다차원 공간의 효율적인 그리드 분할을 통한 디클러스터링 알고리즘 성능향상 기법

(Performance Improvement of Declustering Algorithm by Efficient Grid-Partitioning Multi-Dimensional Space)

김 학 철*

(Hak Cheol Kim)

요 약 본 논문에서는 그리드 분할과 매핑함수에 기반하여 영역질의 성능향상을 위해서 기존에 제시된 디클러스터링 방법들을 다차원 공간에 대해서 적용할 때의 문제점을 분석하고 해결법을 제시한다. 다차원 공간에 대해서 기존에 제시된 방법들을 적용할 때의 문제점은 각 차원의 분할 횟수가 적고(대부분 이진 분할이 발생함) 극히 작은 선택률에 대해서도 영역질의 각 차원의 길이가 커지기 때문에 발생한다. 본 논문에서는 이를 해결하기 위하여 다차원 공간의 다양한 그리드 분할방법에 대해서 수학적으로 성능을 예측하는 모델을 제시한다. 제시한 수학 모델을 이용하여 가능한 다양한 그리드 분할 방법들 가운데 영역질의와 겹치는 그리드 셀의 수를 감소시키는 분할 방법을 선택할 수 있으며, 이는 디클러스터링 알고리즘의 전체 성능향상으로 귀결된다. 다양한 실험결과, 본 논문에서 제시한 분할 방법을 적용할 때, 기존에 제시된 디클러스터링 알고리즘의 성능을 최대 2.7배까지 향상시킬 수 있음을 알 수 있었다.

키워드 : 디클러스터링, 다차원 데이터, 영역질의 성능 모델

Abstract In this paper, we analyze the shortcomings of the previous declustering methods, which are based on grid-like partitioning and a mapping function from a cell to a disk number, for high-dimensional space and propose a solution. The problems arise from the fact that the number of splitting is small(for the most part, binary-partitioning is sufficient), and the side length of a range query whose selectivity is small is quite large. To solve this problem, we propose a mathematical model to estimate the performance of a grid-like partitioning method. With the proposed estimation model, we can choose a good grid-like partitioning method among the possible schemes and this results in overall improvement in declustering performance. Several experimental results show that we can improve the performance of a previous declustering method up to 2.7 times.

Key words : declustering, multi-dimensioanl data, analytical model for range query

1. 서론

전통적인 데이터베이스 관리시스템은 주로 텍스트 위주의 저차원 데이터를 다루는 반면 CPU 속도 및 디스크 저장용량의 비약적인 발전으로 인해 지리정보시스템, CAD, 멀티미디어데이터베이스, 생물학 데이터베이스, 기상 데이터베이스 등 근래의 데이터베이스 관리 시스템에서는 대용량의 고차원 데이터를 효과적으로 저장하고 처리하기 위한 새로운 기술들이 요구되고 있다. 이를 위해서, 저차원 데이터에 대해서 효율적인 색인 방법을 다차원 데이터에 적용할 경우 성능이 저하되는 문제점을 해결하기 위하여 다차원 데이터에 대한 다양한 색인방법들이 제시되고 평가되었다[1,2,3,4,5,6,7,8].

다차원 데이터에 대한 효율적인 색인방법과 함께 디스크 입출력의 성능향상을 위해서 데이터 블록들을 여러 개의 병렬 디스크에 저장한 후 질의 처리 시 여러 개의 디스크로부터 병렬적으로 적재함으로써 질의 처리 시간을 감소시키기 위한 연구들이 진행되었으며 이를 우리는 "디클러스터링(*declustering*)"이라고 부른다.

디클러스터링 관련 기존 연구의 대부분은 균등 분포를 이루는 다차원 공간이 정형의 그리드 형태로 분할되어 있다고 가정하고 각 그리드 셀에 대해서 매핑함수를 이용하여 효과적으로 디스크 번호를 할당함으로써 영역질의의 성능향상을 도모하고자 하였다. 다양한 매핑함수들이 제시되었으며, Disk Modulo(DM)[9,10], Fieldwise Xor (FX)[11], Error Correcting Code(ECC)[12], Hilbert Curve Allocation Method(HCAM)[13], Cyclic Allocation[14,15], GDM[16], Coloring scheme[17], Golden Ratio Sequence(GRS)[18], Discrepancy based allocation[19], Kronekcer Sequence[20] 등이 이에 해당된

† 본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신사업과제의 연구비지원(07국토정보C05)에 의해 수행되었읍니다.

* 한국전자통신연구원 융합기술연구부문 선임연구원, hakcheol@etri.re.kr(교신저자)

논문접수 : 2009.12.09 수정일 : 2010.02.10 심사완료 : 2010.02.17

다. 또한, 그리드 분할에 대해서 디스크 매핑 함수의 성능들에 대해서 일부 제시되었다[21,22,23,24]. 다양한 매핑 함수 가운데, 다차원 공간에 대해서 Kronecker Sequence에 의한 디스크 할당 알고리즘[20]이 가장 좋은 성능을 보이는 것으로 알려져 있다. 그림 1은 디스크 수가 4일 때, 2차원 8×8 그리드 분할에 대해서 일부 매핑 함수에 의해서 할당된 디스크 번호의 예를 보여 준다.

3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1
1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3
3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1
1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3

(a) DM

3	2	1	0	3	2	1	0
2	3	0	1	2	3	0	1
1	0	3	2	1	0	3	2
0	1	2	3	0	1	2	3
3	2	1	0	3	2	1	0
2	3	0	1	2	3	0	1
1	0	3	2	1	0	3	2
0	1	2	3	0	1	2	3

(b) FX

3	2	1	0	3	0	3	2
0	1	2	3	2	1	0	1
3	0	3	0	1	2	3	2
2	1	2	1	0	3	0	1
1	2	1	2	3	0	3	2
0	3	0	3	2	1	0	1
3	2	1	0	1	2	3	2
0	1	2	3	0	3	0	1

(c) HCAM

1	2	3	0	1	2	3	0
3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1
0	1	2	3	0	1	2	3
1	2	3	0	1	2	3	0
3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1
0	1	2	3	0	1	2	3

(d)GRS

그림 1. 매핑함수에 의한 디스크 번호 할당 예제 (디스크 수 : 4)

이와 같이 각 그리드 셀에 디스크 번호가 할당되었을 때 질의영역과 겹치는 그리드 셀들 가운데 가장 많은 디스크 번호가 할당된 디스크의 접근횟수에 의해서 디클러스터링 알고리즘의 성능이 결정된다. 따라서, 가능한 모든 질의에 대해서 각 그리드 셀에 할당되는 디스크 번호가 균등하게 분포될 때 디클러스터링 알고리즘은 절대 최적의 성능을 보장한다. 하지만, 제한적인 조건을 만족하는 극히 일부의 경우를 제외하고 모든 질의에 대해서 절대 최적의 디스크 접근의 병렬성을 보장하는 알고리즘은 존재하지 않는 것으로 알려져 있으며 추가적인 디스크 접근이 필요하다. M개의 디스크에 대해서 2차원 $M\times M$ 그리드 분할에 대해서도 절대 최적의 디스크 접근횟수에 대한 추가 디스크 접근횟수의 하한값이 $\Omega(M)$으로 알려져 있다.

단일 복사본에 대해서 디스크 매핑 함수를 이용한 디스크 접근의 병렬성을 향상시키는데 한계가 있으므로, 최근에는 그리드 분할 결과에 대해서 다수의 복사본을 생성하여 동일한 그리드 셀을 여러 개의 디스크에 분산시켜 저장함으로써 디스크 접근의 병렬성을 높이기 위한 연구들이 제시되고 있다[25,26,27,28,29,30].

비록, 제시된 알고리즘들이 디스크 접근의 병렬성 향상에는 도움이 되지만, 다수의 복사본을 저장함으로 디스크 공간이 추가로 필요할 뿐만 아니라, 차원이 증가함에 따라 디스크 매핑 알고리즘의 개선을 통한 디클러스터링 알고리즘의 성능향상에는 한계가 있으며 데이터 공간을 분할하는 방법에 의해서 많은 영향을 받게 된다. 2장에서 이를 자세히 설명하도록 한다.

이와 별도로, 그리드 분할과 다른 형태로 데이터 블록을 구성한 결과에 대해서 디스크 번호를 할당하기 위한 디클러스터링 알고리즘 일부가 제시되었으며 대부분 그래프 이론에 기초하여 디스크 번호를 할당하며 계산 비용이 많이 드는 문제점이 있다[31,32,33,34,35].

