16. 클러스터링 알고리즘

1. Introduction

- Cluster analysis
  - Statistical technique to generate a category structure which fits a set of observations.
  - High degree of association between members of the same group and a low degree between members of different groups.
  - Similar to automatic classification, but different in that classes are not known prior to processing
  - Methods and algorithms are found in
    - statistical analysis packages: SAS, SPSSX, BMDP
    - cluster analysis packages: CLUSTAN, CLUSTAR/CLUSTID
- Applications in I.R.
  - Documents may be clustered on the basis of the *terms* that they contain.
  - Documents may be clustered based on *co-occurring citations* in order to provide insights into the nature of the literature of a field.
  - *Terms may be clustered* on the basis of the documents in which they co-occur.

2. Measures of Association

- Means of quantifying the degree of association between documents(terms).
  - → Distance measure, or a measure of similarity/dissimilarity
  - → Some methods use a specific measure

    e.g. Euclidean distance for Ward's method
- Weighting of document terms is not as significant in improving performance in cluster-based retrieval.
- 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**

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

Similarity Matrix

- Pairwise coupling of the most similar documents or clusters
- 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

- Similarity matrix can be the basis for identifying a *nearest neighbor(NN)* → find the closest vector to a given vector from a set of *N* multidimensional vectors.
- Efficient NN-finding algorithm → inverted file algorithm

3. Clustering Methods

- Goal: N objects → M groups
  - N >> M and M is usually unknown
- Agglomerative vs. Divisive
  - Agg. : unclustered data set → N-1 pairwise joins
  - Div. : all objects in a single cluster → N-1 divisions of some cluster into a smaller cluster
- Nonhierarchical methods
  - Partitioning and reallocating items until some criterion is optimized.
  - 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.
- Hierarchical methods

4. Nonhierarchical Methods (1/3)

- 경험적인 결정을 요구
- 최적의 해결책을 구하기는 불가능
- 자료집합 N이 클러스터 M보다 매우 크면(M<<N) 큰 자료 집합을 분할하는데 계층적 방법보다 효율적
- 계산자원에 한계가 있었던 초창기 문헌 클러스터링 연구에 사용

- 단일패스 방법
- 재배치 방법

Nonhierarchical Methods (2/3)

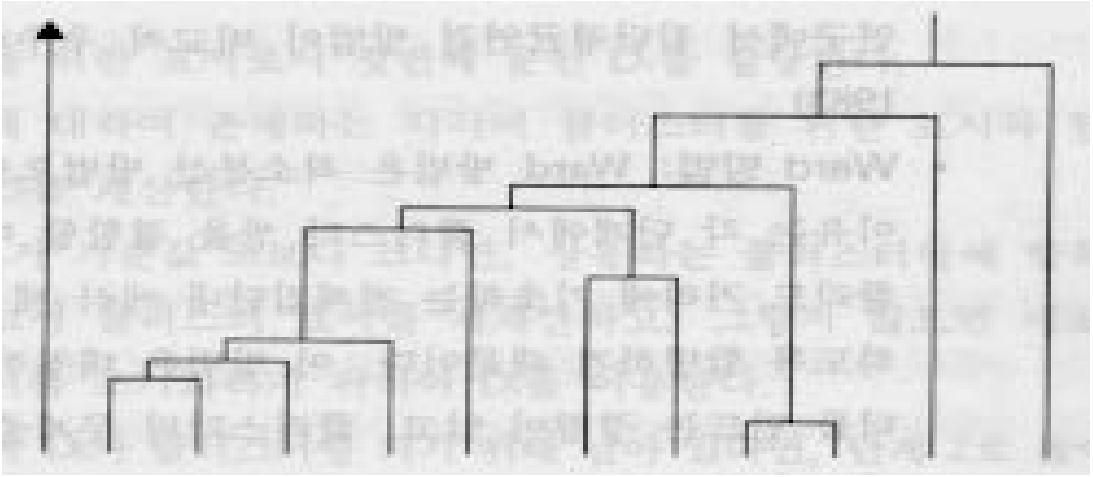
- **Single Pass Methods**
  1. Assign the first document D_1 as the representative for C_1.
  2. For D_i, calculate the similarity S with the representative for each existing cluster.
  3. If $S_{max} > S_T$(threshold), add the item to the corresponding cluster and recalculate the cluster representative;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

Nonhierarchical Methods (3/3)

- **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.

5. 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

General algorithm for HACM

- 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.
- Lance-Williams dissimilarity update formula
  - If objects C_i and C_j have just been merged to form cluster $C_{i,j}$, the dissimilarity d between the new cluster and any existing cluster C_k is given by:

$$d_{C_i,\ _jC_k} = \alpha_i d_{C_i C_k} + \alpha_j d_{C_j C_k} + \beta d_{C_i C_j} + \gamma \left| d_{C_i C_k} - d_{C_j C_k} \right|$$

