속성값 구간 배열을 이용한 계층 상이값수의 계산 기법

(Estimating the Number of Hierarchical Distinct Values Using Arrays of Attribute Value Intervals)

송 하 주　김 형 주

서울대학교 컴퓨터공학과

요약

관계형 데이터베이스 시스템의 각 테이블은 레코드의 집합이며 각 레코드는 일련의 속성들의 집합으로 이루어진다. 속성에 대한 상이값수란 레코드의 속성에 대해 실제로 데이터베이스 내에 사용되고 있는 서로 다른 속성값의 개수를 나타내며 질의 최적화나 통계적 질의의 지원에 유용하게 사용된다. 한편 기존 관계형 데이터베이스 시스템과는 달리 객체 관계 데이타베이스 시스템은 테이블 간의 계승관계를 지원하므로 상위 테이블에서 정의된 속성을 하위 테이블에서 계승받게 된다. 따라서 상이값수 또한 단일 테이블에 관한 정보뿐만 아니라 하위 테이블의 속성 정보를 모두 반영하는 계층 상이값수가 필요하다.

본 논문은 기존 상이값수 측정 방법을 그대로 사용하되 계층 상이값수을 계산할 수 있는 방법으로써 속성값 구간 배열을 이용하는 기법을 제안한다. 이 기법은 해당 테이블과 하위테이블에 대하여 각각 속성값 구간 배열을 구성하고 그것을 합병함으로써 계층 상이값수를 계산한다. 이 기법은 작은 양의 저장 공간만을 사용하여 계층 상이값수를 정확히 구할 수 있게 하며 계층 내의 각 테이블에 대한 갱신 연산이 불균등하게 이루어지는 환경에서 더욱 효과적으로 이용될 수 있다.

Abstract

In relational database management systems(RDBMS), a table consists of sets of records which are composed of a set of attributes. The number of distinct values(NDV) of an attribute denotes the number of distinct attribute values that actually appear in the database records, and is widely used in optimizing queries and supporting statistic queries. An object-relational database management systems(ORDBMSs), however, support the inheritance between tables which enforces an attribute defined in a super-table to be inherited in sub-tables automatically.

Hence, in ORDBMSs, not only the NDV of an attribute in a single table but also the NDV of an attribute in multiple tables(HNDV) is needed.

In this paper, we propose a method that calculate the HNDV using arrays of attribute value intervals. In this method, an array of attribute value intervals is created for an attribute of interest in each table in a table hierarchy, and HNDV can be calculated or estimated by merging the arrays of attribute value intervals. The method accurately calculates the HNDV using small additional storage space and is very efficient for an environment where only some of the tables in a table hierarchy are frequently updated.

1 서론

1.1 계층 상이값수의 정의와 문제점

대다수 데이터베이스 시스템의 사용자 혹은 응용프로그램 개발자는 질의어를 사용하여 데이터베이스 내의 데이터를 검색한다. 질의어는 검색을 원하는 데이터가 가지는 조건을 표현하기 위해 사용되며 질의처리기는 질의 조건에 맞는 데이터를 실제로 데이터베이스에서 추출해내기 위한 데이터베이스 시스템의 내부 모듈이다. 질의어처리기가 주어진 질의를 처리하는 과정은 파싱 단계, 최적화 단계, 실행계획의 생성 단계, 실행 단계로 이루어진다. 이중 최적화 단계는 가능한 여러 가지의 질의 수행 방법 중에서 비용 모델에 근거하여 최소의 비용으로 질의 결과를 도출해낼 수 있는 수행방법을 찾는 과정이다. 이를 위해서는 인덱스의 존재 유무, 인덱스의 종류, 테이블 내의 각 속성에 대한 최소값, 최대값, 히스토그램과 상이값수(NDV)와 같은 데이터에 관한 각종 통계 자료들이 사용된다 [1, 2, 7, 10].

여기서 속성에 대한 히스토그램이란 해당 속성값의 분포 상황을 기록하여 특정 속성값 또는 특정 범위의 속성값에 해당하는 레코드가 몇 개정도 존재하는가를 기록한 것이고 상이값수란 특정 속성에 대해 테이블 내에 실제로 존재하는 속성값들 중 서로 다른 것이 몇 개나 되는지를 나타낸 것이다. 이와같은 통계 값들을 사용하는 경우 질의처리기는 더욱 효율적인 실행계획을 생성할 수 있으며 데이터베이스에 대한 통계적인 질의에 대해서는 질의를 실제로 수행하지 않고 결과를 구할 수 있는 장점이 있다. 기존 관계형 데이터베이스 시스템에서는 단일 테이블에 대한 속성 히스토그램과 상이값수 정보만을 유지하므로 각각 단일 속성 히스토그램과 단일 상이값수(TNDV)라 하자.

한편 객체관계 데이터베이스 시스템이나 객체지향 데이터베이스 시스템(Object-Oriented Database Management Systems, ODBMS)은 테이블(혹은 클래스)들 간의 계층구조를 제공한다. 그림 1는 간단한 테이블 계층구조를 보인 것이다. 여기서 Person 테이블은 최상위 테이블이고 Employee 테이블,

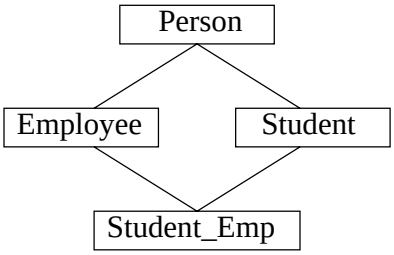


그림 1: 예제 테이블 계층구조

Student 테이블, Student_Emp 테이블은 모두 Person 테이블의 하위 테이블이며 Student_Emp 테이블은 Employee 테이블과 Student 테이블에 동시에 하위테이블이 된다. 이와 같은 계층구조의 지원은 최상위 테이블에서 정의된 모든 속성이 하위 테이블에도 포함되게 한다. 따라서 계층구조 내의 특정 테이블을 질의의 조건으로 사용하거나 질의의 결과로 추출하는 경우, 해당 테이블뿐만 아니라 그것의 하위 테이블까지 질의의 대상이 될 수가 있다[8, 12]. 그러므로 계층구조를 지원하는 데이터베이스 시스템에서 효과적인 질의어 최적화를 수행하기 위해서는 각 테이블과 그것의 하위 테이블을 논리적으로 하나의 테이블로 간주하고 그것에 포함된 속성들에 대한 히스토그램(계층 히스토그램) 및 상이값수(계층 상이값수)에 대한 정보를 필요로 한다.

