장애물을 고려한 밀도 기반의 공간 클러스터링 기법

임현숙[†] · 김호숙[††] · 용환승[†††] · 이상호[††††] · 박승수[††††]

요 약

공간 마이닝에서 클러스터링은 오브젝트간의 거리나 연결 상태, 또는 공간상에서의 상대적인 밀도를 기반으로 서로 비슷한 오브젝트들을 하나의 그룹으로 묶는 과정이다. 실세계에서 공간 상에 분포하는 강이나 호수, 고속도로와 같은 장애물들은 클러스터링의 결과에 영향을 줄 수 있다. 본 논문은 장애물을 고려한 오브젝트 사이의 거리를 정의하고, 이를 이용하여 공간 오브젝트들을 밀도를 기반으로 클러스터링 하면서 동시에 공간상에 존재하는 장애물을 고려하는 새로운 공간 클러스터링 알고리즘(DBSCAN-O)을 제안한다. 또한 실험을 통해 DBSCAN-O가 기존의 밀도 기반 알고리즘인 DBSCAN에서 찾아내지 못한 새로운 형태의 클러스터링 결과를 도출하는 것을 보인다.

Density Based Spatial Clustering Method considering Obstruction

Hyun-Sook Lim[†], Ho-Sook Kim[††], Hwan-Seung Yong[†††],
Sang-Ho Lee[††††] and Seung-Soo Park[††††]

ABSTRACT

Clustering in spatial mining is to group similar objects based on their distance, connectivity or their relative density in space. In the real world, there exist many physical objects such as rivers, lakes and highways, and their presence may affect the result of clustering. In this paper, we define distance to handle obstacles, and using that we propose the density based clustering algorithm called DBSCAN-O to handle obstacles. We show that DBSCAN-O produce different clustering results from previous density based clustering algorithm DBSCAN by our experiment result.

Key words: 데이터 마이닝, 공간 클러스터링, 장애물

1. 서 론

클러스터링이란 주어진 오브젝트 집합을 서로 유사성을 가지는 몇 개의 클러스터로 분할해 나가는 과정으로, 하나의 클러스터에 속하는 오브젝트들 사이에는 서로 다른 클러스터 내의 오브젝트들과는 구분되는 유사성을 갖게 된다[1]. 데이터 마이닝에서 클러스터링 방법은 기존의 통계, 기계 학습, 패턴인식에서 쓰이던 방법에 부가적으로 데이터베이스 지향적인 제약 사항들(제한된 메모리 양, I/O 시간 최소화 등)을 첨가 시킨 것으로서, 최근의 멀티미디어 데이터와 같이 혼합되고 다양한 다차원 데이터를 효율적으로 분류해 나가기 위한 방안으로 연구되고 있다. 클러스터링 방법은 크게 분할 방법[2]과 계층적 방법[3,4], 밀도 기반 방법[5,6], 격자기반 방법[7]과 그래프 기반 방법[8]으로 나눌 수 있다.

클러스터링은 각각의 응용에 따라 적합한 기준에 의해 오브젝트들을 서로 다른 그룹으로 분류하는데 공간상의 오브젝트들을 클러스터링 할 때 그 분류 기준으로 Euclidian 거리가 많이 사용된다[2,5,8]. 그러나 실세계의 응용에서는 공간 오브젝트들과 함께

접수일 : 2002년 7월 25일, 완료일 : 2003년 1월 8일
[†] LG CNS 연구원
[††] 동의공업대학 컴퓨터정보계열 전임강사
[†††] 종신회원, 이화여자대학교 컴퓨터학과 부교수
[††††] 이화여자대학교 컴퓨터학과 교수

존재하는 장애물들로 인해 Euclidian 거리가 유효하지 않은 경우가 빈번하게 발생하게 된다. 다음의 예1은 공간 오브젝트들이 분포하는 지역 내에 장애물들이 존재하는 경우, 장애물을 우회하는 거리가 오브젝트들 사이의 실제 거리로 반영되어야 하는 예를 보여준다.

예 1. 한 백화점에서 셔틀 버스를 운행하기 위해 대상 지역에 속해있는 고객들을 클러스터링 하려고 한다. 이때 운행 지역은 고객의 지역적인 위치에 따른 밀집된 패턴을 기반으로 하며, 보다 효율적인 버스 운행을 위해 대상 영역 내에 존재하는 하천이나 공원 등과 같은 자연적, 인공적 장애물을 고려하여 노선을 결정해야 한다.

