다차원 색인을 이용한 밀도 기반 클러스터링의 근사적 접근 방법*

황재준[1] 문양세[2] 황규영[1] 장주현[2]○ 김진호[2]
[1]KAIST 전산학과/AITrc, [2]강원대학교 컴퓨터과학과 및 KAIST AITrc
[1]{jjhwang, kywhang}@mozart.kaist.ac.kr, [2]{ysmoon, jhjang○, jnkim}@kangwon.ac.kr

An Approximate Approach for Density-Based Clustering Using Multidimensional Indexes

Jae-Joon Hwang[1], Yang-Sae Moon[2], Kyu-Young Whang[1], Joo-Hyun Jang[2], Jin-Ho Kim[2]
[1] Dept. of CS & AITrc, KAIST, [2] Dept. of CS, Kangwon Nat'l Univ. and AITrc, KAIST

요 약

본 논문에서는 기존의 밀도 기반 전지 클러스터링 알고리즘의 성능을 개선한 **밀도 기반 클러스터링의 근사적 접근법**을 제안한다. 기존의 밀도 기반 전지 알고리즘은 다차원 색인의 많은 검색 공간을 빠르게 전지하면서도 원하는 클러스터를 정확히 찾아내는 특징을 가지고 있다. 그러나 기존 알고리즘은 전지를 위한 한계 값 설정을 위하여 단말 영역들의 밀도 값을 사용함으로써, 내부 영역에 속한 단말 영역들 간의 밀도 편차가 큰 경우 전지 여부에 대한 판별이 빨리 이루어지지 않는다. 또한, 최악의 경우에는 모든 단말 페이지를 검색하여야 하고, 이에 따라 성능이 저하될 수 있다. 반면에, 제안하는 근사적 접근법에서는 한계 값 설정을 위해 단말 영역이 아닌 내부 영역의 밀도 값을 사용한다. 일반적으로, 내부 영역들 간의 밀도 편차는 단말 영역들 간의 밀도 편차보다 크지 않으므로, 근사 밀도 기반 전지 알고리즘에서는 더욱 많은 검색 공간의 전지 여부의 빨리 판별할 수 있게 된다. 성능 평가 실험을 수행한 결과, 제안한 알고리즘은 기존의 알고리즘과 비교하여 정확성 측면에서는 큰 차이가 없는 반면 수행 시간 측면에서는 최대 17%의 성능 향상 효과가 있는 것으로 나타났다.

1. 서 론

본 논문에서는 다차원 색인을 이용한 클러스터링 문제[4,6]를 다룬다. 일반적으로 데이터 웨어하우스, 지리 정보 시스템 등 대부분의 데이터베이스 응용에서 대용량의 데이터베이스 질의 처리를 위하여 다차원 색인을 유지 및 관리 한다. 이러한 다차원 색인에서는 거리가 가까운 객체들은 동일한 단말 페이지 혹은 내부 페이지에 저장될 가능성이 크며, 이를 다차원 색인의 **클러스터링 성질**이라 부른다[1]. 이러한 다차원 색인의 클러스터링 성질을 사용하면 전체 데이터베이스의 액세스와 많은 거리 계산 없이 정확하면서도 빠른 클러스터링을 수행할 수 있다.

다차원 색인의 클러스터링 성질을 사용하여 클러스터를 정의하고, 이들 클러스터를 효과적으로 찾는 방법이 밀도 기반 전지 알고리즘이다[1]. 밀도 기반 전지 알고리즘에서는 데이터베이스 객체들의 밀도에 기반하여 클러스터를 정형적으로 정의하고, 이들 클러스터를 빠르게 찾는 하향식 클러스터링 방법을 사용한다. 그런데 이러한 기존의 알고리즘에서는 색인의 단말 영역들이 갖는 밀도 값을 기준으로 한계 값을 설정한다. 따라서 색인의 내부 영역에 속한 단말 영역들 간의 밀도 편차가 큰 경우에는 전지 여부에 대한 판별이 빨리 이루어지지 않는다. 또한 최악의 경우, 모든 단말 페이지를 검색하여야 하고, 이에 따라 성능이 저하되는 단점이 있다.