계층 히스토그램은 기존 관계형 데이터베이스 시스템에서와 동일하게 각 테이블에 대한 단일 히스토그램을 유지하고 그것들의 합을 이용하여 효과적으로 지원할 수가 있다(부록 참조). 그러나 계층 상이값수(HNDV)를 구하는 것은 단순하지 않다. 만일 계층 상이값수를 계산하기 위해 별도의 기법을 사용하지 않고 기존 방식을 그대로 이용하는 경우에는 다음과 같은 방법을 사용할 수 있을 것이다.

1. 각각의 테이블에 대해 독립적으로 표본조사나 전수조사(표본율 100%) 통해 단일상이값수를 계산한 뒤 각 테이블의 계층 상이값수는 해당 테이블과 하위 테이블의 단일 상이값수의 합을 사용한다.

2. 각각의 테이블에 대해 해당 테이블과 모든 하위 테이블을 하나의 논리적인 테이블로 간주하고 모든 테이블로부터 고르게 표본을 구성하거나 모든 테이블에 대한 전수조사를 수행한다.

첫 번째 방법은 동일한 속성값이 다수의 테이블에 존재할 수 있기 때문에 정확한 계층 상이값수를 계산해낼 수는 없을 것이다. 그에 반해 두번째 방법은 비교적 정확한 계층 상이값수를 구할 수는 있다. 그러나 최신의 계층 상이값수를 유지하기 위해서는 많은 부하를 유발하게 된다. 왜냐하면 히스토그램이나 상이값수와 같은 통계정보는 항상 데이터베이스의 최근 상태를 반영하도록 유지하는 것이 중요한 관건인데, 만일 계층구조상의 어느 한 테이블에 대해 상당한 변경이 이루어진 경우에도 계층 상이값수를 갱신하기 위해서는 해당 테이블 이외에도 그것을 하위 클래스로 포함하고 있는 모든 테이블에 대한 계층 상이값수를 다시 계산해 주어야만 하기 때문이다. 반면 첫 번째 방법은 변경이 이루어진 해당 테이블의

단일 상이값수를 갱신하고 그 외의 테이블에 대해서는 변화된 개수만을 가감하면 된다.

본 논문은 위의 두 방법의 이점을 살릴 수 있는 속성값 구간 배열을 이용하는 방안을 제시한다. 이 방법을 이용하면 소량의 저장공간을 필요로 하면서도 오차가 크지 않게 계층 상이값수를 계산할 수 있으며 계층구조내의 각 테이블에 대해 특정 테이블에 편중된 갱신이 일어나더라도 효과적으로 계층 상이값수를 파악할 수 있다.

1.2 관련 연구

System R과 같은 초창기의 데이터베이스 시스템에서는 조인의 선택율을 예측하기 위해서 해당 속성에 대한 상이값수를 사용했다[11]. 그러나 이것은 전수조사나 표본조사를 통한 실제적인 통계 값을 이용하지 않고 일정간격 분포(uniform)를 가정한 것이었기 때문에 오차율도 높고 경우에 따라서는 최악의 실행 계획을 선택하는 경우도 있다[2, 9]. 따라서 근래의 대다수의 상용시스템은 직접적인 조사를 통해 히스토그램을 구성하고 상이값수 파악하는 것이 일반화되었다 [1, 6, 10].

[3, 4, 5]는 질의처리의 기본 연산인 선택, 조인, 프로젝션 연산 등에 관한 결과로 생성되는 레코드의 개수를 표본조사를 통해 예측하는 기법을 제안했다. 특히 [5]는 시간적 제한을 감안한 표본조사를 통해 집합(aggregation) 질의를 처리할 수 있음을 보였다.

[2]는 표본조사에의해 상이값수를 추정함에 있어 통계학적 기법들과 데이터베이스 관련 연구에서 제시되었던 기법들의 성능을 실제 데이터를 이용해 비교하였다. 그리하여 데이터의 분포 형태에따라 추정 기법의 성능은 크게 달라지며 실험에 제시된 모든 형태의 분포에대해 만족스런 성능을 보이는 추정 기법은 없다는 것을 보이고 기존 추정 기법들을 짜맞춘 하이브리드 추정 기법을 제안하였다.

[1]은 기존에 제시된 어떠한 기법도 오차가 매우 작은 정도로 예측하는 것은 불가능함을 수학적으로 증명하였다. 그러나 오차율이 적당한 수준으로 제공되면 실제적으로 사용하는데는 문제가 없는 것으로 보고 기존 방식보다는 간략한 형태의 추정방식을 사용하였다.

이와 같이 기존 연구는 단일 테이블 내에서의 전수조사를 효과적으로 수행하는 방법 또는 표본조사에 의해 비교적 정확하게 특정 속성에 대한 상이값수를 추정하는 방법에 대한 연구가 주를 이루고 있으며 아직 테이블 계층구조를 감안한 상이값수 계산 기법은 제안되고 있지 않다.