위와 같은 응용에서 클러스터링을 수행 할 때는 고객들의 지역적인 위치를 기준으로 클러스터링을 수행하되, 장애물로 가로막힌 경우에는 장애물을 우회하는 최단 경로를 찾아내어 그 거리를 실제 고객 사이의 거리로 사용해야 한다. 또한 생성되는 클러스터링 결과는 공간상에 분포한 오브젝트에 대하여 다양한 모양과 크기의 클러스터를 구분해 내어야 하고, 적절한 기준 이하의 오브젝트는 클러스터에 포함되지 않도록 잡음으로 처리되어야 하며, 공간 오브젝트와 장애물의 수와 분포에 대한 특성에 따라 효율적으로 수행되어야 한다. 이러한 목표를 달성하기 위하여 본 논문에서는 새로운 공간 클러스터링 알고리즘 DBSCAN-O를 제안하고 이를 구현하였다. DBSCAN-O는 밀도 기반의 클러스터링 알고리즘인 DBSCAN을 확장한 것이다. DBSCAN에서는 대상물 사이의 거리로 Euclidian 거리를 사용한 반면, DBSCAN-O은 영역 내에 존재하는 장애물의 수와 분포의 특성을 고려하여 오브젝트 사이의 거리 계산 시 장애물을 우회하는 거리를 적용한다.

논문의 구성은 다음과 같다. 2장에서는 밀도를 기반으로 하는 공간 클러스터링 알고리즘인 DBSCAN과 클러스터링 수행 시 장애물을 고려하는 기존의 연구들을 소개하고, 3장에서는 DBSCAN-O에서 장애물을 고려하는 방법을 제안하며, 4장에서는 제안된 방법에 대한 구현 결과를 보여준다. 마지막으로 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 밀도 기반 공간 클러스터링 알고리즘: DBSCAN

DBSCAN(Density Based Spatial Clustering of Applications with Noise)[5]은 밀도를 기반으로 하는 클러스터링 알고리즘으로, 잡음을 포함한 공간 데이터를 다루는데 적합하며 다양한 모양과 크기의 클러스터를 구분할 수 있는 장점을 갖는다. DBSCAN은 대상 오브젝트들을 1번씩 읽으면서 기준 오브젝트로 부터 사용자가 설정한 반경 Eps내에 위치하는 이웃 오브젝트들을 찾아서 하나의 클러스터로 구분하는 과정을 수행한다. 그러므로 오브젝트의 전체 수가 n일 때 이웃을 찾아내는 겹침 연산을 log n 시간에 수행할 수 있도록 지원하는 R*-tree 인덱스를 이용하는 경우, DBSCAN 알고리즘에서 전체 클러스터링을 수행하는 시간 복잡도는 $O(n*\log n)$이다.

2.2. 장애물을 고려한 클러스터링 알고리즘 : COD-CLARANS

공간상의 오브젝트에 대한 클러스터링 수행 시 장애물을 고려하는 연구로 COD-CLARANS (Clustering with Obstructed Distance-CLARANS)[9]가 있다. COD-CLARANS는 분할 방법의 클러스터링 알고리즘인 CLARANS를 기반으로 하며 두 오브젝트 또는 오브젝트와 클러스터의 중심 사이에 장애물이 존재할 경우 장애물을 우회하는 최단 경로의 길이를 거리로 사용하는 새로운 거리 함수를 적용함으로써 장애물로 인한 거리 문제를 처리한다. COD-CLARANS에서는 장애물을 고려한 클러스터링 문제를 다음과 같이 정의한다.