본 논문은 균등분포 데이터에 대해서 그리드 분할 및 매핑함수에 기초하여 영역질의에 대해서 기존에 제시되었던 디클러스터링 알고리즘들을 다차원 공간에 적용할 때의 문제점을 제시하고 이를 해결하기 위해서 다차원 공간을 효율적으로 분할하기 위한 방법을 제시한다. 먼저, 다차원 공간의 분할에 대해서 영역질의의 성능을 예측하는 수학적 모델을 제시하고 이를 활용하여 효율적으로 분할함으로써 기존에 제시되었던 단일 복사본을 이용하는 디클러스터링 알고리즘의 성능을 최대 2.7배까지 향상시킬 수 있음을 보인다.

본 논문은 다음과 같이 구성된다. 2장에서는 디클러스터링 관련 개념정의와 함께 기존 연구의 문제점을 살펴보고 3장에서는 다차원 공간을 효율적으로 분할하기 위해서 필요한 수학적 분석 모델을 제시하고 4장에서는 이를 활용한 분할 방법을 제시한다. 5장에서는 본 연구에서 제시한 분할 방법을 적용하였을 때 기존에 제시되었던 디클러스터링 알고리즘의 성능향상을 다양한 실험을 통하여 보이고 마지막으로 6장에서는 본 논문의 결론 및 향후 연구 과제를 제시한다.

2. 개념정의 및 연구동기

2.1. 개념 정의

본 절에서는 디클러스터링에 대한 기본 개념들을 정의한다. 표 1은 본 논문에서 사용할 기호 및 의미를 정리한 것이다. N개의 데이터와 M개의 디스크에 대해서 디클러스터링 알고리즘은 다음과 같이 2단계로 구성된다.

- 데이터 분할 단계 : $\{v|v\in N\}\rightarrow\left\{G_1^\pi, G_2^\pi, \cdots, G_{B_N}^\pi\right\}$
  여기서, $|G_i^\pi|\le Bf_{\max}$, $\cup_{i=1}^{B_N} G_i^\pi = N$,
  그리고 $i\neq j$일 때 $G_i^\pi\cap G_j^\pi=\varnothing$
- 디스크 할당 단계 :
  $\left\{G_1^\pi, G_2^\pi, \cdots, G_{B_N}^\pi\right\}\rightarrow\{0,1,2,\cdots,M-1\}$

분할 결과 G_i^π는 1개의 물리적 디스크 페이지에 저장하고 각 디스크로부터 1개의 디스크페이지(데이터블록)을 적재하는데 소요되는 비용이 동일하다고 가정할 때 질의 Q를 처리하기 위한 응답 시간은 M개의 디스크 가운데 가장 많은 데이터 블록을 접근하는 디스크에 저장되어 있는 데이트 블록 수에 의해서 결정된다. 즉, 디클러스터링 알고리즘의 질의 Q를 처리하기 위한 응답시간은 다음과 같이 결정된다.

표 1. 기호 및 의미

기호	의미
d	차원
$\|d_i\|$	i번째 차원 정보의 크기(바이트 수)
N	데이터 수
B_N	N개의 데이터를 저장하기 위해서 필요한 블록 수
$\|B\|$	디스크 블록 크기 (바이트 수)
$Bf_{\max}$	1개의 디스크 블록에 저장 가능한 최대 데이터 수
λ_i	i번째 차원의 분할 횟수
s	선택률
M	디스크 수
Q	영역질의
B_Q	영역질의 Q와 겹치는 블록 수
q_i	영역질의의 i번째 차원의 길이
$DA_i(Q)$	영역질의 Q를 처리하기 위해서 i번째 디스크 접근 횟수
$RT(Q)$	영역질의 Q를 처리하기 위한 응답시간

정의 1. 디클러스터링 알고리즘의 응답시간

질의 Q를 처리하기 위한 디스크 접근 시간 $RT(Q)$는 다음과 같다.

$RT(Q) = \max_{i=o}^{M-1} DA_i(Q)$

여기서 $DA_i(Q)$는 질의 Q를 처리하기 위한 i번째 디스크 접근 횟수

이를 바탕으로, 절대 최적의 디클러스터링 알고리즘을 다음과 같이 정의할 수 있다.

정의 2. 절대 최적의 디클러스터링 알고리즘

다음 조건을 만족하는 디클러스터링 알고리즘은 절대 최적이다.

$$\forall_Q,\ RT(Q) = \left\lceil \frac{B_Q}{M} \right\rceil$$

정의 2를 만족하는 절대 최적의 디클러스터링 알고리즘은 일반적으로 존재하지 않으며, 모든 디클러스터링 알고리즘의 실제 응답 시간은 다음과 같이 결정된다.

$$RT(Q) = \left\lceil \frac{B_Q}{M} \right\rceil + \alpha (> 0)$$

즉, 디클러스터링 알고리즘의 성능에 영향을 주는 요소는 분할 방법에 의해서 결정되는 B_Q와 디스크 할당 방법에 의해서 결정되는 절대 최적의 디클러스터링 알고리즘의 응답시간 대비 추가적인 디스크 접근 횟수 α이다.

2.2. 연구동기

지금까지 제시된 그리드 분할과 매핑 함수를 이용하는 디클러스터링 알고리즘들은 디스크 할당 알고리즘의 성능 향상을 통해서 추가적인 디스크 접근 횟수를 감소시키는 데 집중되어 있었다. 하지만, 다차원 공간에 대해서는 그리드 분할 방법이 매핑함수를 이용하는 디클러스터링 알고리즘의 전체 성능에 큰 영향을 미치게 되며 본 절에서는 이를 설명하도록 한다.

본 논문에서는 기본적으로 $[0,1)^d$차원의 균일분포 데이터를 가정한다. $[0,1)^d$ 공간에 균일하게 분포하는 N개의 데이터에 대해서 i번째 차원의 정보를 저장하기 위한 공간의 크기 $|d_i|$가 동일하다고 가정할 때, 1개의 디스크 블록에 저장할 수 있는 최대 데이터 수 $Bf_{\max}$는 다음과 같이 결정된다.

$$Bf_{\max} = \left\lfloor \frac{|B|}{|d_i| \times d} \right\rfloor$$

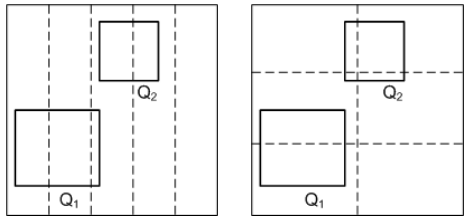
이 때, N개의 데이터를 저장하기 위해서 필요한 최소 블록 수 N_B는 다음과 같다.

$$N_B = \left\lceil \frac{N}{Bf_{\max}} \right\rceil$$

본 논문에서는 데이터 블록 수와 그리드 셀의 수를 동일한 의미로 해석한다. N_B개의 그리드 셀을 생성하기 위해서 각 차원을 동일하게 분할할 때 각 차원의 분할 횟수 λ_i는 다음과 같이 결정된다.

$$\lambda_i = \left\lceil \sqrt[d]{N_B} \right\rceil$$

그림 2는 2차원 공간에 대해서 필요한 그리드 셀의 수가 5일 때, 분할방법이 영역질의에 미치는 영향의 한 예를 보여 준다.



$(a)\ \lambda_0 = 5, \lambda_1 = 1$ $(b)\ \lambda_0 = 2, \lambda_1 = 3$

그림 2. 분할방법에 따른 영역질의 성능 예

예에서 알 수 있듯이 분할 방법에 따라서 영역질의와 겹치는 그리드 셀의 수가 차이가 있음을 알 수 있다. 동일한 디스크 할당 알고리즘을 적용할 때 영역질의와 겹치는 그리드 셀의 수를 최소로 하는 것이 디클러스터링 알고리즘 성능 향상을 위해서 필요하다

저차원 공간에 대해서는 분할 횟수가 영역질의에 많은 영향을 주지 않지만, 차원이 증가할수록 이른 바 "차원의 저주(*Curse of the Dimensionality*)"현상으로 인해 분할 방법이 많은 영향을 미친다. 이러한 현상은 크게 2가지 요인으로 인해 발생한다. 첫 째, 다차원 공간에 대해서는 일부 차원에 대해서만 2번만 분할이 발생한다. 예를 들어, 15차원 공간에 대해서 각 차원에 대해서 2번의 분할이 발생해도 32,768개의 그리드 셀이 생성된다. 좀 더 자세히 살펴보면 그리드 형태로 분할할 때에는 $\lceil \log_2 N_B \rceil$ 차원에 대해서는 2번의 분할만 발생하고 나머지 차원에

대해서는 전혀 분할이 발생하지 않는다. 둘 째, 다차원 공간에서는 매우 작은 선택률(*selectivity*)의 영역질의에 대해서도 각 차원의 길이가 매우 커지는 문제점이 있다. $[0,1)^d$공간에 대해서 선택률이 s인 정방형 영역질의에 대해서 각 차원의 길이 q_i는 $\sqrt[d]{s}$와 같이 결정되며 차원이 증가할수록 극히 작은 선택률의 영역 질의에 대해서도 0.5보다 커진다. 다음 추론은 분할 횟수와 영역질의의 길이에 대한 관계에 따라서 고차원 공간의 그리드 분할의 문제점을 잘 설명해 준다.

