이화여자대학교 과학기술대학원
2001학년도
석사학위 청구 논문

가중치와 장애물을 고려한 밀도기반의 공간 클러스터링 알고리즘

컴 퓨 터 학 과

임 현 숙

2 0 0 2

가중치와 장애물을 고려한
밀도기반의 공간 클러스터링 알고리즘

이 論文을 碩士學位 論文으로 提出 함

2002 年 1月

梨花女子大學校 科學技術大學院

컴퓨터學科 林 賢 淑

임 현 숙의 碩 士 學 位 論 文을 認准 함

指導敎授 용 환 승

審査委員 김 명 _____

이 상 호 _____

용 환 승 _____

梨 花 女 子 大 學 校 科 學 技 術 大 學 院

목 차

그 림 목 차

표 목 차

論 文 槪 要

공간 데이터 마이닝이란 공간 데이터베이스 내에 함축적으로 존재하는 흥미 있는 관계와 특징을 발견하는 과정이라고 정의 할 수 있다. 다양한 데이터 마이닝 알고리즘 중에서 클러스터링은 대량의 데이터에 적용하기 용이하고 다양한 타입의 데이터에 적용할 수 있다는 장점으로 인해 공간 데이터 마이닝 분야에서 적극적으로 활용되고 있으며, 특히 밀도 기반의 클러스터링 기법은 공간 데이터의 지역성을 처리하는 데에 효과적으로 사용되고 있다.

기존의 밀도 기반 공간 클러스터링 기법인 DBSCAN은 공간상에 분포한 대량의 데이터들에 대하여 다양한 크기와 모양의 클러스터를 발견하면서 동시에 잡음을 처리할 수 있다는 장점을 갖는다. 그러나 데이터의 위치 속성만을 고려한다는 제한을 가지며, 공간적인 속성과 함께 가중치로서의 비공간적 속성을 고려하는 방법이나 데이터와 함께 존재하는 장애물들을 처리하는 방법에 대한 연구는 부족하였다.

본 논문은 DBSCAN을 기반으로 하는 두 가지의 새로운 공간 클러스터링 알고리즘인 DBSCAN-W와 COD-DBSCAN을 제안한다. DBSCAN-W는 클러스터링의 응용 목적에 따라 특정 속성에 대한 가중치를 영역으로 확장하여 밀도에 반영하는 알고리즘이며, COD-DBSCAN은 장애물이 존재하는 경우 장애물을 우회하는 새로운 오브젝트 간 거리 함수를 적용하는 알고리즘이다. 또한 실험을 통해 두 가지의 확장 알고리즘이 기존의 밀도기반 알고리즘에서 찾아내지 못한 새로운 형태의 클러스터링 결과를 도출하는 것을 보인다.

I. 서 론

1.1 연구 배경 및 목적

클러스터링(clustering)이란 주어진 데이터 집합을 서로 유사성을 가지는 몇 개의 클러스터로 구분해 나가는 과정으로, 하나의 클러스터에 속하는 데이터 점들 간에는 서로 다른 클러스터 내의 점들과는 구분되는 유사성을 갖게 된다[1]. 데이터 마이닝(data mining)에서 클러스터링 방법은 기존의 통계, 기계학습, 패턴인식에서 쓰이던 방법에 부가적으로 데이터베이스 지향적인 제약 사항들(제한된 메모리 양, I/O 시간 최소화 등)을 첨가시킨 것으로서, 최근의 멀티미디어 데이터와 같이 혼합되고 다양한 다차원 데이터를 효율적으로 분류해 나가기 위한 방안으로 연구되고 있다. 클러스터링 방법은 크게 분할(partitioning) 방법[2]과 계층적(hierarchical) 방법[3,4], 밀도기반(density-based) 방법[5,6], 격자기반(grid-based) 방법[7]과 모델기반(model-based) 방법으로 나눌 수 있다[8].

클러스터링은 대량의 데이터 베이스에 적용하기 용이하고 다양한 타입의 데이터에 적용할 수 있다는 장점으로 인해 공간 데이터 마이닝 분야에서 많이 사용되고 있으며, 특히 밀도 기반의 클러스터링 기법은 공간 데이터에서 흔히 나타나는 다양한 형태의 클러스터를 찾아낼 수 있다는 장점을 갖는다[5]. 그러나 현재까지 진행되어 온 연구들은 대부분 데이터의 공간적인 속성에 치중하여 공간데이터가 갖고 있는 비공간적인 속성을 고려하지 않는 경우가 많으며, 또한 실제 공간 데이터베이스에서 빈번하게 발견되는 장애물로 인한 오브젝트간 거리 문제에 대한 연구도 부족하였다.

그러므로, 응용에 보다 적합한 클러스터링 결과를 도출하기 위해서는 공간 데이터의 공간적인 속성을 충분히 활용함과 동시에 응용 시스템의 목적에 따라 중요하게 간주되는 비공간적 속성을 함께 고려할 수 있는 방안이 연구되어야 하며, 대상 오브젝트 뿐만 아니라 오브젝트 사이에 존재하는 장애물로 인해 발생하는 문제를 효과적으로 처리할 수 있는 방안이 필요하다.

1.2 연구 내용

본 논문은 기존의 밀도기반 클러스터링 방법을 개선하여 보다 효과적으로 클러스터링을 수행하기 위한 두 가지의 새로운 공간 클러스터링 알고리즘을 제안한다. 첫째, 데이터의 공간 속성을 위치로만 한정하고 비공간적 속성을 고려하지 못하는 단점을 보완하기 위해 특정 속성에 대한 가중치를 데이터의 영역으로 확장하여 밀도에 반영하는 알고리즘인 DBSCAN-W를 제안한다. 둘째, 기존의 방법이 direct Euclidian distance를 사용함으로써 오브젝트들과 함께 존재하는 장애물을 고려하지 못했던 문제를 해결하기 위해 오브젝트 간 거리를 장애물을 우회하는 거리로 재정의하는 방법을 사용한 알고리즘인 COD-DBSCAN을 제안한다. 그리고 본 논문에서 제안하는 두 가지 확장 알고리즘의 효율성을 검증하기 위해 기존의 밀도기반 클러스터링 알고리즘인 DBSCAN과 두 가지의 확장 알고리즘을 모두 구현하여 동일한 데이터에 적용한 결과를 분석하였다.

본 논문의 구성은 다음과 같다. II장은 기존의 밀도를 기반으로 하는 공간 클러스터링 알고리즘인 DBSCAN과 장애물을 우회하는 경로를 찾기 위한 가시 그래프

(Visibility Graph) 알고리즘에 대해 기술한다. III장에서는 DBSCAN-W에서 특정 속성에 대한 가중치를 고려하는 방법과 COD-DBSCAN에서 오브젝트 사이에 장애물이 존재하는 경우 장애물을 우회하는 거리를 계산하는 방법을 제안하고, IV장에서는 제안한 알고리즘들에 대한 성능을 평가하고 그 결과에 대해 기술한다. 마지막으로 V장에서는 본 연구의 결론과 함께 앞으로의 연구 방향을 제시한다.

II. 관련 기술 및 연구동향

본 장에서는 공간 데이터(Spatial Data)의 정의와 특성에 대해 살펴보고, 일반적인 데이터와 달리 공간 데이터를 마이닝할 때에 특히 고려되어야 할 사항에 대해 살펴본다. 그리고 이러한 공간 데이터를 효과적으로 마이닝하는 방법으로써, 기존의 밀도기반 클러스터링 알고리즘인 DBSCAN을 살펴보고, 다음에는 공간상에 존재하는 장애물을 처리하기 위해 로봇 모션 플래닝(robot motion planning) 분야에서 많이 사용되는 가시 그래프에 대해 살펴보도록 한다.

2.1 공간 데이터 마이닝

본 절에서는 공간 데이터의 정의와 특성에 대해 살펴보고, 본 논문에서 기본 알고리즘으로 사용한 밀도기반의 공간 클러스터링 알고리즘에 대해 자세히 살펴보도록 한다.

2.1.1 공간 데이터의 정의 및 특성

공간 데이터(spatial data)란 오브젝트들에 의해 점유된 공간에 연관된 정보를 말한다. 데이터는 기하학적인 정보로 구성되며, 이산적(discrete)이거나 연속적(continuous)인 특성을 갖는다. 이산 데이터는 다차원 공간에서 하나의 점(point)으로 나타낼 수 있는데, 이것은 데이터를 공간상에 위치시킬 수 있는 거리 속성 (distance attribute)을 갖는다는 점에서 비공간 데이터(non-spatial data)와 다르다. 연속 데이터

는 공간에서 영역(region)으로 나타낼 수 있다[9].

