Clustering

What is clustering?

- **Clustering** is the process of grouping a set of objects into classes of similar objects
  - » Documents within a cluster should be similar.
  - » Documents from different clusters should be dissimilar.
- **Clustering** is the commonest form of *unsupervised learning*
  - » Unsupervised learning = learning from raw data where any classification of examples is not given
  - » Supervised learning = learning from supervised data where a classification of examples is given
- **Clustering** is a common and important task that finds many applications in IR and other places
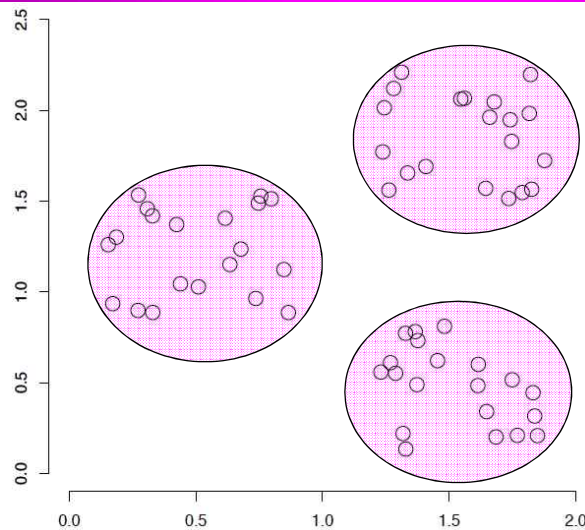
클러스터링을
이걸 주면 이건 뭘까? 하는거

클러스터링 레이블링을 해주는 알고리즘
레이블 자체를 안주기 때문에 비슷한 것끼리 묶어서 레이블링을 할 수 있다

suprevised learning과는 다르게 레이블링이 주어지진 않는다

이거는 뭐다 이거는뭐다
레이블링한 테이블을 주면
새로운 인스턴스가 주면 이건 뭘깡?

A data set with clear cluster structure

- How would you design an algorithm for finding these three clusters in this case?
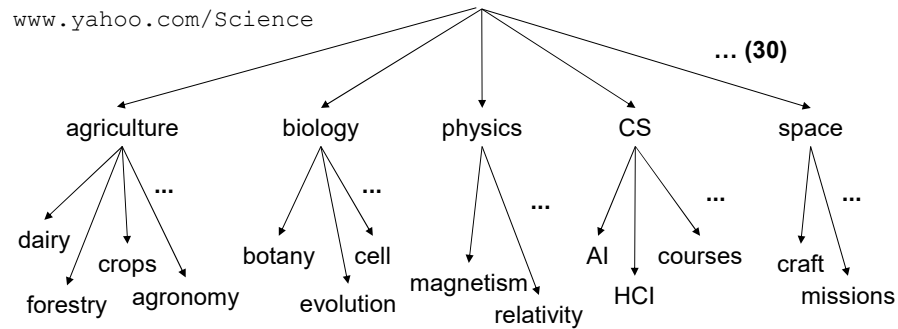- We are going to learn such clustering algorithms in this lecture!

Clustering is widely used in many IR applications

Topic Hierarchy of Yahoo!
: the kind of output we obtain from clustering

`www.yahoo.com/Science`

... (30)

agriculture   biology   physics   CS   space

...   .

dairy   crops   botany   cell   magnetism   AI   courses   craft   missions

forestry   agronomy   evolution   relativity   HCI

Yahoo Science hierarchy, consisting of web pages that were gathered in summer of 1997 and pointed to by Yahoo
This hierarchy has totally 264 classes, but only samples are shown here