• 2차원 공간 R에 n개의 오브젝트들의 집합 $P=p_1,p_2,\ldots,p_n$와 서로 겹쳐지지 않는 m개의 장애물들의 집합 $O=o_1,o_2,\ldots,o_m$가 있다고 할 때, 두 오브젝트 p_j,p_k 사이의 장애물을 고려하지 않는 Euclidian 거리는 $d(p_j,p_k)$로, 장애물을 고려한 거리는 $d'(p_j,p_k)$로 표시한다. 클러스터 Cl_i의 중심을 c_i라고 할 때, 장애물을 고려한 클러스터링 문제는 다음의 함수 E를 최소화 하도록 P를 k개의 클러스터 $Cl_1,\ldots,Cl_k$로 분할한다.

$$E = \sum_{i=1}^{k} \sum_{p \in C_{l_i}}^{k} (d'(p, c_i))^2$$

COD-CLARANS는 기반이 되는 알고리즘인 CLARANS가 갖는 높은 시간 복잡도의 제약 사항을 극복하기 위하여 Micro-cluster 생성, 공간 조인 인덱스 생성 등의 다양한 전처리 과정과 가지치기(pruning)방법을 필요로 하는 단점이 있다. 또한 COD-CLARANS에 의해 생성되는 클러스터링 결과는 잡음을 처리하지 못하고 다양한 모양의 클러스터를 발견하지 못하는 단점을 갖는다.

2.3 가시 그래프

가시 그래프는 장애물로 가로막힌 공간에서 로봇의 이동 경로를 결정하는 로봇 모션 플래닝에 많이 사용되는 알고리즘이다[11]. 장애물들의 꼭지점의 집합을 V라 하고, 서로 보이는 두개의 꼭지점을 잇는 선의 집합을 E라고 할 때 가시 그래프 VG는 다음과 같이 정의된다.

VG = (V, E)

이때 영역 R 내에 위치한 두 오브젝트 p, q가 서로 보이는지를 효과적으로 판단할 수 있는 자료 구조로서, BSP 트리를 이용한다. BSP(Binary-Space-Partitioning) 트리는 p, q를 연결하는 직선이 장애물과 겹치지 않을 때 두 오브젝트 p, q를 서로 보인다고 정의한다[12].

그림 1은 두 장애물에 대한 가시 그래프가 존재할 때, 추가되는 오브젝트들에 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 ∪ {p, q}이고 E′ = E ∪ {e_1, e_2, e_3, e_4, e_5}일 때,

VG′ = (V′, E′)

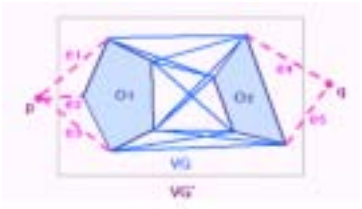
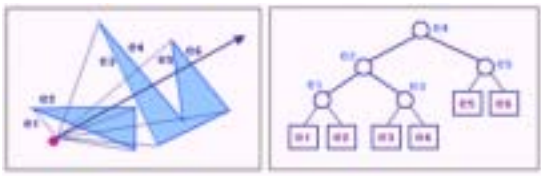


그림 1. 가시 그래프

가시 그래프를 생성하기 위한 기본적 알고리즘은 장애물을 구성하는 꼭지점들의 모든 쌍에 대해 가시성을 검사하는 것으로 꼭지점의 수가 n일 때 $O(n^3)$의 시간 복잡도를 갖는다. 이때 장애물을 구성하는 선들을 회전 방향으로 정렬하여 순차적으로 가시성을 검사하면 시간복잡도를 $O(n^2 * \log n)$으로 줄일 수 있다. 그림 2(a)는 기준 오브젝트를 중심으로 주위의 장애물들을 시계방향으로 정렬한 예를 보여주며, 정렬된 선들은 그림 2(b)와 같이 저장된다.



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

그림 2. 시계 방향으로 정렬된 선들

3. DBSCAN-O의 설계

본 논문에서는 밀도 기반의 공간 클러스터링 알고리즘이 장애물을 고려하기 위해서 기존의 DBSCAN을 확장한 DBSCAN-O (DBSCAN considering Obstruction)를 제안한다. DBSCAN에서는 클러스터링의 대상이 되는 데이터 사이의 거리로 Euclidian 거리를 사용한다. 반면, DBSCAN-O은 영역 내에 장애물이 존재하는 경우 장애물을 우회하는 그래프 상의 최단 거리를 계산하여 이를 오브젝트간 거리로 적용한다.