1.3 논문의 구성

본 논문의 구성은 다음과 같다. 2장에서는 계층 상이값을 계산하기위해 본 논문에서 제안하는 속성값 구간 배열을 이용한 기법을 설명하고 3 장에서는 실험을 통해 제안된 기법의 성능을 검증한다. 마지막으로 4 장으로 결론을 맺는다. 부록은 기존 RDBMS에서 사용하는 히스토그램을 이용하여 계층 선택율

을 계산하는 방법과 그 타당성에 대해 간략히 설명한다.

2 속성값 구간 배열을 이용한 계층 상이값수의 계산

제안하고자 하는 계층 상이값수의 계산 기법은 다음과 같이 두 단계를 거쳐 계층구조상의 각 테이블에 대해 주어진 속성에 대한 계층 상이값수를 계산한다.

1. 단일 속성값 구간 배열 구성 단계 :

   각각의 테이블에 대해 조사(전수조사 또는 표본조사))를 통해 단일 상이값수와 단일 속성값 구간 배열을 구성한 후 레코드 형태로 저장한다. 표본 조사의 경우에는 단일상이값수는 기존 연구를 통해 제안된 수식을 사용하여 추정한다.

2. 속성값 구간 배열 합병 및 계층 상이값수의 계산 단계 :

   각 테이블에 대해 해당 테이블과 하위 테이블의 단일 속성값 구간 배열을 합병하고 그 결과를 이용하여 해당 테이블의 계층 상이값수를 계산한다.

즉 각 테이블 속성에 대해 해당 테이블 내에 존재하는 모든 속성값을 나타내는 속성값 구간 배열을 구성한 다음, 해당 테이블과 그것의 하위 테이블들의 속성값 구간 배열을 합병하여 계층 상이값수를 계산하는 것이다.

이와 같이 속성값 구간 배열을 사용하면 테이블 계층 구조상의 일부 테이블에 대한 변경만이 이루어진 경우에도 계층 구조상의 모든 테이블을 다시 조사해야할 필요가 없다. 빈번히 변경이 일어나는 테이블에 대해서만 주기적으로 속성값 배열을 재 구축하면 된다. 따라서 상이값수 계산에 따르는 부하를 감소시킬 수 있다. 반면에 계층 구조상의 모든 테이블에 대한 조사를 수행한 것과 동일한 효과를 가지므로 정확한 상이값수의 계산이 가능하다. 이어지는 두 장에서는 위의 두 과정에 대해 자세하게 설명한다.

2.1 속성값 구간 배열 구성 단계

계층 상이값수를 구하고자 하는 속성값에 대해 각 테이블별로 속성값 구간 배열을 구성하는 단계이다. 조사를 통해 추출된 속성값들과 최소구간간격(w)이 입력으로 주어지며 계층구조내의 각 테이블 $T_i (i = 1, \ldots, N)$에 대해 다음과 같은 자료를 생성한다.

- 표본 상이값수($SNDV_i$): 표본조사를 통해 추출된 속성값에서의 상이값의 개수
- 단일 상이값수($TNDV_i$): 테이블 내의 단일 상이값의 개수[1]

[1] 표본조사의 경우 기존 연구에서 제안된 추정공식을 사용하여 추정된다. 본 논문의 성능평가에서는 참고문헌 [1]에서 제시한 상이값의 개수 예측 공식을 사용했다. 단 전수조사의 경우에는 $TNDV_i = SNDV_i$ 이다.

- 구간 배열내의 상이값수($INDV_i$): 속성값 구간 배열에 포함된 상이값의 개수
- 단일 속성값 구간 배열

이 단계에서는 표본 조사에 의해 선택된 속성값들에 대해 다음과 같은 과정을 수행한다.

1. 주어진 모든 속성값을 정렬한다.
2. 정렬된 속성값 중 중복된 값은 모두 제거한다.
3. 중복이 제거된 속성값들에 대해 속성값 구간 배열을 구성하여 레코드 형태로 저장한다.

속성값 구간 배열은 테이블 내에 존재하는 모든 속성값을 유지하는 것으로 개개의 속성값을 저장하는 것이 아니라, 속성값이 연속적으로 이어지는 여러 개의 구간에 대해 각 구간의 처음과 끝 값만을 유지하는 것이다. 즉 <구간의 시작, 구간의 끝> 형태로 저장한다.

여기서 연속된 두 구간의 간격이 주어진 w보다 작은 경우에는 두 구간을 합쳐서 하나의 구간으로 설정한다. 따라서 결과적으로 생성되는 속성값 구간 배열은 $w = 1$인 경우에 표본내에 존재하는 속성값을 오차 없이 유지한다($SNDV_i = INDV_i$). $w > 1$ 인 경우에는 실제 존재하지 않는 속성값을 포함할 수 있으므로 오차를 유발할 수 있다.

속성값 구간 배열에 포함되고 실제로 테이블 내의 레코드에도 존재하는 값을 참 속성값이라하고 속성값 구간 배열에는 포함되나 그값을 갖는 실제 레코드가 존재하지 않는 값을 거짓 속성값이라 하자. T_i에 대해 거짓 속성값에 의해 발생하는 $SNDV_i$ 와 $INDV_i$의 오차를 구간오차라 하고 $SNDV_i$에 대한 구간 오차($IE_i = INDV_i - SNDV_i$)의 비를 구간 오차율($IR_i = \frac{IE_i}{SNDV_i} = \frac{INDV_i - SNDV_i}{SNDV_i}$)로 정의한다.

$w > 1$인 경우에는 구간오차가 발생할 가능성이 늘어나지만 그에 비례하여 속성값 구간 배열의 개수도 줄어드는 장점이 있다. w의 값은 데이터베이스 시스템 관리자에 의해 지정된다. 예제 1은 속성값 구간 배열의 구성에 대한 간단한 예를 보인 것이다.

예제 1 $w = 2$이고, T_i에 대한 전수조사를 통해 추출된 속성값들이 다음과 같은 경우,

2, 11, 6, 14, 3, 1, 10, 2, 11, 15, 6, 4, 10, 7, 2

속성값 정렬과 중복 제거를 거치면,

1, 2, 3, 4, 6, 7, 10, 11, 14, 15

이 되고, 이로부터 생성되는 속성값 구간 배열은 다음과 같다.