본 논문에서는 밀도 기반 전지에 사용되는 한계 값의 기준 영역을 색인의 단말 영역이 아닌 내부 영역까지 포함시켜 재정의함으로써, 알고리즘의 성능을 더욱 향상시키는 근사 밀도 기반 전지 알고리즘을 제안한다. 다차원 파일 구조에서 임의의 내부 영역을 가정할 경우, 자신에 속한 하위 계층의 내부 영역들 간의 밀도 편차는 자신에 속한 단말 영역들 간의 밀도 편차보다 크지 않으므로, 자신에 속한 하위 계층의 내부 영역 밀도 값을 한계 값으로 사용하게 되면, 밀도 편차가 상대적으로 작아지게 되어 전지 여부의 판별이 보다 빨리 이루어질 가능성이 높게 된다. 본 논문에서는 근사 밀도 기반 전지의 개념을 제시하고, 이에 기반한 근사 밀도 기반 전지 알고리즘을 제안한다. 그리고 실험을 통하여 제안한 알고리즘이 기존의 알고리즘과 비교하여 성능을 향상을 시킴을 보인다.

2. 관련 연구

밀도 기반 클러스터의 정의 및 관련 알고리즘의 이해를 위하여, 먼저 다차원 파일과 관련된 주요 용어를 설명한다. 일반적으로 다차원 파일 구조는 **색인 페이지**와 **데이터 페이지**의 두 부분으로 크게 나눌 수 있다. 색인 페이지는 다시 루트, 내부, 단말 페이지로 구분된다. 우선, **루트 페이지**는 가장 상위 계층에 존재하는 색인 페이지로서, <키, 자식 포인터>의 구조를 갖는 **루트 엔트리**들로 구성된다. 다음으로, **단말 페이지**는 가장 하위 계층을 구성하는 색인 페이지로서, <키, 객체 식별자>의 구조를 갖는 **단말 엔트리**들로 구성된다. **내부 페이지**는 루트 페이지와 단말 페이지 사이의 중간 계층에 존재하는 색인 페이지로서, <키, 자식 포인터>의 구조를 갖는 **내부 엔트리**들로 구성된다.

색인 페이지의 특정 엔트리가 표현하는 영역을 **색인 영역**이라 한다. 그리고 단말 엔트리가 나타내는 영역을 **단말 영역**, 내부 엔트리가 나타내는 영역을 **내부 영역**이라 정의한다. 그리고 대부분의 다차원 색인은 영역의 형태가 초사각형이므로[3, 5], 본 논문에서도 모든 영역의 형태가 초사각형이라 가정하고, 클러스터링 대상이 되는 다차원 색인은 검색 공간을 분할하는 방법으로 **영역 기준 분할전략**[2]을 사용한다고 가정한다.

참고문헌 [1]에서는 밀도 기반 클러스터를 밀도가 높은 인접한 단말 영역들의 집합으로 정의하였다. 클러스터의 정의를 위하여, 우선 단말 영역들을 주어진 **클러스터링 인수**를 사용하여 **밀집 단말 영역**과 **희소 단말 영역**으로 구분하였다. 여기에서 클러스터링 인수란 밀집 단말 영역에 포함되어야 하는 최소 객체 수와 데이터베이스에 저장된 총 객체 수의 비율로 정의한다. 그리고 밀집 단말 영역과 희소 단말 영역을 구분하는 경계 영역, 즉 밀집 단말 영역들 중에서 최저 밀도를 가지는 영역을 **분할 경계 영역**이라 한다. 이와 같이 분할 경계 영역을 기준으로 판별된 밀집 단말 영역을 대상으로 인접 여부를 판단하여, 서로 인접한 밀집 단말 영역들을 묶어서 밀도 기반 클러스터로 구성하는 것이다.