그림 3은 DBSCAN과 DBSCAN-O에서 오브젝트 p를 중심으로 기준반경(Eps) 내에 존재하는 이웃을 찾기 위해서 두 오브젝트 p와 q 사이의 거리를 계산하는 방법을 나타낸다. 그림 3(a)는 DBSCAN 알고리즘에 의해 클러스터링을 수행하는 경우로서, 이때는

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

그림 3. 장애물을 고려한 DBSCAN 알고리즘의 확장

오브젝트 사이에 장애물의 존재를 고려하지 않는다. 그림 3(b)는 기준 반경으로 설정되는 영역 내에 장애물이 존재하는 경우에 오브젝트 p를 중심으로 이웃을 결정할 때는 기존의 거리 계산 방법과는 다르게 장애물을 고려 하여야 한다는 것을 보인다. 그림3(c)는 두 오브젝트 p와 q의 실제 거리를 p와 q를 잇는 선분의 길이가 아니라 장애물 O_1을 우회하는 경로인 a, b, c의 길이의 합으로 결정 되는 것을 보인다. 즉 a+b+c > Eps이므로 오브젝트 q는 오브젝트 p의 이웃으로 포함되지 못한다.

3.1 가시 그래프 적용의 3가지 방안

두 오브젝트 사이의 거리 계산 시 장애물을 고려하기 위하여 본 논문에서는 가시 그래프를 이용하였다. 이때 해당 영역내의 장애물의 특성과 응용 프로그램에서의 밀도 결정을 위한 입력 변수의 변경 빈도 등에 따라서 가시 그래프의 생성 방법은 그 시기와 범위를 달리하는 세가지로 나누어 적용 할 수 있다.

첫째, 전처리 과정에서 전체 데이터에 대해 생성하는 방법이다. 밀도 기반 클러스터링은 데이터의 지역적인 밀도에 기반 함으로 전역적인 가시 그래프의 생성은 불필요한 계산을 포함하게 되어 비효율적일 수 있다. 그러나 데이터에 대한 사전 지식이 없을 경우, 또는 밀도를 결정하기 위한 입력 변수를 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, 클러스터링 수행 중에 한 오브젝트를 기준으로 기준 반경 내에 존재하는 이웃을 결정하는 단계에서 기준 반경 내에 존재하는 오브젝트와 장애물들에 대해서만 부분적으로 가시 그래프를 생성하는 방법이다. 이것은 기준 영역의 중심점이 바뀌거나 기준 반경을 변경할 때 마다 가시 그래프를 다시 생성해야 한다는 점에서 중복 계산의 위험이 있으나, 기준 반경의 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 클러스터링 수행 시에 부분적으로 가시 그래프를 생성하는 것이 효율적이다.

셋째는 위의 두 방법을 혼합한 것으로, 전처리 과정에서 전체 장애물들의 가시 그래프를 생성하고 클러스터링 수행 시에 기준 반경 내의 오브젝트들만을 추가하여 가시 그래프를 확장하는 방법이다. 이것은 가시 그래프의 확장성에 기반을 둔 방법이며, 전체 장애물의 가시 그래프를 생성함으로써 중복 계산을 방지하고, 생성된 가시 그래프의 부분적인 확장을 통해 불필요한 계산을 줄일 수 있다.

COD-CRARANS에서도 장애물을 고려한 거리 계산을 위해 BSP 트리 생성과 가시 그래프 생성을 수행한다. 그러나 바탕이 되는 CLARANS 알고리즘은 대상이 되는 오브젝트의 수에 제곱에 가까운 시간 복잡도를 갖기 때문에 $O(n \log n)$ 의 시간 복잡도를 갖는 DBSCAN을 기반으로 하는 DBSCAN-O에 비해 효율성이 떨어진다. 또한 COD-CRARANS에 의해 생성되는 클러스터링 결과는 볼록 다각형의 모양으로 제한되고, 대상 오브젝트 집합이 포함하고 있는 잡음을 발견하지 못하는 등의 단점을 갖는다.