2.1.2 공간 데이터 클러스터링

비공간 데이터와 공간 데이터의 차이와 마찬가지로 공간 데이터 마이닝 역시 일반적인 데이터 마이닝과 다르다. 데이터베이스에 저장된 공간 오브젝트의 속성은 공간적으로 이웃에 위치한 다른 오브젝트들로부터 영향을 받을 수 있으며, 오브젝트의 공간적 위치나 그 위치에 대한 잠재적 정보가 바로 공간 데이터 마이닝을 통해 추출되는 정보일 수 있다[10].

다양한 마이닝 알고리즘 중 클러스터링 기법은 대량의 데이터에 적용하여 구성원들 사이의 유사성을 기반으로 대상들의 군집을 생성해나가는 데에 효과적이므로, 공간데이터에 적용할 경우 데이터의 공간적 요소를 일반화하여 성공적으로 마이닝을 수행할 수 있게 한다[10].

공간 데이터를 클러스터링 할 때에 특히 요구되는 사항은 다음과 같다.

1) 입력변수(input parameter) 등 클러스터링을 수행하기 위한 사전지식을 최소화하여야 한다[11].

2) 구형(ball-shaped), 선형 등을 포함하여 다양한 형태(arbitrary shape)와 크기(size)의 클러스터를 구분해내야 한다[5,11].

3) 적절한 기준 이하의 데이터는 클러스터에 포함되지 않도록 잡음(noise)으로 처리해야 한다[5].

4) 대량의 데이터베이스에서 효율적으로 수행되어야 한다[5,11].

2.1.3 밀도 기반 공간 클러스터링 알고리즘 - DBSCAN

DBSCAN(Density Based Spatial Clustering of Applications with Noise) [5]은 밀도를 기반으로 하는 클러스터링 알고리즘으로서, 잡음(noise)을 포함한 공간 데이터를 다루는데 적합하며 다양한 모양(arbitrary shape)과 크기(size)의 클러스터를 구분할 수 있다. 이때 클러스터와 잡음은 직관적으로 점들의 밀도(density)를 기준으로 구분한다. 이를 위해 다음의 몇 가지를 정의한다.

[정의 1] (Eps-neighborhood of a point) 한 점 p의 Eps-neighborhood는 p로부터 반경 Eps 내에 있는 이웃(neighborhood)의 집합이다. 즉,

$$N_{Eps}(p) = \{ q \in D \mid dist(p, q) \leq Eps \}$$

이다.

[정의 2] (directly density-reachable) Eps와 MinPts가 다음의 두 조건을 만족할 때 점 p는 점 q로부터 directly density-reachable하다.

1) $p \in N_{Eps}(q)$

2) $|N_{Eps}(q)| \geq MinPts$ (core point 조건)

[정의 3] (density-reachable) p와 q 사이에 directly density-reachable한 점들로 이루어진 연결(chain)이 존재하면 p는 q로부터 density-reachable하다.

[정의 4] (density-connected) 점 p와 점 q로부터 density-reachable한 점 o가 존재하면 점

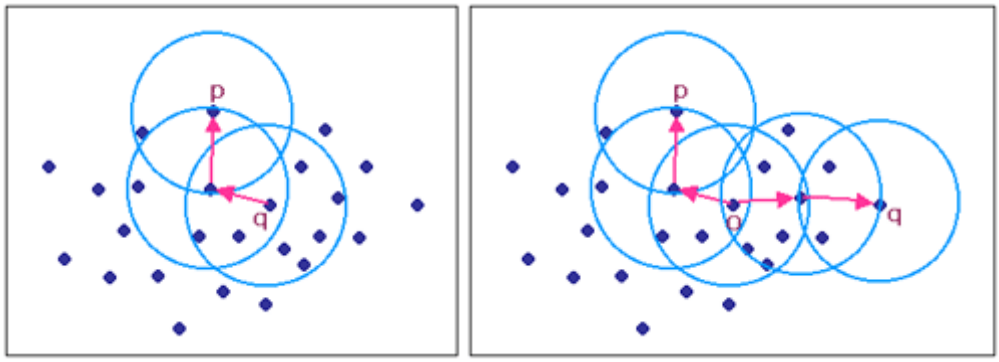
p는 점 q로부터 density-connected하다.

[정의 5] (cluster) D를 점들의 데이터베이스이라고 할 때, 클러스터 C는 다음의 두 조건을 만족하는 D의 non-empty subset이다.

1) 모든 점 p, q에 대해 $p \in C$ 이고 q가 p로부터 density-reachable하면 $q \in C$ 이다.

2) 모든 점 $p, q \in C$ 에 대해 p는 q에 density-connected 하다.

[정의 6] (noise) 잡음은 데이터베이스의 점들 중 어느 클러스터에도 속하지 않는 점들의 집합이다.

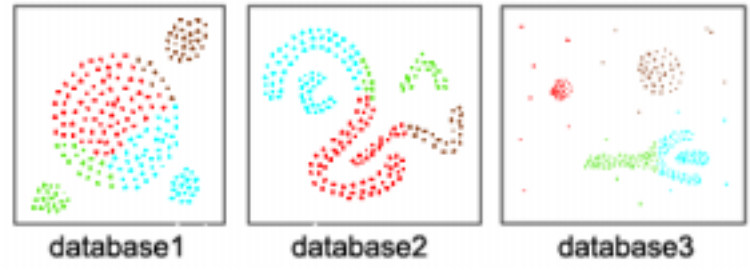


(a) density-reachable (b) density-connected

[그림 2.1] density-reachable과 density-connected

그림 2.1에서 (a)는 점 p가 q로부터 density-reachable한 것을 나타내며, (b)는 점 p

와 q가 각기 o에 의해서 density-connected한 것을 보인다. 결론적으로, 각각의 클러스터는 밀집-연결된(density-connected) 점들의 최대집합으로 정의되며 한 클러스터내의 각 점들은 주어진 반경 Eps 내에 최소 이웃 수 MinPts 이상의 이웃을 갖는다. R*-Tree 인덱스를 이용하는 경우, 전체 DBSCAN의 시간 복잡도는 대상 데이터의 수가 n일 때 $O(n * \log n)$이다



(a) CLARANS를 적용했을 경우 클러스터링 결과 (k = 4)

(b) DBSCAN을 적용했을 경우 클러스터링 결과

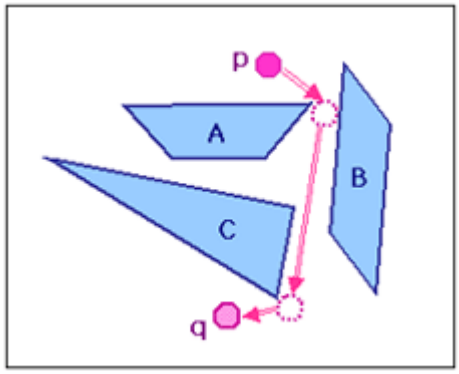
[그림 2.2] CLARANS와 DBSCAN의 클러스터링 결과 비교

그림 2.2는 크기가 다른 4개의 구형(ball-shaped) 클러스터들이 분포한 데이터베이스1, 볼록하지 않은 형태의(non-convex shape) 클러스터들이 분포한 데이터베이스2,

서로 다른 모양(arbitrary shape)과 크기의 클러스터들과 잡음이 공존하는 데이터베이스3을 대상으로, 분할 방법인 CLARANS와 밀도기반 방법인 DBSCAN을 적용하였을 때 각각의 클러스터링 결과를 보여준다.

2.2 가시 그래프 (Visibility Graph)

가시 그래프(Visibility Graph)는 장애물로 가로막힌 공간에서 로봇의 이동 경로를 결정하는 로봇 모션 플래닝(robot motion planning)에 많이 사용되는 알고리즘으로, 그림 2.3은 로봇이 p에서 q로 이동하기 위해 장애물 A, B, C를 우회하는 경로의 예이다[12].



[그림 2.3] 로봇의 이동 경로

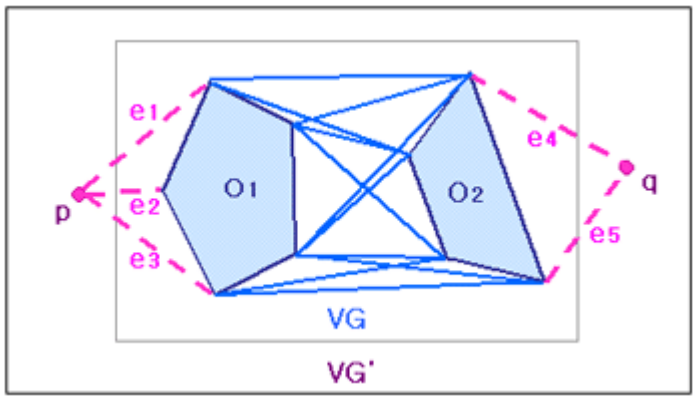
장애물들의 꼭지점의 집합을 node의 집합 V라 하고, 서로 보이는(visible) 두개의 꼭지점을 잇는 선의 집합을 edge의 집합 E라고 할 때 가시 그래프 VG는 다음과 같이 정의된다.

$$VG = (V, E)$$

그림 2.4는 두 장애물에 대한 가시 그래프가 존재할 때, 추가되는 점들에 의해 그래프가 확장되는 방식을 나타낸다. 두 장애물 O_1, O_2에 대한 가시 그래프를 VG = (V, E)라고 하면, p와 q를 포함하도록 확장된 가시 그래프는 다음과 같이 나타낼 수 있다.

