論文
4-12-2

데이터 마이닝을 위한 고차원 클러스터링 기법에 관한 비교 분석 연구

A Comparison and Analysis on High-Dimensional Clustering Techniques for Data Mining

김홍일(Hong-IL Kim)[1] 이혜명(Hye-Myung Lee)[2]

요 약

데이터베이스의 많은 응용분야에서 대용량 고차원 데이터의 클러스터링을 요구하고 있다. 이에 따라 클러스터링 알고리즘에 대한 많은 연구가 이루어지고 있으나 기존의 알고리즘들은 "차원의 저주"에 기인하여 고차원 공간에서 효과적 및 효율적으로 수행하지 못하는 경향이 있다. 더욱이, 고차원 데이터는 상당한 양의 잡음 데이터를 포함하고 있으므로 알고리즘의 효과성 문제를 야기한다. 그러므로 고차원 데이터의 구조와 다양한 특성을 지원하는 적합한 클러스터링 알고리즘이 개발되어야 한다. 본 논문에서는 지금까지 연구된 고차원 클러스터링 기법을 조사한 후, 각 기법의 장단점과 적합한 응용 분야에 대한 비교 및 분석을 통하여 분류한다. 특히 본 논문에서는 최근의 연구를 통하여 개발한 점진적 프로젝션 기반의 클러스터링 알고리즘인 CLIP의 성능을 기존의 알고리즘과 비교 분석함으로써 그 효율성 및 효과성을 입증한다. 이러한 알리즘들의 소개 및 분류를 통하여 향후의 더욱 향상된 클러스터링 알고리즘 개발에 기반이 되고자 한다.

ABSTRACT

Many applications require the clustering of large amounts of high dimensional data. Most automated clustering techniques have been developed but they do not work effectively and/or efficiently on high dimensional (numerical) data, which is due to the so-called "curse of dimensionality". Moreover, the high dimensional data often contain a significant amount of noise, which causes additional ineffectiveness of algorithms. Therefore, it is necessary to look over the structure and various characteristics of high dimensional data and to develop algorithm that support clustering adapted to applications of the high dimensional database. In this paper, we investigate and classify the existing high dimensional clustering methods by analyzing the strength and weakness of each method for specific applications and comparing them. Especially, in terms of efficiency and effectiveness, we compare the traditional algorithms with CLIP which are developed by us. This study will contribute to develop more advanced algorithms than the current algorithms.

1) 정회원 : 대진대학교 컴퓨터공학과 조교수
2) 정회원 : 경문대학교 인터넷미디어정보과 조교수

논문접수 : 2003. 12. 3.
심사완료 : 2003. 12. 12.

1. 서 론

정보기술의 획기적인 발달로 인해 대용량의 데이터들이 데이터베이스에 수집되어 저장되고 있다. 이것은 전형적인 관계형 데이터베이스의 정형화된 데이터뿐 아니라 이미지, CAD, 지리 데이터 등과 같은 복잡한 멀티미디어 데이터에서 또한 그러하다. 데이터마이닝은 이와 같은 대용량 데이터베이스에 암시적으로 존재하는 흥미있는 관계나 특징의 탐사로서 데이터마이닝을 위한 다양한 기법들이 연구되고 있다. 그 중 클러스터링은 중요한 분석방법으로서 데이터 집합에 있는 본래의 분류나 구조를 쉽게 이해할 수 있도록 한다.

관계형 데이터베이스는 각 애트리뷰트가 데이터 집합의 차원에 해당하는 고차원 데이터베이스로 취급된다. 이때 클러스터링은 객체들의 애트리뷰트(차원) 값에 근거하여 객체들을 유사한 그룹으로 식별하는 기술적인 작업으로서 유사검색, 고객 세그먼테이션, 패턴 인식, 경향 분석 등의 데이터베이스 연구에서 폭 넓게 논의되고 있다.

고차원 데이터베이스에서 자동화된 클러스터링은 매우 중요한 문제이며 고차원 데이터를 적용할 수 있는 다양한 클러스터링 기법이 연구되고 있다. 데이터 집합에서 클러스터를 탐색하는 기본적인 방법으로는 분할(partitioning) 기반 기법, 밀도(density)기반 기법, 계층적(hierarchical) 기법으로 분류할 수 있다. 각각에 관한 대표적 알고리즘을 살펴보면, 분할기반 기법으로는 PAM[11], CLARA[11], CLARANS[9], 밀도기반 기법으로는 DBSCAN[5], OPTICS[6] 등이 있다. 계층적 클러스터링은 통상 트리 구조에 의하여 데이터베이스를 몇 단계의 분할로 분해한다. 이러한 계층적 알고리즘은 지식탐사에서 매우 효과적이지만, 트리를 생성하는 비용 때문에 대용량 데이터베이스에 대해서는 비현실적이다. 본 논문에서 계층적인 방법론은 특별히 언급하지 않았는데, 그것은 대부분의 접근방법들이 계층적 구조로 확장 가능하기 때문이다.

대부분의 접근방법들은 고차원 데이터의 클러스터링에 대해 적절히 설계되지 않았으며 따라서 앞서 기술한 알고리즘들의 성능도 차원이 증가함에 따라 급격하게 저하된다. 이와 같은 문제를 개선하기 위해 최적화된 클러스터링 기법들이 제안되었는데 요약정보 기반(condensation-based), 그리드-기반(grid-based) 등이 그것이다. 요약정보 기반의 알고리즘으로는 클러스터-특징 트리를 이용하는 BIRCH[15], 추가적인 통계정보를 사용하는 STING[7] 등이 있다. 그리고 클러스터링의 효율성을 증대하기 위해 일정한 그리드(grid)를 기반으로 하는 알고리즘으로는 DENCLUE[8], WaveCluster[14] 등이 있다.

그러나 위의 방법론에서도 고차원의 문제(curse of dimensionality)[2]는 여전히 결과적인 클러스터링의 정확성 측면에 심각한 영향을 준다. 이와 같이 많은 클러스터링 알고리즘들이 고차원 공간에서 효율적으로 수행하지 못하는 중요한 이유는 데이터 고유의 희소성 때문이다[3,10]. 즉 고차원 데이터의 응용에 있어서, 임의 데이터 점들은 적어도 일부 차원에서는 서로 떨어져 있는 점들이 존재하기 쉽다는 개념이다. 그러므로 데이터 점들이 서로 연관되어 있는 특정 차원에서 클러스터를 탐색하는 부분차원 기반의 클러스터링 기법이 제안되었다. 즉 클러스터 형성에 관련이 적은 차원들을 제거하여 데이터의 잡음을 감소시킨다는 개념으로서 이와 같은 클러스터링 기법으로는 CLIQUE[3], PROCLUS[4], CLIP[13] 등이 있다. 특히 [13]에서 제안한 CLIP은 데이터의 점진적인 프로젝션을 이용하여 고차원 공간에서 클러스터링의 효율성 문제를 해결하려 하였으며, 차원 전체뿐 아니라 부분차원에 존재하는 클러스터의 탐색을 목적으로 한다.