3.2 DBSCAN-O의 수행 단계

DBSCAN-O는 클러스터링 수행 시 영역내의 데이터와 함께 존재하는 다양한 장애물을 처리하기 위해서 오브젝트 간 거리를 계산 할 때에 Euclidian 거리 대신 장애물을 고려한 거리를 적용함으로써 장애물을 고려한 클러스터링을 가능하게 한다. 이때 참조하는 거리의 생성은 3.1절에서와 같이 가시 그래프의 적용 방안에 따라 3가지 단계에서 수행될 수 있다. 첫번째 방법으로 전처리 과정에서 모든 데이터와 오브젝트에 대해 가시 그래프를 생성하는 경우에는 생성한 가시 그래프에 Dijkstras 알고리즘[10]을 적용하여 가시 그래프 상의 최단 거리를 거리 인덱스로 저장하는 전처리 과정을 수행하고 이를 이용하여 DBSCAN 알고리즘을 수행한다. 두번째 방법으로는 전처리 과정 없이 DBSCAN 알고리즘을 이용하여 클러스터링을 수행 하면서 기준 반경과 겹치는 오브젝트와 장애물들에 대해서만 부분적으로 가시 그래프를 생성하여 장애물을 고려한 거리를 적용시키는 방법이다. 기준 반경의 크기가 작고 장애물의 수가 작아서, 해당 반경 내에 포함되는 장애물의 수가 적은 경우 가장 효율적으로 수행될 수 있는 방법이다. 세번째는 전처리 과정에서 전체 장애물들의 가시 그래프를 생성한 후, DBSCAN 알고리즘을 이용하여 클러스터링을 수행 하는 단계에서 반경 내의 오브젝

트들을 추가하여 가시 그래프를 확장하는 방법으로 장애물을 고려하는 방법이다.

이때 DBSCAN-O의 수행 단계를 정리하면 알고리즘 1과 같다.

```
Input : 오브젝트 집합, 장애물집합, 기준반경, 최소 이웃 수, VG적용방법
Output : 클러스터링 결과

오브젝트 집합, 장애물집합, 기준반경, 최소 이웃 수, VG 적용방법을 입력한다.
If (VG적용방법 ==1)
{ 모든 오브젝트와 장애물을 대상으로 가시 그래프를 생성한다.
Dijkstras 알고리즘을 이용하여 가시 그래프 상의 최단 거리를 생성한다.
거리인덱스를 저장한다.
거리인덱스를 이용하여 DBSCAN 알고리즘을 수행한다.
}
else if (VG적용방법 ==2)
{ DBSCAN알고리즘을 수행하면서
if (오브젝트를 중심으로 하는 기준 반경 내에 포함되는 장애물이 있으면)
기준 반경 내에 존재하는 모든 오브젝트와 장애물을 대상으로 가시 그래프를 생성하고
Dijkstras 알고리즘을 이용하여 가시 그래프 상의 최단 거리를 생성하여
장애물을 고려한 거리가 기준 반경보다 작은 오브젝트를 이웃으로 결정한다.
Else Euclidian 거리를 이용하여 이웃을 결정한다.
}
else if (VG적용방법 ==3)
{ 모든 장애물을 대상으로 가시 그래프를 생성한다.
DBSCAN알고리즘을 수행하면서
if (오브젝트를 중심으로 하는 기준 반경 내에 포함되는 장애물이 있으면)
기준 반경 내에 존재하는 오브젝트를 전처리 과정에서 생성한 가시 그래프에 추가한다.
Dijkstras 알고리즘을 이용하여 가시 그래프 상의 최단 거리를 생성하여
장애물을 고려한 거리가 기준 반경보다 작은 오브젝트를 이웃으로 결정한다.
Else Euclidian 거리를 이용하여 이웃을 결정한다.
}
클러스터링 결과를 저장한다.
```

〔알고리즘 1〕 DBSCAN-O의 수행 단계

4. 실 험

본 장에서는 3장에서 제안한 DBSCAN-O를 이용하여 클러스터링을 수행한 결과를 보인다. 실험은 Sun사의 Solaris 2.6 상에서 객체-관계 DBMS인 Informix Universal Server를 사용하여 수행되었다. 이때 사용된 공간 데이터베이스 엔진인 Informix Spatial Datablade Module은 겹침 연산 등을 포함한 2차원 연산을 지원하는 9개의 데이터 타입과 여러 가지 공간 함수를 제공하며 공간 색인을 위해 R-Tree를 지원한다[13].