$V' = V \cup \{p, q\}$이고 $E' = E \cup \{e_1, e_2, e_3, e_4, e_5\}$ 일 때,

$VG' = (V', E')$



[그림 2.4] 가시 그래프

가시 그래프를 생성하기 위한 naïve 알고리즘은 장애물을 구성하는 꼭지점들의 모든 쌍에 대해 가시성을 검사하는 것으로 $O(n^3)$의 시간 복잡도를 갖는다. 이때 장애물을 구성하는 edge들을 회전방향(cyclic order)으로 정렬하여 순차적으로 가시성을 검사하면 시간복잡도를 $O(n^2 * \log n)$으로 줄일 수 있는데, 이를 위한 알고리즘은 그림 2.5와 같다[12,13]. 그림 2.6의 (a)는 기준 오브젝트를 중심으로 주위의 장애물들을 시계방향으로 정렬한 예를 보여주며, 정렬된 edge들은 (b)와 같이 저장된다.

VisibilityGraph()

for all vertices w_i where i = 1 to n

if VISIBLE(w_i) **then Add** w_i to the list of visibility edges

Insert into T the obstacle edges incident to w_i that lie on the clockwise side of the half-line from p to w_i

Delete from T the obstacle edges incident to w_i on the counterclockwise side of the half-line from p to w_i

VISIBLE(w_i)

if pw_i intersects the interior of the obstacle of which w_i is a vertex

then return false

else if i = 1 or w_{i-1} is not on the segment pw_i

then Search in T for the edge e in the leftmost left

(the edge p intersects first)

if e exists and pw_i intersects e then return false

else return true

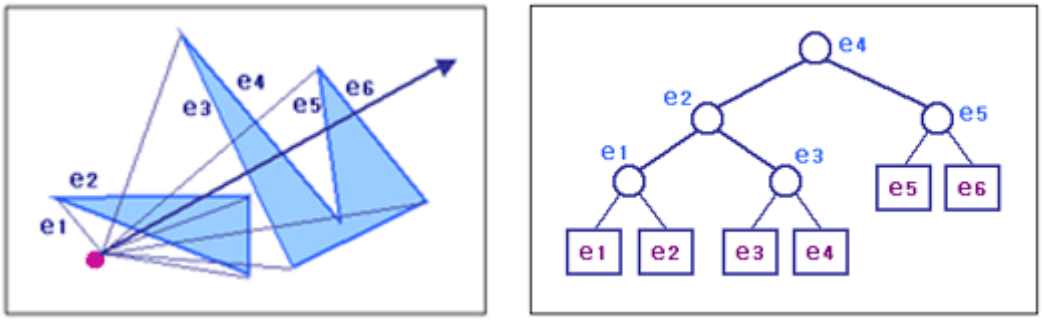
else if w_{i-1} is not visible **then return false**

else Search in T for an edge e that intersects $w_{i-1}w$

if e exists then return false

else return true

[그림 2.5] 가시 그래프 알고리즘



(a) 시계방향으로 정렬된 장애물 (b) 장애물을 구성하는 edge들의 정렬

[그림 2.6] 시계방향으로 정렬된 edge

III. 가중치와 장애물을 고려하도록 확장된 밀도기반 공간 클러스터링 알고리즘

본 장에서는 기존의 밀도기반 공간 클러스터링 알고리즘을 기반으로 하는 두 가지 확장 알고리즘을 제안한다. 먼저, 데이터의 특정 속성에 대한 가중치를 고려하기 위한 방법을 사용하는 알고리즘을 설계하고, 그 다음에는 오브젝트간 거리를 장애물을 고려한 새로운 거리함수로 정의하는 방법을 사용하여 확장된 밀도기반 클러스터링 알고리즘을 설계한다.

3.1 데이터의 특정 속성에 대한 가중치를 고려하는 방법

클러스터링은 각각의 응용에 따라 적합한 가중치를 적용하여 수행한다. 예를 들어, 텍스트 클러스터링의 경우 여러 문서들 내에 공통으로 속해있는 단어의 개수를 기준으로 클러스터링을 수행하고 이때 분류에 중요한 기준이 되는 단어에 가중치를 둘 수 있다. 계층적 클러스터링을 수행할 때에는 유사한 두개의 클러스터를 선택하여 하나의 클러스터로 만드는 연결과정에서 각 클러스터간의 거리를 구할 때 각각의 클러스터의 크기를 가중치로 이용하기도 한다.

본 절에서는 공간 데이터를 대상으로 데이터의 위치정보와 함께 특정 속성에 대한 가중치를 고려해야 하는 예를 살펴보고 그 방법으로써 가중치를 영역으로 고려하는 알고리즘인 DBSCAN-W를 제안한다.

3.1.1 공간 데이터의 비공간적 속성

많은 공간 물체를 다루는 실제적인 응용에서는 다음의 예와 같이 대상물의 공간적인 위치 정보와 함께 비공간적 속성도 다루어져야 한다.

[예 3.1] 거리를 따라 형성된 지역의 상권을 분석하는 응용 프로그램에서는 지리적인 상가의 위치를 고려하면서 동시에 상가 내에 포함된 점포의 수나 총 매출액에 따라 각 데이터에 가중치를 부여하여 클러스터링을 수행하는 것이 요구된다.

[예 3.2] 백화점의 고객 분포를 분석하는 응용 프로그램에서 고객의 지리적 위치를 고려하면서 동시에 고객의 특정 속성(매출액, 방문빈도, 백화점 내의 스포츠센터 회원 가입 여부 등)에 가중치를 두고 클러스터링을 수행하는 것이 요구된다.

3.1.2 가중치를 영역으로 고려하는 방법

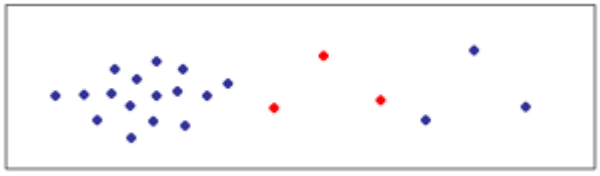
본 절에서는 밀도기반의 공간 클러스터링 알고리즘에서 가중치를 부여하기 위해 다음과 같은 4가지 방법을 제안한다.

첫째, 대상 오브젝트에 대한 가중치가 증가하면 기준이 되는 반경(Eps)을 늘여서 적용한다. 즉, 반경을 늘여서 해당 반경 내에 존재하는 이웃의 수가 많도록 하여 해당 오브젝트가 클러스터에 포함될 확률을 높인다.

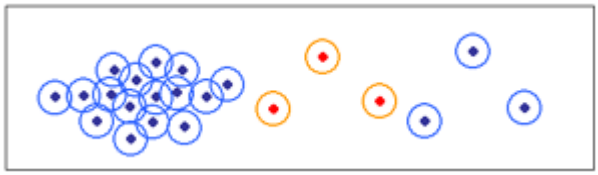
둘째, 대상 오브젝트의 가중치가 증가하면 최소 이웃의 수(MinPts)를 줄여준다. 즉, 클러스터를 구성하는 기준을 완화시켜줌으로 해서 오브젝트가 클러스터에 포함될 확률을 높인다.

위의 두 가지 방법은 DBSCAN에서 범용 파라미터(global parameter)를 적용하는 대신 각 오브젝트 별로 가중치에 따라 변화하는 파라미터를 사용하여 클러스터링을 수행하는 방법으로, 기존의 DBSCAN 알고리즘과 마찬가지로 각 오브젝트를 위치(location) 특성만을 갖는 점으로 처리하는 방법이다.

