반복적 2차원 프로젝션 필터링을 이용한 확장 고차원 클러스터링

이 혜 명† · 박 영 배††

요 약

대용량의 고차원 데이터 집합은 고차원 데이터 고유의 희소성에 의하여 상당한 양의 잡음을 포함하므로 효과적인 고차원 클러스터링에 어려움을 더한다. CLIP은 이와 같은 고차원 데이터의 특성을 지원하는 클러스터링 알고리즘으로 개발되었다. CLIP은 1차원 선형변환 프로젝션을 점진적으로 적용하여, 각 프로젝션 공간에서 얻어진 1차원 클러스터들의 곱집합을 찾는다. 이 집합은 클러스터를 포함할 뿐 아니라 잡음도 포함할 수 있다. 본 논문에서는 클러스터를 포함하는 곱집합을 정제하는 확장된 CLIP 알고리즘을 제안한다. 이미 CLIP에서 찾은 곱집합에 반복적인 2차원 프로젝션을 적용하여 클러스터의 고차원적 잡음을 제거한다. 확장된 알고리즘의 성능을 평가하기 위해 합성 데이터를 이용한 일련의 실험을 통하여 효과성을 증명한다.

Extended High Dimensional Clustering using Iterative Two Dimensional Projection Filtering

Hye-Myung Lee† · Young-Bae Park††

ABSTRACT

The large amounts of high dimensional data contains a significant amount of noises by its own sparsity, which adds difficulties in high dimensional clustering. The CLIP is developed as a clustering algorithm to support characteristics of the high dimensional data. The CLIP is based on the incremental one dimensional projection on each axis and find product sets of one dimensional clusters. These product sets contain not only all high dimensional clusters but also they may contain noises. In this paper, we propose extended CLIP algorithm which refines the product sets that contain clusters. We remove high dimensional noises by applying two dimensional projections iteratively on the already found product sets by CLIP. To evaluate the performance of extended algorithm, we demonstrate its effectiveness through a series of experiments on synthetic data sets.

키워드 : 고차원 클러스터링(high dimensional clustering), 2차원 프로젝션(two-dimensional projection), 잡음 필터링(noise filtering)

1. 서 론

데이터마이닝 기법 중에서 클러스터링은 데이터베이스 분야에서 유사성 검색, 고객 분류, 경향 분석 등을 위한 도구로서 널리 연구되고 있다. 그러나 대부분의 클러스터링 알고리즘들은 고차원 데이터 공간에서 클러스터를 탐사하는데 실패하는 경향이 있다. 그것은 대부분의 알고리즘들이 고차원 데이터에 대해 적절히 설계되지 않은데 있으며, 기존 알고리즘들의 성능은 차원이 증가할수록 빠르게 저하된다.

대용량 고차원 데이터 집합에 대한 클러스터링에 있어서 중요한 문제점 중의 하나는 데이터 집합에 존재하는 많은 양의 잡음으로서 클러스터링의 효과성에 심각한 영향을 준다. 따라서 잡음을 포함하는 고차원 데이터 집합에 대한 정확한 클러스터링 시간은 선형적인 시간 복잡도 내에서는 불가능한 것으로 연구되고 있다[12]. 이와 같이 대부분 클러스터링 알고리즘들이 고차원 공간에서 많은 양의 잡음으로 인해 실패하는 주요 원인은 고차원 데이터 점들이 갖는 고유의 희소성(sparsity) 때문이다[1, 5]. 즉 차원의 전체가 클러스터 형성에 관련되지 않을 수 있다는 개념으로서 최근에는 이를 해결하기 위한 방법으로 연관성 있는 차원을 선택하고 대응하는 차원에서 클러스터를 탐사하는 부분차원 클러스터링 기법에 대한 연구가 진행되고 있다. 대표적인 부분차원 클러스터링 알고리즘으로는 CLIQUE[3], PROCLUS[1], CLIP[15, 16] 등이 있으나 많은 양의 잡음을 포함하는 고차원 데이터에 대해서는 알고리즘의 효과성 측면에서 문제점을 내포한다. 특히

† 정 회 원 : 경문대학 컴퓨터정보과 교수
†† 정 회 원 : 명지대학교 컴퓨터공학과 교수
논문접수 : 2001년 9월 21일, 심사완료 : 2001년 10월 8일

CLIP은 고차원 데이터를 선형변환하기 위해 각 차원을 프로젝션한다. 데이터의 밀도에 근거한 CLIP의 점진적인 프로젝션은 잡음을 단계적으로 제거할 수 있으므로 많은 양의 잡음을 포함한 데이터 집합에서 높은 효과성을 보장할 수 있다. 그러나 선형 프로젝션에 의한 클러스터링은 하이퍼큐브 형태로 클러스터를 명세하므로 클러스터의 형태 식별에 어려움이 있다. 또한 1차원 프로젝션 방법에 의하여 1차원적 잡음은 존재하지 않으나 고차원의 잡음은 여전히 존재한다.

본 논문에서는 반복적인 2차원 프로젝션을 이용하여 클러스터를 정제하는 확장된 CLIP 알고리즘을 제안한다. 제안하는 알고리즘은 프로젝션 방법에서 간과하기 쉬운 클러스터의 형태를 보다 구체화하고, 고차원 필터링을 근사적으로 구현하며, 사용자가 요구하는 수준의 클러스터링을 목적으로 2차원 프로젝션을 반복적으로 적용한다.

2. 관련 연구

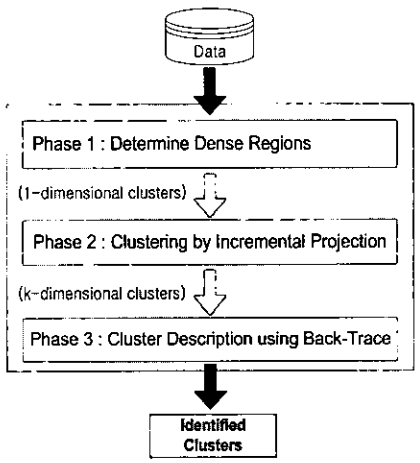
클러스터링을 위한 대부분의 알고리즘들은 고차원 데이터에 대해 적절히 설계되지 않았으며 기존의 알고리즘들의 성능은 차원이 증가함에 따라 빠르게 하락하는 것이 문제점이다. 데이터 공간의 차원, 데이터베이스의 크기에 따른 확장성을 의미하는 알고리즘의 효율성을 개선하기 위해 최적화된 클러스터링 기법들이 제안되고 있다. 그러나 고차원의 문제는 특히 잡음의 존재로 인해 결과의 정확성을 의미하는 클러스터링의 효과성에 있어서도 심각한 영향을 준다.