5

Google News: automatic clustering gives
an effective news presentation style
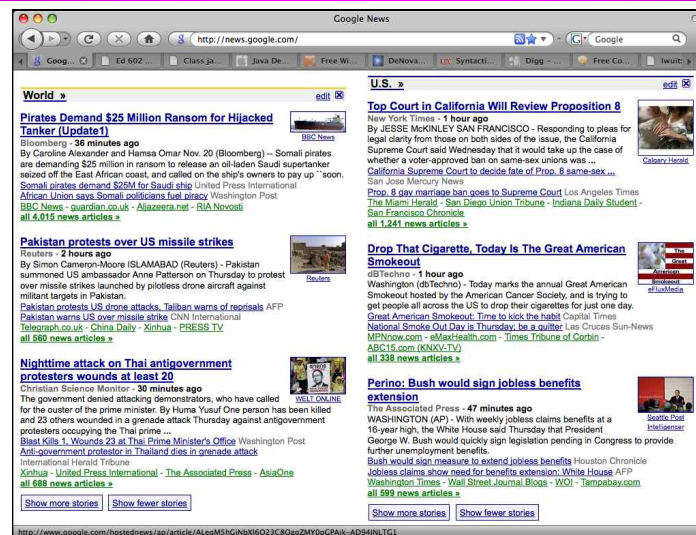
6

Clustering is utilized for visualizing a document collection based on its themes

» In this contour map, each mountain stands for a certain cluster (with its most frequent term) and mountain height stands for its cluster size

Clustering can be used for improving recall in IR system

- *Basic Cluster hypothesis* : Documents in the same cluster behave similarly with respect to relevance to information needs
- Therefore, to improve recall in IR system
  - » Prior process: we can cluster all the documents in collection
  - » When a query matches a document *D*(that is, D is relevant to that query), IR system return other documents in the cluster containing that document *D*
- By doing this, we hope that querying with "car" will also return documents containing "*automobile*"
  - » Because clustering groups documents containing "*car"* together with documents containing "*automobile"*   Why might this happen?

A document containing "*car"* is likely to have many common terms(or words) with a document containing "*automobile"* so that the similarity between the two documents is apt to be very high in vector space model

Clustering can be also used for better navigation of search results in IR system

- As shown in the picture, this search engine gives well-categorized search results that are grouped thematically by clustering
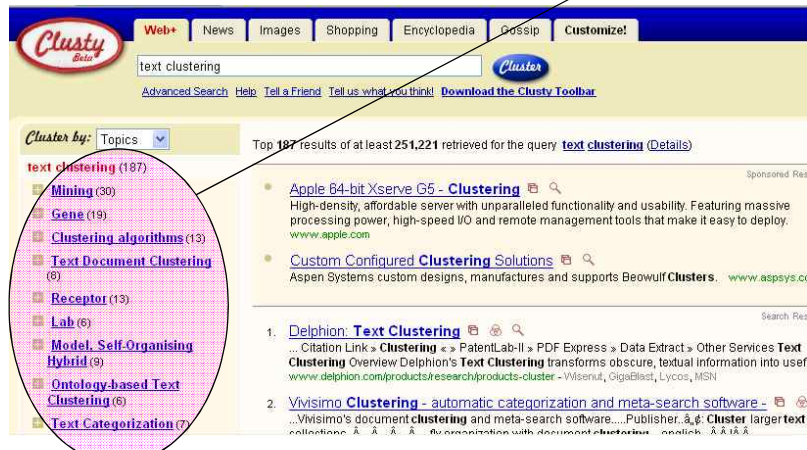
9

Document clustering

- Given a set of documents, document clustering groups those documents into a number of clusters based on their content similarity

10

Issues for document clustering

- Representation and similarity Metric for clustering
  - » Document representation
    - – Is it based on vector space model or boolean model?
  - » Similarity or distance metrics
    - – Is cosine similarity, Euclidean distance or anything else used ?
- How many clusters?
  - » Is it fixed a priori?
  - » Is it completely driven from document collection?
  - » Basic principle: just **avoid "trivial" cluster that is too large or too small**
    - – In an application, if a cluster is too large or too small, then we generally waste an extra user click without narrowing down the set of documents enough when navigating document collection

11

What makes documents "related"?