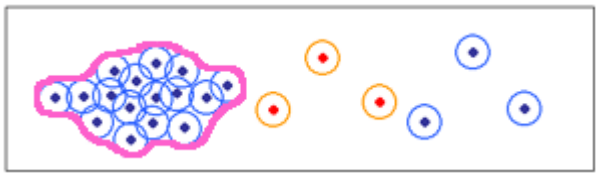
셋째, 대상 오브젝트가 위치 특성 뿐 아니라 가중치에 따른 영역 특성을 갖는다고 보고, 각 영역이 겹쳐지는 오브젝트들을 하나의 클러스터로 만들어 준다. 그림 3.1은 가중치에 따라 각 오브젝트에 영역을 부여한 경우로 세번째 방법에 따라 클러스터링을 수행할 때 결과가 달라지는 예를 보여준다. 그림 3.1의 (a)에는 두 가지 가중치를 갖는 오브젝트들이 분포되어있다. 파란 점은 가중치가 1인 기본 오브젝트들이고 이에 반해 빨간 점은 가중치가 2인 오브젝트들이다. (b)는 각 오브젝트에 영역에 부여할 때 가중치를 고려하지 않고 동일한 영역을 부여한 경우이며 이를 이용한 클러스터링 결과는 (c)와 같다. (d)는 각 오브젝트마다, 가중치에 따라 서로 다른 크기의 영역을 부여한 경우이고, 이때는 가중치를 각 영역의 반지름으로 설정하였다. 이를 이용하여 오브젝트의 영역이 겹치는 경우 하나의 클러스터로 생성한 결과는 (e)와 같다. 영역의 오버랩(overlap)을 이용한 클러스터링 방법은 데이터의 가중치를 공간적 특성인 영역으로 반영하였지만 오브젝트 간의 밀도를 고려하지 못하는 단점을 갖는다.



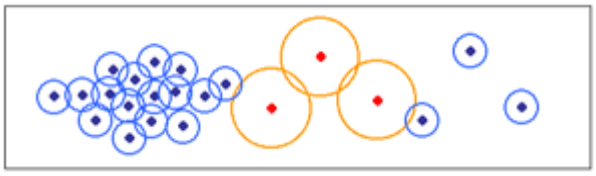
(a) 두 가지 가중치를 갖는 오브젝트들의 분포



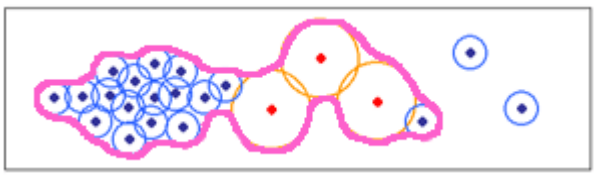
(b) 가중치를 고려하지 않고 동일한 크기의 데이터 영역을 부여한 결과



(c) 가중치를 고려하지 않은 클러스터링 결과



(d) 가중치에 따라 서로 다른 크기의 데이터 영역을 부여한 결과



(e) 가중치를 고려한 클러스터링 결과

[그림 3.1] 영역의 오버랩(overlap)을 이용한 클러스터링에서 가중치에 따라 변화하는 클러스터링 결과

넷째는 오브젝트의 가중치를 공간적 영역 특성으로 처리하면서 동시에 밀도를 고려 하도록 하는 방법으로써 이 논문에서 제안하는 DBSCAN-W(DBSCAN algorithm considering Weight) 알고리즘이다. 즉, 각 오브젝트에 가중치를 기준으로 영역을 부여하고, 기준이 되는 반경 내에 최소 이웃 수 이상의 영역이 겹쳐지는 경우 하나의 클러스터로 처리한다. 이는 기존의 DBSCAN방법에서 모든 오브젝트들을 점으로 처리하는 것과 비교하면 가중치에 따라서 오브젝트들이 서로 다른 크기의 영역을 갖는 원으로 확장된 개념으로, 이를 위해 다음의 몇 가지를 재 정의 한다.

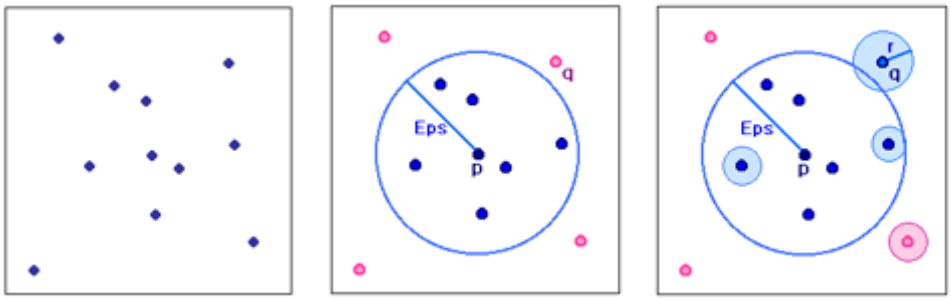
[정의 1] 오브젝트는 (x, y) 좌표값인 공간적 위치를 중심으로 하고, 가중치를 갖는 비공간 속성에 의해 결정된 값을 반지름으로 하는 원으로 표현된다.

[정의 2] 한 오브젝트 p의 Eps-neighborhood는 p의 중심점으로부터 반경 Eps내에 오브젝트를 표현하는 원이 겹쳐지는 이웃들의 집합이다.

[정의 3] 클러스터는 밀집-연결된 원들의 최대 집합(maximal set of density-connected circles)이다.

그림 3.2는 DBSCAN-W를 적용하는 경우 기존의 DBSCAN과는 다른 형태의 클러스터를 생성하는 것을 보인다. (a)는 데이터의 위치 속성만으로 데이터의 분포를 나타낸 것이며, (b)는 각 오브젝트들을 가중치를 갖지 않는 동일한 점으로 간주하는 DBSCAN의 적용 과정을 보인다. 최소 이웃 수를 6이라고 할 때, 점 p를 중심으로

반경 Eps 내의 영역이 기준 밀도를 만족하므로 영역 내부에 존재하는 6개의 점들을 모두 점 p의 Eps-neighborhood에 포함시킬 수 있다. (c)는 각 오브젝트들의 가중치에 따라 생성된 원 영역을 밀도에 반영하는 DBSCAN-W 방법을 나타낸 것으로, (b)의 결과와 달리, 점 p를 중심으로 반경 Eps 내의 영역에는 존재하지 않지만 다른 점에 비해 높은 가중치를 가진 점 q가 가중치에 의해 생성된 영역이 겹쳐짐으로 인해 점 p의 Eps-neighborhood에 포함된다. 즉, 밀도와 가중치를 함께 고려하는 방법을 통해 높은 가중치를 가진 대상이 기준 밀도를 만족하지 않아 클러스터링 과정에서 누락되는 것을 효과적으로 방지할 수 있다.



(a) 데이터의 분포 (b) DBSCAN 적용 (c) DBCAN-W 적용

[그림 3.2] DBSCAN과 DBSCAN-W의 적용 결과

3.2 장애물을 고려한 거리계산 함수를 사용하는 방법

전형적인 클러스터링 작업은 응용에 따라 결정된 기준에 의해 오브젝트들의 집합을 몇 개의 서로 다른 그룹으로 분리한다. 이 때, 분리의 기준으로 오브젝트와 오브젝트, 또는 오브젝트와 그 오브젝트가 속한 클러스터와의 거리로 direct Euclidian

distance를 사용하는 것이 일반적이다. 그러나 실제로 공간상에 분포한 데이터를 대상으로 할 때에는 데이터의 분포 영역 내에 대상 오브젝트들과 함께 존재하는 장애물들로 인해 direct Euclidian distance를 적용하는 것이 적합하지 않는 경우가 빈번하게 발생한다.

본 절에서는 공간 데이터와 함께 장애물이 존재하는 경우 가시 그래프를 구축하고 그래프 상의 최단거리를 탐색하는 새로운 거리계산 함수를 적용함으로써 이러한 문제를 해결하는 알고리즘인 COD-DBSCAN을 제안한다.

3.2.1 장애물이 존재하는 영역의 데이터의 분포

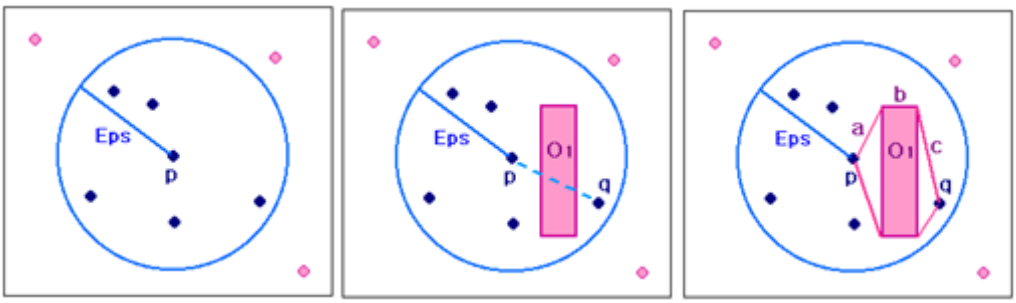
공간상에 분포한 물체를 다루는 많은 실제 응용에서는 다음의 예와 같이 오브젝트들이 데이터의 분포 영역 내에 존재하는 장애물들로 인해 가로막혀있는 경우가 고려되어야 한다.

[예 3.3] 고객의 분포에 따라 백화점의 셔틀 버스 운행 노선을 계획하는 응용 프로그램에서 고객의 지역적인 위치와 함께 영역 내에 존재하는 강이나 공원 등과 같은 자연적, 인공적 장애물을 고려하여 고객의 밀집된 패턴을 분석하는 것이 요구된다.