정의한 밀도 기반 클러스터를 찾기 위하여, 참고문헌 [1]에서는 먼저 직관적 클러스터링 방법인 단말 대조 알고리즘을 제안하였다. 단말 대조 알고리즘은 다차원 색인의 모든 단말 페이지를 액세스하고, 주어진 클러스터링 인수를 사용하여 단말 영역들을 밀집 단말 영역과 희소 단말 영역들로 구분한다. 그런 다음, 구분된 밀집 단말 영역 대상으로 인접한 영역을 합병하여 클러스터를 형성하는 방법을 사용한다.

다음으로, 다차원 색인의 계층 구조를 사용하여 색인의 일부분만을 검색하면서 클러스터를 찾아내는 밀도 기반 전지 알고리즘을 제안하였다[1]. 밀도 기반 전지 알고리즘에서는 색인의 모든 단말 엔트리를 액세스하지 않고서도 정확한 클러스터링을 수행하는 밀도 기반 전지 개념을 사용한다. 여기에서 **밀도 기반 전지**란 색인 내부 영역에서 자신의 단말 영역들이 모두 밀집 단말 영역이거나 모두 희소 단말 영역임을 판단하여 하위 영역들에 대한 검색을 생략, 즉 전지하는 방법을 의미한다. 이때, 어떠한 내부 영역에 속한 단말 영역들이 모두 밀집 단말 영역들인 경우 이를 **밀집 내부 영역**, 반대로 모두 희소 단말 영역들인 경우 이를 **희소 내부 영역**이라 정의하였다. 그리고 이들 밀집 내부 영역 및 희소 내부 영역을 판별하고 이를 전지하는 방법을 제안하였다.

밀집 내부 영역과 희소 내부 영역을 판별하기 위하여, 밀도 기반 전지 알고리즘에서는 다차원 색인의 내부 엔트리에 최고 밀도(*d_highest*) 및 최

* 본 연구는 첨단정보기술연구센터를 통해 한국과학재단의 지원을 받았음

저 밀도(*d_lowest*)를 관리한다[2]. 여기에서, 최고 밀도와 최저 밀도는 각각 내부 영역에 포함된 단말 영역 중에서 최고 밀도를 갖는 영역과 최저 밀도를 갖는 영역의 밀도를 나타낸다. 밀도 기반 전지 알고리즘은 하향식 넓이 우선 탐색 방식으로 색인을 검색하면서, 각 내부 엔트리에서 관리하는 *d_highest* 및 *d_lowest* 값을 사용하여 하위 계층의 전지 여부를 결정한다.

3. 밀도 기반 클러스터링의 근사적 접근 방법

본 장에서는 단말 영역의 밀도가 아닌 내부 영역의 밀도를 이용하여 밀도 기반 전지를 수행하는 근사 밀도 기반 전지 알고리즘을 제안한다.

제안하는 근사 밀도 기반 전지 알고리즘에서는 "내부 페이지에 포함된 내부 영역들의 밀도 차이는 해당 내부 페이지에 포함된 단말 영역들의 밀도 차이보다는 작다"는 관찰에 기반하여 클러스터링을 수행한다. 다차원 파일 구조에서 임의의 내부 영역의 밀도는 자신에 속한 단말 영역들의 최고 밀도 값인 *d_highest*와 최저 밀도 값인 *d_lowest* 사이의 값을 갖는다. 이는 자신에 속한 임의 계층의 내부 영역들 간의 밀도 편차가 자신에 속한 단말 영역들 간의 밀도 편차보다 크지 않음을 의미한다. 따라서 자신에 속한 임의 계층의 내부 영역들 중 최고 밀도 값을 *d_highest*로 사용하고 최저 밀도 값을 *d_lowest*로 사용할 경우, 단말 영역을 사용하는 경우에 비하여 *d_highest*와 *d_lowest*의 차이가 상대적으로 작아지게 되어 밀집 혹은 희소 내부 영역의 판별이 보다 빨리 이루어질 수 있고, 결국 보다 많은 영역을 빠르게 전지할 수 있게 된다.

3.1 근사 밀도 기반 전지 개념

밀도 기반 전지의 개념과 알고리즘의 설명을 위하여 다음 표 1과 같은 표기법을 사용한다.