<1, 7>, <10, 11>, <14, 15>

이 때 $TNDV_i = SNDV_i = 10,\ INDVi = 11$ 이 된다. 따라서 이 경우 $w = 2$에 따른 속성값 구간 배열의 오차율: $IR_i = \frac{11-10}{10} \times 100 = 10\%$ 가 된다.

2.2 속성 배열 합병 단계

이 단계는 각각의 테이블에 대한 단일 속성값 구간 배열을 이용하여 각 테이블(T_i)에 대한 계층 상이값수($HNDV_i$)를 계산한다.

T_i에 대한 계층 속성값 구간 배열이란 T_i의 속성값 구간 배열과 그것의 하위 테이블의 속성값 구간 배열을 합병한 것이고 계층 속성값 구간 배열 내의 상이값수를 $HINDV_i$라 하면 $HNDV_i$는 $HINDV_i$를 통해 계산된다.

알고리즘 1은 이 과정에서 사용되는 것으로 특정 속성을 포함하는 계층구조상의 모든 테이블에 대한 속성값 구간 배열들을 입력받아 각 테이블에 대한 $HINDV$를 계산한다.

이 알고리즘은 각각의 속성값 구간 배열들로부터 구간을 버퍼로 읽어들이고 버퍼 내의 구간들을 정렬한다. 그리고 버퍼 내의 구간들을 차례로 읽으면서 속성값의 개수를 계산한다. 따라서 계층 속성값 구간 배열은 실제로는 생성되지 않고 다만 테이블별로 $HINDV$만 계산된다.

Algorithm 1 계층 상이값수를 구하기 위한 병합 알고리즘

```
prev_key 배열은 minimum 값으로, HINDV 배열과 insiders 배열은 0으로 각각 초기화한다.
각 테이블의 속성값 구간 배열의 첫 번째 구간의 시작값과 구간의 끝값을 버퍼로 읽어들여 오름차순으로 정렬한다[2].
while 버퍼에 데이터가 있는 동안 do
  curval ← 버퍼내의 최소 속성값[3].
  for T_i in curval의 테이블과 그것의 상위 테이블들 do
    if insiders_i != 0 then
      HINDV_i ← HINDV_i+구간 [prev_val, curval) 내에서의 상이값의 개수
      if Tag(curval) = 'TO' then
        preval_i ← curval
        insiders_i = insiders_i − 1
        if insiders_i == 0 then
          HINDV_i ← HINDV_i + 1
        end if
        continue
      end if
    end if
    prev_val_i ← curval
    insiders_i = insiders_i + 1
  end for
  if Tag(curval) = 'TO' then
    Table(curval)의 속성값 구간 배열에서 다음 번 속성값 구간을 버퍼로 읽어 들이고 버퍼내의 값을 정렬한다.
  end if
end while
```

그림 2와 예제 2는 상위 테이블 T_1과 그것의 하위 테이블 T_2으로 이루어진 테이블 구조에 대해 각각의 구간 배열을 이용하여 상위 테이블 T_1의 계층 상이값수를 계산하는 과정을 보인 것이다. 여기서 T_1의 구간 배열은 두 개의 구간 $<(a),(b)>$와 $<(e),(g)>$로 이루어지고 T_2의 구간 배열은 하나의 구간 $<(b),(g)>$로 구성된 것으로 한다. 그림 2에서 계층 구간 배열(hierarchy interval array)는 실제로 생성되는 것이 아니며 2.3 절의 내용을 설명하기 위해 그것이 생성된 것으로 가정했을 때의 구조를 나타낸 것이다.

예제 2 다음은 알고리즘 *1*에 의해 그림 *2*의 T_1과 T_2의 구간 배열을 병합하여 T_1의 $HINDV_1$을 구하는 과정이다.

curval	*Tag(curval)*	$prev\_val_1$	$insiders_1$	$HINDV_1$
		minimum	*0*	*0*
(a)	FROM	*(a)*	*1*	*0*
(b)	FROM	*(b)*	*2*	*1*
(b)	TO	*(b)*	*1*	*1*
(e)	FROM	*(e)*	*2*	*4*
(g)	TO	*(g)*	*1*	*6*
(g)	TO	*(g)*	*0*	*7*

따라서 $HINDV_1$은 *7*이 된다.

알고리즘 1에의해 각 테이블에 대한 HINDV가 계산되면 이를 통해 각 테이블에 대한 계층 상이값수를 다음과 같이 계산한다.

- 테이블 T_i의 계층 상이값 구간 배열을 구성하기 위해 사용한 단일 상이값 구간 배열이 모두 전수조사에의해 생성한 것이라면

$$HNDV_i = HINDV_i$$

- 만일 합병에 사용된 단일 속성값 구간 배열을 구성하는데 있어 표본조사를 사용한 것이 하나 이상 존재하는 경우에는 다음과 같은 공식을 사용하여 계층 상이값수를 추정한다.

[2] 각 속성에 대해서는 구간의 시작값인지 끝값인지를 표시하는 TAG('FROM' 또는 'TO')와 테이블 식별자를 표시한다. 속성값이 같은 경우에는 구간의 시작값을 끝값보다 작은 값으로 본다.

[3] 읽혀진 속성값은 버퍼에서 제거된다.