BIRCH는 잡음 데이터를 다루는 첫 번째 알고리즘으로서, 클러스터를 탐사하거나 잡음으로부터 클러스터의 구별을 위해 몇 가지 경험적 정보를 사용한다. BIRCH는 클러스터 특징을 저장하는 균형트리인 CF-트리라는 계층적 구조를 사용하며, 주어진 메모리 자원을 이용하여 최상의 클러스터링에 힘쓴다. BIRCH는 가장 효율적인 알고리즘의 하나이며 데이터베이스를 오직 한번 스캔하는데 이것은 고차원 데이터에 대해서도 그렇다. 그러나 클러스터의 특징을 정의하는데 반지름이나 지름 등의 유사성 개념을 사용하므로 오직 구형의 클러스터만을 탐색하는 한계가 있다. 또한 입력되는 데이터의 순서에 따라 점진적 수정 방법을 사용하여 동적으로 구축되므로 동일한 데이터들에 대해서도 입력순서가 다르면 다른 클러스터를 형성할 수 있는 단점이 있다.

DBSCAN[8]은 잡음을 고려하는 대표적인 알고리즘으로서 지역성(locality)을 고려한 밀도 개념을 이용하며 임의 형태의 클러스터를 보다 효율적으로 탐사한다. 그러나 DBSCAN은 R*-트리 기반으로 구현되므로 고차원 공간에서 R-트리 기반 인덱스의 성능저하로 인해 효율적으로 수행하지 못한다. 만약 고차원 데이터에 관하여 특별한 인덱싱 기법이 사용된다고 해도, nearest neighbor는 고차원 공간에서 데이터의 밀도에 관한 충분한 정보를 포함하지 않기 때문에 이에 근거하여 클러스터를 결정하는 모든 접근방법들은 효과적으로 수행하지 않는다. 따라서 DBSCAN도 효율성에 있어서 심각한 성능저하와 잡음을 포함하는 데이터 집합에 대해 효과성 문제를 보이고 있다.

CLIQUE, PROCLUS, CLIP 등은 클러스터 형성에 관련성 있는 차원이나 공간을 고려하는 부분차원 클러스터링 알고리즘으로 제안되었다. 고차원 데이터의 응용에 있어서 임의 데이터 점들은 적어도 일부 차원에서는 서로 떨어져 있는 점들이 존재하기 쉽다. 이에 따라 데이터 점들이 서로 연관되어 있는 특정 차원을 탐색하고, 클러스터 형성에 관련이 적은 차원들을 제거함으로써 데이터의 잡음을 감소시킨다는 개념이다. CLIQUE는 부분차원 클러스터링의 첫 번째 연구로서 고차원 공간상의 데이터 점들은 차원의 부분집합에 대해 보다 잘 클러스터링될 수 있다는 사실에 근거한다. 그러나 CLIQUE는 그리드 셀의 수가 차원의 증가에 따라 지수적으로 증가하는 그리드 방식의 근본적인 문제점으로 공간 및 시간적인 효율성의 저하를 초래한다. 또한 데이터 점들을 서로소인 집합으로 분할하기 어렵기 때문에 엄밀한 정의의 클러스터링에는 한계가 있다. PROCLUS는 고차원 공간에서 클러스터를 탐색하는데 프로젝트된 클러스터링 개념을 논의한 알고리즘으로서 부분차원에 존재하는 클러스터를 효과적으로 찾을 수 있는 접근방법이다. 그러나 PROCLUS는 주로 특정 차원에 국한된 희소 데이터 분석에만 용이할 수 있으며, 대량의 고차원 데이터에 대해서는 최상의 medoid를 선택하는데 어려움이 따른다.

3. CLIP에 의한 고차원 클러스터링



(그림 1) CLIP의 개요

CLIP에 의한 클러스터링은 (그림 1)과 같다. 1단계(Phase 1)에서는 고차원 데이터를 선형변환하기 위해 차원별 프로젝션을 하여 밀집영역 즉 1차원적 클러스터를 결정한다. 2단계(Phase 2)에서는 1단계에서 결정된 밀집영역에 대해 차원을

증가시키며 점진적으로 프로젝션하는 클러스터링을 수행한다. 이와 같은 클러스터링은 대부분의 잡음을 효과적으로 제거할 수 있다. 또한 차원 전체뿐 아니라 부분차원에 존재하는 클러스터를 탐사할 수 있다. 3단계(Phase 3)에서는 탐사된 클러스터에 대해 백 트레이스를 적용하여 클러스터의 정확한 명세를 얻게 되는데, 이는 하이퍼큐브 형태의 클러스터를 향후 OLAP 등에 응용할 수 있기 때문이다.

3.1 CLIP 알고리즘

3.1.1 밀집영역의 결정

CLIP에 의한 클러스터링에 있어서 중요하게 고려할 사항은 각 차원에서의 밀집영역 즉 1차원적 클러스터를 찾는데 있다. CLIP이 사용하는 방법은 다음과 같다.