4.1 테이블 정의

실험을 위하여 작성된 인위적인 데이터베이스는 오브젝트와 장애물 테이블로 구성되며, 각각의 테이블에 대한 정의는 다음과 같다.

```
Create table Object
( Object_ID     smallint not null,  /* 오브젝트 ID */
Location        sp2Pnt,             /* 오브젝트의 공간적 위치 값 (x, y) */
Cluster_ID      smallint            /* 클러스터링 결과 */
);

Create table Obstacle
( Obstacle_ID   smallint not null,  /* 장애물 ID */
Region          sp2Box              /* 장애물의 영역 (x1, y1, x2, y2) */
);
```

그림 4. 오브젝트와 장애물에 대한 테이블 정의

Object 테이블은 클러스터링의 대상이 되는 공간 오브젝트들의 정보를 저장하는 테이블이다. 구성 필드들을 살펴보면 각 데이터마다 고유한 값을 갖는 Object_ID가 기본키로 설정되어 있고, 데이터의 위치인 Location은 2차원 좌표값 (x, y)로 표현되는 점 타입(sp2Pnt)으로 정의된다. Cluster_ID는 클러스터링 수행 결과 결정된 클러스터의 번호를 저장한다. Obstacle 테이블은 데이터 영역 내에 분포하는 장애물들의 정보를 저장하는 테이블로서, Obstacle_ID와 Region 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 사각형 타입(sp2Box)으로 정의된다.

4.2 실험 데이터의 분포

Object 테이블에 포함된 공간 오브젝트들의 위치 정보는 (x, y) 형태의 2차원 좌표로 표시되며, 모두 (0, 0)~(1000, 1000)의 범위의 값을 갖는 500개의 오브젝트가 밀집하거나 넓게 흩어져 있는 불규칙한 형태의 분포로 구성된다. 또한 영역 내에는 직사각형

형태를 갖는 5개의 장애물이 존재한다. 오브젝트들과 장애물들의 분포는 아래 그림 5와 같다.

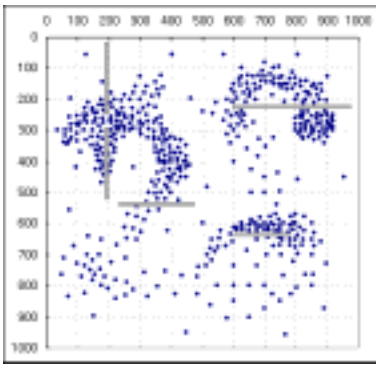
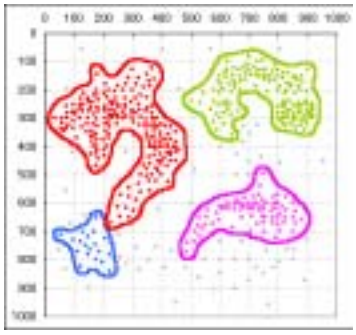


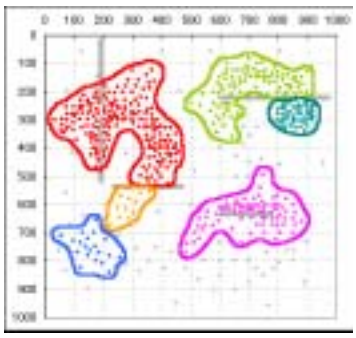
그림 5. 오브젝트와 장애물들의 분포

4.3 클러스터링 결과 비교

그림 6은 동일한 데이터를 대상으로 동일한 입력변수를 사용하여 DBSCAN과 DBSCAN-O를 각각 적용하였을 때의 클러스터링 결과를 보여준다. 이때 밀도의 기준이 되는 두 가지 입력 변수인 기준반경과 최소 이웃 수는 각각 50과 5로 설정하였다. 그림 6(a)는 장애물을 고려하지 않고 대상 데이터만을 사용하여 DBSCAN을 수행한 결과로 4개의 클러스터가 생성된 것을 볼 수 있다. 또한 클러스터에 포함되지 않은 데이터들이 모두 잡음으로 처리된 것을 볼 수 있다. 그림 6(b)는 공간 상에 위치한 장애물을 고려한 DBSCAN-O의 수행 결과이다. DBSCAN에서는 장애물의 존재가 고려되지 않으므로 장애물 주위에 밀집한 데이터들이 하나의 클러스터로 포함되었다. 그러나 DBSCAN-O에서는 데이터 사이에 장애물이 존재하는 경우, 장애물을 우회하는 거리를 계산함으로써 장애물로 인해 발생하는 소규모의 지역적인 분리를 처리할 수 있다. 즉 그림 6(a)에서는 하나의 클러스터인 우측 상단 부분의 오브젝트들이 그림 6(b)에서는 장애물에 의해 두개의 클러스터로 분리된 것을 볼 수 있다. 장애물의 존재가 항상 클러스터를 분리 시키는 것은 아닌데 예를 들면 좌측 상단에서와 같이 클러스터가 장애물에 의해 완전히 분리되지 않고 연결 통로가 존재하거나, 우측 하단에서와 같이 장애물이 클러스터의 내부에 포함되어 있는 경우에는 장애물을 우회하는 거리상에 오브젝트들과 이웃관계를 형성하므로 서로 다른 클러스터로 분리되지