3.2.2 장애물을 고려한 거리 계산 방법

본 절에서는 데이터 영역 내에 장애물이 존재하는 경우 가시 그래프를 생성하여 그래프 상의 최단거리를 오브젝트간 거리로 사용하는 알고리즘인 COD-DBSCAN (Clustering with Obstructed Distance - DBSCAN) 을 제안한다. 기존의 DBSCAN 방법과

같이 각각의 오브젝트들이 하나의 점으로 표현되는 위치 속성 중심의 클러스터링에서 장애물을 고려하는 방법은 그림 3.3과 같다.



(a)DBSCAN 적용 (b)영역 내의 장애물 (c)장애물을 고려한 거리

[그림 3.3] 장애물을 고려한 DBSCAN 알고리즘의 확장

그림 3.3의 (a)는 기존의 DBSCAN 알고리즘에 의해 클러스터링을 하는 경우, 점 p를 중심으로 기준반경 Eps 내에 최소 이웃 수 MinPts 이상의 점들이 존재하는지를 확인하는 것을 나타낸다. 대상 오브젝트들의 기준반경 내 존재 여부는 중심점과의 거리와 기준반경 Eps를 비교함으로써 알 수 있으며, 이때 오브젝트 간 거리는 direct Euclidian distance를 사용한다. 그러나 (b)와 같이 기준 반경으로 설정되는 영역 내에 장애물이 존재하는 경우에는 점 p, q와 마찬가지로 오브젝트들이 장애물로 가로막혀 있는 상황이 고려되어야 한다. 즉, 점 p와 q를 잇는 선분이 장애물 O_1과 겹쳐지므로 실제로 점 p의 위치에서 점 q의 위치로 이동하기 위해서는 장애물을 우회하는 경로를 선택해야 한다. 그림 3.3의 (c)는 p와 q의 실제 거리로 p와 q를 잇는 선분의 길이가 아니라 장애물 O_1을 우회하는 경로인 a, b, c의 길이의 합을 사용함을 보인다.

이와 같이 오브젝트 간 거리로 장애물을 우회하는 거리를 사용하기 위해서는 해당 영역 내의 모든 오브젝트와 장애물에 대해 가시 그래프가 생성되어야 한다. 가시 그래프의 생성 방법은 그 시기와 범위에 따라 다음의 세가지로 나누어 볼 수 있다.

첫째, preprocessing time에 전체 데이터에 대해 생성하는 방법이다. 밀도 기반 클러스터링은 데이터의 지역적인 밀도에 기반하므로 전역적인 가시 그래프의 생성은 불필요한 계산을 포함하게 되어 비효율적일 수 있다. 그러나 데이터에 대한 사전 지식이 없을 경우, 또는 밀도를 결정하기 위한 입력변수를 자주 변경하면서 클러스터링을 수행해야 할 경우이거나 많은 장애물이 광범위한 영역에 걸쳐 존재하는 경우에는 클러스터링을 수행하기 전에 전체적인 가시 그래프를 생성하는 것이 중복 계산을 방지할 수 있다.

둘째, run-time에 기준 반경과 겹치는 오브젝트와 장애물들에 대해서만 부분적으로 생성하는 방법이다. 이것은 기준 영역의 중심점이 바뀌거나 기준 반경을 변경할 때 마다 가시 그래프를 새로 생성해야 한다는 점에서 중복 계산의 위험이 있으나, 기준반경의 외부에 있는 장애물 및 오브젝트들은 가시 그래프의 구성 요소로서 고려하지 않음으로써 불필요한 계산을 줄일 수 있게 된다. 따라서 소수의 작은 장애물들이 분포해 있거나 입력 변수의 변경이 적은 경우에는 run-time에 부분적인 가시 그래프를 생성하는 것이 효율적이다.

셋째는 위의 두 방법을 혼합한 것으로, preprocessing time에 전체 장애물들의 가시 그래프를 생성하고 run-time에 기준반경 내의 오브젝트들만을 추가하여 가시 그래프를 확장하는 방법이다. 이것은 가시 그래프의 확장성에 기반한 방법이며, 전체 장

애물의 가시 그래프를 생성함으로써 중복 계산을 방지하고, 생성된 가시 그래프의 부분적인 확장을 통해 불필요한 계산을 줄일 수 있다.

가시 그래프의 생성 후에는 Dijkstra’s 알고리즘[14]을 통해 최단거리를 계산하여 그 결과를 거리 인덱스로 저장한다. 클러스터링을 수행하기 위한 기본 알고리즘은 DBSCAN과 동일하지만, 오브젝트 간 거리를 계산 할 때에 direct Euclidian distance 대신 거리 인덱스를 사용함으로써 장애물을 고려한 클러스터링을 가능하게 한다.

IV. 성 능 평 가

본 장에서는 전 장에서 제안한 방법들의 성능 평가를 위하여 기존의 밀도기반 공간 클러스터링 알고리즘인 DBSCAN과 함께 가중치와 장애물을 고려하도록 확장한 두 가지 알고리즘인 DBSCAN-W와 COD-DBSCAN을 각각 구현하여 다음과 같은 성능 평가를 수행한다.

먼저 본 논문에서 제안하는 두 가지 확장 알고리즘의 기본 알고리즘인 DBSCAN을 구현하고, 그 다음 가중치를 고려하는 방법으로 전 장에서 제안한 네 가지 방법 중 오브젝트의 가중치를 공간적 영역 특성으로 처리하면서 동시에 밀도를 고려하는 방법인 DBSCAN-W를 구현한다. 마지막으로 장애물을 고려한 거리 계산을 위해 가시 그래프를 구축하고, 그래프 상의 최단거리를 탐색하여 오브젝트간 거리 인덱스를 생성하는 방법인 COD-DBSCAN을 구현한다.

세 알고리즘의 성능의 차이를 확인하기 위해 동일한 테스트 데이터에 대해 구현한 세 가지 알고리즘을 각각 수행시키고 클러스터링 결과를 분석해 본다.

4.1 시스템 구현 환경

본 논문에서 가중치와 장애물을 고려하도록 기존의 밀도 기반 공간 클러스터링 알고리즘을 확장한 두 가지 알고리즘의 효율성을 분석하기 위한 성능 평가 시스템의 구현 환경은 표 4.1과 같다.

[표 4.1] 시스템 구현 환경

서버측 운영 체제	Sun Solaris 2.6
객체-관계 DBMS	Informix Universal Server
개발도구 및 언어	Informix Spatial Datablade Module 2.2 Informix ESQL/C Informix Datablade API GCC Compiler Microsoft VisualBasic 6.0

가중치와 장애물을 고려한 밀도기반의 공간 클러스터링 알고리즘의 성능 평가를 위한 시스템은 Sun사의 Solaris 2.6 기반의 객체-관계 DBMS인 Informix Universal Server를 사용하였다. Informix Universal Server는 Informix-OnLine Dynamic Server의 DSA를 기반으로 2D, 3D, 이미지, 사운드, 비디오 및 전자 문서와 웹 문서 등 다양한 형태의 데이터와 이를 지원할 수 있는 함수를 정의할 수 있도록 기존 RDBMS의 성능에 대한 확장성을 제공한다. 또한 각 데이터 객체에 대한 계층과 이에 기반한 상속성, 함수 중복 등의 객체 지향적 개념을 지원하는 대표적인 객체-관계 DBMS이다 [15].

데이터베이스로부터 공간 데이터를 효율적으로 접근하게 하는 데이터베이스 공간 엔진의 기본 기능으로는 공간 데이터의 효율적인 입출력을 위한 데이터베이스 관리와 실시간 위상을 형성하는 위상 관리, 공간 질의를 처리할 수 있는 질의 처리와 신속한 검색을 제공하는 공간 색인 관리를 들 수 있다. Informix Spatial Datablade

Module은 Informix Universal Server Database의 기존의 질의와 함께 2차원 연산을 지원하기 위해 9개의 data type과 45개의 SQL function을 제공하며 공간색인을 위해 R-tree를 지원한다[16,17,18,19,20,21]. 그 밖에 프로그램에 필요한 다양한 함수 구현을 위해 Informix Datablade API와 ESQL/C를 사용하였고, 클러스터링 결과의 가시화를 위해 Microsoft VisualBasic 6.0을 사용하였다.

4.2 데이터베이스 구축

본 논문에서 구현한 밀도기반의 공간 클러스터링 알고리즘들인 DBSCAN, DBSCAN-W, COD-DBSCAN의 성능 평가를 위한 데이터는 오브젝트와 장애물의 두 가지 타입이 있으며 각각의 데이터에 대한 테이블 구조는 다음과 같다.