표 1. 주요 표기법

기호	정의/의미
n(R), d(R)	영역 R에 포함된 객체 수, R의 밀도
ρ	사용자에 의해 주어지는 클러스터링 인수 (0<ρ<1)
NRO	제거 가능한 객체 수 (number of removable objects)
R_i	영역 R의 i번째 엔트리가 나타내는 영역

근사 밀도 기반 전지는 전지 여부를 빠르게 결정하기 위하여, 각 내부 영역의 *d_highest*와 *d_lowest* 값으로 바로 아래 계층 영역들의 최고 밀도 값과 최저 밀도 값을 사용한다. 기존의 밀도 기반 전지 알고리즘에서 사용한 이들 값은 다음과 같이 재귀 함수 형태로 표현할 수 있다.

$$d\_highest(R) = \begin{cases} \max(\{d\_highest(R_i) \mid R_i \text{ is an internal region in } R\}) & \text{if } R_i \text{ is a region of an internal page.} \\ \max(\{d(R_i) \mid R_i \text{ is a leaf region in } R\}) & \text{if } R_i \text{ is a region of a leaf page.} \end{cases}$$

$$d\_lowest(R) = \begin{cases} \min(\{d\_lowest(R_i) \mid R_i \text{ is an internal region in } R\}) & \text{if } R_i \text{ is a region of an internal page.} \\ \min(\{d(R_i) \mid R_i \text{ is a leaf region in } R\}) & \text{if } R_i \text{ is a region of a leaf page.} \end{cases}$$

반면에, 근사 밀도 기반 전지에서는 *d_highest*와 *d_lowest*를 바로 아래 계층의 영역들을 사용하여 다음과 같이 재 정의하여 표현한다.

$$d\_highest(R) = \max(\{d(R_i) \mid R_i \text{ is a region in } R\})$$

$$d\_lowest(R) = \min(\{d(R_i) \mid R_i \text{ is a region in } R\})$$

이와 같이 단말 영역들의 밀도 값 대신 바로 아래 계층 내부 영역들의 밀도 값을 사용하면, *d_highest*와 *d_lowest*의 차이가 줄어들어 보다 많은 내부 영역들을 빠르게 전지할 수 있다. 또한, 색인의 클러스터링 성질에 의해 내부 영역들의 밀도 값은 단말 영역들의 밀도 값을 반영하고 있으므로, 단말 영역의 밀도 값 대신 내부 영역의 밀도 값을 *d_highest*와 *d_lowest*로 사용하더라도 밀도 기반 전지 알고리즘과 유사한 형태의 클러스터를 찾을 수 있다.

예제 1: 그림 1은 기존의 전지 알고리즘과 제안하는 근사 전지 알고리즘에 있어서 *d_highest*와 *d_lowest* 값을 나타낸다. 먼저, 그림 1(a)의 밀도 기반 전지 알고리즘을 보면, 내부 영역 R_0의 *d_highest*와 *d_lowest* 값은 R_0에 속한 단말 영역들에 대한 밀도의 최대 값과 최소 값으로 구해진다. 반면에, 그림 1(b)의 근사 밀도 기반 전지 알고리즘을 보면, 내부 영역 R_0의 경우 바로 아래 계층인 R_1, R_2, R_3, R_4의 밀도 값 중에서 최대 값 및 최소 값을 *d_highest*와 *d_lowest*로 삼는다. 이 예에서 보듯이, 밀도 기반 전지 알고리즘의 *d_highest*와 *d_lowest*의 차이(= 10)에 비하여 근사 밀도 기반 전지 알고리즘의 *d_highest*와 *d_lowest*의 차이(= 4)가 크게 줄었음을 알 수 있다. 이와 같이, 두 값 사이의 차이가 줄어들어 결국 더욱 많은 내부 영역의 전지가 빠르게 이루어질 수 있게 된다. □

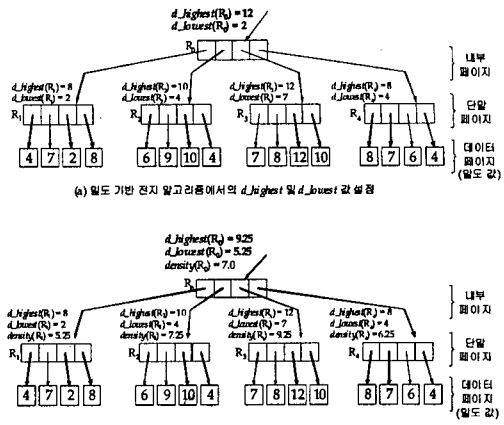

그림 1. 재정의에 따른 *d_highest*와 *d_lowest* 값의 변화.