데이터 공간의 각 차원을 겹침이 없는 단위(u)로 분할하며 그 단위들은 모든 차원을 일정한 ξ 간격으로 나누어 얻어진다. 각 단위 u는 각 애트리뷰트에서 한 간격의 교차점으로서 각 차원은 집합 $\{u_1, u_2, \cdots, u_d\}$로 정의하며, 하나의 데이터 점 $v = \langle v_1, v_2, \cdots, v_d \rangle$는 하나의 단위 $u = \{u_1, u_2, \cdots, u_d\}$에 포함된다. 이때 한 단위의 선택(selectivity) 여부는 단위 안에 포함된 점들의 수로 정의하며, "단위 u가 밀집(*dense*)하다"는 것은 $(selectivity(u) \geq \tau)$을 만족할 때임을 뜻한다. 클러스터를 형성하는 밀집영역은 연결된 밀집단위(dense unit)들의 연결된 최대 집합(maximal set)으로 정의하며, 이에 따라 밀집영역의 최소 및 최대값을 결정한다. 다음의 (그림 2)는 영역의 밀집 여부(1, 0, null)를 판단하여 1차원적 클러스터를 결정하는 알고리즘이며 (그림 1)의 1단계에 해당한다.

```
/* Input  : Data_rec, τ
   Output : Found (1, 0, null) */
procedure Find_Dense(Data_rec) {
  프로젝션된 입력 데이터를 조사한다.
  데이터 영역의 밀도를 계산한다.
  if (밀도 ≥ τ) then
    Found = 1 ;
    밀집영역의 min, max 값을 결정한다.
    dense_region[]의 min, max 값을 갱신한다.
  else if (밀도 < τ) then
    Found = 0 ;
  else Found = null ;
return Found ; }
```

(그림 2) 밀집영역 결정 알고리즘

3.1.2 클러스터의 생성

CLIP에 의한 클러스터링은 (그림 3)과 같으며 (그림 1)의 2단계 및 3단계에 해당한다. 데이터 공간에서 각 차원을 형성하는 차원에 대하여 축-평행하게 프로젝션한 후, 데이터의 밀도를 계산하여 밀도 임계값 τ 이상인 밀집영역을 (그림 2)의 방법에 의해 찾는다. 그 다음, 밀집영역에 해당하는 초월사각형(hyper-rectangle) 부분에 존재하는 레코드에 대해서만 그 다음 차원 값을 프로젝션한다. 부분차원에서 이러한 초월사각형 안의 데이터 점들의 밀도는 평균 밀도보다 매우 크다. 이처럼 CLIP은 전체 k차원 데이터 공간에서, 1차원에서 k차원까지 각 차원에 해당하는 밀집영역을 그 다음 차원에 반영하여 점차적으로 조사할 데이터 공간 및 잡음 데이터를 감소시킨다. 만약 각 차원에 있어서 이전 차원으로부터 종속적으로 결정된 영역의 밀도가 τ를 초과하지 않는다면, 그 차원은 제외시키고 그 다음 차원을 조사한다. 그 이유는 데이터가 균등하게 분포한 영역이거나 또는 임계값을 초과하지 않는 영역은 클러스터 형성에 관련성이 매우 적은 차원이기 때문이다. CLIP은 클러스터 형성에 관련된 부분차원에서 클러스터를 탐사하지만 후보 차원의 모든 조합을 고려할 수 없으므로 클러스터로 인정할 차원의 개수(Num_subspace)를 결정한다. state_list는 클러스터를 형성하는 차원의 연관성을 나타내는 배열로서 각 요소는 밀집영역에 관한 통계 정보를 저장한다. dense_region은 각 차원에 존재하는 밀집영역의 최소, 최대값을 저장하는 배열이다. 차원이 증가함에 따라 dense_region에 저장된 값은 갱신된다.

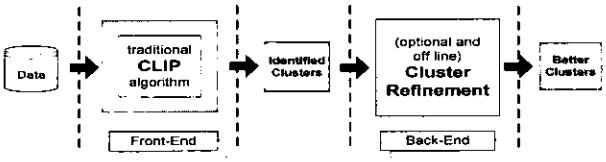
```
/* Input : Data_rec(입력 데이터 레코드 집합), ξ(그리드의 크기)
           Num_subspace(클러스터로 인정할 부분차원 개수)
   Output : Identified Cluster(명세된 클러스터), 데이터 ID */
procedure GenerateCluster (Data_rec, Num_subspace) {
  Data_rec의 모든 차원에 대해 선형 프로젝션한다.
  state_list[k][k], dense_region[ ]을 초기화한다.
  for 전체 k-차원에 대해 {
    for i = 1 to (k-Num_subspace+1) do {
      이전 차원에서 조사된 데이터 ID를 제외한 i-차원의 데이터
      ID를 읽는다.
      Found = Find_Dense(Data_rec) ;
      if (Found equal to 0) then
        i = i + 1 ;
      for j = i + 1 to k do {
        i-차원에 종속적인 (i + 1)차원의 데이터 ID를 읽는다.
        Found = Find_Dense ( ) ;
        state_list[i][j]의 각 요소의 값(0, 1, null)을 결정한다.
        if (j equal to (k-Num_subspace+2)) then
          state_list의 "1"의 수를 검사한다.
          if ("1"의 수 = 1) then
            break For (i+1 ≤ j ≤ k) ;
        else j = j + 1 ;
      }end For j = i + 1 to k
      i = i + 1 ;
    }end For i = 1 to (k-Num_subspace+1)
    dense_region[ ]에 포함된 데이터 ID들에 대한 min, max 값을
    읽는다.
    백 트레이스를 이용하여 각 차원의 min, max 값을 명세한다.
  }end For (k-dimensions)
}end Procedure GenerateCluster
```

(그림 3) CLIP에 의한 클러스터링

3.2 제안하는 확장된 CLIP 알고리즘

본 논문에서 제안하는 확장된 CLIP 알고리즘은 (그림 4)와 같이 크게 전반부(front-end)와 후반부(back-end)로 나눌 수 있다. 전반부는 기존의 CLIP 알고리즘이며 후반부는 확장된

CLIP에서 추가된 부분으로서 반복적인 2차원 프로젝션에 의해 사용자가 원하는 수준으로 클러스터 형태를 구체화하고 고차원의 잡음을 필터링한다. 즉 후반부는 보다 정제된 품질의 클러스터를 얻기 위한 과정으로서 도메인이나 응용분야에 따라 선택적으로 수행한다.