Table 16.1 Characteristics of HACM

HACM	Lance-Williams parameters	Cluster centers
Single link	$\alpha_i = \frac{1}{2}$ $\beta = 0$ $\gamma = -\frac{1}{2}$	—
Complete link	$\alpha_i = \frac{1}{2}$ $\beta = 0$ $\gamma = \frac{1}{2}$	—
Group average	$\alpha_i = \frac{m_i}{m_i + m_j}$ $\beta = 0$ $\gamma = 0$	—
Median	$\alpha_i = \frac{1}{2}$ $\beta = -\frac{1}{4}$ $\gamma = 0$	$C_{i,j} = \frac{C_i + C_j}{2}$
Centroid	$\alpha_i = \frac{m_i}{m_i + m_j}$ $\beta = -\frac{m_i m_j}{(m_i + m_j)^2}$ $\gamma = 0$	$C_{i,j} = \frac{m_i C_i + m_j C_j}{m_i + m_j}$
Ward's method	$\alpha_i = \frac{m_i + m_k}{m_i + m_j + m_k}$ $\beta = -\frac{m_k}{m_i + m_j + m_k}$ $\gamma = 0$	$C_{i,j} = \frac{m_i C_i + m_j C_j}{m_i + m_j}$

Notes: m_i is the number of items in C_i; the dissimilarity measure used for Ward's method must be the increase in variance (section 16.5.5).

Single Link Method (1)

- Characteristics
  - Joins the most similar pair of objects.
  - It has some attractive theoretical properties.
  - It can be implemented relatively efficiently. → widely used
  - Long straggly clusters, or chaining
  - Suitable for delineating ellipsoidal clusters
  - Unsuitable for isolating spherical or poorly separated clusters
  - Complexity: $O(NlogN) \sim O(N^5)$
- Van Rijsbergen algorithm
  - Do not require the storage of the similarity matrix
  - $O(N^2)$ in time, $O(N)$ in space
- SLINK algorithm
  - Dendrogram is built by inserting one point at a time into the representation.
  - The hierarchy is generated in a form of *pointer representation*.
  - 3 arrays
    - *pi* : hold the pointer representation
    - *lambda* : hold the distance value associated with each pointer
    - *distance* : process the current row of the distance matrix

Single Link Method (2)

- Minimal spanning tree(MST) algorithm
  - Tree linking *N* objects with *N-1* connections → no loops]
  - The sum of the *N-1* dissimilarities is minimized.
- Fundamental construction principles for MST
  1. Any isolated point can be connected to a nearest neighbor.
  2. Any isolated fragment(subset of MST) can be connected to a nearest neighbor by a shortest available link.
- Prim-Dijkstra algorithm for MST
  1. Place an arbitrary point in MST and connect its nearest neighbor to it.
  2. Find the point not in MST closest to any point in MST, and add it to the fragment.
  3. If a point remains that is not in the fragment, return to step 2.

Complete Link Method

- Characteristics
  - Use the least similar pair between each of two clusters to determine the intercluster similarity
  - All entities in a cluster are linked to on e another within some minimum similarity.
  - Small, tightly bound clusters
  - Difficult to apply to large data sets
- Defay's CLINK algorithm
  - Analogous to SLINK algorithm
    - Uses the same three arrays: pi, lambda, distance
    - $O(N^2)$ in time, $O(N)$ in space
- Voorhees algorithm
  - Efficient for relatively large document collections
  - It is a variation on the sorted matrix approach
  - This requires a sorted list of document-document similarities, and a means of counting the number of similarities seen between any two active clusters.

Group Average Link Methods

- Characteristics
  - The similarity between two clusters is determined by the average value of all the pairwise links between points.
  - The general HACM algorithm can be used → impractical for large collection.
  - More efficient special case algorithm is available.
    - Vorhees algorithm

Ward's Method

- Characteristics
  - ***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.
  - Follows the general HACM algorithm
- 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.

6. Evaluation and Validation

- 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

Evaluation and Validation (2)

- Validation
  - Is the data matrix random?
  - How well does a hierarchy fit a proximity matrix?
  - Is a partition valid?
  - Which individual clusters appearing in a hierachy are valid?
- Three tests for clustering tendency
  1. Clustering tendency: is retrieval performance achieved?
  2. Overlap test: query-relevance test → RR, RNR overlap
  3. Nearest neighbor test: how many of its n nearest neighbors are also relevant?

7. Updating the Cluster Structure

- When new items are added, updating the cluster without the need to recluster the entire collection is desirable.

- Crouch's reallocation algorithm(1975)
  - Includes a mechanism for cluster maintenance

- Can and Ozkarahan(1989)
  - Strategy for dynamic cluster maintenance based on their cover coefficient concept.

8. Document Retrieval From A Clustered Data Set

- Document clustering
  - Improves the efficiency of retrieval.
  - Improves the effectiveness of retrieval.
  - Provides an alternative to Boolean or best match retrieval.
- Approaches to retrieval
  - Top-down search
    1. Enter the tree at the root and matching the query against the cluster at each node.
    2. Move down the tree following the path of greater similarity.
  - Bottom-up search
    1. Begins with some document or cluster at the base of the tree.
       - beginning document → an item known to be relevant
       - It can be obtained by a best match search of documents or lowest-level clusters.
    2. Moves up until the retrieval criterion is satisfied.

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
    - 이를 방지하려면, 서로 다른 초기값으로 여러 번 시도하여 가장 좋은 결과를 얻는 기법 등을 사용
- 이 알고리즘은 클러스터 개수를 미리 정해야 함
  - 클러스터 개수를 많게 하면 큰 클러스터가 여러 개로 분할될 수 있음