본 논문에서는 대표적인 클러스터링 알고리즘을 고찰하여 각각의 특징 및 장·단점을 비교하고 분석하였다. 특히 최근의 연구를 통하여 저자가 개발한 점진적 프로젝션 기반의 부분차원 클러스터링 알고리즘인 CLIP(**CL**ustering based on **I**ncremental **P**rojection)을 포함시켜서 다른 알고리즘들과 비교하도록 하였다. CLIP은 각 차원에 대해 프로젝션 선형변환을 이용하는 클러스터링 기법이다. 특히 CLIP은 전자상거래에서 고객데이터 분류를 목적으로 개발되었으며, 각 차원의 중

요도에 따라 우선순위를 부여하여 프로젝션의 순서를 결정하여 차원의 순서가 임의적일 수 있는 응용분야에 적합하도록 설계되었다.

본 논문의 알고리즘 고찰 및 분석 연구는 더욱 향상된 클러스터링 알고리즘 개발에 기반이 될 것이며, 클러스터링 기법의 특성별로 최적의 응용분야에 적용될 것으로 기대한다. 단, 본 논문에서는 등급, 학년, 성별 등과 같은 범주형(categorical) 데이터로 분류되는 데이터에 관련된 알고리즘은 다루지 않았으며 일반적인 수치데이터에 관련된 알고리즘으로 그 범위를 한정하였다.

2 고차원 데이터 클러스터링 기법

2.1 분할기반 알고리즘

분할 알고리즘은 데이터베이스를 k개의 클러스터 즉 클러스터의 무게중심(k-means) 또는 클러스터 내의 대표 객체(k-medoid)에 의하여 분할(partition)을 형성하는 것으로 각 객체는 가장 가까운 클러스터에 할당된다. 초기의 클러스터링 알고리즘들은 다음과 같은 이유들로 인해 k-medoid 방법을 선택하였다[9]. 첫째, 다른 많은 분할 기법들과는 달리 k-medoid 방법들은 잡음(outlier)의 존재에 대해 고려하고 있다. 즉, 데이터 점들의 나머지와 매우 떨어져 있는 데이터 점들을 잡음으로 간주한다. 두 번째, k-medoid 방법들에 의해 식별된 클러스터들은 객체들이 조사되는 순서에 의존하지 않는다. 더욱이, 식별된 클러스터들은 데이터 점들을 변형하거나 전이시켜도 달라지지 않는다. 다음은 k-medoid 방법을 이용한 분할 알고리즘 중에서 잘 알려진 PAM[11], CLARA[11], CLARANS[9]를 소개한다.

(1) PAM[11]

PAM(Partitioning Around Medoids)은 k개의 클러스터들을 찾기 위해서 각각의 클러스터에 대해 하나의 대표 객체를 결정한다. 이 대표 객체를 medoid라고 부르는데 이것은 클러스터 내에서 가장 중심에 위치하는 객체를 의미한다. PAM에서는 자체적으로 k개의 medoid들을 선택한 후 선택되지 않은 나머지 객체들에 대해 선택된 medoid와의 거리를 측정하여, 최소의 거리를 갖는 medoid가 속한 클러스터에 포함시킨다. 즉 만약에 O_j가 하나의 선택되지 않은 객체이고, O_i가 선택된 객체이며, $d(O_j, O_i) = \min_{Oe} d(O_j, O_i)$라면 O_j가 O_i로 대표되는 클러스터에 속한다고 말할 수 있다. (여기서, $\min_{Oe}$는 모든 medoid들 Oe에 걸친 최소값이고, $d(O_a, O_b)$는 비유사도(dissimilarity) 또는 객체 O_a와 O_b 사이의 거리를 의미한다.) 이 때, 클러스터의 품질은 임의의 객체와 클러스터 내의 medoid 사이의 평균 비유사도를 계산함으로써 측정된다. 여기서 k개 medoid들을 찾기 위해서는 임의 방법으로 k 객체들의 선택으로 시작한 다음, 각 단계에서 객체간의 비유사도에 따라 객체를 교체하게 되는데 이러한 교체작업(swap)은 클러스터링의 품질을 개선할 수 있을 때까지 수행한다.

(2) CLARA[11]

CLARA(Clustering LARge Applications)는 보다 대용량의 데이터 집합을 다루기 위해 설계되었으며 데이터의 샘플링에 의존한다. CLARA는 전체 데이터 집합에 대한 대표 객체들을 찾는 대신, 데이터 집합에 대한 샘플을 무작위로 추출하고 그 샘플에 대해서 PAM을 적용하여 샘플의 medoid들을 찾는다. 이때 샘플 데이터의 신뢰성을 위하여 복수 개의 샘플들을 추출하여 medoid들을 구함으로써 최상의 클러스터를 탐색하도록 한다. 단, CLARA는 클러스터링의 품질을 측정하는데 있어서는 샘플들에 대해서뿐만 아니라 전체 데이터 집합에 대해서 모든 객체들의 평균 비유사도를 계산한다.

(3) CLARANS[9]

CLARANS(Clustering Large Applications based on RANdomized Search)는 PAM과 CLARA의 장점을 혼합한 알고리즘이다. CLARA가 전체 데이터 집합에 대해 추출한 샘플을 검색의 각 단계에서 고정적으로 이용하는 반면 CLARANS는 검색의 단계별로 샘플을 동적으로 추출한다. 여기서 단계의 의미는 클러스터를 형성하는 k개의 medoid들을 찾는 과정으로서, 이 결과 찾아진 medoid들 집합은 그