(a) DBSCAN 결과



(b) DBSCAN-O 결과

그림 6. DBSCAN과 DBSCAN-O의 클러스터링 결과

않는 결과를 볼 수 있다. 즉 DBSCAN은 4개의 클러스터를 생성하는 반면, DBSCAN-O은 6개의 클러스터를 생성하였다.

실험의 결과 장애물을 고려하지 않은 DBSCAN에 의해서 생성된 하나의 클러스터에 DBSCAN-O를 적용하였을 때, 장애물의 크기와 장애물 주변의 오브젝트들의 분포에 따라서 클러스터의 분리 결과가 달라지는 것을 볼 수 있다. 그림 7은 하나의 클러스터

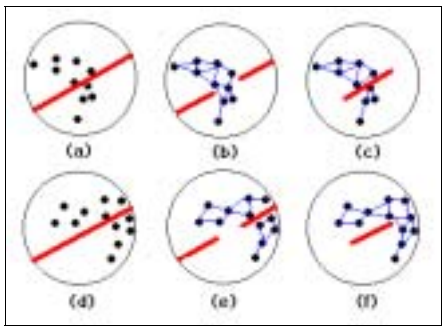


그림 7. 장애물과 오브젝트의 위치와 분포에 따른 클러스터의 분리

내에 존재하는 오브젝트와 장애물의 위치와 분포에 따라 클러스터 분리 결과가 달라지는 것을 보여준다. 그림 7의 (a)(d)와 같이 클러스터를 관통하는 장애물이 존재하는 경우에는 항상 하나의 클러스터가 2개로 분리되는 결과를 보여준다. 그러나 그림 7의 (b)(f)와 같이 오브젝트 사이에 장애물이 존재할 때도 클러스터가 분리되지 않는 경우가 발생하는 데, 이때는 클러스터내의 오브젝트 사이의 연결이 장애물에 의해 분리되지 않도록 장애물 사이에 통로가 존재하거나 장애물을 우회하는 연결 오브젝트들이 존재하는 경우이다. 그림 7의 (c)(e)의 경우는 장애물이 오브젝트들 사이에 포함되거나 통로가 존재하더라도 클러스터의 중심 오브젝트들이 장애물에 의해 가장자리 오브젝트로 바뀌면서 하나의 클러스터가 서로 다른 두개의 클러스터로 분리되는 결과를 보여준다. 즉 DBSCAN에서 클러스터의 정의를 밀집 연결된 오브젝트의 최대 집합으로 정의하므로, 비록 클러스터 내에 장애물이 존재하더라도 오브젝트 사이의 밀집 연결이 지속되는 경우에는 하나의 클러스터로 남아있게 된다.

동일한 데이터를 대상으로 동일한 입력 파라미터를 사용하여 DBSCAN과 DBSCAN-O 알고리즘을 수행시킨 결과, 두 알고리즘 모두 COD-CLARANS에서는 처리하지 못하는 불규칙하고 다양한 모양의 클러스터를 구분해 내면서 동시에 기준 이하의 데이터는 잡음으로 처리하는 우수한 결과를 볼 수 있다. 그러나 데이터가 분포한 영역 내에 장애물이 존재하는 경우, DBSCAN은 장애물이 실제 오브젝트간 거리에 미치는 영향을 전혀 반영하지 못하는데 반해 DBSCAN-O은 장애물의 크기 및 위치에 따라 다양한 형태의 클러스터들을 찾아낸 것을 볼 수 있다. 공간 마이닝의 응용 시스템에 따라서는 대상 오브젝트의 분포를 기반으로 하지만 오브젝트와 함께 존재하는 물리적 환경 요소의 영향을 고려해야 하는 경우가 있다. 이때 본 논문에서 제시한 DBSCAN-O를 사용하여 기존의 DBSCAN과는 다른 실제 응용에 보다 적합한 새로운 클러스터링 결과를 도출할 수 있다.