- Ideally, semantic similarity makes documents related each other
- But, practically, statistical similarity makes documents related each other
  - » We can use cosine similarity
  - » To make it effective, documents are represented as term vectors
  - » Incidentally, for many IR algorithms, it is easier to think similarity in terms of a *distance* (rather than similarity) between documents
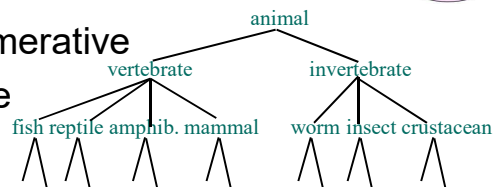 - » We often use Euclidean distance as a similarity metric (more exactly, dissimilarity metric)

12

Clustering Algorithms

- Flat algorithm
  - » Usually, it start with a random partitioning
  - » Gradually, it is refined iteratively
    - – *e.g. K*-means clustering
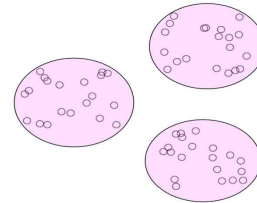
- Hierarchical algorithms
  - » Bottom-up, agglomerative
  - » Top-down, divisive

animal

vertebrate   invertebrate

fish reptile amphib. mammal   worm insect crustacean

K-Means (flat clustering algorithm)

- It assumes that documents are real-valued term vectors
- Cluster is based on *centroid* (the center of gravity or mean) of term vectors:

  In a cluster c , $\vec{\mu}(c) = \dfrac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$

- Re-assigning instances to clusters is based on Euclidean distance to each cluster centroid.

  Note that henceforth, we can equivalently phrase Euclidean distance in terms of dissimilarity)

K-Means Algorithm
for making *K* clusters

select K random docs $\{s_1, s_2, \ldots s_K\}$ as seeds.
do
{
    for each doc d_i 랜덤시드를 정한 후에 assign해라
클러스터 계산
다시 assign후
클러스터링 계산
    {   converged될 때까지 계속 반복ㄱ
      assign d_i to the cluster c_j such that $dist(d_i, s_j)$ is minimal ;
    }
    for each cluster c_j // *update the seeds to the centroid of each cluster*
    {
      $s_j = \mu(c_j)$;
    }
} until(clustering is converged enough or termination condition is met)

Much simpler and easier than expected !

15

K Means Example
(*K*=2)
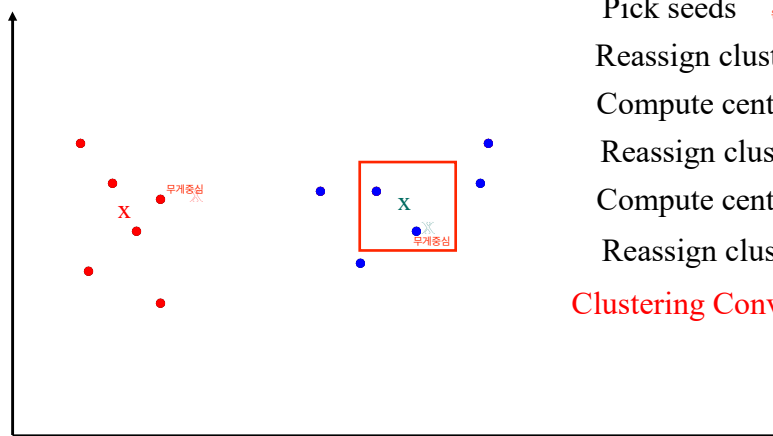
Pick seeds   왼쪽위 빨강 오른쪽아래 파랑
Reassign clusters
Compute centroids
Reassign clusters   무게중심 이동하면 다시 assign
Compute centroids
Reassign clusters
Clustering Converged!

무게중심
x
x
무게중심

16

8

Termination conditions

- We may have several options for termination condition of K-means algorithm

- e.g.
  - » just terminate in a fixed number of iterations
  - » terminate when all clusters are unchanged at all 모든 클러스터가 버냥지 않는다는 것은 새롭녹 assignment가 변하지 않는다는 것
  - » terminate when all cluster centroid positions are unchanged at all 클러스터의 센터에 있는 것들이 변하지 않는다는 것

Does the second termination condition is equivalent to the third one?

yes or **no** ?
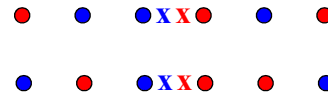
대부분 yes. 가끔 no

Convergence

- Why should the *K*-means algorithm ever reach a *fixed point*?
  - » There is always a state in which all clusters don't change