래프의 특정 노드를 형성한다. 이러한 샘플링 방법은 탐색의 지역적 제한을 초래하지 않는다는 이점이 있다. 하나의 노드를 형성하는 과정에서, 해당 노드에 속한 medoid들을 다른 노드에 속한 medoid들과 비교하여 나머지는 같고 오직 하나만 다를 때 이 두 노드는 "이웃하다"고 정의한다. CLARANS는 이웃노드를 무작위로 찾는데 이때 이웃노드의 최대수는 입력매개변수 *maxneighbor*값으로 제한한다. 만약 이웃노드와 비교하여 이웃노드의 품질이 우수하면 이웃노드로 이동하여 다시 그 노드의 이웃노드들을 찾는 과정을 반복한다. 만약 더 좋은 이웃노드가 발견되지 않으면 현재의 노드를 클러스터링의 지역적 최적치(local optimum)로 간주한다. 이와 같이 지역적 최적치를 구하게 되면 CLARANS는 새로운 지역적 최적치를 찾기 위해 임의의 노드를 형성하고 위의 과정을 반복한다. 이러한 과정을 통하여 구해진 지역적 최적치의 개수는 입력매개변수 *numlocal* 값에 의해 제한된다. 이처럼 CLARANS는 CLARA처럼 노드마다 이웃 노드들을 모두 검사하지 않으나, 샘플을 단계별로 구성한다는 점에서 CLARA의 지역적 탐색의 단점을 개선하였다. 또한 PAM과 비교하였을 때, PAM은 모든 이웃 노드들을 검사하기 때문에 대용량의 데이터에 대해서 매우 비효율적이다. 반면 CLARANS는 단지 제한된 수의 이웃 노드들에 대한 샘플만을 검사하여 상대적으로 적은 비용이 소요된다. 단, *maxneighbor*값이 높을수록 CLARANS는 PAM과 유사해지고 각각의 지역적 최적치를 탐색하는 시간이 길어진다는 사실에 주목할 수 있다.

2.2 밀도기반 알고리즘

밀도기반 클러스터링은 보다 효율적인 기법으로서 지역성(locality)을 고려하여 인접한 데이터 요소들을 지역적 조건에 따라 클러스터로 그룹화한다. 또한 데이터베이스를 한번만 스캔하여 클러스터링할 수 있는 장점이 있다. 이 기법은 밀도함수를 이용하여 데이터 점들의 공간적인 분포를 계산한 후 클러스터를 탐색하는 것으로서 다음은 대표적인 밀도기반 알고리즘이다.

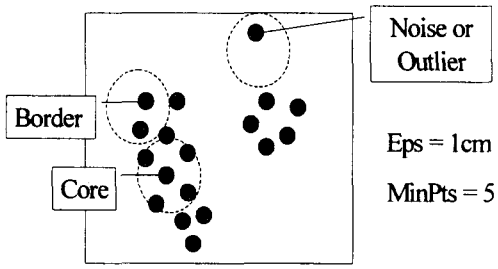
(1) DBSCAN[5]

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)은 데이터 집합이 주어지면 클러스터와 어느 클러스터에도 속하지 않은 데이터 점들인 잡음을 식별한다. 여기서 클러스터들을 인식할 수 있는 중요한 이유는, 각 클러스터 내 데이터 점들의 밀도는 클러스터의 외부에서 보다 상당히 높다는 것이다. 더욱이, 잡음 영역에서의 밀도는 어떤 다른 클러스터 내에서보다 밀도가 낮다.

DBSCAN은 k차원 공간의 점들로 된 데이터베이스에서 이러한 직관적 견해의 클러스터와 잡음을 정형화하였다. 기본 개념은 클러스터 내의 각 점에 대하여, 주어진 반경의 이웃영역(neighborhood)이 적어도 최소 수의 점들을 포함해야 한다. 즉 이웃영역에서의 밀도가 임의 임계값을 초과해야 하는 것으로 이를 위해 DBSCAN은 매개변수 *Eps*, *MinPts*에 관한 전역변수를 사용하는데, 이것은 한 클러스터 내의 각 데이터 점들에 대하여 그 점의 이웃 영역 *Eps* 거리 안에 있는 점들의 개수는 최소한 *MinPts*가 있음을 의미한다. 이웃영역의 형태는, 두 점 p와 q의 거리함수의 선택에 의해 결정되는데 이 방법은 주어진 응용에 따라 적당한 함수를 선택하도록 한다. 그리고 밀도 임계값을 입력 매개변수로 결정한다. DBSCAN의 중요한 특징으로는 둥근 구면, 잡아당겨 늘어진 형태, 직선 형태, 길게 늘어진 형태 등의 임의 형체를 가진 클러스터의 발견할 수 있다는 것이다. 또한 DBSCAN은 R^*-트리를 사용하여 보다 나은 효율성을 얻을 수 있다. 그림 1은 매개변수 *Eps*, *MinPts*에 의해 클러스터와 잡음을 분리하는 예이다.

그러나 DBSCAN은 입력 매개변수의 결정을 위해 도메인에 대한 사용자의 분석이 필요하고, R^*-트리 기반으로 구현되므로 고차원 공간에서는 R-트리 기반 인덱스의 성능저하로 인해 효율적으로 수행하지 못한다. 만약 고차원 데이터에 관하여 특별한 인덱싱 기법이 사용된다고 해도, 최근접 이웃들(nearest neighbors)은 고차원 공간에서 데이터의 밀도에 관한 충분한 정보를 포함하지 못하므로 근접한 이웃의

정보에 근거한 클러스터링 방법들은 효과적으로 수행하지 못한다. 따라서 DBSCAN과 같은 알고리즘도 고차원 데이터 집합에서는 효과성 문제를 보이고 있다.



(그림 1) *Eps*, *MinPts*에 의한 잡음 분리

(2) OPTICS[6]

OPTICS(Ordering Points To Identify the Clustering Structure)는 DBSCAN을 확장한 알고리즘으로서 무한개의 거리 매개변수 ε_i를 사용한다. OPTICS의 개발 동기로는, 많은 실제 데이터 집합들의 중요한 특성은 그들이 갖는 고유의 클러스터 구조를 전역적인 밀도 매개변수에 의해서 특징지을 수 없다는 것이다. 따라서 데이터 공간의 다른 영역에 존재하는 클러스터를 발견하기 위해서는 다양한 지역적 밀도가 요구될 수 있다. 이를 위해 OPTICS는 밀도기반의 클러스터 순서화(density-based cluster ordering)를 제안하였다. 이 개념은 상수 MinPts값에 대해, 고밀도(ε에 관해 낮은 값)에 관한 밀도기반 클러스터는 저밀도(ε에 관해 높은 값)에 관한 밀도기반 클러스터에 완전히 포함된다는 것이다. 결론적으로, OPTICS는 DBSCAN을 확장하여 몇 개의 거리 매개변수를 동시에 처리함으로써 다른 밀도의 클러스터들을 동시에 탐색할 수 있다. 단, 일관성 있는 결과를 생성하기 위해, 특정 순서를 준수해야 하며 순서에 따라 객체들은 클러스터를 확장할 때 처리된다. 이와 같이 OPTICS는 차별화된 밀도개념에 의하여 클러스터를 분석하는 목적으로 사용될 수 있다.