(그림 4) 확장된 CLIP의 개요

3.2.1 반복적 2차원 프로젝션에 의한 클러스터 정제

(그림 4)의 후반부인 클러스터 정제 과정은 기존의 CLIP 알고리즘으로 식별된 클러스터에 속한 데이터 집합을 대상으로 한다. 이 과정의 목적은 첫째 프로젝션 방법에서 간과하기 쉬운 클러스터의 실제 형태를 예측하고, 둘째 고차원의 잡음을 필터링하며 셋째, 사용자가 원하는 수준의 클러스터를 찾는데 있다.

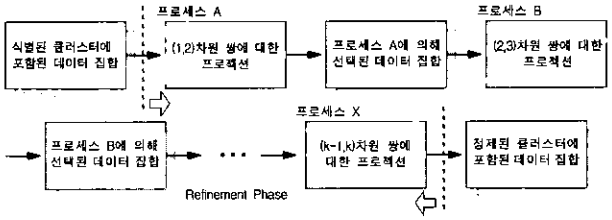
점진적인 1차원적 프로젝션으로 결정된 클러스터는 각 차원에 대해 초월사각형 형태의 영역이다. 이때 실제 클러스터에 형성하는 영역의 크기는 이보다 작을 가능성이 크다. 즉 *Size*(*CLIP* 에 의한 초월사각형 영역) ≥ *Size*(실제 클러스터 영역)이며, 두 영역의 차이는 잡음으로 간주할 수 있다. 잡음 데이터 판단의 경우, 1차원 프로젝션 방법에 의하여 1차원에서는 완전한 클러스터를 찾았으므로 더 이상 1차원적 잡음은 존재하지 않으나 고차원의 잡음은 존재할 수 있다. 그러나 직접적인 고차원 데이터 필터링은 연산의 복잡성에 의하여 매우 큰 계산비용이 소요된다. 따라서 적은 비용으로 실현 가능한 2차원 필터링을 반복적으로 적용하여 고차원 필터링을 근사적으로 구현하고자 하는데, (정의 1)과 같은 2차원 프로젝션을 적용한다.

(정의 1) 2차원 프로젝션 (two-dimensional projection)
전체 k차원 공간 R^k의 데이터 집합에서 (i,j)축으로의 2차원 프로젝션 $P_{i,j}$는 다음과 같이 정의한다.

$P_{i,j} : R^k \rightarrow R^k$;
$P_{i,j}(x_1, x_2, \cdots, x_k) = (0, \cdots, 0, x_i, 0, \cdots, 0, x_j, 0 \cdots, 0)$이다.
즉 (x_i, x_j) 차원을 제외한 모든 항은 0(zero)이다.

방법은 전체 k차원 클러스터의 정제를 위하여 임의 2차원의 쌍 (i,j)에 대하여 반복적으로 프로젝션을 하는 것이다. 이 경우 연산횟수가 총 ${}_kC_2 = k(k-1)/2$ 번이라는 많은 반복이 필요하다. 따라서 본 논문에서는 이 중 일부의 (i,j) 쌍을 선택하는 부분적 정제(partial refining)를 적용한다. CLIP의 전반부에 의해 결정된 클러스터를 대상으로 하는 클러스터 정제 알고리즘은 (그림 6)과 같다. (그림 5)는 연산횟수가 (k-1)번인 클러스터 정제 과정을 보이고 있다.

2차원 프로젝션에 적용한 그리드(ξ')는 알고리즘 전반부의 클러스터링에서 사용했던 것보다 세분화된 크기이며, 클러스터링을 조절하는 밀도 임계값(τ')은 입력받는다. 이때 임계값을 크게 하면 많은 객체가 포함된 셀들이 선택되어 보다 일반화되고, 작게 하면 적은 객체가 포함된 셀들도 선택되어 좀더 세분화된다. 그러므로 사용자는 임계값을 변화시키면서 원하는 수준의 클러스터를 발견할 수 있다. (i,j)축으로 이루어진 2차원 부분 공간에서, 각 i 축($1 \le i \le k$)은 n_i 개의 그리드 셀로 나뉘며, 셀들의 총 개수는 $n_i \times n_j$이다. (그림 7)의 *min* 과 *max* 는 CLIP에 의한 클러스터에 존재하는 값에 대한 각 축의 최소값 및 최대값을 의미한다. 따라서 i 축 그리드 셀의 크기(ξ')는 $\lceil \frac{(\max_i - \min_i)}{n_i} \rceil$ 로 계산된다.



(그림 5) 클러스터 정제 과정

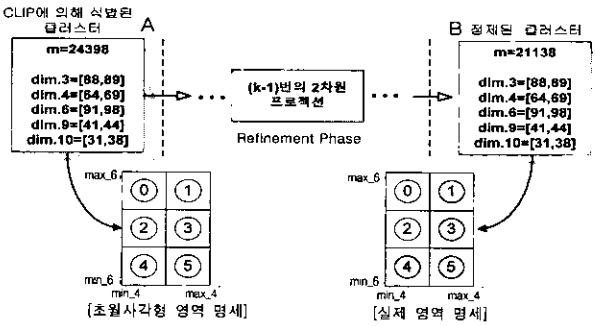
```
/* Input : object IDs in identified Cluster
           (식별된 클러스터 안의 데이터 점들),
           ξ'(그리드 크기), τ'(밀도 임계값), 프로젝션할 차원 쌍
   Output : Better Cluster (향상된 품질의 클러스터) */
 procedure RefinedCluster( ) {
    for 2차원 쌍에 대해 {
       데이터 집합을 2차원 프로젝션한다.
       각 차원의 (밀도 ≥ τ')인 영역을 교차-곱 연산한다.
       교차-곱 영역에 해당하는 2차원 셀을 DFS로 방문한다.
       for 모든 셀에 대해 {
         if (셀 밀도 ≥ τ') then
           셀에 포함된 데이터들을 선택한다.
         else
           셀에 포함된 데이터들을 제외한다. /*잡음 데이터 필터링*/
       }
    }
 return Better Cluster ; }
```

