문서 클러스터링

국민대학교 소프트웨어학부
강 승 식

클러스터 분석(Cluster analysis)

- 범주 구조를 생성하는 통계적 기법
  - Statistical technique to generate a category structure which fits a set of observations.

- 동일 그룹에 속하는 개체들은 결속력이 강하고, 다른 그룹에 속하는 개체들은 결속력이 약함
  - High degree of association between members of the same group and a low degree between members of different groups.

- 자동 문서 분류와 유사하지만 클러스터 그룹이 알려져 있지 않다는 점에서 다름
  - Similar to automatic classification, but different in that classes are not known prior to processing

Document clustering, term clustering

- 문서 클러스터링은 문서에 출현한 term들을 기반으로 함
  - Documents may be clustered on the basis of the *terms* that they contain.
- 문서 클러스터링은 함께 인용(citation)되는 것을 기반으로 함
  - Documents may be clustered based on *co-occurring citations* in order to provide insights into the nature of the literature of a field.
- 어휘 클러스터링(term clustering)은 어휘들이 함께 출현한 문서들을 기반으로 함
  - *Terms may be clustered* on the basis of the documents in which they co-occur.

유사도 계산 기법

- Means of quantifying the degree of association between documents(terms)

  → Distance measure, or a measure of similarity/dissimilarity

- Similarity measures
  - Dice coefficient
  - Jaccard coefficient
  - Cosine coefficient

Similarity Measures

- **Dice coefficient**

$$S_{D_i,D_j} = \frac{2\sum_{k=1}^{L}(weight_{ik}\ weight_{jk})}{\sum_{k=1}^{L} weight_{ik}^2 + \sum_{k=1}^{L} weight_{jk}^2}$$

Binary term weights

$$S_{D_i,D_j} = \frac{2C}{A+B}$$

A : the number of terms in D_i

B : the number of terms in D_j

C : D_i 와 D_j 의 공통 용어 수

- **Jaccard coefficient**

$$S_{D_i,D_j} = \frac{\sum_{k=1}^{L}(weight_{ij}\ weight_{jk})}{\sum_{k=1}^{L} weight_{ik}^2 + \sum_{k=1}^{L} weight_{jk}^2 - \sum_{k=1}^{L}(weight_{ik}\ weight_{jk})}$$

- **Cosine coefficient**

$$S_{D_i,D_j} = \frac{\sum_{k=1}^{L}(weight_{ij}\ weight_{jk})}{\sqrt{\sum_{k=1}^{L} weight_{ik}^2 \sum_{k=1}^{L} weight_{jk}^2}}$$

유사도 행렬(Similarity Matrix)

- The similarity between every pair of documents
- Symmetric → *lower triangular matrix*

$$S = \begin{vmatrix} & & & & \\ s_{21} & & & & \\ s_{31} & s_{32} & & & \\ s_{41} & s_{42} & s_{43} & & \\ \vdots & \vdots & \vdots & \ddots & \\ s_{N1} & s_{N2} & s_{N3} & \cdots & s_{N(N-1)} \end{vmatrix}$$

Figure 16.1 Similarity matrix

- Identifying a *nearest neighbor(NN)*
  - Find the closest vector to a given vector from a set of *N* vectors.
  - Efficient NN-finding algorithm → inverted file algorithm

Clustering 방법

- Goal: N objects → M groups
  - N >> M and M is usually unknown

- Agglomerative vs. Divisive(응집과 분할)
  - 응집 : unclustered data set → N-1 pairwise joins
  - 분해 : all objects in a single cluster → N-1 divisions of some cluster into a smaller cluster

- Nonhierarchical methods(비계층적 방법)
- Hierarchical methods(계층적 방법)

Clustering Algorithms

- Connectivity-based clustering
  - Hierarchical clustering: SLINK, CLINK, GA Link
- Centroid-based clustering
  - k-means algorthm
- Distribution-based clustering
  - Cluster의 분포를 중요시함
  - 각 cluster들의 크기가 일정함
- Density-based clustering
  - Cluster의 밀집도를 중요시함

Nonhierarchical Methods: 비계층적

- Partitioning and reallocating items until some criterion is optimized.
  - 자료집합 N이 클러스터 M보다 매우 큰(M<<N) 자료 집합을 분할하는데 계층적 방법보다 효율적
  - 계산자원에 한계가 있었던 초기의 클러스터링 연구에 사용
- 경험적인 결정을 요구
  - Heuristic in nature, since a priori decisions about *the number of clusters, cluster size, criterion for cluster membership, and form of cluster representation* are required.
- 단일패스 방법 vs. 재배치 방법

비계층적 방법: 단일패스 알고리즘

- **Single Pass Methods**
  1. Assign the first document D_1 as the representative for C_1.
  2. For D_i, calculate the similarity S with the representative for each existing cluster.
  3. If $S_{max} > S_T$(threshold), add the item to the corresponding cluster and recalculate the cluster representative; otherwise, use D_i to initiate a new cluster.
  4. If an item D_i remains to be clustered, return to step 2.