2.3 요약정보 기반 알고리즘

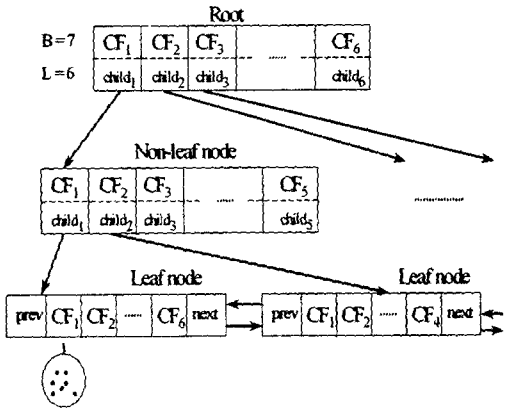
BIRCH의 경우와 같이 클러스터 특징-트리, STING과 같이 그리드에 저장되는 임의 종류의 요약(aggregated or condensed) 정보를 이용하는 효율적이고 효과적인 알고리즘이 제안되었다. 이들은 공통적으로 한가지 방법 또는 그 이상으로 이용 가능한 정보를 요약하므로 요약기반의 접근방법이라 한다.

(1) BIRCH[15]

BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies)는 클러스터 특징(clustering feature)을 저장하는 균형트리인 CF-트리라 불리는 계층적 데이터 구조를 사용한다. CF는 전체 데이터 점들을 모두 저장하는 대신 서브클러스터(sub-cluster)에 대한 정보를 요약한 $CF=(N, \overrightarrow{LS}, SS)$로 정의되는데, 여기서 N은 클러스터 내의 데이터 점들의 개수이고, $\overrightarrow{LS}$는 N개 데이터 점들의 선형 합계($\sum_{i=1}^{N} \overrightarrow{X_i}$)이다. 그리고 SS는 데이터 점들의 제곱의 합($\sum_{i=1}^{N} \overrightarrow{X_i^2}$)이다. BIRCH는 주어진 제한된 메모리 자원을 이용하여 가능한 최상의 클러스터링을 형성하도록 한다. CF-트리는 자식 노드의 최대 개수인 B(branching factor)와 리프 노드에 저장된 서브 클러스터의 최대 반지름인 T(threshold)를 갖으며, 알고리즘의 수행 중 주기억장치가 부족하게 되면 CF-트리의 노드에 있는 유사한 데이터 항목들은 요약된다. CF-트리의 중간 노드는 자식 노드들의 CF에 대한 합을 저장하고 있으므로 자식 노드에 대한 정보를 요약하고 있다. 다음의 그림 2는 B=7이고, L(리프 노드의 엔트리 크기)=6인 CF-트리의 예를 보이고 있다. BIRCH는 잡음 데이터를 다루는 첫 번째 알고리즘으로서, 클러스터를 발견하거나 잡음으로부터 클러스터의 구별을 위해 몇 가지 경험적 정보를 사용한다. 또한 매우 효율적인 알고리즘의 하나이며 데이터베이스를 오직 한번 스캔하여 트리를 구성하는 장점이 있는데, 이것은 고차원 데이터에 대해서도 그러하다.

그러나 클러스터의 특징을 정의하는데 반지름이나 지름 등의 유사성 개념을 사용하므로 오직 구형의 클러스터만을 발견하는 한계가 있

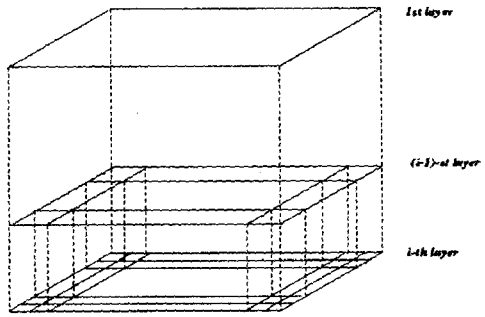
다. 또한 클러스터의 생성이 데이터의 입력순서에 따라 점진적 수정 방법을 사용하여 동적으로 구축되므로 동일한 데이터들에 대해서도 입력순서가 다르면 다른 클러스터를 형성할 수 있는 단점이 있다.



(그림 2) $B=7$, $L=6$인 CF-트리

(2) STING[7]

STING(STatistical INformation Grid-based method)에서는 그리드 셀 계층(grid cell hierarchy)구조를 이용하여 데이터 공간을 다중 레벨의 사각형 셀들로 나눈다. 1번째 레이어에서는 오직 하나의 셀만을 가지며 i번째 레이어에서는 (i-1)번째 레이어의 4배에 해당하는 셀들을 가진다. 각각의 셀은 셀에 속한 객체의 개수, 평균, 표준편차, 최소값, 최대값, 분포형태 등 통계적 매개변수를 저장한다. 그림 3은 STING의 계층 구조를 나타내고 있다. 이러한 정보는 효율적으로 클러스터를 결정하는데 사용된다. STING이 각 셀의 통계적 매개변수를 계산하는데는 데이터 집합에 대하여 오직 한번의 스캔이라는 선형적 수행시간을 필요로 한다. 이와 같이 STING의 계층구조는 질의에 대하여 보다 빠른 응답시간을 제공하지만 셀 간의 공간적인 관계를 고려하지는 않았다. 이로 인해 식별된 모든 클러스터의 경계가 수직적 또는 수평적이다. 따라서 대각선 모양의 경계 등 클러스터의 정확한 경계를 찾을 수 없으므로 클러스터의 품질을 저하시킬 수 있다. 또한 STING은 저차원 데이터를 위해 설계되었으므로 고차원 데이터를 위하여 바로 확장하기는 어렵다.



(그림 3) STING의 계층구조

2.4 그리드 기반 알고리즘

그리드 기반 알고리즘은 데이터 공간을 유한개의 셀로 나누어 정량화한 후, 그 셀에 대하여 연산이 이루어지는 방법론이다. 중요한 특징으로는 알고리즘의 처리속도가 데이터 객체의 수와는 독립적으로 빠르다는 점이며, 처리속도는 단지 각 차원에 존재하는 셀들의 수에 좌우된다.

