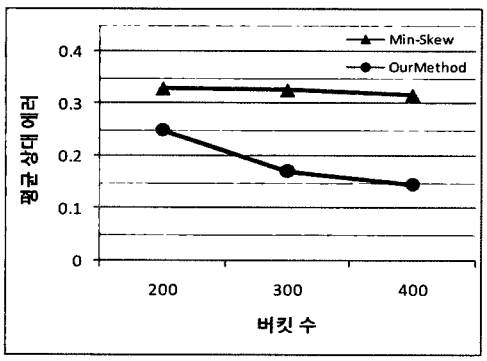


그림 8 버킷 수에 대한 성능 평가

확인할 수 있다. 그림에서 비교적 상당히 균일한 데이타 불균등 정도 0.1의 데이타셋에서는 두 히스토그램 모두 매우 높은 정확도를 보인다. 그러나, 데이타 불균등 정도가 증가함에 따라 제안하는 히스토그램이 보다 안정적인 성능을 제공함을 확인할 수 있다. 이 실험에서 우리는 최대 68%의 성능 향상을 확인하였다. 예를 들어, 데이타 불균등 정도 0.7 데이타셋에 대한 Min-Skew 히스토그램의 에러는 1.08인 반면 제안하는 히스토그램의 에러는 0.34이며, 이로써 68% 성능 향상을 확인하였다.

그림 8은 다양한 버킷 수에 대한 성능 평가 결과를 보여준다. 이 성능 평가에는 200~400개의 버킷이 이용되었고, 데이타 불균등 정도 0.5의 데이타셋이 이용되었다. 질의는 전체 영역 15% 크기의 질의가 이용되었다. 그림 8에서 히스토그램 성능은 버킷 수가 증가함에 따라 향상되는 즉 평균 상대 에러가 감소하는 경향을 보여준다. 그러나, 제안하는 히스토그램이 모든 버킷 수에서 향상된 성능을 보임을 확인할 수 있다. 우리는 이 실험에서 최대 54%의 성능 향상을 확인하였다. 이러한 성능 향상의 요인은 제안하는 기법이 버킷 수가 증가함에 따라 더욱 정확한 버킷 통계정보를 획득하기 때문이다. 그림에서 400개 버킷으로 구성된 제안하는 히스토그램은 15% 에러 이하의 우수한 성능을 제공한다. 반면, Min-Skew 히스토그램은 32% 에러를 보이고 있으며, 이는 제안하는 기법에 비해 대략 200% 높은 수치이다.

5. 결론 및 향후 계획

히스토그램은 질의 결과의 크기를 추정하는 데 널리 이용되고 있다. 이러한 질의 결과 크기의 추정치는 데이타베이스 시스템의 질의 최적화에 유용하게 활용된다. 본 연구는 데이타가 균등하게 분포하지 않은 상황에서 기존 방식보다 향상된 정확도를 제공하는 새로운 히스토그램 기법을 제안하였다. 제안하는 히스토그램 기법은 주어진 데이타 분포 내에 존재하는 고 밀도 영역을 탐색하고 그 정보를 히스토그램 생성에 활용한다. 이를 통해 제안하는 기법은 고 밀도 영역에 의한 정확도 저하를 효과적으로 감소시킬 수 있다.

본 연구는 고 밀도 데이타 영역 정보를 히스토그램을 이용한 질의 결과 추정에 활용하기 위해, 우선 핫스팟의 개념을 정의하였고 핫스팟 기반의 히스토그램 생성법을 제안하였다. 그리고, 제안하는 히스토그램을 이용한 질의 결과 크기의 추정법을 제시하였다. 본 연구는 성능 평가 실험을 통하여 제안하는 히스토그램이 데이타가 균등하게 분포하지 않은 여러 상황에서 가장 널리 활용되고 있는 기존 기법보다 정확한 추정을 제공함을 보였다.

향후 연구로서, 우리는 추정 성능 향상을 위한 연구를 진행하고자 한다. 우선, 주어진 데이타 분포의 특성에

따른 핫스팟 조건 즉 크기 및 객체빈도수 설정에 관한 연구가 요구된다. 또한, 버킷 계층구조 생성 과정에서 버킷 수의 균등 할당에 대한 개선이 필요할 것으로 판단된다. 이에 대한 개선안으로 우리는 버킷 내 데이타의 불균등에 기반한 차등적 분배를 고려하고 있다.

참 고 문 헌

[1] Y. E. Ioannidis, "The History of Histograms," VLDB, pp. 19-30, 2003.

[2] Y. E. Ioannidis, "Query Optimization," Computing Surveys, Vol.28, No.1, pp. 121-123, 1996.

[3] V. Poosala and Y. E. Ioannidis, "Selectivity estimation without the attribute value independence assumption," VLDB, pp. 486-495, 1997.

[4] N. Bruno, S. Chaudhuri, and L. Gravano, "ST Holes: A Multidimensional Workload-Aware Histogram," ACM SIGMOD, pp. 211-222, 2001.

[5] U. Srivastava, P. Haas, V. Markl, N. Megiddo, M. Kutsch, and T. Tran, "ISOMER: Consistent Histogram Construction Using Query Feedback," ICDE, pp. 39-44, 2006.

[6] S. Acharya, V. Poosala, and S. Ramaswamy, "Selectivity estimation in spatial databases," ACM SIGMOD, pp. 13-24, 1999.

[7] D. Gunopulos, G. Kollios, V. J. Tsotras, and C. Domeniconi, "Approximating Multi-Dimensional Aggregate Range Queries over Real Attributes," ACM SIGMOD, pp. 463-474, 2000.

[8] V. Markl, P.J. Hass, M. Kutsch, N. Megiddo, U. Srivastava, and T. M. Tran, "Consistent selectivity estimation via maximum entropy," Journal of VLDB, Vol. 16, No.1, pp. 55-76, 2007.