(그림 6) 클러스터 정제 알고리즘

3.2.2 클러스터 정제 예

(그림 7)은 정제 과정에 의해 클러스터를 형성하는 영역을 더욱 구체화하는 예를 보인다. 전체 10차원인 데이터 집합에 대해 CLIP의 수행으로 식별된 클러스터인 A에 포함된 데이터 개수 m은 24,398이며, 클러스터 형성에 관련된 차원은 3, 4, 6, 9, 10 차원이다. 또한 각 차원의 최소, 최대값에 의한 클러스터 명세는 그림에 표현한 바와 같다. 그 다음 반복적 2차원 프로젝션에 의해 정제된 클러스터인 B를 살펴보면, 각 차원

의 명세는 같으나 데이터 개수는 21,138개로 감소되었다. 이는 (4,6)차원의 프로젝션에서 정제된 영역으로 설명되는데, 클러스터A는 초월사각형 형태인 반면 클러스터B는 2차원 프로젝션에 의하여 밀집한 그리드가 선택되었기 때문이다. 즉 (4,6)차원으로 형성된 ⓪~⑤번 그리드 중에서 ①, ③, ④, ⑤번이 더욱 밀집한 것으로 조사됨에 따라 ⓪, ②번 그리드는 제외된다.



(그림 7) 클러스터 정제 예

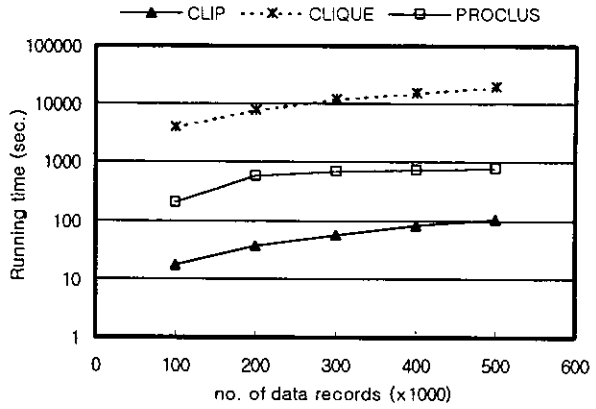
4. 실험 및 성능 분석

확장된 CLIP의 성능을 평가하기 위해 [3]에서 사용한 합성 데이터 집합을 이용하여 실험하였으며, 실험의 목표는 알고리즘의 효율성 및 효과성을 실험적으로 평가하는데 있다. 특히 효과성은 많은 양의 잡음을 포함한 데이터 집합에서 클러스터링 결과의 정확성을 평가한 것으로, 제안하는 알고리즘이 클러스터 형성에 밀접하게 관련된 그리드를 선택하여 더욱 의미있는 클러스터를 찾을 수 있는가를 실험하는 것이다.

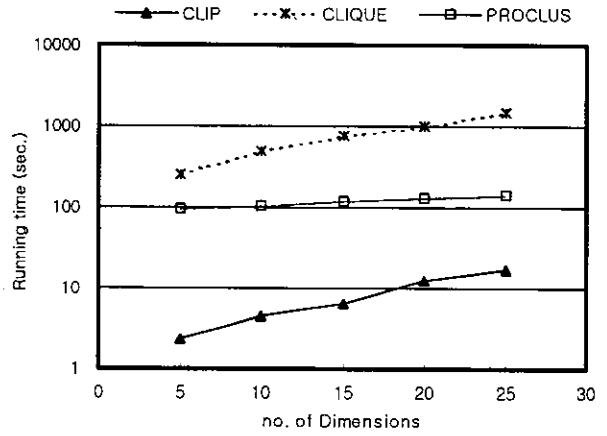
실험 환경은 256M 메인 메모리의 200-MHz SUN-Ultra SPARCⅡ워크스테이션에서, 데이터는 12GB SCSI 디스크에 저장하였다. 또한 C++언어와 LEDA-4.2 라이브러리 그리고 GNU g++ 컴파일러를 사용하여 구현하였다. 실험에서 CLIP에 대한 입력 값의 범위는 모든 애트리뷰트에 대해 [0,100]으로 고정하였으며, ξ=10으로 수행되었다.

4.1 효율성 실험 결과

본 절에서는 CLIP 알고리즘의 효율성을 기존의 부분차원 클러스터링 알고리즘과 비교한다. (그림 8)은 100,000에서 500,000 레코드까지 데이터베이스의 크기를 증가시킴에 따른 확장성을 보인다. 데이터 공간은 50차원이며 5개의 클러스터가 있다. 예상대로 수행시간은 데이터베이스 크기에 선형적으로 증가하였는데, 이는 데이터베이스를 스캔하는 횟수는 변하지 않기 때문이다. (그림 9)는 5개 클러스터가 있는 100,000 레코드의 데이터베이스에서 데이터 차원을 5에서 25까지 증가시킴에 따른 확장성을 보인다. CLIP에 의한 부분차원 클러스터링은 데이터 차원이 증가에 따라 수행시간도 선형적으로 확장한다는 것에 주목할 수 있다.