추론 1. $[0,1)^d$ 공간을 이루는 각 차원이 λ개의 동일한 너비로 분할되었을 때, 정방형 영역질의 Q의 한 변의 길이 q_i가 다음 조건을 만족하면 영역질의 Q는 모든 그리드 셀과 겹치게 된다.

$$q_i \geq 1 - \frac{1}{\lambda}$$

그림 3은 각 차원이 4바이트로 구성된 10^6개의 데이터에 대해서 디스크 블록 크기가 4KB일 때 차원의 증가에 따라서 정방형 영역질의 한 변의 길이와 분할 횟수와의 관계를 나타낸 것이다.

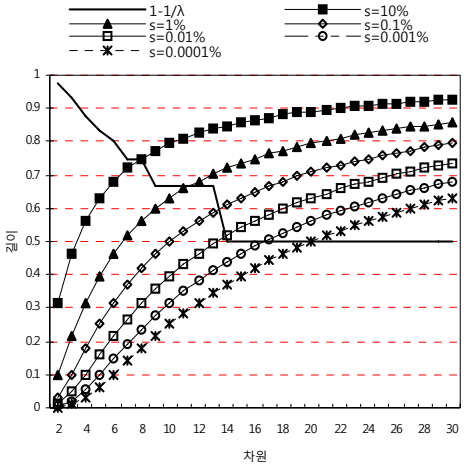


그림 3. 다차원 공간의 그리드 분할과 영역 질의 크기의 관계

차원이 증가함에 따라서 각 차원의 분할횟수는 급격히 감소하여 14차원 이상의 데이터에 대해서는 일부 차원에 대해서 단 2번의 분할만 발생하고 나머지 차원은 전혀 분할이 발생하지 않는다. 또한, 차원이 증가함에 따라 영역질의 한 변의 길이가 급격히 증가하여 선택률이 10%인 영역질의의 경우, 4차원 이상의 공간에서는 영역질의 한 변의 길이가 0.5보다 큰 것을 알 수 있으며 20차원 이상의 경우에는 선택률이 10^{-6}인 영역질의에 대해서도 한 변의 길이가 0.5보다 크게 된다. 즉, 어떠한 영역질의에 대해서도 모든 그리드 셀과 겹치게 되어 더 이상 디스크 매핑 알고리즘의 개선으로 인한 디클러스터링 알고리즘의 성능 향상은 기대할 수 없게 된다.

본 논문에서는 다차원 공간에 대해서 그리드 분할 방법의 개선을 통하여 기존에 제시되었던 매핑함수를 적용하는 디클러스터링 알고리즘의 성능을 개선하고자 한다. d차원 공간에 대한 그리드 분할은 각 차원의 분할 횟수 λ_i를 결정하는 문제와 동일하며 그림 3에서 알 수 있듯이 모든 차원에 대해서 분할이 발생하면 다차원 공간에 대해서는 극히 작은 영역질의에 대해서도 모든 그리드 셀이 영역질의와 겹치게 된다. 따라서, 디클러스터링 알고리즘의 성능 향상을 위하여 다차원 공간의 특성을 반영하여 그리드 분할을 하여야 하며 이를 위해서 주어진 그리드 분할에 대해서 영역질의 성능을 예측하는 모델이 필요하게 되며 다음 장에서 이를 제시하도록 한다.

3. 다차원 공간 그리드 분할에 대한 영역질의 성능예측 모델

본 논문에서는 다차원 공간에 대해서 효과적인 그리드 분할방법을 결정하기 위하여 질의크기에 따른 그리드 분할 방법의 성능을 수학적으로 계산하는 방법을 제시하고 이를 활용한 다차원 공간의 그리드 분할 방법을 제시하도록 한다.

본 장에서는 다차원 공간에 대한 그리드 분할 방법의 성능예측 모델을 제시한다. 먼저, 정방형 영역질의에 대해서 성능예측 모델을 제시하고 이를 확장하여 각 차원의 길이가 다른 경우에 대해서 일반화하도록 한다. 각 차원이 λ개의 동일한 너비로 분할되어 있을 때, 1개의 구간의 길이는 $\frac{1}{\lambda}$이며 구간 질의 q는 $(q_{\min}, q_{\max})$로 정의된다. 본 논문에서는 질의 q의 상한값인 $q_{\max}$를 기준으로 성능예측 모델을 도출하도록 한다. 먼저, 구간 질의 q의 길이가 분할 구간의 너비 $\frac{1}{\lambda}$보다 작을 경우에 대해서 정의한다. 그림 4는 1차원 $[0,1)$을 λ개의 동일한 구간으로 분할한 예를 보여준다.

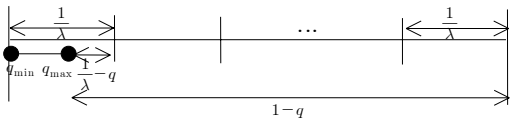


그림 4. 1차원 공간분할 예(질의 변의 길이가 구간 1개의 너비보다 작은 경우)

보조정리 1. 한 변의 길이 q가 $\frac{1}{\lambda}$보다 작은 정방형 영역질의 Q에 대해서 특정 분할 축에 대해서 1개의 구간과 겹치는 확률은 다음과 같다.

$$\frac{1-\lambda q}{1-q}$$

증명. 그림 4에서 알 수 있듯이 $q_{\max}$는 $[0,1)$ 영역 가운데 $(1-q)$영역에 존재할 수 있다. 첫 번째 영역을 예를 들면, $q_{\max}$가 $\left(\frac{1}{\lambda}-q\right)$ 영역에 존재할 경우에는 1개의 영역과 겹치게 되며 $q_{\max}$가 $\frac{1}{\lambda}$을 초과할 경우에는 2개의 영역과

겹치게 된다. 즉, q의 상한값 $q_{\max}$가 다음의 영역에 존재할 경우에는 1개의 구간과 겹치게 된다.

$$\left[\frac{1}{\lambda}\bullet i-q,\ \frac{1}{\lambda}\bullet i\right),\quad 1\le i\le \lambda$$

또한, 위의 영역을 제외한 나머지 영역에 대해서는 2개의 구간과 겹치게 된다. X가 크기가 $\frac{1}{\lambda}$보다 작은 구간 질의 q가 겹치는 영역의 수를 나타내는 확률변수라고 하면 다음 식을 만족한다.

$$\Pr(X=1)=\frac{\left(\frac{1}{\lambda}-q\right)\bullet\lambda}{1-q}=\frac{1-\lambda q}{1-q}$$

또한, 보조정리 1에 따라서 한 변의 길이 q가 $\frac{1}{\lambda}$보다 작은 정방형 영역질의 Q에 대해서 특정 분할 축에 대해서 2개의 구간과 겹치는 확률은 다음과 같이 계산될 수 있다.

$$\Pr(X=2)=1-\Pr(X=1)=\frac{q(\lambda-1)}{1-q}$$

각 차원이 λ개의 동일한 너비로 분할되어 있는 $[0,1)^d$ 그리드 공간에 대해서 각 차원의 크기가 $\frac{1}{\lambda}$보다 작은 d차원 정방형 영역질의 Q에 대해서 1개의 구간과 겹치는 차원의 수 Z는 $p=\Pr(Z=1)$을 만족하는 이항분포 $B(d,p)$를 따른다. 따라서, d차원 가운데 z개의 차원이 1개의 구간과 겹치는 확률은 다음과 같다.