3.2 밀도 기반 클러스터링의 근사적 알고리즘

제안하는 근사 밀도 기반 전지 알고리즘의 수행 과정은 그림 2와 같다. 그림 2의 알고리즘은 *d_highest*와 *d_lowest*의 재정의를 제외하면 밀도 기반 전지 알고리즘[1]과 동일하므로 자세한 설명은 생략한다.

결과적으로, 기존의 밀도 기반 전지 알고리즘 *d_highest*와 *d_lowest*를 단말들의 밀도를 사용하여 계산 및 관리한 반면에, 근사 밀도 기반 전지 알고리즘에서는 이들 값을 바로 아래 계층 영역들의 밀도 값을 기반으로 계산 및 관리하는 부분이 상이하다. 밀도 기반 전지 알고리즘에서, 그림 2의 알고리즘을 사용할 경우, 전지가 정확하게 이루어지며, 이에 따라 정의한 클러스터와 동일한 클러스터를 정확하게 찾음이 증명되었다[1]. 반면에, 근사 밀도 알고리즘은 빠른 전지를 위하여, *d_highest* 및 *d_lowest* 값 계산에 있어서 단말 영역의 밀도 값이 아닌 내부 영역의 밀도 값을 사용하므로, 전지되는 영역이 밀집 내부 영역이나 희소 내부 영역일 가능성이 높은 영역이다. 따라서 근사 밀도 기반 전지 알고리즘은 궁극적으로 밀도 기반 클러스터와 유사한 클러스터를 찾게 된다.

```
Algorithm Approximate_Density_Pruning_Clustering
Input:  md_index : multidimensional index to be used for clustering
        N        : the number of objects stored in the database
        ρ        : clustering factor
Output: Clusters
begin
1:   R_dense := { };
2:   NRO := (1-ρ) x N;
3:   curr_level := root level of md_index;
4:   while (NRO > 0 and curr_level is an internal level) begin
5:      Construct a list L of internal regions at the current level;
6:      Make a sorted list L_high (={R_1^high, R_2^high, ..., R_m^high}) in order of d_highest;
7:      p := Find_Partition_Boundary_Region(L_high, NRO);
8:      B_high := d_highest(R_p^high);
9:      Make a sorted list L_low (={R_1^low, R_2^low, ..., R_m^low}) in order of d_lowest;
10:     p := Find_Partition_Boundary_Region(L_low, NRO);
11:     B_low := d_lowest(R_p^low);
12:     for each R_i^low whose d_lowest(R_i^low) > B_high begin
13:        R_dense := R_dense ∪ R_i^low;
14:        Prune the subtree whose root represents the internal page of R_i^low;
15:     end
16:     for each R_i^high whose d_highest(R_i^high) < B_low begin
17:        NRO := NRO - n(R_i^high);
18:        Prune the subtree whose root represents the internal page of R_i^high;
19:     end
20:     curr_level := curr_level + 1;
21:  end
22:  if (NRO > 0) begin
23:     Construct a list L of leaf regions by reading all remaining leaf pages;
24:     Make a sorted list L_sort (={R_1, R_2, ..., R_m}) in order of region density;
25:     p := Find_Partition_Region(L_sort, NRO);
26:     R_dense := R_dense ∪ {R_p, R_p+1, ..., R_m};
27:  end
28:  C := Find_Clusters(R_dense);
29:  Return C;
end
```

그림 2. 근사 밀도 기반 전지 알고리즘.