(그림 8) 데이터 개수에 대한 클러스터링 시간



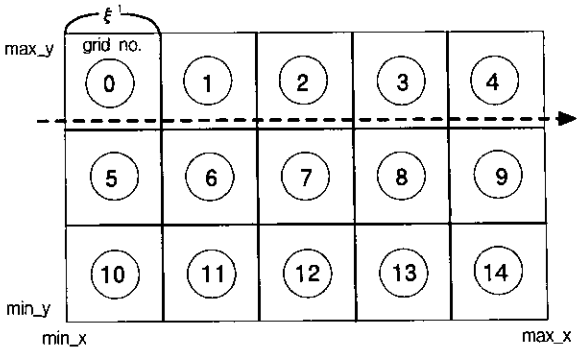
(그림 9) 데이터 차원에 대한 클러스터링 시간

4.2 효과성 실험 결과

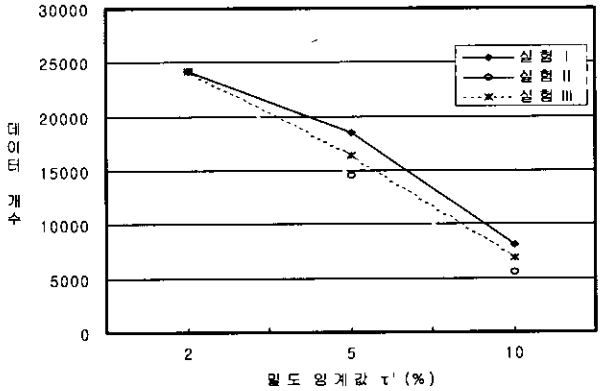
확장된 CLIP의 효과성 실험에서 사용한 2차원 쌍은 <표 1>과 같다. 2차원 프로젝션을 반복적으로 적용한 결과는 <표 2>, <표 3>과 같다. 데이터 집합은 전체 10차원이며 전체차원 및 부분차원으로 형성된 클러스터를 대상으로 한다. 표에서 나타내는 그리드 번호는 각 2차원에서 선택된 셀 번호를, m은 데이터 개수를 의미하며 (그림 10)의 방식에 의하여 부여한다. 예를 들어, <표 2>의 실험Ⅰ은 2차원 프로젝션을 (k/2)번 반복한 실험이며, 실험Ⅰ-1에서 실험Ⅰ-3은 밀도 임계값(τ')를 2%에서 10%까지 증가시킴에 따른 변화를 보이고 있다. ξ'=2이고 τ'=2%일 경우는 본래 클러스터링에서와 같은 비율의 밀도이므로 2차원 프로젝션을 반복하여도 데이터 개수에 변화가 없다. 그러나 τ'를 5%, 10%로 증가시키면 제외되는 그리드가 발생하며 이에 따라 데이터의 수도 감소하는 현상을 볼 수 있다. (k-1)번의 반복수행인 <표 3>에서 클러스터 ID가 A인 경우를 분석해 보면 다음과 같다. CLIP에 의한 클러스터링으로 식별된 클러스터의 개수는 24,130인데, τ'가 2%인 실험Ⅱ-1의 수행에서는 데이터 개수에 변화가 없다. 그러나 τ'를 5%로 증가시키면 (5,6)차원에

서는 ⓪, ①번 그리드가 제외되고, (6, 7)차원에서는 ⓪, ①, ②번 그리드가 제외되므로 결과적인 데이터 개수는 14,524개로 감소된다. 또한 τ'를 10%로 증가시키면 (4, 5) 및 (5, 6)차원에서는 ⓪번 그리드가 제외되고, (6, 7)차원에서는 ⓪~⑭번 그리드가 제외되며 결과적인 데이터 개수는 5,625로 감소된다. 이는 상대적으로 밀도가 희소한 그리드 셀은 제외된 결과이며, 밀도 임계값이 클수록 많은 데이터 객체가 포함된 그리드 셀들이 선택되어 더욱 일반화된 클러스터가 조사된 것이다. 참고로, 표에서 '동일'로 표시한 것은 CLIP에 의해 식별된 클러스터와 같은 그리드 셀을 포함한다는 것이다.

<표 4>는 CLIP에 의한 클러스터에 대해 각각의 정제 실험을 수행한 결과이다. 반복적인 2차원 프로젝션 후의 데이터 개수 m은 CLIP에 의한 클러스터의 데이터 수보다 감소하는 현상을 볼 수 있는데, 이는 밀도 임계값(τ')의 크기에 따라 그리드 셀이 제외되기 때문이다. 각 실험에서 τ'의 값이 2%일 경우에는 데이터 개수의 변화가 거의 없는 반면, 5%나 10%일 경우에는 데이터 개수의 차이를 점차 크게 보인다. 이는 각각 사용자가 원하는 수준의 클러스터가 조사된 것이다. 또한 프로젝션할 2차원 쌍에 따른 결과적인 데이터 개수를 분석해 보면, 프로젝션의 반복횟수가 많을수록 데이터 개수의 차이를 보이며 이는 보다 근사적인 정제임을 의미한다. 각 차원의 [최소, 최대]값 명세를 살펴보면 클러스터 정제 과정을 통해 다소 변화가 있으며, 이는 사용자가 원하는 수준의 클러스터 실제 형태에 더욱 근접하게 차원을 명세한 것이다. 표에서 '불변'으로 표시한 것은 CLIP에 의해 식별된 클러스터의 최소, 최대값에 변화가 없음을 의미한다. (그림 11)은 클러스터 A에 관해 실험 I, II, III을 수행한 다음 클러스터에 포함된 데이터 개수의 변화를 나타낸 것이다.



(그림 10) 2차원 프로젝션에 의한 (3×5) 그리드



(그림 11) 밀도 임계값에 따른 데이터 개수

〈표 1〉 프로젝션할 대상 차원

실험	반복 횟수	적용 차원
I	$(k/2)$번	(1,2) (3,4) (5,6) (7,8) (9,10)
II	$(k-1)$번	(1,2) (2,3) (3,4) (4,5) (5,6) (6,7) (7,8) (8,9) (9,10)
III	k번	(1,3) (2,4) (3,5) (4,6) (5,7) (6,8) (7,9) … (10,2)

〈표 2〉 반복적 2차원 프로젝션 결과(I)

Cluster ID	차원 쌍	CLIP에 의해 선택된 그리드		(k/2)번의 반복적 2차원 프로젝션					
				실험 I-1		실험 I-2		실험 I-3	
		총 그리드	그리드 번호	$\xi'=2$, $\tau'=2\%$		$\xi'=2$, $\tau'=5\%$		$\xi'=2$, $\tau'=10\%$	
				그리드	m	그리드	m	그리드	m
A	(1, 2)	(1×2)	⓪, ①	동일	24,130	동일	24,130	동일	24,130
	(3, 4)	(2×1)	⓪, ①	동일	24,130	동일	24,130	동일	24,130
	(5, 6)	(4×4)	⓪~⑮	동일	24,130	⓪,① 제외	18,478	⓪ 제외	8,136
	(7, 8)	(1×4)	⓪~③	동일	24,130	동일	18,478	동일	8,136
	(9, 10)	(1×2)	⓪, ①	동일	**24,130**	동일	**18,478**	동일	**8,136**
B	(1, 2)	(2×3)	⓪~⑤	동일	67,293	동일	67,293	⓪ 제외	56,648
	(3, 4)	(1×1)	⓪	동일	67,293	동일	67,293	동일	56,648
	(5, 6)	(2×2)	⓪~③	동일	67,293	동일	67,293	동일	56,648
	(7, 8)	(1×1)	⓪	동일	**67,293**	동일	**67,293**	동일	**56,648**
	10								
C	(3, 4)	(2×3)	⓪~⑤	②, ⑤	122,139	②, ⑤	122,139	②, ⑤	122,139
	(6, 9)	(3×3)	⓪~⑧	⑥,⑦,⑧ 제외	**122,139**	⑥,⑦,⑧ 제외	**122,139**	⑥,⑦,⑧ 제외	**122,139**
	10								