5. 결론 및 향후 연구과제

지식탐사 프로세스의 핵심적인 역할을 담당하는 데이터 마이닝 단계에서는 여러 가지 목적에 따라 알고리즘을 선택하여 사용한다. 최근 통계, 비즈니스, 전자 상거래, 의학, 생물학 등의 분야에서 데이터 마이닝 기술이 적극적으로 활용되고 있으며 이를 위해 다양한 알고리즘들이 계속해서 연구 개발되고 있다. 본 논문은 대상 오브젝트들의 공간적인 위치 속성을 기반으로 하는 공간 데이터 마이닝을 수행할 때에 데이터의 밀도를 고려하면서 실제 응용에서 빈번하게 나타나는 장애물로 인한 오브젝트간 거리 문제를 해결하는 알고리즘인 DBSCAN-O를 제안하였다. 또한 실험을 통해 기존의 DBSCAN 알고리즘에서 찾아내지 못한 새로운 형태의 클러스터링 결과를 도출하는 것을 보였다.

향후 장애물을 포함하는 실제 공간 데이터를 이용하여 제안한 DBSACN-O의 성능을 COD-CLARANS 등을 포함한 기존의 클러스터링 방법과 비교하는 연구가 필요할 것으로 보인다.

참 고 문 헌

[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] V.Estivill-Castro, and I.Lee, AUTOCLUST: Automatic Clustering via Boundary Extraction for Mining Massive Point-Data Sets, In proc. of the 5th International Conference on Geocomputation, 2000.

[9] Anthony K. H. Tung, Jean Hou, Jiawei Han, Spatial Clustering in the Presence of Obstacles, In proc. of International Conference on Data Engineering, 2001.

[10] 이상호, 박승수, C언어에 의한 자료구조론 , 생릉출판사, pp.199-202, 1995.

[11] M.de Berg, M.van Kreveld, M.Overmars, O. Schwarzkopf, Computational Geometry Algorithms and Applications, Springer, pp305-314, 1997.

[12] BSP Tree Frequently Asked Questions-http://reality.sgi.com/bspfaq/whole.shtml

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

임 현 숙

1999년 이화여자대학교 컴퓨터학과 학사
2002년 이화여자대학교 컴퓨터학과 공학석사
2002~현재 LG CNS 연구원

관심분야 : 데이터 마이닝, CRM

김 호 숙

1993년 이화여자대학교 전자계산학과 학사
1993년~1997년 삼성 SDS 전임연구원
1999년 이화여자대학교 컴퓨터학과 공학석사
1999년~현재 이화여자대학교 컴퓨터학과 박사과정
2001년~현재 동의공업대학 컴퓨터정보계열 전임강사
관심분야 : 공간 데이터베이스, 데이터 마이닝

용 환 승

1983년 서울대학교 컴퓨터학과 학사
1985년 서울대학교 컴퓨터학과 공학석사
1985년~1989년 한국 전자 통신연구소 연구원
1994년 서울대학교 컴퓨터학과 공학박사
1995년~현재 이화여자대학교 컴퓨터학과 부교수
관심분야 : 데이터베이스, 데이터 마이닝, 바이오 인포매틱스

이 상 호

1979년 서울대학교 계산통계학과 학사
1981년 한국과학기술원 전산학과 석사
1987년 한국과학기술원 전산학과 박사
1990년 일리노이대 전산학과 방문 교수
1983년~현재 이화여자대학교 컴퓨터학과 교수
관심분야 : 알고리즘 설계, 정보보호, 바이오 인포매틱스, 데이터마이닝

박 승 수

1974년 서울대학교 문리대학 수학전공 학사
1976년 한국과학기술원 전산학전공 석사
1981년 Computer Science, New York University. USA 석사
1988년 Computer Science, University of Texas. USA
1991년 3월~현재 이화여자대학교 컴퓨터학과 교수
관심분야 : 인공지능, 데이터마이닝, 바이오 인포매틱스

교 신 저 자

용 환 승 120-750 이화여자대학교 컴퓨터학과