$$\Pr(Z=z)=\binom{d}{z}p^z(1-p)^{d-z}$$

1개의 구간과 겹치는 차원의 수가 Z일 때, 영역질의 Q와 겹치는 전체 그리드 셀의 수 Y는 $Y=1^Z\bullet 2^{d-Z}$를 만족하며 기대값 $E(Y)$는 다음과 같다.

$$E(Y)=E(2^{d-Z})=\sum_{z=0}^{d}2^{d-z}\binom{d}{z}p^z(1-p)^{d-z}$$

이항분포를 $B(d,p)$ 따르는 Z의 적률생성함수(*moment generating function*)는 $E[e^{tZ}]=(1-p+pe^t)^d$을 만족하며 $2^{d-Z}=2^de^{-Z\ln 2}$이다. 따라서, 한 변의 길이 q가 1개의 구간 길이 $\frac{1}{\lambda}$보다 작은 정방형 영역질의 Q와 겹치는 그리드셀 수에 대한 기대값은 다음과 같다.

$$\begin{aligned}E[Y]&=2^dE[e^{-Z\ln 2}]\\&=2^d(1-p+pe^{-\ln 2})^d\\&=2^d\left(1-\frac{p}{2}\right)^d\\&=(2-p)^d\end{aligned}$$

다음은 한 변의 길이 q가 임의의 길이를 가지는 정방형 영역질의 Q에 대해서 겹치는 그리드 셀의 수를 예측하는 수학적 모델을 제시하도록 한다. 영역질의 Q의 한 변의 길이 q가 다음을 만족한다고 하면 그림 5와 같이 $i=\lambda$인 경우를 제외하고 q는 i개 또는 $(i+1)$개의 구간과 겹치게 된다.

$$\frac{1}{\lambda}\bullet(i-1)\le q\le\frac{1}{\lambda}\bullet i,\qquad 1\le i\le\lambda \tag{1}$$

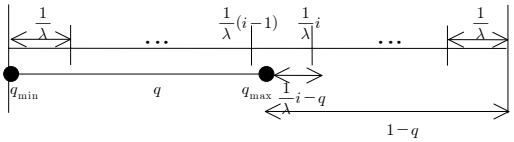


그림 5. 1차원 공간의 분할 및 질의 예
(1개의 구간 너비보다 큰 경우)

변수 X가 특정 분할 축에 대해서 한 변의 길이가 q인 정방형 영역 질의 Q와 겹치는 구간의 수를 나타내는 확률변수라고 하자.

보조정리 2. 한 변의 길이 q가 식 (1)을 만족하는 정방형 영역질의 Q에 대해서 특정 분할 축에 대해서 i개의 구간과 겹치는 확률은 다음과 같다.

$$\frac{(i-\lambda q)(\lambda-i+1)}{\lambda(1-q)}$$

증명. 한 변의 길이가 q인 정방형 영역 질의 Q에 대해서 특정 분할 축에 대해서 그림 5와 같이 q의 상한값 $q_{\max}$가 존재할 수 있는 구간의 길이는 $(1-q)$이다. $(1-q)$구간 가운데 $q_{\max}$가 다음의 구간에 존재할 때 i개의 구간과 겹치게 된다.

$$\left[q+\frac{1}{\lambda}\bullet k,\ \frac{1}{\lambda}\bullet(i+k)\right),\quad k=0,1,2,\ldots,(\lambda-i)$$

따라서, q가 i개의 구간과 겹치는 확률은 다음과 같다.

$$\frac{\left(\frac{1}{\lambda}\bullet i-q\right)(\lambda-i+1)}{1-q}=\frac{(i-\lambda q)(\lambda-i+1)}{\lambda(1-q)}$$

보조정리 2에 의하여 $(i+1)$개의 구간과 겹치는 확률은 다음과 같다.

$$1-\Pr(X=i)=\frac{(i-\lambda q)(\lambda-i+1)}{\lambda(1-q)}=\frac{(q\lambda-i+1)(\lambda-i)}{\lambda(1-q)}$$

정리 1. $[0,1)^d$ 데이터 공간의 모든 차원이 동일한 λ개의 구간으로 분할되어 λ^d개의 그리드 셀로 분할되어 있을 때, 정방형 영역질의 Q의 한 변의 길이 q가 $\frac{1}{\lambda}\bullet(i-1)\le q\le\frac{1}{\lambda}\bullet i$ 를 만족할 때, 정방형 영역질의 Q와 겹치는 그리드 셀의 수에 대한 기대값 $E(Q)$는 다음과 같다.

$$E(Q)=(1+i-p)^d$$

여기서, p는 특정 분할축에 대해서 i개의 구간과 겹치는 확률을 나타내며 보조정리 2의 결과와 같다.

증명. 변수 Z가 d개의 차원 가운데 i개의 구간과 겹치는 차원의 수를 나타내는 확률변수라고 하자. Z는 이항분포 $B(d,p)$를 만족하며 $p=\Pr(X=i)$이다. 이 때, d개의 차원 가운데, z개의 차원이 i개의 구간과 겹치는 확률은 다음과 같다.

$$\Pr(Z=z)=\binom{d}{z}p^z(1-p)^{d-z}$$

d차원 정방형 영역질의 Q에 대해서 i개의 구간과 겹치는 차원의 수가 Z일 때, 영역질의와 겹치는 그리드 셀의 수는 $i^Z \cdot (i+1)^{d-Z}$와 같다. 확률변수 Y가 정방형 영역질의 Q와 겹치는 그리드 셀의 수를 나타낸다고 할 때 그 기대값은 다음과 같이 계산된다.

$$\begin{aligned} E[Y] &= E[i^Z \cdot (i+1)^{d-Z}] \\ &= E\left[(i+1)^d \cdot \left(\frac{i}{i+1}\right)^Z\right] \\ &= (i+d)^d \cdot E\left[\left(\frac{i}{i+1}\right)^Z\right] \end{aligned}$$

이항분포를 따르는 확률변수 Z의 적률생성함수(*moment generating function*)$\psi(t)$는 다음을 만족한다[36].

$$\begin{aligned} \psi(t) &= E[e^{tZ}] \\ &= \sum_{z=0}^{d} e^{tz}\binom{d}{z}p^z(1-p)^{d-z} \\ &= \sum_{z=0}^{d}\binom{d}{z}(pe^t)^z(1-p)^{d-z} \\ &= (pe^t+(1-p))^d \end{aligned}$$

e^t를 $\frac{i}{i+1}$라고 가정하면,

$$\begin{aligned} E[Y] &= (i+1)^d \cdot E\left[\left(\frac{i}{i+1}\right)^Z\right] \\ &= (i+1)^d \cdot \left(p \cdot \frac{i}{i+1} + (1-p)\right)^d \\ &= (i+1-p)^d \end{aligned}$$

마지막으로 각 차원에 대한 분할횟수가 상이하고 영역질의 한 변의 길이가 상이한 일반적인 경우에 대해서 영역질의와 겹치는 그리드 셀의 수에 대해서 기대값을 수학적으로 제시한다.

$[0,1)^d$ 공간을 이루는 i번째 차원이 λ_i개의 동일한 너비로 분할되었다고 가정하자. 또한, 영역질의 Q의 i번째 차원의 길이 q_i가 다음 조건을 만족하는 정수라고 가정한다.

$$\frac{h_i-1}{\lambda_i} \le q_i < \frac{h_i}{\lambda_i}, \quad 1 \le h_i \le \lambda_i$$

따라서, 영역질의 Q는 i번째 차원에 대해서 h_i개 또는 (h_i+1)개의 구간과 겹치게 된다. 변수 X_i를 i번째 차원에 대해서 겹치는 구간의 수를 나타내는 확률변수라고 하면 다음을 만족한다.

보조정리 3. 영역질의 Q의 i번째 차원의 길이가 q_i일 때, 영역질의 Q가 i번째 차원에 대해서 h_i개의 구간과 겹치는 확률은 다음과 같다.

$$\Pr(X_i = h_i) = \frac{(h_i - \lambda_i q)(\lambda_i - h_i + 1)}{\lambda_i(1-q_i)}$$

증명. 보조정리 2의 증명과 동일한다.