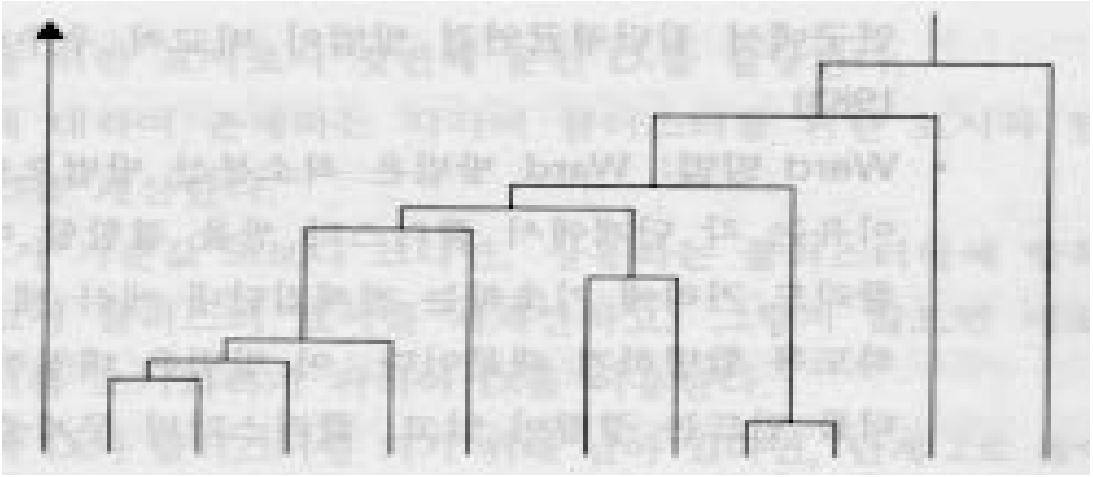
비계층적 방법: 재배치 알고리즘

- **Reallocation Methods**
  - Beginning initial partition of the data set
  - Moving items from cluster to cluster to obtain an improved partition.

1. Select *M* cluster representatives or centroids.
2. For i = 1 to *N*, assign D_i to the most similar centroid.
3. For j = 1 to *M*, recalculate the cluster centroid C_j.
4. Repeat steps 2 and 3 until there is little or no change in cluster membership.

Hierarchical Methods

- *Hierarchical Agglomerative Clustering Method : HACM*
- Dendrogram(역수형도) → 생성된 클러스터의 구조
  - The order of pairwise coupling of the objects is shown and the value of the similarity function(level) at which each fusion occurred.



- **Single/complete/group-average link, Ward's method**

계층적 클러스터링 알고리즘

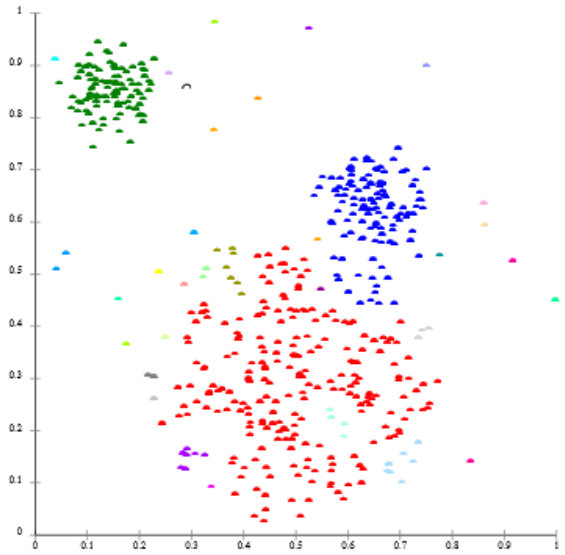
- General algorithm for HACM
  1. Identify the two closest points and combine them in a cluster.
  2. Identify and combine the next two closest points(treating existing clusters as points).
  3. If more than one cluster remains, return to step 1.