- Indirect Answer: *K*-means algorithm is a special case of a general procedure known as *Expectation Maximization (EM) algorithm*
  - » EM is proved to converge to a stable state
    (We assume you've seen EM algorithm in AI class before)
  - » Number of iterations for convergence can be large, but in practice it is usually not be large (much smaller than expected)

Time Complexity of *K*-Means Algorithm

```
select K random docs {s₁, s₂,… s_K} as seeds.
do
{
    for each doc d_i
        assign d_i to the cluster c_j such that dist(d_i, s_j) is minimal ;
    for each cluster c_j
        s_j = μ(c_j) ;
} until(clustering is converged enough)
```

- Computing distance between two term vectors is O(m) where m is the dimensionality of the vectors
- Reassigning clusters needs O(Kn) distance computations and thus O(Knm) where n is the number of all document vectors
- Computing centroids requires each document vector to be added once to some centroid and thus needs O(nm)
- Because these two steps(reassigning clusters and computing centroids) are each performed once for *I* iterations, the total time complexity is O$(IKnm)$ → *Totally linear complexity !*

19

Initial Seed Choice in *K*-Means Algorithm

- Results can vary based on initial random seed selection.
- It means that some initial seeds can result in poor convergence rate, or convergence to a sub-optimal clustering state
  - » Solutions
    - – Select good initial seeds using a heuristic (e.g. select a document vector that is most distant from any (already) selected seed)
    - – Try multiple times with different initial seeds because this algorithm apparently uses hill-climbing searching
      (We assume you've seen hill-climbing searching in algorithm or AI classes before)

an example showing sensitivity to initial seeds
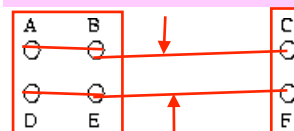
In the above,
i) if you start with B and E as centroids, you converge to {A,B,C} and {D,E,F}

ii) If you start with D and F, you converge to {A,B,D,E} and {C,F}

20

Hierarchical Clustering Algorithm

- Two ways
  - » Divisive Clustering (top down)
    - – The complete document collection is assumed to represent one complete cluster
    - – Then, each cluster is subsequently and recursively broken down into smaller clusters using flat clustering algorithm like k-means algorithm
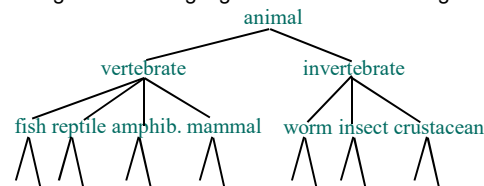
  - » Hierarchical Agglomerative Clustering (bottom up)
    - – Individual document similarities are used as a starting point.
    - – Based on it, a gluing operation agglomeratively collects similar documents (or similar document clusters) into a super-ordinate (or larger) cluster

---

Hierarchical Agglomerative Clustering

- Basic procedure
  1. Place each of N document vectors into a cluster of its own ;
  2. Compute all pairwise cluster-to-cluster similarities ;
  3. Form a new larger cluster by combining the pair of most similar clusters ;
  4. Goto step 2 while the number of clusters left is greater than 1 ;

How to Combine Clusters?

- Basic procedure
  1. Place each of N document vectors into a cluster of its own ;
  2. Compute all pairwise cluster-to-cluster similarities ;
  3. Form a new larger cluster by combining the pair of most similar clusters ;
  4. Goto step 2 if the number of clusters left is greater than 1 ;

- Just combine two clusters using intercluster similarity
  - » Three options
    - Single-link clustering
    - Complete-link clustering
    - (Group) average-link clustering

23

How to Combine Clusters?