보조 정리 3에 의하여 영역질의 Q의 i번째 차원의 길이가 q_i일 때, 영역질의 Q가 i번째 차원에 대해서 (h_i+1)개의 구간과 겹치는 확률은 다음과 같다.

$$\Pr(X_i = h_i+1) = 1 - \Pr(X_i = h_i) = \frac{(\lambda_i q - h_i + 1)(\lambda_i - h_i)}{\lambda_i(1-q_i)}$$

정리 2. $[0,1)^d$ 공간을 이루는 i번째 차원이 λ_i개의 동일한 너비로 분할되었다고 가정하자. 또한, 영역질의 Q의 i번째 차원의 길이 q_i가 $\left(\frac{h_i-1}{\lambda_i} \le q_i < \frac{h_i}{\lambda_i},\ 1 \le h_i \le \lambda_i\right)$일 때, 영역질의 Q와 겹치는 그리드 셀 수의 기대값은 다음과 같다.

$$\prod_{i=1}^{d}(h_i+1-p_i),$$

여기서, p_i는 i번째 차원에 대해서 h_i개의 구간과 겹치는 확률을 나타낸다.

증명. 각각의 $i(<d)$에 대해서 모집단 특성값(*parameter*) $p_i = \Pr(X_i = h_i)$를 가지는 베르누이 확률변수(*Bernoulli random variable*) Z_i를 가정한다. 다시 말하면,

$$Z_i = \begin{cases} 1, & i\text{번째 차원이 } h_i\text{개의 구간과 겹치는 경우} \\ 0, & i\text{번째 차원이 } (h_i+1)\text{개의 구간과 겹치는 경우} \end{cases}$$

Z_i들은 서로 독립적인 사건이며 Q와 겹치는 그리드 셀의 수 $Y = \prod_{i=0}^{d-1} Y_i$는 각 확률변수 Y_i가 다음을 만족하는 확률변수이다.

$$Y_i = h_i^{Z_i} \cdot (h_i+1)^{1-Z_i}$$

Y_i는 단지 i번째 차원에 대해서 겹치는 구간의 수를 나타낸다. 모집단 특성값으로 p_i를 가지는 베르누이 시행에 대한 적률생성함수는 $(1-p_i+p_ie^t)$이므로 $t = \ln\frac{h_i}{h_i+1}$라고 가정하면 Y_i의 기댓값은 다음과 같다.

$$\begin{aligned} E[Y_i] &= E\left[h_i^{Z_i} \cdot (h_i+1)^{1-Z_i}\right] \\ &= (h_i+1)E\left[\left(\frac{h_i}{h_i+1}\right)^{Z_i}\right] \\ &= (h_i+1)\left(1 - \frac{p_i}{h_i+1}\right) \\ &= 1 + h_i - p_i \end{aligned}$$

마지막으로, Y_i는 서로 독립적인 사건이므로

$$\begin{aligned} E[Y] &= E\left[\prod_{i=d}^{d} Y_i\right] \\ &= \prod_{i=1}^{d} E[Y_i] \\ &= \prod_{i=1}^{d}(h_i+1-p_i) \end{aligned}$$

정리 2에 정의된 수식은 다차원 공간에 대해서 가능한 여러 가지 분할방법 가운데 영역질의를 만족하는 그리드 셀의 수를 감소시키는 분할방법을 결정하는 데 적용될 수 있다.

4. 분할차원 감소를 이용한 다차원 공간의 효율적인 그리드 분할방법

본 장에서는 3장에서 제시한 성능예측 모델을 이용하여 다차원 공간을 효율적으로 분할하는 방법을 제시한다. 각 차원의 정보를 저장하기 위해서 필요한 공간이 동일할 때, 균등분포를 이루는 d차원 N개의 데이터를 저장하기 위한 그리드 분할은 다음을 만족하는 각 차원의 분할 횟수인 정수 λ_i를 결정하는 문제와 동일하다.

$$\prod_{i=0}^{d-1}\lambda_i \geq \left\lceil \frac{N}{Bf_{\max}} \right\rceil,\ \lambda_i \in Z^+,\ 1 \leq \lambda_i \leq \left\lceil \frac{N}{Bf_{\max}} \right\rceil \quad (2)$$

식 (2)를 만족하는 λ_i의 조합은 매우 다양하며 기존 연구들은 모든 차원을 이용하여 필요한 개수의 셀이 생성될 때까지 순차적으로 분할하는 것을 가정하였다. 하지만, 그림 3에서 알 수 있듯이 다차원 공간을 모든 차원을 이용하여 순차적으로 분할할 경우 차원이 증가함에 따라 각 차원에 대한 분할횟수가 급격히 감소하고 이에 반하여 극히 작은 선택률의 영역질의에 대해서도 영역질의 각 차원의 길이가 급격히 증가하여 모든 그리드 셀이 영역질의를 만족함으로 디클러스터링 알고리즘의 성능향상에 제한이 있다.

이를 해결하기 위해서는 일부 차원에 대해서 여러 번 분할하고 나머지 차원은 분할하지 않는 방법을 적용할 수 있다. 예를 들어, 15차원 데이터에 대해서 14,706개의 그리드 셀이 필요한 경우 모든 차원을 이용하여 분할하면 $2^{14}\times 1$형태의 그리드가 생성된다. 즉, 14개의 차원에 대해서는 단 2번의 분할이 발생하고 나머지 차원은 분할이 발생하지 않는다. 하지만, 14개의 차원보다 적은 수의 차원을 선택하여 여러 번 분할하고 나머지 차원은 분할하지 않을 경우 그림 6과 같이 영역질의와 겹치는 평균 그리드 셀의 수를 감소시킬 수 있음을 알 수 있다.

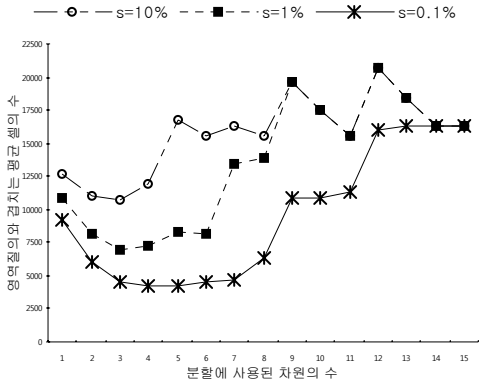


그림 6. 다차원 그리드 분할방법이 영역질의 성능에 미치는 영향 ($d=15$)

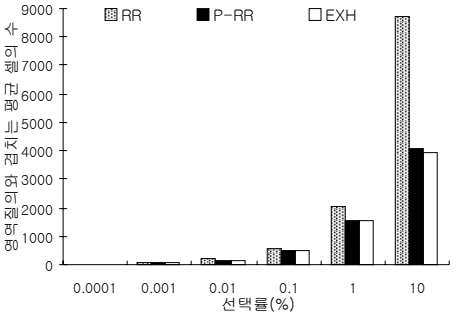
그림 6에서 알 수 있듯이 선택률이 10%, 1%인 영역질의의 경우에는 $25^2\times 24$, 0.1%인 영역질의의 경우 12×11^3 형태의 그리드 분할이 가장 좋은 성능을 보임을 알 수 있다. 즉, 3개 또는 4개의 차원만 이용하여 여러 번 분할하는 것이 성능향상에 도움이 되는 것을 알 수 있다. 그림 6의 결과는 다차원 공간에 대한 정방형 영역질의의 경우, 질의 크기가 증가할 수록 작은 수의 차원에 대해서 여러 번 분할하는 것이 성능향상에 도움을 주는 것을 알 수 있다.

위와 같이 다차원 공간을 그리드 형태로 분할하는 방법은 매우 다양하며 분할 방법에 따라서 디클러스터링 알고리즘의 성능에 많은 영향을 미치게 된다. 본 논문에서는 3장에서 제시한 수학적 모델을 이용하여 영역질의의 크기를 미리 안다고 가정할 경우, 가능한 분할 방법들 가운데 영역질의와 겹치는 평균 그리드 셀의 수를 최소로 하는 분할방법을 결정하는 알고리즘을 제시한다. 식 (2)의 모든 경우에 대해서 고려하는 것은 계산 시간이 많이 필요하므로 정방형 영역질의에 대해서는 순차적으로 분할할 때 필요한 차원의 수보다 적은 수의 차원을 선택하여 선택한 차원에 대해서 순차적으로 여러 번 분할하는 모든 경우에 대해서 성능예측 모델을 적용하여 최적의 분할방법을 결정하는 것이다. 즉, 필요한 그리드 셀의 수가 N_B일 때, 다음 조건을 만족하는 분할에 사용되는 차원의 수 d_p 가운데 영역질의와 겹치는 그리드 셀의 수를 최소로 하는 d_p를 결정하는 문제와 동일하다.