(1) DENCLUE[8]

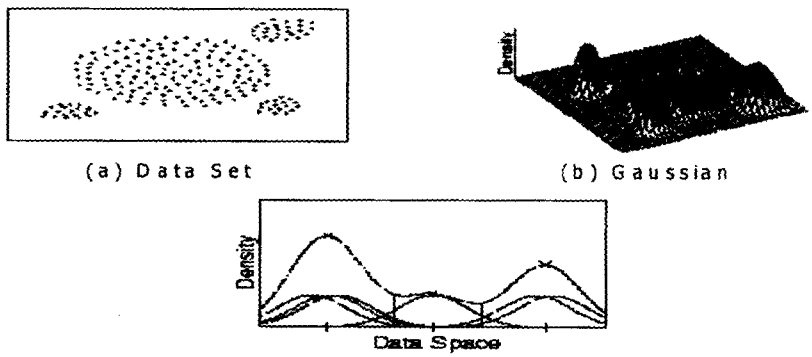
대용량 멀티미디어 데이터베이스에 일부 클러스터링 알고리즘들이 적용될 수 있다. 그러나 기존 알고리즘들은 효과성과 효율성 면에서 다소 한계성을 갖는데, 그 이유는 멀티미디어 데이터베이스는 고차원 특징벡터의 클러스터링을 요구하며 흔히 많은 양의 잡음을 포함하고 있기 때문이다. DENCLUE (DENsity-based CLUstEring)는 대용량 멀티미디어 데이터베이스를 위하여 개발되었으며, 이 접근방법은 데이터 점들의 영향함수(influence function)의 합으로써 총체적인 점밀도를 설계하는 것을 기본 개념으로 한다. 영향함수에는 Parabolic function, Square wave function, Gaussian function 등이 있다. 그 다음으로, 클러스터는 밀도 유인자(density-attractor)의 결정에 의해 식별되며 임의 형태의 클러스터는 전체 밀도 함수에 기초한 수학식에 의하여 간단하게 명세

될 수 있다. 즉 클러스터링을 위한 새로운 접근방법으로서 고차원 공간상의 이산 데이터를 연속 구조체로 변환한 후 미분하여 0(zero)이 되는 점을 밀도의 최고 값인 클러스터의 중심으로 간주하여 클러스터를 식별하는 방법론이다. 그림 4는 데이터 집합 (a)에 대해 Gaussian 영향함수를 적용한 예이다. 이 알고리즘들의 공헌으로는 우선 확고한 수학적 기반을 마련했으며, 많은 양의 잡음을 가진 데이터 집합에서 비교적 좋은 클러스터링 특성을 갖는다. 또한 고차원 데이터 집합에서 임의형태의 클러스터에 관하여 간결히 수학적 명세를 할 수 있다. DENCLUE는 데이터 공간을 일정한 간격으로 나누는 그리드를 사용하는데, 일반적인 그리드 기반 알고리즘보다 효율적이다. 그것은 실제로 데이터를 포함하는 그리드 셀만을 유지하고 있으며, 이 셀들을 트리 기반의 구조로 관리하기 때문이다.

다. 이와 같이 웨이브렛 변환을 이용하는 장점은, 클러스터를 효율적으로 결정할 수 있도록 자동적으로 데이터 그리드의 다중 해상도 표현(multiresolution representation)을 제공하는데 있다. 이 방법에서 결과적인 클러스터는 잡음 데이터의 영향을 적게 받으며, 입력되는 데이터의 순서에 민감하지 않다. 또한 복잡한 구조를 가진 임의 형태의 클러스터 탐색에 용이하다. 그러나 WaveCluster는 저차원 데이터를 위해 설계되었으며 고차원 데이터를 위하여 곧바로 확장하는 것은 어렵다. 그것은, 그리드 셀의 수는 차원의 수에 대해 지수적으로 증가하며 연결 요소의 결정은 많은 수의 인접하는 셀들로 인해 상당한 비용이 소요되기 때문이다.

2.5 부분차원 기반 알고리즘

대부분 알고리즘들은 고차원 공간에서 클러스터링에 실패하는 경향이 있는데 이는 데이터



(그림 4) Gaussian 영향함수에 의한 데이터 변환

(2) WaveCluster[14]

다차원 공간 데이터를 위하여 설계된 WaveCluster는 효율적인 클러스터링을 위하여 일정한 간격의 그리드 상에서 웨이브 렛 변환(wavelet transformation)을 이용하는 접근방법이다. 즉 공간 데이터를 다차원 그리드 위에 사상하여, 이 그리드 셀에 공간 데이터를 주파수 도메인으로 변환하는 웨이브렛 신호처리 기법을 적용한다. 그 다음은 클러스터를 의미하는 연결 요소(connected component)를 검색하여 변환된 도메인에 있는 밀집영역을 결정한

점들이 갖는 고유의 희소성 때문이다[10]. 즉 고차원 공간에서 차원의 전체가 주어진 클러스터에 관련되지 않을 수 있다는 개념으로서 이것을 다루는 방법 중 하나가 연관된 차원을 고르는 것으로 해당하는 부분차원에서 클러스터를 탐색한다.

(1) CLIQUE[3]

CLIQUE(CLustering In QUEst)는 고차원 공간상의 데이터 점들은 차원의 부분집합에 대

하여 보다 잘 클러스터링될 수 있다는 사실에 근거한 첫 번째 연구이다. 즉 CLIQUE는 고차원 데이터에서 클러스터를 찾는 효과적인 방법으로서 부분공간 즉 부분차원에서의 클러스터링 기법을 제시하였다. CLIQUE는 데이터 공간의 밀도를 간단하게 산정하기 위하여 각 차원을 다수의 일정한 간격으로 나누고 분할된 각 셀(unit) 안에 놓여진 점들의 수를 찾는다. 이것은 각 셀이 동일한 크기를 가지므로 그 안의 점들의 수는 셀의 밀도를 의미하기 때문이다. CLIQUE는 다음의 세 단계로 수행된다. **1) 부분공간**(부분차원) **식별** 단계에서는 우선 부분공간에 존재하는 밀집영역(dense unit)을 찾는다. 이를 위해 CLIQUE는 bottom-up 알고리즘을 사용하는데, 이 알고리즘은 만약 데이터 점들의 모임 S가 k-차원 공간에서 하나의 클러스터이면 S는 또한 이 공간의 임의의 (k-1)차원적인 프로젝션에서 클러스터의 부분이라는 개념의 Monotonicity 정리에 근거하여 검색공간을 제거(prune)해 나간다. **2) 클러스터 식별** 단계에서는 1)의 결과인 밀집 단위들의 집합을 분할하여 출력하는데 각 분할들은 하나의 클러스터이다. 이 문제를 그래프에서 연결 요소의 탐색으로 보면, 그래프의 노드를 밀집 단위로, 두 노드간의 간선은 공통 면을 갖는 밀집 단위로 간주한다. 그래프에서 동일한 연결 요소의 노드에 해당하는 단위들은 연결되어 있는데, 이것은 그들이 같은 클러스터 내에 있음을 의미한다. 반면에, 다른 요소에 있는 노드들에 해당하는 단위들은 서로 연결되지 않으므로 같은 클러스터 안에 속하지 않는다. 그래프의 연결요소를 찾기 위해서는 깊이우선 탐색 알고리즘을 사용한다. **3) 클러스터의 최소명세(minimal description) 생성** 단계의 입력은 동일한 부분공간에서 연결된 k 차원 단위들의 서로소인 합으로서 각 집합은 하나의 클러스터이며, 그것을 위한 간결한 명세를 생성하는 것이 이 단계의 목적이다. CLIQUE는 greedy growth 알고리즘을 사용하여 각 클러스터의 최소명세를 생성하는데, 클러스터를 구성하는 모든 밀집단위들을 최대한 포함해야 한다. 마지막으로, 최대영역에 대한 최소한의 명세를 DNF식을 사용하여 클러스터를 표현한다. CLIQUE는 데이터의 흥미있는 특성을 발견할 수 있는 접근방법이지만, 데이터 점들을 서로소인 집합으로 분할하기 어렵기 때문에 엄밀한 정의의 클러스터 탐색에는 한계가 있다. 그리고 주어진 밀집영역에 대하여, 저차원 부분공간에서 그것의 모든 프로젝션들은 또한 밀집하다 라고 조사되므로 조사된 밀집영역 사이에는 큰 오버랩이 존재할 수 있는 문제점을 내포하고 있다.