- Basic procedure
  1. Place each of N document vectors into a cluster of its own ;
  2. Compute all pairwise cluster-to-cluster similarities ;
  3. Form a new larger cluster by combining the pair of most similar clusters ;
  4. Goto step 2 if the number of clusters left is greater than 1 ;

- Single-link clustering is based on the following three criteria
  - » Each document vector must have a similarity exceeding a threshold value with at least one other document vector in the same cluster
  - » The similarity between two clusters is taken to be the similarity between the most similar pair of document vectors in the two clusters
  - » Each cluster member(=document vector) will be more similar to at least one member in that same cluster than to any member of another cluster

24

How to Combine Clusters? *(Continued)*

• Basic procedure
1. Place each of N document vectors into a cluster of its own ;
2. Compute all pairwise cluster-to-cluster similarities ;
3. Form a new larger cluster by combining the pair of most similar clusters ;
4. Goto step 2 if the number of clusters left is greater than 1 ;

● Complete-link clustering is based on the following three criteria

» Each document vector must have a similarity exceeding a threshold value with all other document vectors in the same cluster

» The similarity between two clusters is taken to be the similarity between the least similar pair of document vectors in the two clusters

» Each cluster member(=document vector) is more similar to the most dissimilar member of that same cluster than to the most dissimilar member of any other cluster

---

How to Combine Clusters? *(Continued)*

• Basic procedure
1. Place each of N document vectors into a cluster of its own ;
2. Compute all pairwise cluster-to-cluster similarities ;
3. Form a new larger cluster by combining the pair of most similar clusters ;
4. Goto step 2 if the number of clusters left is greater than 1 ;

● Group average-link clustering

» Makes a compromise between the extreme stuffs of single-link and complete-link clusterings

» Each cluster member(=document vector) has a greater average similarity to the remaining members of that same cluster than it does to all members of any other cluster

Single-link, complete-link, and group average-link clusterings

- Note: for easy understanding of single-link, complete-link, and group average-link clusterings, we may need some examples and try out each of them for ourselves

Detailed examples for single-link clusterings in terms of run-time data structure

A-F (6 items) 6(6-1)/2 (15) pairwise similarities

Rank	Pair	Similarity	Rank	Pair	Similarity
1	AF	0.9	9	BC	0.4
2	AE	0.8	10	DE	0.4
3	BF	0.8	11	AB	0.3
4	BE	0.7	12	CD	0.3
5	AD	0.6	13	EF	0.3
6	AC	0.5	14	CF	0.2
7	BD	0.5	15	DF	0.1
8	CE	0.5			

decreasing order

상상각 행렬
하상각 행렬

Actually only a upper or lower triangular matrix is enough !

```
    A   B   C   D   E   F
A   .   .3  .5  .6  .8  (.9)
B  .3   .   .4  .5  .7  .8
C  .5  .4   .   .3  .5  .2
D  .6  .5  .3   .   .4  .1
E  .8  .7  .5  .4   .   .3
F  .9  .8  .2  .1  .3   .
```

initial data structure

Single Link Clustering

largest one

1. AF: 0.9
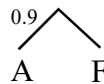
$sim(AF,X)=max(sim(A,X),sim(F,X))$

	A	B	C	D	E	F
A	.	.3	.5	.6	.8	.9
B	.3	.	.4	.5	.7	.8
C	.5	.4	.	.3	.5	.2
D	.6	.5	.3	.	.4	.1
E	.8	.7	.5	.4	.	.3
F	.9	.8	.2	.1	.3	.

largest one

2. AF,E: 0.8
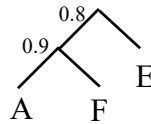
$sim(AEF,X)=max(sim(AF,X),sim(E,X))$

	AF	B	C	D	E
AF	.	.8	.5	.6	.8
B	.8	.	.4	.5	.7
C	.5	.4	.	.3	.5
D	.6	.5	.3	.	.4
E	.8	.7	.5	.4	.

29

Single Link Clustering (Cont.)

3. AEF, B: 0.8

Note E and B are on the same level.
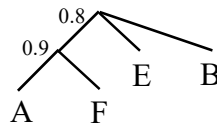
$sim(ABEF,X)=max(sim(AEF,X), sim(B,X))$

largest one

	AEF	B	C	D
AEF	.	.8	.5	.6
B	.8	.	.4	.5
C	.5	.4	.	.3
D	.6	.5	.3	.