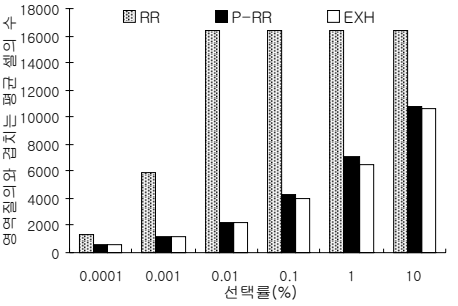
$$\prod_{i=0}^{d_p}\lambda_i \geq N_B,\ d_p = \min(d,\ \lceil \log_2 N_B \rceil),$$
$$\forall_{i,j(\leq d_p)},\ |\lambda_i - \lambda_j| \leq 1,\ \forall_{k(>d_p)},\ \lambda_k = 1 \quad (3)$$

영역질의 Q의 각 변의 길이를 안다고 가정할 때, 위의 조건을 만족하는 각 λ_i 조합에 대해서 정의 2에 의해서 영역질의와 겹치는 그리드 셀의 수를 예측하여 그 중에서 최소로 하는 d_p를 결정한다.

식 (3)과 같은 부분 순차 분할 방법은 정방형 영역 질의에 대해서는 일정 부분 성능향상을 보장하지만, 영역질의의 각 변의 길이가 상이할 경우에는 성능의 저하를 초래한다. 본 논문에서는 이를 고려하여 식 (2)를 만족하는 모든 λ_i 조합에 대해서 성능 예측 모델을 적용하여 그 중에서 최소로 하는 λ_i 조합을 결정하는 방법 또한 적용하였다. 이 방법은 비록 계산 비용이 많이 필요로 하지만, 한 번 구축 후 갱신작업이 필요하지 않는 경우에는 초기에 영역질의 특성을 반영하여 최적의 형태로 분할할 경우 영역질의 성능 향상에 크게 기여할 수 있다. 예를 들어, 공간을 그리드 형태로 분할 후 현재 화면을 기준으로 화면 이동을 고려하여 미리 데이터 블록(그리드 셀)을 캐쉬에 적재할 경우 일반적으로 데이터를 적재하는 형태(영역질의 형태)가 일정하다. 이러한 경우에는 본 논문에서 제시한 방법을 이용하여 캐쉬 정책을 반영하여 미리 데이터 공간을 분할해 놓으면 캐쉬 성능을 개선할 수 있다. 이 외에도 본 논문에서 제시한 방법은 그리드 형태로 분할하고 미리 정해진 형태로 데이터 블록을 접근하는 다양한 응용 분야에 적용 가능하다.



(a) 8차원 그리드 분할



(b) 15차원 그리드 분할

그림 7. 정방형 영역질의와 겹치는 평균 그리드 셀의 수

5. 실험결과

5장에서는 본 논문에서 제안한 방법에 대해서 다양한 시뮬레이션을 통하여 성능평가 결과를 제시한다. 이를 위해서 각 차원의 정보가 4Byte로 구성된 $[0,1)^d$공간에 균등하게 분포하는 10^6개의 데이터를 가정하였으며 디스크 페이지 크기는 4KB를 가정하였다. 따라서, 1개의 블록(그리드 셀)에 저장할 수 있는 d차원 데이터의 최대 수 $Bf_{\max} = \lfloor 4096/4d \rfloor$ 이며 필요한 그리드 셀의 수는 $\lceil 10^6/Bf_{\max} \rceil$ 이다.

분할방법이 디클러스터링 알고리즘의 성능에 미치는 영향을 분석하기 위하여 필요한 그리드 셀을 생성하기 위하여 기존 방법들에서 가정하는 가능한 모든 차원을 이용하여 순차적으로 분할하는 방법(RR), 본 논문에서 제시한 식 (3)을 만족하는 분할차원 감소 후 순차 분할방법 가운데 성능 예측 모델을 적용하여 가장 성능이 좋은 분할 방법을 결정하는 방법 (P-RR), 식(2)를 만족하는 모든 분할 조합에 대해서 최적의 분할 방법을 선택하는 (EXH)의 3가지 형태로 분할하였다.

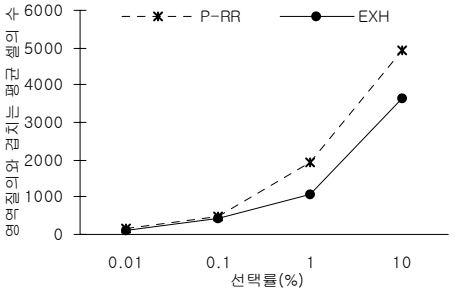
질의크기가 미치는 영향을 분석하기 위하여 $[0,1)^d$ 공간에 랜덤하게 분포하는 다양한 크기의 영역질의 100개를 생성하였다. 선택률이 s인 d차원 정방형 영역질의 Q의 한 변의 길이는 $\sqrt[d]{s}$와 같이 결정하였다. 또한, 영역질의 유형이 미치는 영향을 분석하기 위하여 동일한 선택률에 대해서 각 변의 길이가 서로 다른 영역질의도 생성하였다. 이를 위하여 선택률이 s인 영역질의에 대해서 각 변의 길이가 $\sqrt[d]{s} \pm \delta$ 가 되도록 생성하였으며 δ는 정방형 영역질의의 한 변의 길이인 $\sqrt[d]{s}$와 차이가 20%, 40%가 되도록 결정하였다.

각각의 영역질의와 겹치는 그리드 셀에 대하여 디스크 할당 알고리즘에 의한 디스크 번호를 결정한 후 영역질의와 겹치는 그리드 셀에 할당된 디스크 번호 가운데 가장 많이 발생하는 디스크 번호의 빈도수를 디클러스터링 알고리즘의 응답시간에 대한 성능척도로 적용하였다.

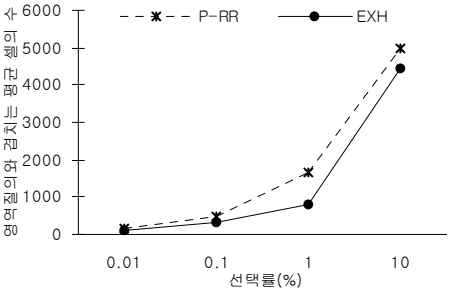
그림 7은 8차원, 15차원 공간에 대해서 서로 다른 크기의 선택률에 해당하는 정방형 영역질의에 대해서 겹치는 평균 그리드 셀의 수를 나타낸 것이다. 가능한 모든 차원을 이용하여 분할하는 경우 8차원 데이터의 경우에는 $4^1 \times 3^7$, 15차원 데이터의 경우에는 $2^{14} \times 1^1$ 형태의 분할이 발생한다. 8차원 데이터의 경우, 작은 영역질의에 대해서는 기존의 모든 차원을 이용하여 분할하는 방법들과 비교하여 성능 차이가 미세하지만, 영역질의 크기가 증가할수록 본 논문에서 제안한 방법들이 성능이 우수함을 알 수 있다. 특히 선택률이 10%인 경우에는 본 논문에서 제안한 방법에 의해서 분할할 경우 영역질의와 겹치는 평균 그리드 셀의 수를 $\frac{1}{2}$이상 감소시킬 수 있다. 15차원 데이터의 경우에는 매우 작은 선택률$(s = 10^{-6})$의 영역질의에 대해서도 본 논문에서 제안한 방법들이 기존 분할방법에 비해서 좋은 성능을 보임을 알 수 있다. 이러한 현상은 차원이 증가할수록 극히 작은 선택률의 영역질의에 대해서도 영역질의 한 변의 길이가 커지기 때문에 발생한다. 예를 들어, 15차원 정방형 영역질의의 경우 선택률이 10^{-6}인 영역질의에 대해서도 한 변의 길이가 0.3981 이상이다. 15차원 데이터의 경우 선택률이 0.01% 이상인 영역질의의 경우 기존 분할 방법의 경우 모든 셀이 겹치는 것을 알 수 있다. 이는 영역질의 한 변의 길이가 0.5이상이기 때문에 발생한다. 그림 7에서 알 수 있듯이 정방형 영역질의의 경우, 가능한 모든 경우의 분할 조합 가운데 가장 좋은 것을 선택하는 방법(EXH)와 분할차원 감소 후 순차분할(P-RR)하는 방법의 성능이 큰 차이가 없음을 알 수 있다.