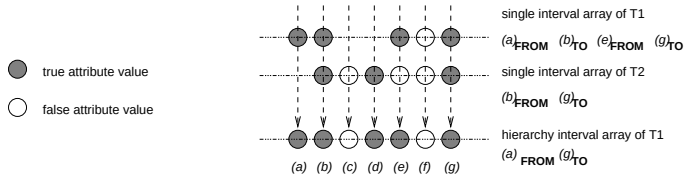


그림 2: 구간 배열의 병합 과정

$$HNDV_i = \sum TNDV_j \times \frac{HINDV}{\sum INDV_j} \tag{1}$$

여기서 j는 테이블 T_i와 그것의 하위 테이블을 나타낸다. 표본조사를 이용하는 경우 실제로 테이블내에 존재하는 속성값의 개수가 표본을 통해 생성된 단일 속성값 구간 배열에 나타난 속성값의 개수보다 많다. 따라서 식 1은 표본의 합병에 따른 감소 비율을 이용하여 HNDV를 추정하는 것을 나타낸다.

2.3 최소 구간 간격 값에 따른 오차의 분석

제안하는 기법에 따라 계층 상이값수를 구하면 실제 계층 상이값수와는 오차가 발생할 수 있다. 오차의 원인은 두가지가 있으며 그 중 하나는 표본조사에 따른 오차이고 나머지 하나는 w를 1보다 크게 설정함에 따른 오차이다. 표본조사에 따른 오차는 표본의 추출 방법이나 데이터의 분포 특성에 좌우되므로 제안하는 기법과는 관계가 없다. 따라서 이 절에서는 w를 1보다 크게 설정하는 것이 계층 상이값수의 계산에 미치는 오차에 대해 설명한다.

먼저 거짓 속성값이 계층 상이값수를 계산하는데 어떻게 영향을 미치는지 알아본다. N개의 단일 테이블에서 생성된 속성값 구간 배열을 병합했을 때, 병합의 결과로 생성된 속성값 구간 배열에 속하는 속성값(v)은 다음의 다섯 가지 경우에서 하나에 해당하는 값이다(그림 2 참조).

1. v는 두개 이상의 속성값 구간 배열에 속하는 값이며 v를 포함하는 모든 속성값 구간 배열에서 참 속성값이다 - 그림 2의 (b)와 (g)

2. v는 두개 이상의 속성값 구간 배열에 속하는 값이며 v를 포함하는 일부 속성값 구간 배열에서는 거짓 속성값이고 일부에서는 참 속성값이다 - 그림 2의 (e)

3. v는 두개 이상의 속성값 구간 배열에 속하는 값이며 v를 포함하는 모든 속성값 구간 배열에서 거짓 속성값이다 - 그림 2의 (f)

4. v는 오직 하나의 구간 배열에만 속하는 값이며 해당 구간 배열에서 참 속성값이다 - 그림 2의 (d)

5. v는 오직 하나의 구간 배열에만 속하는 값이며 해당 구간 배열에서 거짓 속성값이다 - 그림 2의 (c)

거짓 속성값으로 인한 오차가 가장 작게 발생하는 경우는 모든 거짓 속성값에 대해 (2)의 경우만이 적용되는 것이다. 이 경우에는 거짓 속성값에의한 계층 속성값 구간 배열의 구간 오차율은 0이 된다. 반면 거짓 속성값으로 인한 오차가 가장 크게 발생하는 경우는 모든 거짓 속성값은 (5)의 경우에 해당하고 참 속성값은 모두 (1)의 경우에 해당하는 경우이다.

그러면 거짓 속성값에 의한 오차가 가장 커지는 상황에서 단일 속성값 구간 오차율이 계층 속성값 구간 오차율에 미치는 영향을 알아본다. 아래의 식에서 $1, \ldots, K$는 T_i와 그것의 하위 테이블들을 의미한다.

$$\begin{aligned} HIR_i &= \frac{\text{계층 속성값 구간 배열에 포함된 거짓 속성값의 개수}}{\text{계층 속성값 구간 배열에 포함된 참 속성값의 개수}} \\ &< \frac{\sum IE_j}{max(SNDV_1,\cdots,SNDV_K)^4} \\ &= \sum \frac{IE_j}{max(SNDV_1,\cdots,SNDV_K)} \\ &\leq \sum \frac{IE_j}{SNDV_j} \\ &= \sum IR_j \end{aligned}$$

즉 어떠한 경우에도 거짓 속성값에 의해 발생한 계층 속성값 구간 배열의 오차율은 단일 속성값 구간 오차율의 합보다는 작게 된다. 따라서 최악의 경우일지라도 거짓 속성값에 의한 오차는 테이블의 개수가 늘어남에 따라 산술적으로만 증가함을 알 수 있다.

이 결과는 전수조사를 사용한 경우에 대해서는 계층 상이값수의 오차율에 직접 적용될 수 있다. 그러나 표본조사에 의한 경우에는 단일 상이값수의 추정에서도 이미 추정공식에 의해 오차가 발생하기 때문에 이 경우에는 위의 결과보다 더 큰 값의 오차율이 발생할 수 있다.

3 성능평가

성능 평가는 실제 데이터가 아닌 합성된 데이터를 이용한 결과이다. 표 1은 성능평가를 위해 실험 환경을 나타낸다. 각 테이블의 속성값이 가질 수 있는 영역을 속성값 영역이라 할 때, 표 1의 테이블간 속성값의 분포 형태는 각 테이블들간의 속성값 영역이 어느 정도 겹치는가를 나타낸 것이다. 그러므로 완전겹침이란 테이블들간의 속성값 영역이 모두 같은 것이고 무겹침은 속성값의 영역이 서로 달라 두개이상의 테이블에서 동시에 나타나는 속성값이 없는 경우를 나타낸다. 따라서 완전겹침과 무겹침은 속성값

[4] $max(v_1, \cdots, v_K)$은 $v_1, \cdots, v_K$ 중에서 가장 큰 값을 의미한다.