4. ABEF, D: 0.6
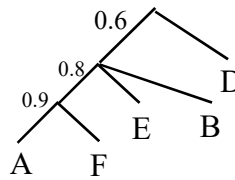
$sim(ABDEF,X)=max(sim(ABEF,X), sim(D,X))$

largest one

	ABEF	C	D
ABEF	.	.5	.6
C	.5	.	.3
D	.6	.3	.

30

Single Link Clustering (Cont.)

5.   ABDEF, C: 0.5

	ABDEF	C
ABDEF	.	.5
C	.5	

final one

ABCDEF	.
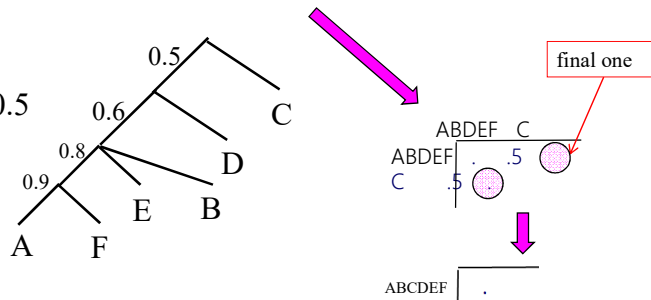
Single-Link Clusters

- Similarity level 0.85 (i.e., similarity threshold)
  - » AF, E, B, D, C

- Similarity level 0.65 (i.e., similarity threshold)
  - » AFEB, D, C

- Similarity level 0.55 (i.e., similarity threshold)
  - » AFEBD, C
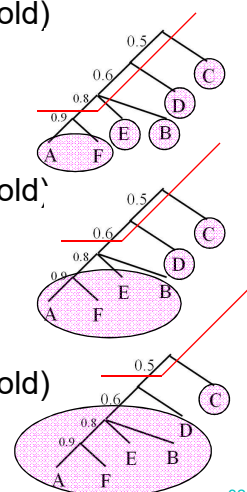
Detailed examples for complete-link clusterings in terms of run-time data structure

Rank	Pair	Similarity	Rank	Pair	Similarity
1	AF	0.9	9	BC	0.4
2	AE	0.8	10	DE	0.4
3	BF	0.8	11	AB	0.3
4	BE	0.7	12	CD	0.3
5	AD	0.6	13	EF	0.3
6	AC	0.5	14	CF	0.2
7	BD	0.5	15	DF	0.1
8	CE	0.5			

Step Number	Similarity Pair	Checked List	Link Structure	Similarity Matrix

1. AF 0.9   AF

클러스터가 크면 거리가 길다
월 스트리트를 재면 siimlarity가….

a랑 f합치고 싶으면 a하고 어떤놈, f하고 어떤놈에 유사성이 있어야한다
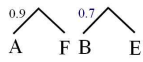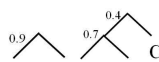
A and F are clustered

$sim(AF,x)=\mathbf{min}(sim(A,x),\ sim(F,x))$

Similarity Matrix:

	A	B	C	D	E	F
A	.	3	.5	.6	.8	.9
B	.3	.	.4	.5	.7	.8
C	.5	.4	.	.3	.5	.2
D	.6	.5	.3	.	.4	.1
E	.8	.7	.5	.4	.	.3
F	.9	.8	2	1	3	.

AF하기로했으니

2. AE 0.8   AE, AF

a랑 e하고싶으면
e랑 f와서 더 안좋은거 알아야한다

Note: EF is required in checked list for clustering AF and E
ef가 아직 없어서 패스해야 한다
단 방금 나온 edge AE 기억은 해둔다
제일 나쁜 조건으로 모들시켜야한다

	AF	B	C	D	E
AF	.	.3	.2	.1	.3
B	.3	.	.4	.5	.7
C	.2	.4	.	.3	.5
D	.1	.5	.3	.	.4
E	.3	.7	.5	.4	.

그래서 변하는 것 없다

3. BF 0.8   AE, AF, BF

이것도 패스
AB가 아직 없다

	AF	B	C	D	E
AF	.	.3	.2	.1	.3
B	.3	.	.4	.5	.7
C	.2	.4	.	.3	.5
D	.1	.5	.3	.	.4
E	.3	.7	.5	.4	.