4. 성능평가

4.1 실험 환경

본 논문에서는 그림 3과 같은 두 가지 종류의 이차원 데이터 집합을 생성하여 실험에 사용하였다. DS1 및 DS2의 각 데이터 집합에 대한 자세한 생성 방법은 참고문헌[1]과 동일하므로 설명을 생략한다.

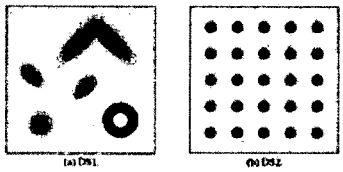

그림 3. 실험 데이터 집합.

실험은 512M 바이트 메모리를 가진 SUN Ultra 60 워크스테이션에서 수행하였다. 데이터를 저장하는 다차원 파일 구조로는 MLGF[2, 5]를 사용하고, 데이터 및 색인 페이지의 크기는 1024 바이트로 하였다.

4.2 성능 시험 결과

본 절에서는 알고리즘 *Approximate_Density_Pruning_Clustering*(이하 *ADP* 라 한다)에 의한 성능 평가 결과를 기존의 밀도 기반 전지 알고리즘인 *Density_Pruning_Clustering*[1](이하 *DP* 라 한다)의 결과와 비교하여 설명한다. 실험을 위하여 생성된 DS1과DS2는 각각 10만, 100만, 1000만 개의 객체를 포함하고 있으며, 노이즈 비율은 전체 객체수의 8%이다. 노이즈 비율이 8%이므로 알고리즘에서의 클러스터링 인수 ρ는 0.92를 사용하였다.

그림 4는 서로 다른 크기의 DS1에 대한 ADP와 DP의 수행시간을 나타낸다. 그림 4를 보면, ADP는 DP보다도 수행시간이 단축됨을 알 수 있는데, 이는 ADP에서 더 많은 전지가 일어나기 때문이다. 즉, ADP에서 색인 공간에 대한 많은 전지가 일어나 클러스터 대상이 되는 밀집 영역의 개수를 더욱 줄임으로써, 알고리즘 수행에 따른 CPU 처리 시간을 줄이기 때문이다. 데이터 집합의 크기에 따라 ADP는 DP에 비하여 클러스터링 수행시간을 최대 17.0%까지 줄일 수 있는 것으로 나타났다.

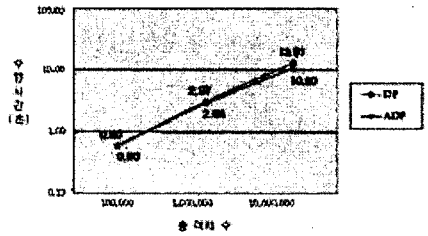

그림 4. 서로 다른 크기의 DS1에 대한 알고리즘별 수행시간.

다음으로 서로 다른 크기의 DS2에 대한 알고리즘 DP와 ADP의 수행시간도 DS1과 유사한 결과를 나타냈다. DS2의 경우, 데이터 집합의 크기에 따라 ADP는 DP에 비하여 클러스터링 수행시간을 최대 10.2%까지 줄인 것으로 나타났다.

4.3 정확성 시험 결과

그림 5는 DS1에 대해 ADP가 찾은 클러스터를 DP가 찾은 클러스터와 비교한 것이다. 그림 5(a)는 DP가 찾은 클러스터이고, 그림 5(b)는 ADP가 찾은 클러스터로서 두 경우 모두 찾은 클러스터의 모양이 매우 유사하게 나타났음을 알 수 있다. 이는 DS1의 경우 제안한 ADP에 의해서도 매우 정확하게 전지가 이루어짐을 의미한다. 그림 5(a)와 (b)를 서로 비교해 보면, 그림 5(b)에서는 클러스터 내부의 많은 사각형들이 더 크다는 것을 알 수 있다. 이는 ADP가 내부 영역의 밀도를 이용하여 영역 대조 분할을 수행하기 때문에, DP에 비하여 색인의 보다 상위 계층에서도 전지가 이루어지는 데 따른 것이다. DS2의 경우도 DS1과 같이 유사한 클러스터를 찾았으며, 자세한 결과는 생략한다.