[표 4.2] 오브젝트 테이블 정의

DataPoint 테이블		
필드명	데이터 타입	설 명
id	smallint (not null)	오브젝트의 ID
location	sp2Pnt	오브젝트의 공간 속성인 위치 좌표 (x,y)
region	sp2Circ	가중치를 반영한 오브젝트의 영역 (x, y, r)
weight1	int	가중치로 사용할 비공간적 속성 1
weight2	int	가중치로 사용할 비공간적 속성 2
cluster	smallint	클러스터링 결과

[표 4.3] 장애물 테이블 정의

Obstacle 테이블		
필드명	데이터 타입	설 명
id	smallint (not null)	장애물의 ID
location	sp2Pnt	장애물의 좌측상단 꼭지점의 위치 좌표 (x, y)
region	sp2Box	장애물의 영역 (x1, y1, x2, y2)

DataPoint 테이블은 클러스터링의 대상이 되는 오브젝트들의 정보를 저장하는 테이블로서, 구성하는 필드들을 살펴보면 각 데이터마다 고유한 값을 갖는 id가 primary key로 설정되어 있고, 데이터의 위치인 location은 2차원 좌표값 (x, y)로 표현되는 sp2Pnt type으로 정의된다. 데이터의 영역인 region은 중심 좌표와 반경으로 표현되는 원 형태의 sp2Circ type으로 정의된다. 응용 목적에 따라 다르게 적용될 가중치로는 integer type의 weight1과 weight2의 두 필드가 있다. cluster는 클러스터링의 수행 결과를 저장하기 위한 필드이며 smallint type으로 정의된다.

Obstacle 테이블은 데이터 영역 내에 분포하는 장애물들의 정보를 저장하는 테이블로서, DataPoint 테이블과 마찬가지로 id와 location 필드를 갖는다. 이때, 장애물의 형태는 실제 장애물을 둘러싸는 최소의 사각형, 즉 MBR(minimum boundary rectangle)로 간편화하여 처리하므로 장애물의 영역인 region은 사각형의 좌측 상단과 우측 하단의 두 점의 좌표 값을 갖는 sp2Box type으로 정의된다.

4.3 성능 평가 프로그램 알고리즘

4.3.1 프로그램의 순서도

성능 평가를 위한 프로그램은 대상 데이터들에 대한 정보를 로딩하고 프로그램에 사용되는 변수들을 초기화하는 작업으로 시작된다. 초기화 작업을 완료한 뒤에는 세 가지 알고리즘에서 공통적으로 요구되는 input parameter인 기준반경 Eps와 최소 이웃 수 MinPts를 사용자로부터 입력받는다. 다음으로 수행할 클러스터링 알고리즘을 선택하게 되는데, 기존의 밀도기반 공간 클러스터링 알고리즘인 DBSCAN과 확장된 두 가지 알고리즘인 DBSCAN-W, COD-DBSCAN 중 사용자의 선택에 따라 해당 클러스터링 작업을 수행하도록 한다.

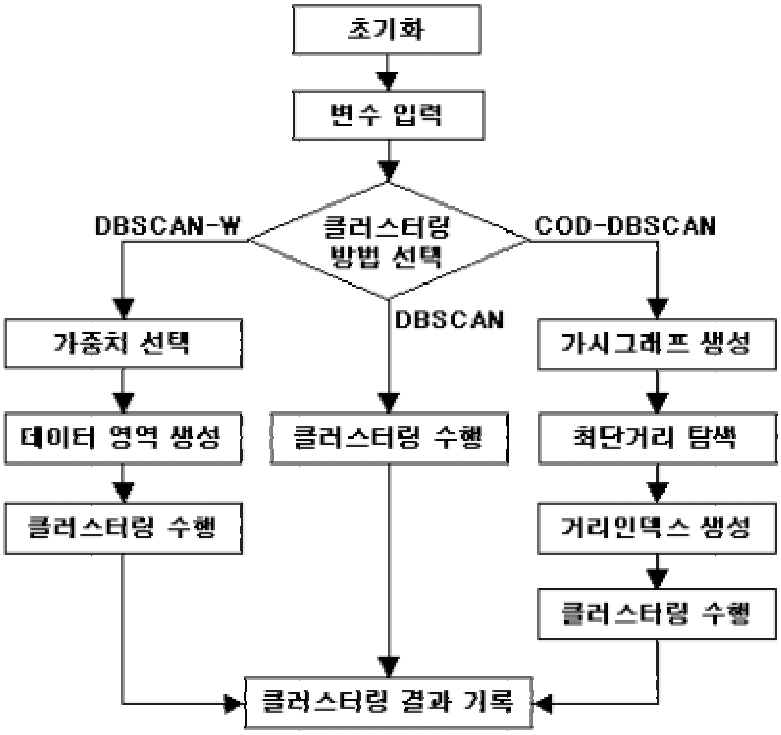
DBSCAN 알고리즘을 선택하면, 데이터 영역 내의 장애물의 존재 여부와 무관하게 기준 반경 Eps 내에 최소 이웃 수 MinPts 이상의 오브젝트가 존재하는지를 확인하여 클러스터링을 수행하고 결과를 데이터베이스에 저장한다.

DBSCAN-W 알고리즘을 선택하면 DBSCAN과 마찬가지로 데이터 영역 내의 장애물의 존재 여부는 고려하지 않으며, 클러스터링의 응용에 적합하게 선택된 가중치만 고려한다. 데이터의 영역을 생성하는 데에 기준으로 사용하기 위해 미리 지정된 두 가지의 가중치 중 하나를 선택하면 그것을 기준으로 데이터마다 각 데이터의 위치를 중심으로 하는 원 형태의 영역을 생성한다. 다음으로, 기준 반경 Eps 내에 최소 이웃 수 MinPts 이상의 영역이 존재하는 경우 하나의 클러스터로 포함시키는 DBSCAN-W 알고리즘을 수행하고 그 결과를 데이터베이스에 저장한다.

COD-DBSCAN 알고리즘을 선택하면 영역 내에 데이터와 함께 존재하는 장애물

을 처리하기 위해 먼저 모든 데이터와 장애물을 대상으로 가시 그래프를 생성한다. 다음으로 Dijkstra's 알고리즘을 적용하여 전 단계에서 생성된 가시 그래프 상의 최단 거리를 계산하여 계산 결과를 거리 인덱스로 저장한다. 마지막으로 DBSCAN 알고리즘을 사용하여 클러스터링을 수행하되, 데이터간 거리는 거리 인덱스를 참조하도록 한다. 클러스터링이 완료되면 그 결과를 데이터베이스에 저장한다.

다음의 그림 4.1은 성능 평가 프로그램의 대략적인 순서도이다.



[그림 4.1] 성능 평가 프로그램 순서도

4.3.2 입력 파라미터

성능평가를 위한 프로그램에서 사용되는 입력 파라미터는 표4.4와 같다. 구현된 두 가지 확장 알고리즘이 모두 밀도기반 클러스터링 알고리즘인 DBSCAN을 기본으로 하기 때문에 DBSCAN에서 기준 밀도를 지정하기 위해 사용하는 두 개의 파라미터, 즉 기준 반경 Eps와 최소 이웃 수 MinPts가 요구된다. DBSCAN-W의 경우 데이터의 영역을 생성하기 위한 기준이 되는 가중치의 종류를 선택하는 것이 추가로 요구된다.

[표 4.4] 입력 파라미터

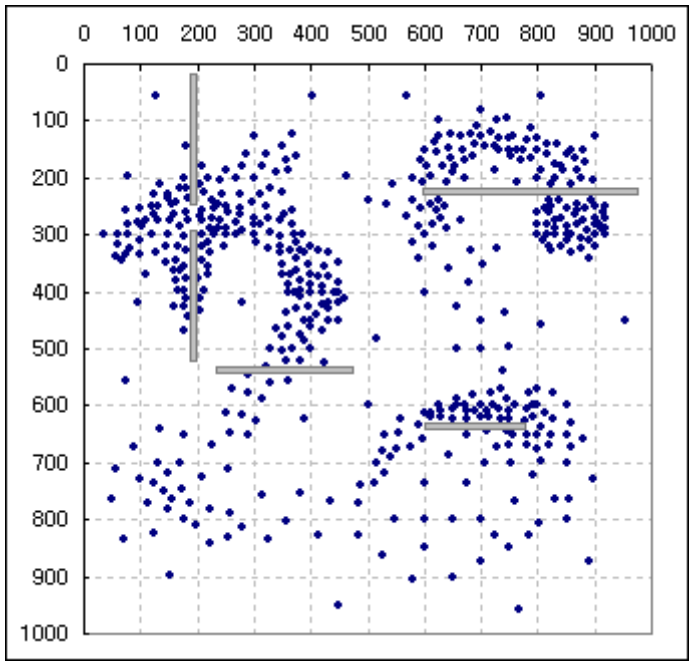
파라미터	값의 범위	설 명
Eps	1~500	기준 반경
MinPts	1~500	최소 이웃 수
weight	A, B	응용에 따라 데이터에 부여하는 가중치