(2) PROCLUS[4]

PROCLUS(PROjected CLUStering) 알고리즘은 [3]에 이어 고차원 공간에서 클러스터를 탐색하는데 프로젝트된 클러스터링 개념을 논의한 것으로서, 데이터 점 및 차원을 기반으로 클러스터를 산출한다. 특히 고차원 데이터에 관한 클러스터링에서는 질적인 향상을 가져올 수 있다. PROCLUS에서 제안한 프로젝트된 클러스터링 알고리즘은 우선, CLARANS에서 제안한 mediod 탐색기법을 이용하여 클러스터와 차원의 적당한 집합을 찾는다. 그 다음, 찾아진 각 mediod와 연관된 차원의 집합을 찾기 위하여 지역성 분석(locality analysis)기법을 사용한다. 즉 PROCLUS 알고리즘은 다음의 3가지 단계로 진행된다. **1) 초기화 단계**에서는 CLARANS를 이용하여 데이터 점들의 집합을 줄이는데 것을 목적으로 한다. **2) 반복 단계**에서는 보다 좋은 medoid 집합의 탐색을 시도하고, 각 medoid에 해당하는 차원의 집합을 계산한다. 즉 각 medoid에 대응하는 차원의 집합을 계산하여 medoid에 할당된 점들은 결정된 부분차원에서 최상의 클러스터를 형성한다. **3) 클러스터 정제 단계**에서는 클러스터링의 질적인 향상을 위하여 데이터를 한번씩 검사한다. 단, PROCLUS에서 탐색하고자 하는 부분차원의 클러스터 수(k)는 매개변수로 입력받는다.

(3) CLIP[13]

CLIP(**CL**ustering based on **I**ncremental **P**rojection)은 저자가 개발 및 제안하는 기법으로서 각 차원에 대해 프로젝션 선형변환을 이용하는 클러스터링 기법이다. CLIP은 각 애트리뷰트 값의 분포에 의존적으로 점진적인 프로

젝션을 하여 클러스터 형성에 연관성이 적은 차원 및 영역은 제외시켜 클러스터가 포함될 후보공간을 결정한다. 그런 다음, 결정된 후보 공간에서 데이터 점들의 평균값을 이용하여 보다 정확한 클러스터 형태를 식별하고자 하는 기법이다. · 프로젝션 $P_l : R^k \rightarrow R$은 $P_l(x_n^{(1)}, x_n^{(2)}, \cdots, x_n^{(k)}) = x_n^{(l)}$ 인 선형변환으로 정의하며, CLIP은 k차원의 전체 데이터 공간 R^k에 클러스터가 존재하면, 그 클러스터의 (k-1) 부분차원에서도 클러스터를 형성하는 밀집영역을 포함한다는 정리[3]에 근거한다. 특히 CLIP은 전자상거래에서 고객데이터 분류를 목적으로 개발되었으며, 각 차원의 중요도에 따라 우선순위를 부여하여 프로젝션의 순서를 결정할 수 있다. 즉 차원의 순서가 임의적일 수 있는 응용분야에 적합하도록 설계되었다. CLIP은 크게 후보영역을 결정하기 위한 부분차원 탐색 단계와 부분차원에 포함된 클러스터를 식별하는 단계로 나누며 수행과정은 다음과 같다. **1) 부분차원 탐색** (후보공간 결정) : 데이터 공간의 점들은 데이터베이스에서 각각의 레코드를 의미한다. 이때 레코드의 각 애트리뷰트는 데이터 공간에서 하나의 차원을 의미하는데, 하나의 애트리뷰트에 해당하는 한 차원에 대하여 축-평행하게 프로젝션한다. 그리고 데이터 분포에 따라 밀도 임계값을 초과하는 밀집영역을 구한 후, 그 밀집영역에 해당하는 초월 사각형(hyper-rectangle) 부분의 레코드에 대해서만 그 다음 차원의 애트리뷰트 값으로 프로젝션시킨다. 전체를 k차원이라 할 때, 1차원에서 k차원에 이르기까지 순차적으로 각 차원에 해당하는 밀집영역을 그 다음 차원에 반영시킴으로써 최종적으로는 탐색할 데이터공간을 줄여 가는 것이다. 이 때, 임의 차원에서 구해진 영역의 분포가 균등분포에 근사하면 이러한 영역은 일단 제외시킨다. 왜냐하면 비교적 균등하게 데이터가 분포하게 되면 클러스터를 탐색하는데 있어서는 방해가 되므로, 이와 같은 차원에 대해서는 후 순위로 처리하거나 제외할 수 있다. **2) 클러스터 식별**(cluster identification) 단계는 1)에서 결정된 후보공간에서 클러스터의 형태를 구체화하는 과정으로서, 점들의 중심부에서 탐색을 시작하기 위해 영역에 속한 데이터 점들의 대수적인 평균값을 계산한다. 이때 평균은 후보영역의 중심으로서 다른 데이터 점들과의 거리의 합이 최소가 되는 점이다. 그 다음, 평균을 중심으로 2^k 개만큼 공간을 분할하는데 평균은 전체 공간을 2^k 개로 분할할 수 있기 때문이다. 이러한 방법은 각 분할 영역에서 평균을 중심으로 한 공간의 지역성(locality)을 고려한 것이다. 즉 정규분포에 의해 생성된 클러스터를 탐색한다고 가정할 때, 데이터 점들은 그 평균을 중심으로 ε거리 내에 비교적 많이 분포하기 때문이다. 따라서 점들의 평균값을 중심으로 데이터를 조사하는 것은 클러스터의 형태(shape)를 보다 효과적으로 구체화할 수 있다.