실험 조건	값
테이블당 레코드 수	1,000,000
테이블의 개수	30
속성값의 분포형태	정규분포
테이블간 속성값 분포 형태	완전겹침, 무겹침
조사방식	표본조사(0.1%, 0.5%, 1%, 5%, 10% 표본율), 전수조사
최소 속성값 구간 간격(w)	1, 2, 4, 8, 16

표 1: 실험 조건

분포의 양극단을 표현한 것이며 실제 데이터베이스에서는 양자의 중간 형태도 존재할 수 있다. 표본조사의 경우 전체 테이블의 상이값수를 예측하기 위해 [1]에서 제안된 기법을 사용하였다.

일반적으로 히스토그램 구성 기법 또는 단일 테이블에 있어 상이값수를 추측하는 기법의 성능 평가를 위해서는 속성값의 빈도에 대해 Zipfian 분포가 주로 사용되고 [1, 10], 속성값 자체의 분포에 대해서는 등간격, 임의 간격, Zipfian 간격 등을 사용하기도 한다. 그러나 본 논문이 제시하는 기법의 목적은 기존 연구의 대상이었던 표본조사를 효과적으로 수행하는 기법이나 표본조사를 통해 단일 테이블의 상이값수를 구하는 추정 공식을 새로이 제안하는 것이 아니라 단일 테이블에 대한 추정 방법은 기존 연구 기법들을 그대로 사용하되 계층 구조에 적용할 수 있는 방법을 제안하는 것이다. 따라서 속성값 분포에 대한 여러 형태 중의 하나인 정규분포만을 사용한 실험으로 그 특징을 간략히 나타내고자 하였다.

그림 3과 그림 4는 각각 완전겹침과 무겹침 조건으로 하위 테이블의 개수가 늘어남에 따라 제안하는 방식에의한 최상위 테이블의 계층 상이값수와 그것의 실제값과의 오차율을 나타낸 것이다. 각 그림의 우상단은 대응하는 그래프에서 사용된 표본율($0.5\% \sim 100\%$(전수조사))과 w의 값을 나타낸 것이다[5].

전수조사(그림 3-(a), 그림 4-(a))의 경우 $w = 16$으로 설정해도 최대 10%대의 오차율을 넘지 않음을 알 수 있다. 표본조사(그림 3-(b), 그림 4-(b))의 경우에는 0.1%의 표본조사 이외에는 최대 100% 가량의 오차율을 보인다.

제시된 실험결과에서 w값이 증가함에 따라 오차율이 증가하는 것을 볼 수 있는데 이것은 w의 값을 크게 할수록 거짓 속성값이 늘어났기 때문이다. 예외적으로 무겹침 조건에서 표본 조사를 수행한 경우(그림 4-(b)) w값이 영향을 거의 미치지 않는 것으로 나타난다. 그 이유는 표본조사의 경우 계층 상이값수를 계산하기 위해 식 1를 사용하는데, 무겹침의 경우 $HINDV_i = \sum INDV_j$이 되므로 식 1의 $ENDV_i = \sum TNDV_j$가 되고, w는 개개의 테이블에 대한 $TNDV_j$(단일 상이값수)를 추정하는데 영향을 미치지 않기 때문이다.

전수조사의 경우(그림 3-(a), 그림 4-(a)) 동일한 w 값을 적용하면 완전겹침에서는 하위 테이블의 개수가 증가할수록 오차율이 감소하는 경향을 보이나 무겹침은 별다른 변화를 보이지 않는다. 이와 같은

[5] 0.1%는 오차율이 500%를 넘어서는 관계로 그림에는 나타내지 않았다.

표본조사 비율	$w=1$	$w=2$	$w=4$	$w=8$	$w=16$
0.1% 표본조사	74.1(0%)	36.9(10%)	18.0(22%)	8.4(35%)	4.2(47%)
0.5% 표본조사	51.7(0%)	27.9(4%)	14.0(11%)	7.4(17%)	3.7(24%)
1% 표본조사	46.8(0%)	25.4(4%)	13.7(9%)	6.9(15%)	3.5(21%)
5% 표본조사	40.6(0%)	22.9(3%)	12.3(6%)	6.0(11%)	2.9(16%)
10% 표본조사	38.9(0%)	21.8(2%)	11.2(6%)	6.0(9%)	2.9(14%)
전수조사	32.4(0%)	18.2(2%)	9.5(4%)	4.7(7%)	2.5(10%)

표 2: 최소 구간값 간격에 따른 테이블당 구간 배열 레코드의 개수

현상을 보이는 이유는 완전겹침의 경우 테이블의 개수가 늘어날수록 거짓 속성값과 같은 값을 갖는 실제 속성값이 존재할 가능성이 커지기 때문이며, 반면 무겹침의 경우에는 테이블들간에 속성값의 영역이 겹치지 않아 이와 같은 가능성이 존재하지 않기 때문인 것으로 파악된다.

표 2는 완전겹침 실험에서 w에 따른 테이블당 구간 배열 레코드의 개수와 테이블당 $w >= 1$로 인한 구간 오차율을 백분율로 나타낸 것이다.

구간 배열 레코드의 개수는 곧 구간 배열을 저장할 저장공간의 필요량을 나타낸다. 제시된 모든 경우에서 $w = 16$을 사용하면 저장공간이 10배 이상 줄어들었다. 반면 그로 인한 오차율도 증가한다. 특히 0.1% 표본 조사의 경우 $w = 16$인 경우 오차율은 47%에 이른다. 그것은 작은 표본율로 표본을 추출하였기 때문에 추출된 속성값들간에 간격이 크고 그에 따라 거짓 속성값의 값이 증가하기 때문이다.