〈표 3〉 반복적 2차원 프로젝션 결과(Ⅱ)

Cluster ID	차원 쌍	CLIP에 의해 선택된 그리드		(k-1)번의 반복적 2차원 프로젝션					
		총 그리드	그리드 번호	실험Ⅱ-1		실험Ⅱ-2		실험Ⅱ-3	
				$\xi' = 2,\ \tau' = 2\%$		$\xi' = 2,\ \tau' = 5\%$		$\xi' = 2,\ \tau' = 10\%$	
				그리드	m	그리드	m	그리드	m
A	(1, 2)	(1×2)	⓪,①	동일	24,130	동일	24,130	동일	24,130
	(2, 3)	(1×1)	⓪	동일	24,130	동일	24,130	동일	24,130
	(3, 4)	(2×1)	⓪, ①	동일	24,130	동일	24,130	동일	24,130
	(4, 5)	(4×2)	⓪~⑦	동일	24,130	동일	24,130	⓪제외	16,740
	(5, 6)	(4×4)	⓪~⑮	동일	24,130	⓪, ① 제외	18,478	⓪제외	5,625
	(6, 7)	(4×4)	⓪~⑮	동일	24,130	⓪,①,② 제외	14,524	⓪,①,②,④,⑤,⑥,⑧,⑨,⑩,⑫,⑬,⑭ 제외	5,625
	(7, 8)	(1×4)	⓪~③	동일	24,130	동일	14,524	동일	5,625
	(8, 9)	(2×1)	⓪~②	동일	24,130	동일	14,524	동일	5,625
	(9, 10)	(1×2)	⓪, ①	동일	**24,130**	동일	**14,524**	동일	**5,625**
B	(1, 2)	(2×3)	⓪~⑤	동일	67,293	동일	67,293	⓪ 제외	56,648
	(2, 3)	(2×1)	⓪, ①	동일	67,293	동일	67,293	동일	56,648
	(3, 4)	(1×1)	⓪	동일	67,293	동일	67,293	동일	56,648
	(4, 5)	(2×1)	⓪, ①	동일	67,293	동일	67,293	동일	56,648
	(5, 6)	(2×2)	⓪~③	동일	67,293	동일	67,293	동일	56,648
	(6, 7)	(1×2)	⓪, ①	동일	67,293	동일	67,293	동일	56,648
	(7, 8)	(1×1)	⓪	동일	67,293	동일	67,293	동일	56,648
	(8. 10)	(1×1)	⓪	동일	**67,293**	동일	**67,293**	동일	**56,648**
C	(3, 4)	(2×3)	⓪~⑤	②, ⑤	122,139	②, ⑤	122,139	②, ⑤	122,139
	(4, 6)	(3×2)	⓪~⑤	동일	122,139	동일	122,139	⓪,② 제외	105,447
	(6, 9)	(3×3)	⓪~⑧	⑥,⑦,⑧ 제외	122,139	⑥,⑦,⑧ 제외	122,139	⑥,⑦,⑧ 제외	86,435
	(9, 10)	(4×3)	⓪~⑪	②,⑤,⑧,⑪ 제외	**122,139**	⓪,①,②,⑤,⑧,⑪ 제외	**113,569**	⓪,①,②,⑤,⑧,⑪ 제외	**80,329**

〈표 4〉 실험별 클러스터링 결과

Cluster ID		데이터 개수(m)	실험별 클러스터의 관련차원 [최소, 최대] 값									
			1	2	3	4	5	6	7	8	9	10
A	**CLIP**	**24,130**	[74,79]	[21,24]	[33,36]	[72,77]	[81,89]	[40,49]	[80,89]	[89,90]	[40,44]	[11,14]
	실험Ⅰ-1	24,130	불변	불변	불변	불변	불변	불변	불변	불변	불변	불변
	실험Ⅰ-2	18,478	불변	불변	불변	불변	불변	[42,49]	불변	불변	불변	불변
	실험Ⅰ-3	8,136	불변	불변	불변	불변	[83,89]	[46,49]	불변	불변	불변	불변
	실험Ⅱ-1	24,130	불변	불변	불변	불변	불변	불변	불변	불변	불변	불변
	실험Ⅱ-2	14,524	불변	불변	불변	불변	불변	[42,49]	불변	불변	불변	불변
	실험Ⅱ-3	5,625	불변	불변	불변	[74,77]	[83,89]	[46,49]	불변	불변	불변	불변
B	**CLIP**	**67,293**	[10,16]	[24,29]	[53,55]	[46,49]	[24,29]	[54,58]	[12,15]	[43,46]		[63,64]
	실험Ⅰ-1	67,293	불변	불변	불변	불변	불변	불변	불변	불변		불변
	실험Ⅰ-2	67,293	불변	불변	불변	불변	불변	불변	불변	불변		불변
	실험Ⅰ-3	56,648	불변	불변	불변	불변	불변	불변	불변	불변		불변
	실험Ⅱ-1	67,293	불변	불변	불변	불변	불변	불변	불변	불변		불변
	실험Ⅱ-2	67,293	불변	불변	불변	불변	불변	불변	불변	불변		불변
	실험Ⅱ-3	56,648	불변	불변	불변	불변	불변	불변	불변	불변		불변
C	**CLIP**	**122,139**			[82,89]	[64,69]		[91,98]			[41,48]	[30,38]
	실험Ⅰ-1	122,139			[88,89]	불변		불변			[41,44]	[31,38]
	실험Ⅰ-2	122,139			[88,89]	불변		불변			[41,44]	[31,38]
	실험Ⅰ-3	122,139			[88,89]	불변		불변			[41,44]	[31,38]
	실험Ⅱ-1	122,139			[88,89]	불변		불변			[41,44]	[31,38]
	실험Ⅱ-2	122,139			[88,89]	불변		불변			[41,44]	[32,38]
	실험Ⅱ-3	86,435			[88,89]	불변		[93,98]			[41,44]	[32,38]