3. 비교 분석

본 논문에서는 고차원 데이터를 위한 클러스터링 알고리즘들을 클러스터 탐색 방법에 따라 분류하였는데 다음과 같이 요약할 수 있다. 첫째, 분할에 의한 알고리즘은 데이터 집합에서 최적의 k개 대표객체를 결정해야 하므로 고차원의 대용량 데이터베이스에 대한 적용은 비현실적이다. 둘째, 밀도 기반의 알고리즘은 고차원 공간에서는 데이터의 밀도에 관한 정보를 충분히 포함하지 못하므로 고차원 데이터 관해서는 적합하지 않음을 의미한다. 셋째, 요약정보 기반의 알고리즘은 입력 데이터에 대한 임의의 총계정보를 이용하여 고차원 공간에서도 좋은 효율성을 갖지만, 효과성 면에서는 취약점을 갖는다. 넷째, 그리드 기반은 객체의 입력 순서와 무관하다는 장점을 갖지만, 그리드 셀의 수가 차원의 수에 대해 지수적으로 증가한다는 일반적인 그리드 기반의 단점으로 인해 고차원 확장에 어려움이 있다. 그러나 DENCLUE의 경우와 같이 데이터를 포함하는 셀만을 저장하는 경우는 보다 효율적으로 수행할 수 있다. 마지막으로, 소위 "차원의 저주" 문제를 해결하기 위한 하나의 방법인 부분 차원을 고려한 알고리즘은 고차원 공간에서 클러스터 형성에 관련된 차원을 고르는 것으로 부분차원에서 클러스터를 탐색한다. 즉 고차원 공간의 클러스터 탐색에서 실패의 원인을 데이

터 점들이 갖는 고유의 희소성에 두고, 클러스터 형성에 연관된 차원을 선택한다는 개념으로서 고차원 공간에서도 매우 효율적이다. 이와 같은 고차원 데이터에 관한 클러스터링 기법의 특징과 장·단점을 표 1과 표 2로 요약할 수 있다. 표 1은 각 알고리즘이 클러스터링하는데 사용하는 방식을 나타낸 것인데, 대부분의 알고리즘들은 단일 방식을 사용하는 것이 아니라 기본 방식이외에도 알고리즘의 효율성 및 효과

(표 1) 알고리즘별 클러스터링 특징

클러스터링 알고리즘	분 할 (k-medoid)	밀도	계층 구조	그리드	요약 정보	변환 함수			부분 차원
						웨이브 렛 변환	influence function	선형변환 프로젝션	
CLARANS	○								
DBSCAN		○	○						
BIRCH			○		○				
STING			○	○	○				
DENCLUE				○	○		○		
WaveCluster				○		○			
Optigrid		○						○	
CLIQUE		○		○					○
PROCLUS	○	○							○
CLIP		○		○	○			○	○

성을 위하여 여러 방식을 혼용하고 있음을 알 수 있다. 표 2에서는 알고리즘별로 장·단점 및 수행성능을 비교하였다. 특히 표 2에서는 각 알고리즘들의 수행능력을 시간복잡도 측면에서 비교하였다.

이 중 저자가 [13]에서 제안한 바 있는 CLIP의 수행능력을 여기서 증명하고자 한다. CLIP의 시간복잡도는 $O(\frac{N}{2} + c \cdot N \cdot k)$이다.

[증명]

다음과 같이 순환식(recurrence formula)을 이용하여 증명할 수 있다.

전체 k차원 데이터 공간에서, N개의 입력 데이터에 대해 m개 차원 이상으로 형성된 클러스터를 조사한다고 하자. 이때 데이터를 액세스하는 총 횟수 $a(k, m)$는 다음과 같이 정의할 수 있다. 여기서, $c^{m-2}N$는 $(m-1)$개 차원으로 형성된 클러스터에 속하여 액세스되는 데이터의 총 개수이다. 참고로, 여기서 상수 c의 의미는 다음과 같다. 첫 번째 차원에서 시작하는 클러스터를 조사한다고 가정하자. 이때 처음 시작하는 차원에서는 N개 데이터 전체에 대한 첫 번째 애트리뷰트 값을 조사한다. 만약 첫 번째 차원에서 밀집영역이 발견되면, 그 다음 차원에서는 처음 N에 대해 $0 \le c_1 < 1$인 c_1의 비율만큼만 애트리뷰트 값을 조사한다. 또 그 다음 차원에서는 c_1에 대한 c_2만큼만 조사하게 된다.

즉 $0 \le c_1, c_2, \cdots, c_{k-1} < 1$ 이며, $c = \max\{c_1, c_2, c_3, \cdots, c_{k-1}\}$이 다. $(0 \le c < 1)$

$$a(k, m) \le a(k-1, m-1) + c^{m-2}N$$
$$a(k, m) \le a(k-2, m-2) + c^{m-3}N + c^{m-2}N$$
$$\vdots$$

$a(k, m) \leq a(k-m+1, 1) + c^{1}N + \cdots + c^{m-3}N + c^{m-2}N$

$a(k, m) \leq (k-m+1)N + N + cN + \cdots + c^{m-2}N$

$a(k, m) \leq (k-m+1)N + \frac{1-c^{m-1}}{1-c}N$

$0 \leq c_1 < 1$이 므로

$\frac{1-c^{m-1}}{1-c} \leq (m-1)$ 이다.

따라서 $\frac{1-c^{m-1}}{1-c} = (m-1)$라 가정할 경우,

$a(k, m) \leq (k-m+1)N + (m-1)N = k \cdot N$ 이다.

그러므로 $O(N \cdot k \cdot c)$이다. 단, $(0 < c \leq 1)$

4. 결론

본 논문에서는 지금까지 연구 발표된 대표적 클러스터링 기법에 대한 수행절차의 특징과 장점 및 단점을 분석하였다. 일부 기법은 저차원 소량의 데이터에 관해서만 좋은 성능을 갖는 반면, 또 일부 기법들은 고차원 데이터에 대해서도 매우 효율적이며 효과적인 성능을 보인다. 그러나 차원이 증가함에 따라 고차원 데이터의 특성으로 말미암아 알고리즘의 성능은 급격히 하락하는 것으로 보고되고 있으며, 이를 위한 해결책의 하나로서 부분차원을 고려하는 알고리즘들이 제안되고 있다. 앞으로도 고차원 공간에서 클러스터의 질적인 향상을 위하여 공간을 감소시키는 방법으로 알고리즘의 효율성 측면을, 클러스터 형태를 더욱 구체화하는 방법으로 효과성 측면을 위한 연구가 계속되어야 할 것이다. 또한 이와 같은 기법들에서 보이는 공통점과 차이점을 일반화하여, 특정 도메인에 국한하지 않고 다양한 종류의 데이터 유형을 지원할 수 있는 클러스터링 알고리즘이 개발되어야 할 것이다. 이에 따라 본 논문은 대표적인 클러스터링 방법론에 대한 분석 및 평가를 통하여 더욱 향상된 알고리즘 연구에 기반을 마련할 것으로 기대한다.