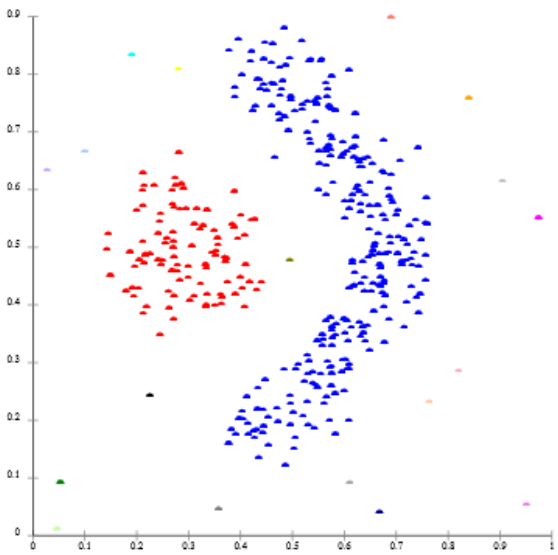
Single Link Method

- Joins the most similar pair of objects.
  - Implemented relatively efficiently
- Long straggly clusters, or chaining
  - Suitable for delineating ellipsoidal clusters
- Unsuitable for isolating spherical or poorly separated clusters
- Complexity: $O(N\log N) \sim O(N^5)$

SLINK on density-based clusters

- 2개의 작은 클러스터로 분할(left)
- 길쭉한 형태의 클러스터 생성(right)





Complete Link Method

- Use the least similar pair between each of two clusters to determine the intercluster similarity
- All entities in a cluster are linked to one another within some minimum similarity.
- Small, tightly bound clusters
- Difficult to apply to large data sets

Group Average Link Methods

- The similarity between two clusters is determined by the average value of all the pairwise links between points.

Ward's Method

- Minimum variance method
  - Minimizes the increase in the total within-group error sum of squares, based on the Euclidean distance between centroids.

- It tends to produce homogeneous clusters and a symmetric hierarchy.

- Its definition of a cluster center of gravity provides a useful way of representing a cluster.

Ward's Method

- Reciprocal Nearest Neighbor algorithm
  - For any point or cluster, there exists a chain of nearest neighbors(NNs)
  1. Select an arbitrary point.
  2. Follow the NN chain from this point till an RNN pair is found.
  3. Merge these two points and replace them with a single point.
  4. If there is a point in NN chain preceding the merged points, return to step 2; otherwise return to step 1. Stop when only one point remains.

Evaluation

- Determine the “best” clustering method by
  - applying a range of clustering methods to test data sets
  - and comparing the quality of the results

- Voohees found that
  - Complete link → most effective for larger collections
  - Complete and group average → comparable for smaller collections
  - Single link → worst performance

- El-Hamdouchi and Willett
  - Group average → most suitable for document clustering
  - Complete link → not as effective as in Voorhees

k-Means 알고리즘

- i번째 클러스터의 중심 μ_i을 , 클러스터에 속하는 점의 집합을 S_i라고 할 때, 전체 분산은 다음과 같이 계산

$$V = \sum_{i=1}^{k} \sum_{j \in S_i} | x_j - \mu_i |^2$$

- 이 값 V를 최소화하는 S_i을 찾는 알고리즘

- 우선 초기의 μ_i를 임의로 설정한 후에 아래 두 단계를 반복
  1) 클러스터 설정: 각 점에 대해, 그 점에서 가장 가까운 클러스터에 할당
  2) 클러스터 중심 재조정: μ_i를 각 클러스터에 있는 점들의 평균값으로 재설정
- 만약 클러스터가 변하지 않는다면 반복을 중지한다.

- 맨 처음, 각 점들을 k개 집합으로 분할
  1) 임의로 분할 혹은 적당한 휴리스틱을 사용
  2) 각 집합의 무게중심 계산
  3) 각 점들을 방금 구한 무게중심 가운데 제일 가까운 것에 연결하여 집합을 재구성
  4) 이 작업을 반복하면 점들이 소속된 집합을 바꾸지 않거나, 무게중심이 변하지 않는 상태로 수렴

- 이 알고리즘은 간단하고 빠르게 수렴하여 널리 사용
  - 다만, superpolynomial 시간이 걸리는 경우도 있음
  - 이 알고리즘은 전역 최적값을 보장해 주지 않음
  - 초기 클러스터 설정에 따라서는 실제 최적값보다 꽤 나쁜 값을 얻을 수도 있음
- 이 알고리즘은 클러스터의 갯수를 미리 정해야 함
  - 클러스터 개수를 많게 하면 큰 클러스터가 여러 개로 분할될 수 있음

Document Clustering

1. Tokenization
2. Stemming and lemmatization
3. Removing stop words and punctuation
4. Computing term frequencies or tf-idf
5. Clustering
6. Evaluation and visualization