본 논문에서는 이를 고려하여 동일한 선택률에 대해서 각 변의 길이가 서로 다른 영역질의에 대해서 실험을 하였으며 그림 8은 그 결과를 보여준다. 영역질의 각 변의 길이가 다를 경우에는 모든 경우의 수에 대해서 가장 좋은 분할 조합을 선택하는 방법(EXH)이 단순히 분할차원을 감소하여 순차적으로 분할하는 방법(P-RR)에 비해서 최대 2배 이상의 성능향상을 나타내는 것을 알 수 있다. 이러한 결과는 모든 경우의 수를 고려하여 최적의 분할 형태를 결정하는 방법이 영역질의의 특성을 잘 반영하여 분할하기 때문이다.

그림 9는 8차원 영역질의에 대해서 각 차원의 길이와

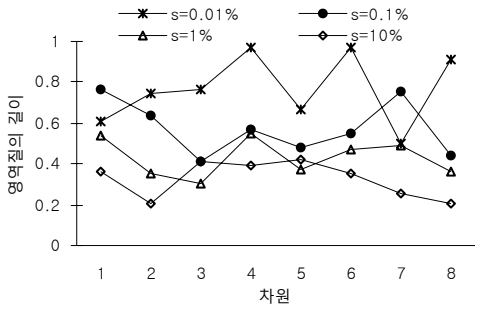


(a) 정방형 질의 길이와 차이 : $\delta = 20\%$

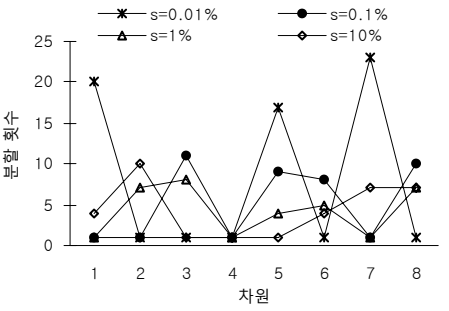


(b) 정방형 질의 길이와 차이 : $\delta = 40\%$

그림 8. 질의 유형에 따른 영역질의와 겹치는 평균 셀의 수 : $(d = 8)$

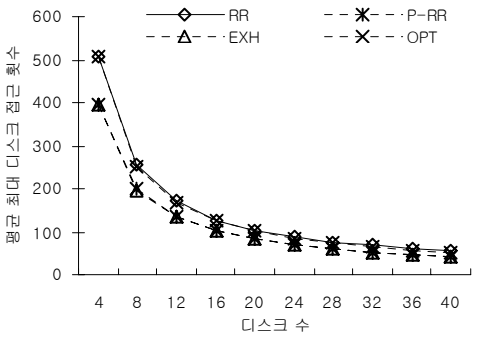


(a) 영역질의 각 차원의 길이

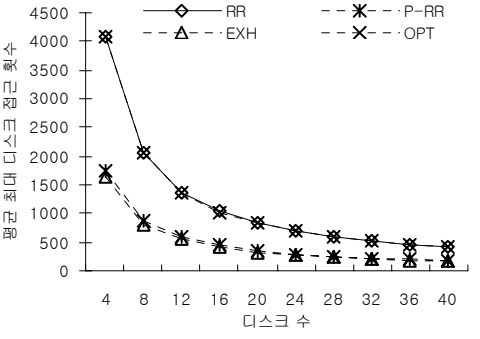


(b) 각 차원의 분할횟수

그림 9. 질의 유형에 따른 영역질의와 겹치는 평균 셀의 수 : $(d = 8, \delta = 40\%)$



(a) $d = 8, s = 1\%$

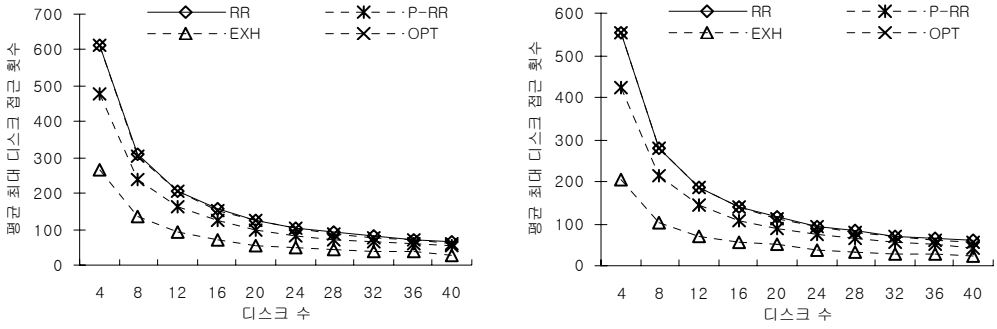


(b) $d = 15, s = 1\%$

그림 10. 정방형 영역질의에 대한 평균 디스크 접근 횟수

각 차원의 분할횟수를 나타낸 것이다. 결과에서 알 수 있듯이 영역질의의 길이가 긴 차원에 대해서는 분할 횟수를 감소시키고 짧은 차원에 대해서는 여러 번 분할하는 것이 효과적임을 알 수 있다. 이는 길이가 긴 차원에 대해서 여러 번 분할할 경우 영역질의의 해당하는 차원에 대해서 많은 구간이 겹치기 때문이다.

영역질의와 겹치는 평균 셀의 수의 감소는 전체 디클러스터링 알고리즘의 성능향상으로 귀결된다. 본 논문에서는 다차원 그리드 분할에 대해서 가장 좋은 디스크 할당 성능을 보여주는 Kronecker Sequence[20]를 이용하



(a) $d = 8, s = 1\%, \delta = 20\%$ (b) $d = 8, s = 1\%, \delta = 40\%$

그림 11. 비정방형 영역질의에 대한 평균 디스크 접근 횟수

여 분할결과에 디스크 번호를 할당한 후 정의 1에 의하여 디클러스터링 알고리즘의 디스크 접근횟수를 계산하였으며 그림 10과 그림 11은 실험 결과를 나타낸다. 그리드 분할에 대해서 여러 개의 복사본을 만들어 디스크 매핑 알고리즘을 적용할 경우 절대 최적의 디스크 매핑이 가능하다고 알려져 있으며 본 논문에서는 이를 고려하여 기존의 알고리즘에서 적용하였던 모든 차원을 이용하여 순차적으로 분할하는 방법에 절대 최적의 디스크 매핑 알고리즘을 적용할 경우와 함께 비교하였으며 OPT는 이를 나타낸다. 그림 10은 선택률이 1%인 정방형 영역질의에 대해서 디스크 수의 증가에 따른 디스크 접근 횟수를 나타낸 것이다. 15차원 데이터에 대해서는 최대 2.5배 이상의 성능향상이 있음을 알 수 있다. 그림 11은 비정방형 영역질의에 대해서 8차원 공간에 대해서 평균 디스크 접근횟수를 나타낸 것이다. 비정방형 영역 질의에 대해서도 디스크 수에 따라 최대 2.7배 이상의 성능향상이 있음을 알 수 있다. 그림 10과 그림 11에서 알 수 있듯이 기존에 적용하였던 모든 차원을 이용하여 순차적으로 분할하는 방법에 대해서 절대 최적의 디스크 할당 알고리즘을 적용할 때 성능이 개선되지만 그 차이는 미세하다. 이러한 결과는 다차원 공간에 대해서 그리드 형태로 분할한 후 디스크 매핑함수를 적용하는 디클러스터링 알고리즘의 경우 디스크 매핑 함수의 성능향상을 통한 성능 개선은 한계가 있음을 보여 준다. 이에 비해서, 본 논문에서 제시한 효율적인 분할 방법을 이용하여 단일 복사본을 생성하고 기존의 디스크 할당 알고리즘을 적용할 경우 디클러스터링 알고리즘의 성능이 향상됨을 알 수 있다.

이는 분할 방법이 다차원 공간에 대해서 매핑함수를 이용하는 디클러스터링 알고리즘의 성능에 많은 영향을 주는 것을 보여 준다.

6. 결론 및 향 후 연구과제

본 논문에서는 그리드 분할 및 매핑함수를 이용하는 기존 디클러스터링 알고리즘들을 다차원 공간에 적용할 때의 문제점을 분석하고 분할방법이 다차원 공간에 미치는 영향을 살펴 보았다.

매핑함수를 이용하는 디클러스터링 알고리즘들을 다차원 공간에 적용할 때, 분할 방법이 중요한 성능 요인임을 제시하고 이를 해결하기 위하여 그리드 분할의 성능예측 모델을 수학적으로 제시하였다. 또한, 제시한 수학적 모델을 이용하여 질의크기를 고려하여 가능한 분할방법들 가운데 영역질의와 겹치는 그리드 셀의 수를 감소시키는 방법들을 선택하는 방법들을 제시하였다.