[9] C. Faloutsos and I. Kamel, "Relaxing the Uniformity and Independence Assumptions Using the Concept of Fractal Dimension," Journal of Computer and System Science, Vol.55, No.2, pp. 229-240, 1997.

[10] M. Garofalakis and P.B. Gibbons, "Approximate Query Processing: Taming the Terabytes," Tutorial in VLDB, 2001.

[11] M. Muralikrishna and D. J. DeWitt, "Equi-depth histograms for estimating selectivity factors for multidimensional queries," ACM SIGMOD, pp. 28-36, 1988.

[12] V. Markl, N. Megiddo, M. Kutsch, T. Tran, P. Hass, and U. Srivastava, "Consistently Estimating the Selectivity of Conjuncts of Predicates," VLDB, pp. 373-384, 2005.

[13] S. Muthukrishnan, V. Poosala, and T. Suel, "On rectangular partitionings in two dimensions: Algorithms, complexity, and applications," 7th International Conference on Database Theory, pp. 236-256. 1999.

[14] Y. J. Choi, and C. W. Chung, "Selectivity Estimation for Spatio-Temporal Queries to Moving Objects," ACM SIGMOD, pp. 440-451, 2002.

[15] Y. Tao, J. Sun, and D. Papadias, "Selectivity Estimation for Predictive Spatio-Temporal Queries," ICDE, pp. 417-428, 2003.

[16] H. K. Park, J. H. Son, and M. H. Kim, "Dynamic histograms for future spatiotemporal range predicates," Information Sciences, Vol. 172, No. 1-2, pp. 195-214, 2005.

[17] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive Skyline Computation in Database Systems," ACM Transactions on Database Systems, Vol. 30, No. 1, pp. 41-82, 2005.

[18] J. Sun, Y. Tao, D. Papadias, and G. Kollios, "Spatio-temporal Join Selectivity," Information Systems, Vol. 31, No. 8, pp. 793-813, 2006.

[19] E. Frentzos, K. Gratsias, and Y. Theodoridis, "On the Effect of Location Uncertainty in Spatial Querying," IEEE Transactions on Knowledge and Data Engineering, preprint, 30 July 2008, doi: 10.1109/TKDE.2008.164.

[20] J. Roh, H. K. Park, K. W. Min, and M. H. Kim, "A Histogram Utilizing the Cluster Information," Technical Report CS/TR-2004-210, KAIST, Nov. 2004.

[21] Y. Theodoridis and M. Naseimento, "Generating Spatiotemporal Datasets on the WWW," ACM SIGMOD, pp. 39-43, 2000.

부 록 A

핫스팟 매개변수 결정에 관한 논의 - GenHist 기법과 비교 중심

본 연구의 핫스팟 매개변수 즉 s와 f는 버킷으로 구성될 영역의 크기 및 객체빈도수에 대한 조건을 지정한다. 이러한 측면에서 핫스팟 매개변수는 GenHist 기법의 매개변수 즉 격자 크기 및 각 격자 당 생성 버킷 수와 유사한 목적을 갖는다. 그러나, 핫스팟 매개변수의 결정은 GenHist 기법의 매개변수 결정과 다음과 같은 차이점이 있다. 첫째, GenHist 히스토그램은 계층구조를 구성하지 않으며 버킷 영역 크기는 전체 도메인 영역에 대한 상대적인 수치로 설정된다. 예를 들어, 100×100 격자 이용 시 버킷 영역 크기는 도메인-영역-크기/10000과 같다. 반면, 제안하는 히스토그램은 계층구조를 구성하며 버킷 영역 크기는 자신을 포함하는 버킷 즉 부모 버킷 영역에 대한 상대적인 수치로 설정된다. 예를 들어, s=10 경우 버킷 영역 크기는 부모버킷-영역-크기/10과 같다. 그리고, GenHist 기법은 특정 수의 최대 객체빈도수 격자 셀이 버킷으로 구성되며 버킷 객체빈도수 조건은 객체빈도수에 대한 셀의 상대적 순위로 설정된다. 반면, 본 연구의 버킷 객체빈도수 조건은 부모 버킷의 객체빈도수에 대한 상대적인 수치로 설정된다. 둘째, GenHist

기법의 격자 크기 및 각 격자 당 생성 버킷 수는 사용자에 의해 결정되는 매개변수이다. 즉, GenHist 기법은 사용자가 버킷 영역 크기와 객체빈도수 조건을 결정하도록 한다. 그러나, 제안하는 기법은 정의 3의 상대-에러와 버킷 수를 고려한 속성 1 및 속성 2에 기반하여 핫스팟 매개변수를 히스토그램 생성 과정에 자동 계산한다. 결국, 본 연구의 핫스팟 매개변수는 GenHist 기법의 매개변수와 유사한 목적을 갖지만 의미 및 결정 주체 측면에서 차이점이 존재한다.

노 요 한

2002년 경북대 컴퓨터공학과 졸업(학사)
2004년 KAIST 전산학과 졸업(석사)
2004년~현재 KAIST 전산학과 박사과정. 관심분야는 데이타베이스, 센서 네트워크, 시맨틱 웹

정 연 돈

정보과학회논문지 : 데이타베이스
제 35 권 제 3 호 참조

김 호 진

2002년 홍익대 컴퓨터공학과 졸업(학사)
2004년 KAIST 전산학과 졸업(석사)
2004년~현재 KAIST 전산학과 박사과정. 관심분야는 컴퓨터 아키텍처, 센서 네트워크, 그리드 컴퓨팅

김 명 호

정보과학회논문지 : 데이타베이스
제 35 권 제 3 호 참조