Note: AB is required in checked list for clustering AF and B

33

Complete-Linke Cluster Generation (Cont.)

Rank	Pair	Similarity	Rank	Pair	Similarity
1	AF	0.9	9	BC	0.4
2	AE	0.8	10	DE	0.4
3	BF	0.8	11	AB	0.3
4	BE	0.7	12	CD	0.3
5	AD	0.6	13	EF	0.3
6	AC	0.5	14	CF	0.2
7	BD	0.5	15	DF	0.1
8	CE	0.5			

Step Number	Similarity Pair	Checked List	Link Structure	Similarity Matrix

4. BE 0.7   AE, AF, **BE**, BF

B and E are clustered

$sim(BE,x)=\mathbf{min}(sim(B,x),\ sim(E,x))$

	AF	B	C	D	E
AF	.	.3	.2	.1	.3
B	.3	.	.4	.5	.7
C	.2	.4	.	.3	.5
D	.1	.5	.3	.	.4
E	.3	.7	.5	.4	.

5. AD 0.6   **AD**, AE, AF, BE, BF

DF 안나와서 패스

Note: DF is required in checked list for clustering AF and D

	AF	BE	C	D
AF	.	.3	.2	.1
BE	.3	.	.4	.4
C	.2	.4	.	.3
D	.1	.4	.3	.

6. AC 0.6   **AC**, AD, AE, AF, BE, BF

CF안나와서 패스

Note: CF is required in checked list for clustering AF and C

	AF	BE	C	D
AF	.	.3	.2	.1
BE	.3	.	.4	.4
C	.2	.4	.	.3
D	.1	.4	.3	.

7. BD 0.5   AC, AD, AE, AF, **BD**, BE, BF

Note: DE is required in checked list for clustering BE and D

	AF	BE	C	D
AF	.	.3	.2	.1
BE	.3	.	.4	.4
C	.2	.4	.	.3
D	.1	.4	.3	.

34

17

Complete-Linke Cluster Generation (Cont.)

Rank	Pair	Similarity	Rank	Pair	Similarity
1	AF	0.9	9	BC	0.4
2	AE	0.8	10	DE	0.4
3	BF	0.8	11	AB	0.3
4	BE	0.7	12	CD	0.3
5	AD	0.6	13	EF	0.3
6	AC	0.5	14	CF	0.2
7	BD	0.5	15	DF	0.1
8	CE	0.5			

Step Number	Similarity Pair	Checked List	Link Structure	Similarity Matrix

8.   CE 0.5   AC, AD, AE, AF, BD, BE, BF, **CE**

	AF	BE	C	D
AF	.	.3	.2	.1
BE	.3	.	.4	.4
C	.2	.4	.	.3
D	.1	.4	.3	.

Note: BC is required in checked list for clustering BE and C

9.   BC 0.4   AC, AD, AE, AF, **BC**, BD, BE, BF, CE

ce있어서 합병가능

BE and C are clustered $sim(BCE,x)=\mathbf{min}(sim(BE,x), sim(C,x))$

	AF	BE	C	D
AF	.	.3	.2	.1
BE	.3	.	.4	.4
C	.2	.4	.	.3
D	.1	.4	.3	.

	AF	BCE	D
AF	.	.2	.1
BCE	.2	.	.3
D	.1	.3	.

10.   DE 0.4   AC, AD, AE, AF, BC, BD, BE, BF, CE, **DE**

Note: CD is required in checked list for clustering BCE and D

11.   AB 0.3   **AB**, AC, AD, AE, AF, BC, BD, BE, BF, CE, DE

	AF	BCE	D
AF	.	.2	.1
BCE	.2	.	.3
D	.1	.3	.

Note: CF and EF are required in checked list for clustering BCE and D

(왼) 2 * (오) 3 = 6
6개 다 있나 없다 cf ef가 없다

35

Complete-Linke Cluster Generation (Cont.)

Step Number	Similarity Pair	Checked List	Link Structure	Similarity Matrix

12.   CD 0.3   AB, AC, AD, AE, AF, BC, BD, BE, BF, **CD**, CE, DE

	AF	BCE	D
AF	.	.2	.1
BCE	.2	.	.3
D	.1	.3	.