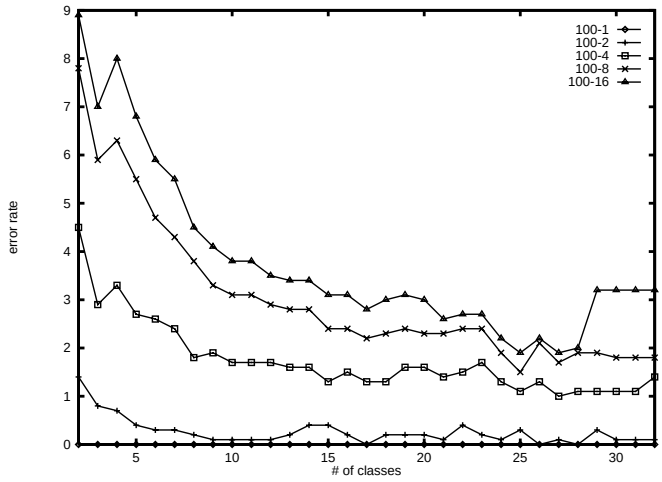
4 결론

계층질의를 지원하는 ORDBMS 또는 ODBMS에서는 효과적인 질의처리를 위해서 단일 테이블(또는 클래스)에 대한 상이값수는 물론 계층구조 전체에 대한 상이값수에 대한 정보를 필요로 한다.

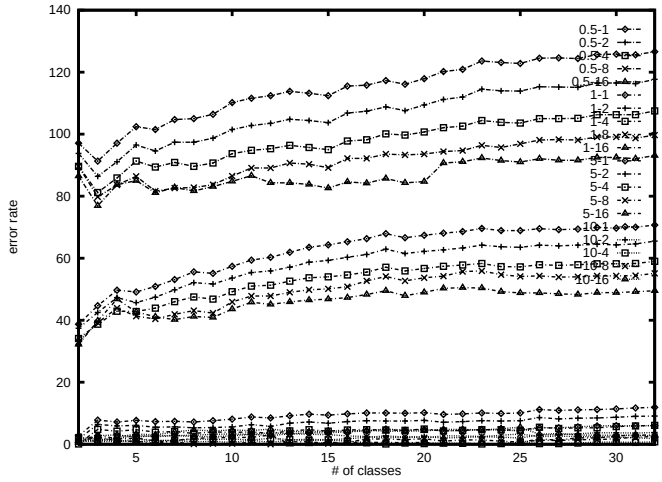
본 연구는 단일 테이블 내에서의 특정 속성에 대한 상이값수의 예측에 대한 연구를 수행하였던 기존 연구와는 달리 테이블 계층구조를 위한 계층 상이값수를 계산하기 위한 기법을 제안하였다. 제안하는 기법은 테이블 계층 구조상의 각 테이블에 대해 전수조사 혹은 표본조사를 통해 추출된 속성값을 나타내는 속성값 구간 배열을 구성하고 각 테이블과 그것의 하위 테이블의 속성값 구간 배열을 병합하여 해당 테이블의 속성에 대한 계층 상이값을 계산하는 기법이다.

제안하는 기법을 사용하면 단일 테이블에서의 상이값수를 구하기 위해서는 기존의 방식을 변형 없이 이용할 수 있으며 많은 저장공간을 사용하지 않고서도 계층 상이값수를 비교적 정확히 계산할 수 있다. 특히 계층구조 내의 일부 테이블에 대해서만 변경이 빈번히 이루어지는 환경에 효과적으로 사용될 수 있다.

추후에는 상이값 구간 배열을 저장하기 위한 별도의 공간을 사용하지 않고 히스토그램의 데이터와

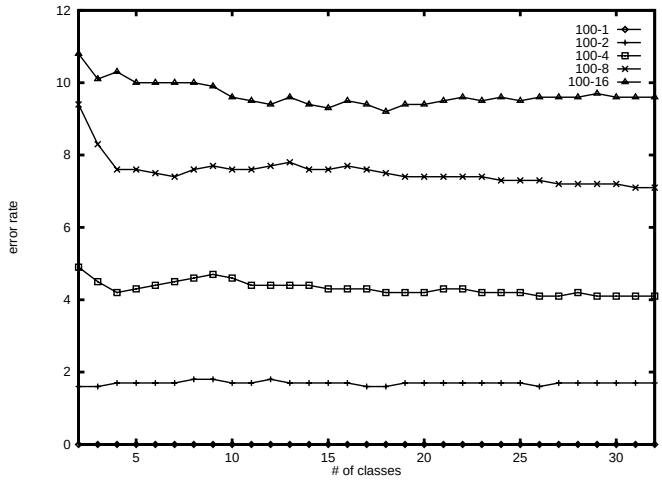


(a) 전수조사

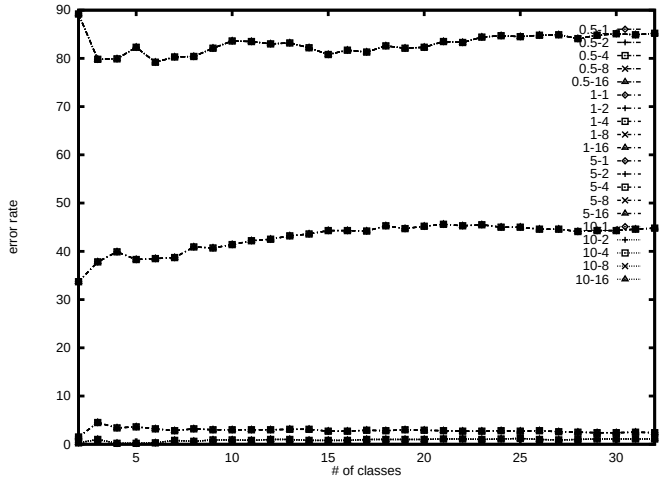


(b) 표본조사

그림 3: 완전겹침 조건에서의 오차율의 변화



(a) 전수조사



(b) 표본조사

그림 4: 무결침 조건에서의 오차율의 변화

합병하여 저장하는 방법에 대해 연구를 수행할 예정이다.

참고문헌

[1] S. Chaudhuri, R. Motwani, and V. R. Narasayya. Random sampling for histogram construction: How much is enough? In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 436–447, 1998.