5. 결 론

본 논문에서는 클러스터 정제 단계를 적용한 CLIP 알고리즘을 제안하였다. 제안하는 확장된 CLIP 알고리즘의 목적은 다음과 같다. 첫째 프로젝션 방법에서 간과하기 쉬운 클러스터의 실제 형태를 추정한다. 둘째 고차원의 잡음을 필터링하여 클러스터의 품질을 향상시킨다. 셋째, 사용자가 원하는 수준의 다양하고 복잡한 형태의 클러스터를 발견할 수 있다.

기존의 CLIP은 선형 프로젝션에 의한 클러스터링으로 인해 오직 하이퍼큐브 형태로 클러스터를 명세하는 것이 문제

점이다. 또한 잡음 데이터 판단의 경우, 1차원적 프로젝션 방법에 의하여 각 차원에서는 완전한 클러스터를 찾았으므로 더 이상 1차원적 잡음은 존재하지 않으나 고차원의 잡음은 존재할 수 있다. 그러나 직접적인 고차원 데이터 정제는 연산의 복잡성에 의하여 매우 큰 계산비용이 소요된다. 따라서 실현 가능한 2차원 프로젝션을 반복적으로 적용하는 과정을 기존의 CLIP에 추가하여 클러스터 정제 및 고차원 클러스터링을 근사적으로 구현하고자 하였다.

기존의 CLIP에 의해 조사된 하나의 클러스터가 상이한 밀도의 클러스터를 여러 개 포함할 경우가 있다. 이때 2차원 프로젝션에 의한 곱집합에서 그리드 크기 및 밀도 임계값을 사용자 의도로 조절하여 클러스터를 재조사할 수 있었다. 즉 밀도가 일정치 않은 데이터 집합에 대해서도 클러스터의 정제 과정에 의해 사용자가 원하는 임계값의 클러스터를 발견할 수 있음을 보였다. 이는 다양한 밀도의 클러스터들이 존재하는 경우에 사용자가 요구한 밀도보다 낮은 클러스터는 세분화되어 제외되고, 해당 밀도보다 높은 부분만이 클러스터로 발견됨을 의미하며 이에 대한 일련의 실험을 통해 알고리즘의 효과성을 입증하였다.

향후 연구는 최적의 클러스터 정제를 위해 프로젝션할 차원의 선택 및 클러스터의 결정에 중요한 영향을 주는 밀도 임계값이나 그리드의 크기 등 입력 매개변수의 선택을 지원하는 시스템을 개발할 계획이다.

참 고 문 헌

[1] Charu C. Aggrawal, Cecilia 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.

[2] Charu C. Aggrawal, Philip S. Yu, "Finding Generalized Projected Clusters in High Dimensional Spaces," *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pp.70-81, 2000.

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

[5] S. Berchtold, D. A. Keim, C. Böhm, H.-P. Kriegel, "A Cost Model For Nearest Neighbor Search in High-Dimensional Data Space," *Proc. of the 16th Symposium on Principles of Database Systems (PODS)*, pp.78-86, 1997.

[6] S. Berchtold, D. A. Keim, "High-dimensional Index Structures, Database Support for Next Decade's Applications," *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, 1998.

[7] Kaushik Chakrabarti, Sharad Mehrotra, "Local Dimensionality Reduction : A New Approach to Indexing High Dimensional Spaces," *Proc. of 26th Int. Conf. on VLDB*, pp. 89-100. 2000.

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

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

[10] Fayyad, U. M., et al., *Advances in Knowlwdge Discovery and Data Mining*, AAAI Press/The MIT Press, pp.307-328, 1996.

[11] Hinneburg A., "Mining for High Dimensional Cluster using Projection and Visualizations," *Proc. of the EDBT 2000 phD Workshop*, 2000.

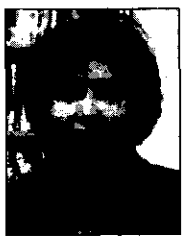
[12] Hinneburg A., Keim D. A, "Opimal Grid-Clustering : Towards breaking the Curse of Dimensionality in High-Dimensional Clustering," *Proc. of 25th Int. Conf. on VLDB*, pp.506-517, 1999.

[13] 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.

[14] 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.

[15] 이혜명, 박영배, "고차원 데이터에서 점진적 프로젝션을 이용한 클러스터링", 한국정보과학회 가을학술발표논문집(Ⅰ), 2000.

[16] 이혜명, 박영배, "점진적 프로젝션을 이용한 고차원 클러스터링", 한국정보과학회논문지, 제28권 제4호, 2001.



이 혜 명

e-mail : hmlee@kmc.ac.kr

1989년 명지대학교 공학사(전자계산학)

1993년 명지대학교 공학석사(전자계산학)

1997년 명지대학교 박사과정 수료 (컴퓨터공학)

1998년~현재 경문대학 컴퓨터정보과 조교수

관심분야 : 데이터마이닝, 웹 DB, 전자상거래 등

박 영 배

e-mail : parkyb@mju.ac.kr

1974년 동아대학교 전기공학(공학사)

1980년 연세대학교 전자계산학(공학석사)

1993년 서울대학교 컴퓨터공학(공학박사)

1974년~1981년 한국전력공사 전자계산소 과장대리

1990년~1992년 명지대학교 전자계산소 소장

1993년~2000년 중앙전산개발경진대회(행정자치부) 심사위원장

1997년~2001년 산업대학원 원장

1981년~현재 명지대학교 컴퓨터공학과 교수로 재직, 데이터베이스, 자료구조, 파일처리 등을 강의

관심분야 : Spatial, Multidimensional, Web, Scientific Databases, Data Warehousing and Data Mining, System Integration 등