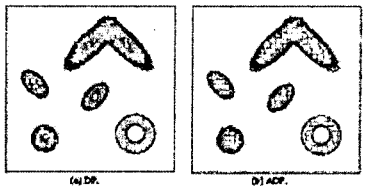

그림 5. DS1에 대한 DP와 ADP의 클러스터링 결과 비교.

실험 결과를 요약하면, 근사 밀도 전지 알고리즘은 기존의 밀도 전지 알고리즘에 비해서 정확성을 크게 해치지 않으면서도 성능을 최대 17.0%까지 줄인 것으로 나타났다. 이는 보다 빠른 전지를 통해서 색인 검색 공간을 줄였기 때문이다.

5. 결론

본 논문에서는 대부분의 대용량 데이터베이스 응용에서 이미 유지하고 있는 다차원 파일 구조를 이용하는 밀도 기반 클러스터링의 새로운 접근법을 제안하였다. 밀도 기반 클러스터란 다차원 색인의 내부 혹은 단말 영역의 밀도를 사용하여 밀도가 높은 영역들을 클러스터로 정의한 것이다. 기존 연구인 밀도 기반 전지 알고리즘[1]에서는 이러한 밀도 기반 클러스터를 정확하게 찾아내는데 연구의 목적이 있었다. 반면에, 본 논문에서는 정확성을 크게 해치지 않으면서도 정의한 밀도 기반 클러스터를 빠르게 찾아내는 알고리즘을 제안하였다.

본 논문에서는 기존의 밀도 기반 전지 알고리즘의 단점인 단말 영역의 밀도 값을 기준으로 전지 여부를 판별하는 방법을 보완하여, 성능을 보다 더 향상시킬 수 있는 근사 밀도 기반 전지의 개념을 제시하였다. 기존의 밀도 기반 전지에서는 내부 영역에 속한 단말 영역들 간의 밀도 편차가 큰 경우에는, 밀집 혹은 희소 내부 영역의 판별이 빨리 이루어지지 않게 된다. 또한, 최악의 경우에는 모든 단말 페이지를 검색하여야 하고, 이에 따라 성능이 저하될 수 있다. 제안한 근사 밀도 기반 전지에서는 단말 영역의 밀도 대신에 내부 영역의 밀도를 이용하여 밀도 기반 전지를 수행함으로써, 유사한 정확도를 유지하면서 보다 높은 성능을 제공할 수 있다. 근사 밀도 기반 전지 알고리즘을 이용하여 성능 실험을 수행한 결과, 기존의 밀도 기반 전지 알고리즘과 비교하여 정확성 측면에서는 큰 차이가 없는 반면 수행 시간 측면에서는 최대 17%의 성능 향상 효과가 있는 것으로 나타났다.

6. 참고문헌

[1] J.-J. Hwang, K.-Y. Whang, Y.-S. Moon, and B.-S. Lee, "A Top-down Approach for Density-Based Clustering Using Multidimensional Indexes," *Journal of Systems and Software*, Vol. 73, Issue 1, pp. 169-180, Sept. 2004.

[2] J.-H. Lee, Y.-K. Lee, K.-Y. Whang, and I.-Y. Song, "A Region Splitting Strategy for Physical Database Design of Multidimensional File Organizations," In *Proc. the 23rd Int'l Conf. on Very Large Data Bases*, Athens, Greece, pp. 416-425, Aug. 1997.

[3] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "The Grid File: An Adaptable, Symmetric Multikey File Structure," *ACM Trans. on Database Systems*, Vol. 9, No. 1, pp. 38-71, March 1984.

[4] C. Ordonez and E. Omiecinski, "Efficient Disk-Based K-Means Clustering for Relational Databases," *IEEE Trans. on Knowledge and Engineering*, Vol. 16, No. 8, pp. 909-921, Aug. 2004.

[5] K.-Y. Whang and R. Krishnamurthy, Multilevel Grid Files, IBM Research Report RC11516, 1985.

[6] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, Montreal, Quebec, Canada, pp. 103-114, June 1996.