다양한 실험결과는 본 논문에서 제시한 분할방법들을 적용할 경우 기존의 매핑함수를 이용하는 디클러스터링 알고리즘의 성능을 최대 2.7배 이상 향상 시킬 수 있음을 알 수 있었다.

본 논문에서 제시한 그리드 분할에 대한 성능예측 모델은 질의 특성을 반영하여 최적의 그리드 분할을 수행하기 위한 다양한 분야에 예측모델로 적용 가능하다.

향후 연구 과제는 본 연구에서 제시한 분할방법을 고려하여 새로운 디스크 할당 알고리즘을 제시하는 것이다.

참 고 문 헌

[1] I. Kamel and C. Faloutsos, “Parallel R-trees,” In Proc. of SIGMOD Conference, 1992, pp.195-204.

[2] S. Berchtold, D. A. Keim and H.-P. Kriegel, “The X-tree: An Index Strucutre for High-Dimensional Data,” In Proc. of VLDB Conference, 1996, pp.28-39.

[3] S. Berchtold, C. Bőhm and H-P. Kriegel, “The Pyramid-Technique: Towards Breaking the Curse of Dimensionlaity,” In Proc. of SIGMOD conference, 1998, pp. 142-153.

[4] R. Zhang, B.C. Ooi and K-L. Tan, “Making the Pyramid Technique to Robust to Query Types and Workloads,” In Proc. of ICDE Conference, 2004, pp.313-324.

[5] 서영덕, “병렬 공간 색인을 위한 검색 기법”, 한국공간정보시스템학회 논문지, 제 7권 제 2호, 2005, pp.

81-89.

[6] 김정준, 강홍구, 김동오, 한기준, "메인 메모리 다차원 인덱스를 위한 효율적인 MBR 압축 기법", 한국공간정보시스템학회 논문지, 제 9권 제 2호, 2007, pp.13-23.

[7] C.-S. Chen, J.-Y. Liang, Y.-K. Lee, M.-H. Fan and C.-H. Huang, "Efficient Searching Algorithm for Multi-Dimensional Space Data Using Hilbert Space-Filling Curves," In Proc. of FCS, 2008, pp.264-269.

[8] H Chen, J. Liu, K. Furuse, J.X. Yu and N. Ohbo, "Indexing the Functions: An Efficient Algorithm for Multi-dimensional Search with Expensive Distance Functions," In Proc. of ADMA Conference, 2009, pp.67-78.

[9] H.C Du and J.S. Soblewski, "Disk Allocation for Cartisian Files on Multiple-Disk Systems," ACMTrans.DatabaseSystems, Vol.7, No.1, 1982, pp.82-102.

[10] J. Li, J. Srivastava, and D. Rotem, "CMD: A Multidimensional Declusteirng Method for Parallel Data Systems," In Proc. of VLDB Conference, 1992, pp.3-14.

[11] M. H. Kim and S. Pramanik, "Optimal File Distribution For Partial Match Retrieval.,"In Proc. of SIGMOD Conference, 1988, pp.173-182.

[12] C. Faloutsos and D. Metaxas, "Disk Allocation Methods Using Error Correcting Codes," IEEE Transon Computers, Vol.40 No.8, 1991, pp. 907-914.

[13] C. Faloutsos and P. Bhagwat, "Declusteirng Using Fractals," In Proc. of Parallel and Distributed Information Systems Conference, 1993, pp.18-25.

[14] S. Prabhakr, K. Abdel-Ghaffar, and A. El Abbadi, "Cyclic Allocation for Two-Dimensional Data," In Proc. of ICDE Conference, 1998, pp.94-101.

[15] S. Prabhakar, D. Agrawal, and A. E. Abbadi, "Disk Allocation for Fast Range and Nearest-Neighbor Queries," Distributed and Parallel Databases, Vol.14 No.2, 2003, pp.107-135.

[16] S-W. Kuo, M. Winslett, Y. Cho, and J. Lee, "New GDM-based Declustering Methods for Parallel Range Queries," In Proc. of IDEAS Symposium, 1999, pp.119-127.

[17] M. J. Atallah and S.Prabhakar,"(Almost) Optimal Parallel Block Access for Range Queries," In Proc. of PODS Conference, 2000, pp.205-215.

[18] R. Bhatia, R.K. Sinha, and C.M. Chen, "Declustering Golden Ratio Sequences," In Proc. of ICDE Conference, 2000, pp.271-280.

[19] C. M. Chen and C. T. Cheng, "From Discrepancy to Declustering: Near optimal multidimensional declustering strategies for range queries," In Proc. of PODS Conference, 2002, pp.29-38.

[20] CM. Chen, R. Bhatia, and R.K. Sinha, "Multidimensional Declustering Schemes Using Golden Ratio and Kronecker Sequences," IEEE Trans. Knowledge and Data Engineering, Vol.15 No.3, 2003, pp.659-670.

[21] B. Himatsingka and J. Srivastava, "Performance Evaluation of Grid Based Multi-Attribute Record Declustering Methods," In Proc. of ICDE Conference, 1994, pp.356-365.

[22] B.K. Moon and J.H. Saltz, "Scalability Analysis of Declustering Methods for Multidimensional Range Queries," IEEE Trans. Knowledge and Data Engineering, Vol.10 No.2, 1998, pp.310-327.

[23] Yuan Y. Sung, "Performance analysis of disk modulo allocation method for Cartisian product files," IEEE Trans. Software Eng, Vol.13 No.9, 1987, pp.1018-1026.

[24] K. Abdel-Ghaffar and A. E. Abbadi, "Optimal Allocation of Two-Dimensional Data," In Proc. of ICDT Conference, 1997, pp.409-418.

[25] A.S. Tosun and H. Ferhatosmanoglu, "Optimal parallel I/O using replications," In Proc. of ICPP Conference, 2002, pp. 506-513

[26] Y. Liu, S. Y. Sung, H. Xiong and P. A. Ng, "Data Declustering with Replications". In Proc. of DASFAA Conference, 2004, pp.682-693

[27] H. Ferhatosmanoglu, A. S. Tosun, A. Ramachandran, "Replicated Declustering of Spatial Data," In Proc. of PODS Conference, 2004, pp.125-135.

[28] A. S. Tosun,, "Threshold Based Declustering in High Dimensions," In Proc. of DEXA Conference, 2005, pp.818-827.

[29] A. S. Tosun, "Analysis and Comparison of Replicated Declustering Schemes," IEEE Trans. Parallel Distributed. System(TPDS), Vol. 18 No.11, 2007, pp.1578-1591

[30] K. Yasin Oktay, A. Turk, C. Aykanat, "Selectivity Replicated Declustering for Arbitrary Queries," In Proc. of Euro-Par Conference, 2009, pp.375-386.

[31] B. Moon, A. Acharya, and J. H. Saltz, "Study of Scalable Declustering Algorithms for Parallel Grid Files," In Proc. of IPPS Symposium, 1996, pp.34-440.

[32] M. T. Fang, R.C.T. Lee, and C.C. Chang, "The Idea of De-Clustering and Its applications," In

Proc. of VLDB Conference, 1986, pp.181-188.

[33] D.R. Liu and S. Shekhar, “Partitioning Similarity Graphs: A Framework for Declustering Problems,” International Journal Information System, Vol.21 No.6, 1996, pp.475-496.

[34] D. R. Liu and M. Y. Wu, “A Hypergraph Based Approach to Declustering Problems,” Distributedand Parallel Databases.,Vol.10 No.3, 2001, pp.269-288.

[35] S.Berchtold, C. Bohm, B. Braunmuller, D.A. Keim, and H.-P. Kriegel, “Fast Parallel Similarity Search in Multimedia Databases,” In Proc. of SIGMOD Conference, 1997, pp.1-12

[36] J. L. Johnson. Probability and Statistics for computer science, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003

김 학 철

1997년 부산대학교 전자계산학과 졸업 (학사)

1999년 부산대학교 전자계산학과 졸업 (석사)

2005년 부산대학교 전자계산학과 졸업 (박사)

2005년~2007년 한국전자통신연구원 텔레매틱스·USN연구단

2007년~2008년 경원대학교 텔레매틱스 응용소프트웨어 인력양성단 연구교수

2008년~현재 한국전자통신연구원 융합기술연구부문 선임연구원

관심분야는 시공간 데이터베이스, GIS, 지오 센서네트워크