4.4 성능 평가 결과

기존의 알고리즘과 본 논문에서 제안한 알고리즘들의 클러스터링 결과의 차이를 효과적으로 분석하기 위하여 동일한 데이터를 대상으로 입력 파라미터인 기준 반경 Eps와 최소 이웃 수MinPts를 세 가지 알고리즘 모두 동일하게 설정하여 수행하였다. 성능 평가에 사용한 데이터는 (x, y)의 2차원 좌표를 나타내는 값으로 모두 (0, 0) ~ (1000, 1000)의 범위의 값을 갖는 500개의 점들이며, 데이터의 분포 영역 내에 존재하

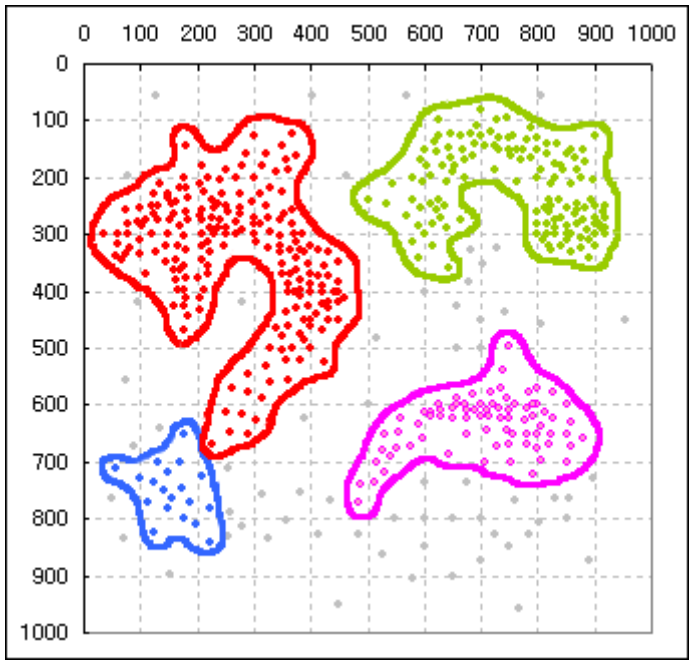
는 장애물은 5개의 사각형 영역이다. 밀도를 결정하는 두 가지 입력 변수인 기준 반경 Eps는 50, 최소 이웃 수 MinPts는 5로 설정하였다.

그림 4.2는 성능평가에 사용한 샘플 데이터로서, 5개의 직사각형 장애물들과 함께 오브젝트들이 밀집하거나 넓게 흩어져 있는 불규칙한 분포 형태임을 보인다.



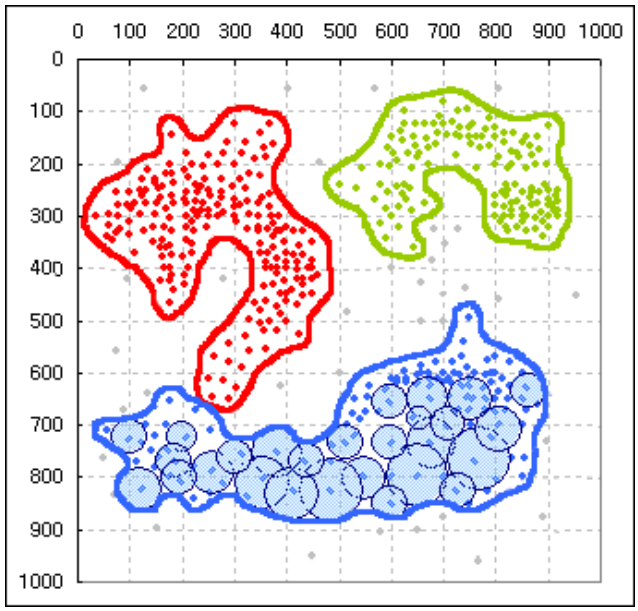
[그림 4.2] 샘플 데이터와 장애물의 분포

그림 4.3은 장애물을 고려하지 않고 대상 데이터만을 사용하여 DBSCAN을 수행한 결과로서, 4개의 클러스터가 생성되었다. 4개의 클러스터에 포함되지 않은 데이터들은 잡음으로 처리되었다.

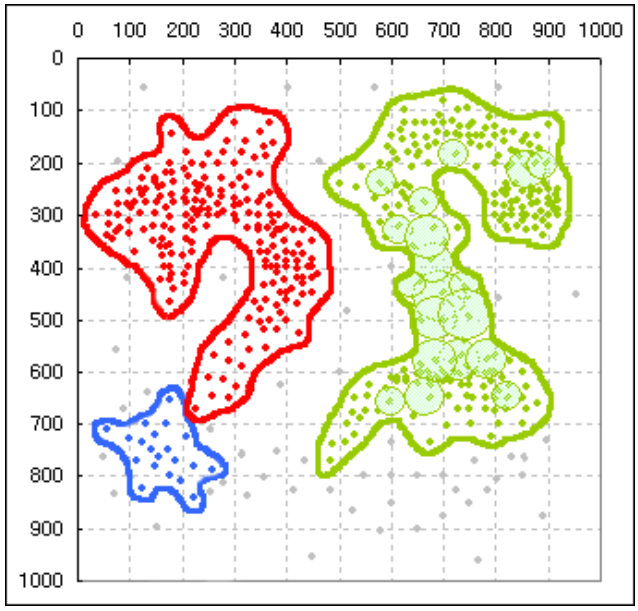


[그림 4.3] DBSCAN 수행 결과 (Eps=50, MinPts=5)

그림 4.4는 가중치를 적용한 클러스터링 방법인 DBSCAN-W의 수행 결과를 나타낸다. 그림의 아래 부분에 보이는 다양한 크기의 원들은 가중치에 의해 데이터 영역이 생성된 결과이다. 이렇게 생성된 영역들은 클러스터링 작업을 수행할 때에 밀도에 반영되므로, 데이터를 위치 속성만 갖는 점으로 다루는 DBSCAN과는 다른 결과를 유도할 수 있다. 실제로 영역을 고려한 클러스터링을 수행한 결과, 그림 4.3에서는 잡음으로 처리되었던 점들이 그림 4.4에서는 가까운 클러스터에 포함되어 새로운 클러스터를 생성하는 것을 볼 수 있다. 그림 4.4의 (a)는 weight1을 가중치로 사용한 경우로 그림의 하단을 중심으로 가중치가 높은 데이터가 분포하며, 클러스터링 결과 좌측 하단과 우측 하단에 분리되어 있던 두개의 클러스터가 인접한 점들을



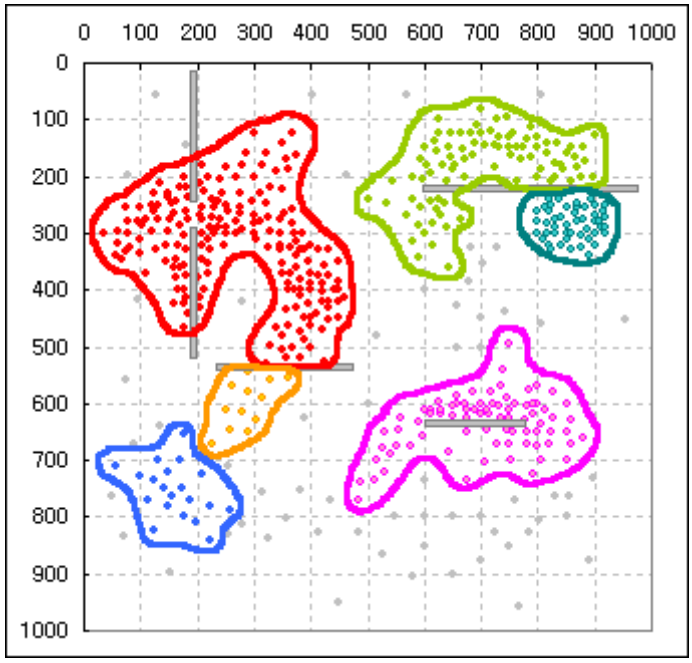
(a) weight1을 가중치로 사용한 DBSCAN-W 수행 결과



(b) weight2를 가중치로 사용한 DBSCAN-W 수행 결과

[그림 4.4] DBSCAN-W 수행 결과 (Eps=50, MinPts=5)

포함하는 하나의 큰 클러스터로 통합되었다. (b)는 weight2를 가중치로 사용한 경우이며, 우측 상단과 우측 하단의 두개의 클러스터가 인접한 점들을 포함하는 세로로 긴 형태의 하나의 큰 클러스터를 형성하였다.