(표 2) 클러스터링 알고리즘의 장·단점 및 수행성능

클러스터링 알고리즘	장 점	단 점	수행 성능 (시간복잡도)
CLARANS	보다 향상된 k-mediod 방법	·DB의 다중스캔 ·무작위 접근방법의 사용으로, N이 클 경우 결과의 품질을 보장 못함 ·객체들의 주기억장치 상주를 전제하므로 대용량 DB에 적용 불가능	$\Omega(KN^2)$ K : 클러스터 수 N: 입력 데이터 수
DBSCAN	·임의 형태 클러스터 탐색 ·잡음(noise) 분리	·고차원 공간에서 R-트리 기반 인덱스의 성능저하로 수행의 효율성 문제 ·고차원 공간에서 최근접 정보에 의한 접근방법의 효과성 저하	$O(NlogN)$
BIRCH	·DB의 크기 및 데이터의 방문 횟수 면에 대한 선형적 시간 복잡도 ·잡음을 고려한 첫 번째 알고리즘	·요약정보 기반이므로 고차원 공간에서 효과성에 관한 성능저하 ·둥근 모양의 클러스터 경우에만 효과적인 탐색 ·데이터 입력순서에 민감	$O(N)$
STING	빠른 프로세스 시간	·요약정보 기반이므로 고차원 공간에서 효과성에 관한 성능저하 ·저차원 데이터를 위해 설계되었으며, 고차원으로 확장이 어려움	$O(N)$
DENCLUE	·데이터 점을 포함하는 그리드 셀만 저장 ·간단한 식에 의해 임의 형태 클러스터 식별	효과성 문제 : 고차원 데이터의 경우, 상당한 비율의 클러스터를 탐색하지 못함	$O(log(\|N\|))$
WaveCluster	·잡음 / 입력 객체의 순서에 무관 ·임의 형태 클러스터 탐색	고차원 데이터 적용을 위한 확장의 어려움	$O(N)$
CLIQUE	·입력 데이터 순서와 무관한 동일한 결과 ·고밀도 클러스터를 갖는 부분공간 자동식별	·많은 부분공간이 제거될 가능성 ·클러스터간에 큰 오버랩이 존재할 가능성	$O(c^k + N\cdot k)$ k : 차원 수 N: 입력 데이터 수 c : 상수
PROCLUS	·클러스터 형성에 연관성이 낮은 차원의 제거로 데이터의 잡음 감소 : 향상된 효율성	·고차원 공간에서 특정 차원에 국한된 희소 데이터 분석에만 용이 ·고차원 공간에서 최상 medoid 선택의 어려움	$O(N\cdot k\cdot m)$ N: 입력 데이터 수 m : 클러스터 수 k : 차원 수
CLIP	·클러스터를 포함할 후보 공간의 탐색 : 효율성 ·부분차원 클러스터링	·차원별 데이터 분포에 대해 점진적으로 프로젝션하므로 처음 차원에서 정해진 영역에 다소 의존적일 가능성	$O(\frac{N}{2} + c\cdot N\cdot k)$ N: 입력 데이터 수 k : 차원 수 $(0 < c < 1)$

참고 문헌

[1] Fayyad, U. M., et al. "Advances in Knowlwdge Discovery and Data Mining", AAAI Press / The MIT Press, 1996.

[2] C. Faloutsos, "Fast Searching by Content in Multimedia Database," Data Engineering Bulletin, 18(4), 1995.

[3] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan, "Automatic subspace Clustering on High Dimensional Data Mining Applications," Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp.94-105, 1998.

[4] Charu C. Agrawal, Ceilia Procopiuc, Joel L. Wolf, Philip S. Yu, and Jong Soo Prk, "Fast Algorithms for Projected Clustering," Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp.61-72, 1999.

[5] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu, "A density-based algorithm for discovering clusters in large spatial database with noise," Int. Conf. on Knowledge Discovery in Databases and Data Mining, 1996.

[6] Mihael Ankerst, Markus M. Breunig, Han-Peter Kriegel, and Jorg Sander, "OPTICS: Ordering points to identify the clustering structure," Proc. of ACM SIGMOD Int. Conf. on Management of Data, 1999.

[7] Wei Wang, Jiong Yang, and Richard Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining," Proc. of 23rd Int. Conf. on VLDB, pp. 186-195, 1997.

[8] Hinneburg A., Keim D. A, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise," Proc. of 4rd Int. Conf. on Knowledge Discovery and Data Mining, AAAI Press, 1998.

[9] R. Ng, J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining," Proc. of 20th Int. Conf. on VLDB, pp. 144-155, 1994.

[10] D. Keim, S. Berchtold, C. Bohm, H.-P. Kriegel, "A cost model for nearest neighbor search in high-dimensional data space," Proc. of the 18th Symposium on Principles of Database Systems(PODS), pp. 78-86, 1997.

[11] L. Kaufman, P.J. Rousseeuw, "Finding Groups in Data: an Introduction to Cluster Analysis," John Wiley & Sons.

[12] Hinneburg A., Keim D. A, "Opimal Grid-Clustering: Towards breaking the Curse of Dimensionality in High-Dimensional Clustering," Proc. of 25th Int. Conf. on VLDB, pp. 506-517, 1999.

[13] 이혜명, 박영배, "점진적 프로젝션을 이용한 고차원 클러스터링," 한국정보과학회 논문지, 제28권 제4호, 2001.

[14] Gholamhosein Sheikholeslami, Surojit Chatterjee and Aidong Zhang, "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases," Proc. of Int. Conf. on VLDB, 1998.

[15] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 103-114, 1996.

이혜명

1989년 2월 명지대학교 공학사 (전자계산학)
1993년 2월 명지대학교 공학석사 (전자계산학)
2002년 2월 명지대학교 공학박사 (컴퓨터공학)

1998년 3월~현재 경문대학 인터넷미디어정보과 조교수

관심분야: 데이터마이닝, 웹 DB, 전자상거래 등

김홍일

1986년 홍익대학교 전자계산학과 (이학사)
1989년 인하대학교 전자계산학과 (이학석사)
2000년 홍익대학교 전자계산학과 (이학박사)

1994년 ~ 현재 : 대진대학교 컴퓨터공학과 조교수

관심분야: IPv6, P2P, 인터넷응용