[2] P. J. Haas, J. F. Naughton, S. Seshadri, and L. Stokes. Sampling-based estimation of the number of distinct values of attribute. In *Proceedings of the International Conference on Very Large Data Bases*, pages 311–322, 1995.

[3] W.-C. Hou and G. Ozsoyoglu. Statistical estimators for aggregate relational algebra queries. *ACM Trans. Database Syst.*, 16(4):600–654, 1991.

[4] W.-C. Hou, G. Ozsoyoglu, and B. K. Taneja. Statistical estimators for relational algebra expressions. In *Proceedings of ACM SIGACT-SIGMOD-SIGART symposium on principles of Database Systems*, pages 276–287, 1988.

[5] W.-C. Hou, G. Ozsoyoglu, and B. K. Taneja. Processing aggregate relational queries with hard time constraints. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 68–77, 1989.

[6] Y. Ioannidis and V. Poosala. Histogram-based solutions to diverse database estimation problems. *Data Engineering Bulletin*, 18(3):10–18, 1995.

[7] Y. E. Ioannidis. Query optimization. *The Computer Science and Engineering Handbook*, pages 1038–1057, 1997.

[8] W. Kim. *Introduction to Object-Oriented Databases*. The MIT Press, 1990.

[9] M. V. Mannino, P. Chu, and T. Sager. Statistical profile estimation in database systems. *ACM Computing Surveys*, pages 191–221, 1995.

[10] V. Poosala. *Histogram-based Estimation Techniques in Database Systems*. PhD thesis, University of Wisconsin Madison, 1997.

[11] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 82–93, 1979.

[12] M. Stonebraker, P. Brown, and D. Moore. *Object-Relational DBMSs : Tracking the Next Great Wave*. Morgan Kaufmann Publishers, 1998.

부록

계층 히스토그램의 지원 방안

테이블 계층 구조에서 테이블의 특정 속성에 대한 계층 선택율은 관계형 데이터베이스 시스템에서와 같이 해당 속성에 대한 히스토그램을 각 테이블마다 유지하는 것으로서 계산이 가능하다. 다음은 이 방법의 내용과 타당성을 보인 것이다.

상위 테이블 T_1 과 하위 테이블 $T_2, \ldots, T_N$으로 이루어진 테이블 계층구조에 대하여 다음과 같이 조건이 주어질 때

- 테이블 $i(1 \leq i \leq N)$의 레코드(인스턴스) 개수 : N_i
- 계층구조상의 모든 레코드의 개수 : $N_H = \sum_{i=1}^{N} N_i$
- 주어진 질의에 대해 테이블 i의 히스토그램을 통해 예측된 선택율:s_i^e
- 주어진 질의에 대해 테이블 i의 히스토그램을 통해 선택될 것으로 예상되는 레코드의 개수 : $S_i^e = s_i^e \times N_i$
- 주어진 질의에 대해 계층 구조 전체에서 선택될 것으로 예상되는 레코드의 개수 : $S_H = \sum_i^N S_i^e$

T_1에서의 예상 계층 선택율(s_H^e)는 다음과 같이 구할 수 있다.

$$\begin{aligned} s_H^e &= \frac{S_H}{N_H} \\ &= \frac{\sum_{i=1}^{N} S_i^e}{N_H} \\ &= \frac{\sum_{i=1}^{N} s_i^e \times N_i}{N_H} \\ &= \frac{\sum_{i=1}^{N} s_i^e \times N_i}{N_H} \end{aligned}$$

만일 위의 각 변수에 대한 실제 값이 다음과 같다면,

- 실제 선택율:s_i
- 실제로 선택될 레코드의 개수 : $S_i = s_i \times N_i$
- 계층 구조 전체에서 선택된 레코드의 개수 : $S_H = \sum_i^N S_i^e$

단일 선택율을 히스토그램을 통해 예측함으로써 발생하는 오차율(d_i)을 일반적으로 다음과 같이 정의하며

$$
\begin{aligned}
d_i &= \frac{S_i^e - S_i}{S_i} \\
&= \frac{s_i^e \times N - s_i \times N}{s_i \times N} \\
&= \frac{s_i^e - s_i}{s_i}
\end{aligned}
$$

이 값은 히스토그램의 구성 방식이나 히스토그램을 저장하기 위해 사용한 저장공간의 크기에 따라 달라진다. 동일한 방법으로 T_1에서의 계층 선택율에 대한 오차율(d_H)을 계산하면 다음과 같다.

$$
\begin{aligned}
d_H &= \frac{S_H^e - S_H}{S_H} \\
&= \frac{\sum_{i=1}^{N} S_i^e - \sum_{i=1}^{N} S_i}{S_H} \\
&= \frac{\sum_{i=1}^{N} (S_i^e - S_i)}{S_H} \\
&= \frac{\sum_{i=1}^{N} d_i S_i}{S_H} \\
&= \frac{\sum_{i=1}^{N} d_i S_i}{S_H} \\
&= \sum_{i=1}^{N} d_i \frac{S_i}{S_H}
\end{aligned}
$$

만일 단일 선택율에서 발생한 따른 오차율의 최대 값을

$d_{max}(d_{max} \geq d_1, \ldots, d_N)$라하면

$$
\begin{aligned}
d_H &\leq \sum_{i=1}^{N} d_{max} \frac{S_i}{S_H} \\
&= d_{max} \sum_{i=1}^{N} \frac{S_i}{S_H} \\
&= d_{max}
\end{aligned}
$$

가 된다. 즉, 계층 선택율은 그것을 계산하기 위해 사용한 개개의 히스토그램에서 발생하는 오차율보다는 크지 않는 오차율을 갖음을 의미한다. 따라서 각각의 테이블에 대한 히스토그램이 비교적 정확한 경우 그것들을 통해 계산된 계층 선택율 또한 충분한 정확성을 가짐을 알 수 있다.