[그림 4.5] COD-DBSCAN 수행 결과 (Eps=50, MinPts=5)

그림 4.5는 장애물을 고려한 클러스터링 방법인 COD-DBSCAN의 수행 결과이다. 앞의 두 방법에서는 장애물의 존재가 고려되지 않으므로 장애물 주위에 밀집한 데이터들이 하나의 클러스터로 포함되었다. 그러나 COD-DBSCAN 알고리즘에서는 데이터 사이에 장애물이 존재하는 경우, 장애물을 우회하는 거리를 계산함으로써 장애물로 인해 발생하는 소규모의 지역적인 분리를 처리할 수 있다. 그림 4.5의 우측 상단

의 클러스터는 장애물에 의해 클러스터가 두개로 분리된 것을 보여준다. 또한 좌측 상단에서와 같이 클러스터가 장애물에 의해 완전히 분리되지 않고 연결 통로가 존재하거나, 우측 하단에서와 같이 장애물이 클러스터의 내부에 포함되어있는 경우에는 장애물의 존재가 오브젝트간 거리에 큰 영향을 주지 않으므로 클러스터를 분리하지 않는다. DBSCAN과 동일한 조건으로 COD-DBSCAN을 수행한 결과 그림 4.3의 4개의 클러스터가 그림 4.5에서는 지역적 분리를 유발하는 장애물을 기준으로 나누어져 전체적으로는 6개의 클러스터를 생성하는 것을 볼 수 있다.

동일한 데이터를 대상으로 동일한 입력 파라미터를 사용하여 세 가지의 클러스터링 알고리즘을 수행시킨 결과, 밀도 기반의 클러스터링 알고리즘의 장점인 다양한 모양의 클러스터를 구분해 내면서 동시에 기준 이하의 데이터는 잡음으로 처리하는 결과가 공통적으로 나타나는 것을 볼 수 있다. 또한 응용 목적에 따라 특정 속성에 가중치를 부여하는 경우에는 DBSCAN-W 알고리즘을 통해 클러스터의 조건을 유연하게 조절하여 목적에 적합한 결과를 도출할 수 있으며, 데이터가 분포한 영역 내에 존재하는 장애물을 고려하기 위해서는 COD-DBSCAN을 적용하는 것이 효과적임을 알 수 있다.

IV. 결론 및 향후과제

지식 탐사 프로세스의 핵심적인 역할을 담당하는 데이터 마이닝 단계에서는 여러 가지 목적에 따라 알고리즘을 선택하여 사용한다. 최근 통계, 비즈니스, 전자상거래, 의학, 생물학 등의 분야에서 데이터 마이닝 기술이 적극적으로 활용되고 있으며 이를 위해 다양한 알고리즘들이 계속해서 연구 개발되고 있다.

본 논문에서는 다양한 모양과 크기의 클러스터를 구분해내고 기준 이하의 데이터는 잡음으로 처리할 수 있는 장점을 갖는 밀도 기반의 클러스터링 알고리즘을 기반으로 하여, 클러스터링의 응용 목적에 따라 특정 속성에 가중치를 부여하는 방법과 장애물이 존재하는 경우 장애물을 우회하는 새로운 오브젝트 간 거리 함수를 적용하는 방법을 제안하였다. 이 두 방법은 목적에 따라 보다 적절한 클러스터링 결과를 유도하고 실제 데이터에의 적합성을 높이기 위한 것으로, 실험을 통해 두 가지의 확장 알고리즘이 기존의 밀도기반 알고리즘에서 찾아내지 못한 새로운 형태의 클러스터링 결과를 도출하는 것을 보였다.

본 논문의 결과를 기반으로 각각의 응용 분야에 따라 오브젝트의 가중치를 적절한 형태의 영역으로 표현하는 방법, 다양한 형태의 장애물을 처리하는 방법, 가시 그래프의 생성 범위 및 생성 시기 조절을 통한 성능의 향상, 다양한 밀도를 갖는 클러스터의 구분 방법 등에 대한 연구가 필요할 것으로 보인다.

참고 문헌

[1] Michael J. A Berry, and Gorden Linoff, “Data Mining Techniques : For Marketing, Sales, and Customer Support”, John Wiley & Sons, Inc., 1997.

[2] Raymond T. Ng, Jiawei Han, "Efficient and Effective Clustering Method for Spatial Data Mining", In Proc. of the VLDB Conference, Santiago, Chile, 20th Int, pp. 144-155, September 1994.

[3] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "BIRCH : An Efficient Data Clustering Method for Very Large Databases", In Proc. of the ACM SIGMOD Conference on Management of Data, Montreal, Canada, pp. 103-114, June 1996.

[4] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, "CURE : An Efficient Clustering Algorithm for Large Databases", In Proc. of the ACM SIGMOD Conference on Management of Data, Seattle, Washinton, USA, pp. 73-84, May 1998.

[5] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", In Proc. of ACM SIGMOD 3rd International Conference on Knowledge Discovery and Data Mining, pp. 226-231, AAAI Press, 1996.

[6] Mihael Ankerst, Markus M. Breuning, Hans-Peter Kriegel, and Jorg Sander, “OPTICS: Ordering Points To Identify the Clustering Structure," In proc. of ACM SIGMOD International Conference on Management of Data, Philadephia, Pennsylvania, USA, pp. 49-60, June 1999.

[7] W.Wang, J.Yang, and R.Muntz, "STING :A statistical information grid approach to spatial data mining", In Proc. 1997 Int. conf. Very Large Data Bases(VLDB'97),Athens,

Greece, pp.186-195, Aug.1997.

[8] Jiawei Han, Micheline Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann publishers,2001.

[9] Samet, Hanan, “Spatial Data Models and Query Processing. In Modern Database Systems : The Object Model, Interoperability, and Beyond”, Addison Wesley / ACM Press, Reading, MA, 1994.

[10] Fayad, Usama M., “Advances in Knowledge Discovery in DataBases”, AAAI Press / MIT Press, Menlo Park, CA, 1996.

[11] Vladimir Estivill-Castro and Ickjai Lee, “AUTOCLUST : Automatic Clustering via Boundary Extraction for Mining Massive Point-Data Sets”, International Conference on Geocomputation, 2000.

[12] M.deBerg et al, “Computational Geometry in C”, Cambridge University Press, 1994.

[13] Michael J. Laszlo, “Computational Geometry and Computer Graphics in C++”, Prentice Hall, 1996.

[14] 이상호, 박승수, “C언어에 의한 자료구조론”, 생능출판사, pp.199-202, 1995.

[15] Informix, Informix Universal Server Guide to SQL: Tutorial Version 9.1, Informix Press, 1997.

[16] Informix, Informix Universal Server Guide to SQL: Reference, Informix Press, 1997.

[17] Informix, Extending Informix Universal Server: Data Types Version 9.1, Informix Press, 1997.

[18] Informix, Informix Spatial Datablade Module: User's Guide, Informix Press, 1997.

[19] Informix, Informix Datablade API Programmer’s Manual, Informix Press, 1997.

[20] Informix, Developing Applications Using Informix-ESQL/C, Informix Press, 1997.

[21] Thomas Brinkhoff, Hans Peter Kriegel, Bernhard Seeger, “Efficient Processing of Spatial Joins Using R-trees”, Proceedings of the ACM, pp.237-246, 1993.

[22] Anthony K. H. Tung, Jean Hou, Jiawei Han, “Spatial Clustering in the Presence of Obstacles”, Proc. 2001 International Conference on Data Engineering, 2001.

[23] 김호숙, 임현숙, 용환승, “가중치를 고려한 밀도 기반의 공간 클러스터링 알고리즘”, Korean Database Conference 학술발표 논문집 2001년 6월 17권 2호, pp.109-113, 2001.

ABSTRACT

A Density-based Spatial Clustering Algorithm

Considering Weight and Obstructed Distance

Department of Computer Science & Engineering

Ewha Institute of Science and Technology

Lim, Hyun Sook

Spatial data mining is a process to discover interesting relationships and characteristics that exist implicitly in a spatial database. Clustering techniques work well with a large amount of data and various types of data, so that it is easy to apply clustering techniques to spatial data. The density-based clustering algorithms are effective especially at handling the locality.

DBSCAN is the locality-based algorithm, relying on a density-based notion of clustering. The density-based notion of clustering states that within each cluster, the density of the points is significantly higher than the density of points outside the cluster. The algorithm can handle the issue of noise and is successful in discovering arbitrary shaped clusters on a large amount of spatial data. But it restricts the attributes that can affect clustering results as a location and cannot manage obstacles exist in real world such as rivers, lakes and highways.

In this thesis, we present two new spatial clustering algorithms, called DBSCAN-W and COD-DBSCAN. DBSCAN-W algorithm assigns regions to objects relying on the weights and the regions have influence on density. COD-DBSCAN generates a distance index for the obstructed distance and uses it instead of the direct Euclidian distance when there exist one or more obstacles between two objects. We conduct various performance studies to show that DBSCAN-W and COD-DBSCAN produce different clustering results from DBSCAN depend on the purpose of applications.