BCE and D are clustered $sim(BCDE,x)=\mathbf{min}(sim(BCE,x), sim(D,x))$

13.   EF 0.3   AB, AC, AD, AE, AF, BC, BD, BE, BF, CD, CE, DE, **EF**

	AF	BCDE
AF	.	.1
BCDE	.1	.

Note: CF and DF are required in checked list for clustering AF and BCDE

14.   CF 0.2   AB, AC, AD, AE, AF, BC, BD, BE, BF, CD, CE, **CF**, DE, EF

	AF	BCDE
AF	.	.1
BCDE	.1	.

Note: DF is required in checked list for clustering AF and BCDE

36

18

Complete-Linke Cluster Generation (Cont.)

Rank	Pair	Similarity	Rank	Pair	Similarity
1	AF	0.9	9	BC	0.4
2	AE	0.8	10	DE	0.4
3	BF	0.8	11	AB	0.3
4	BE	0.7	12	CD	0.3
5	AD	0.6	13	EF	0.3
6	AC	0.5	14	CF	0.2
7	BD	0.5	15	DF	0.1
8	CE	0.5			

Step Number	Similarity Pair	Checked List	Link Structure	Similarity Matrix

15.　DF 0.1　　AB, AC, AD, AE, AF, BC, BD, BE, BF, CD, CE, CF, DE, **DF**, EF
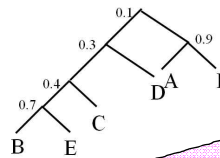
BCDE and AF are clustered

$sim(ABCDEF,x)=\textbf{min}(sim(BCDE,x), sim(AF,x))$

Actually, we don't need to do the final update of matrix

single 15개를 다 볼 필요가 없는데
이건 worst case를 따져야 하기 때문에 끝까지 다 봐야야한다

싱글..은 다 붙여나가는 형식

37

Complete Link Clusters

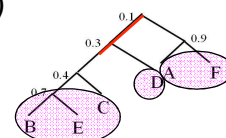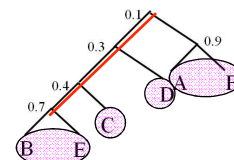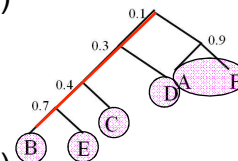- Similarity level 0.85 (i.e., similarity threshold)
  - » AF, B, E, C, D

- Similarity level 0.65 (i.e., similarity threshold)
  - » AF, BE, C, D

- Similarity level 0.35 (i.e., similarity threshold)
  - » AF, BCE, D

38

19

Group Average Link Clustering

- This lecture will not give a detailed example of group average-link clustering
  - » But, in order to formulate the details of such clustering with their data structure
    - – Just, use the average values of pair-wise members within a cluster to update the similarity matrix
    - – remember that all cluster members contribute to inter-cluster similarity
    - – It usually result in a intermediately bound structure between the loosely bound single-link clusters and tightly bound complete-link clusters
  - » Formulate more details of group average-link clustering by yourself, if necessary

39

Comparison

- The Behavior of Single-Link Clustering
  - » The single-link clustering process tends to produce a small number 한쪽으로 막 쏠리는 형태가 되고 of large and loosely linked clusters that are characterized by a 그래서 loosely linked clusters가 된다 chaining effect → This often results in unfair growth of clusters
  - » Each element is usually attached to only one other member of the same cluster at each similarity level
  - » It is sufficient to remember the step sequence of previously clustered example when using single-link clustering

40

Comparison

- The Behavior of Complete-Link Clustering
    - » Complete-link clustering process tends to produce a much larger number of small and tightly linked groupings
    - » It is because each member in a complete-link cluster is firmly guaranteed to resemble all other members in that cluster at the stated similarity level
    - » It is necessary to remember the step sequence of previously clustered example when using complete-link clustering
- Comparison
    - » Generally, the complete-link clustering may be better adapted to information retrieval than the single-link clustering
    - » Of course, a complete-link cluster generation is more expensive to perform than a comparable single-link process

41