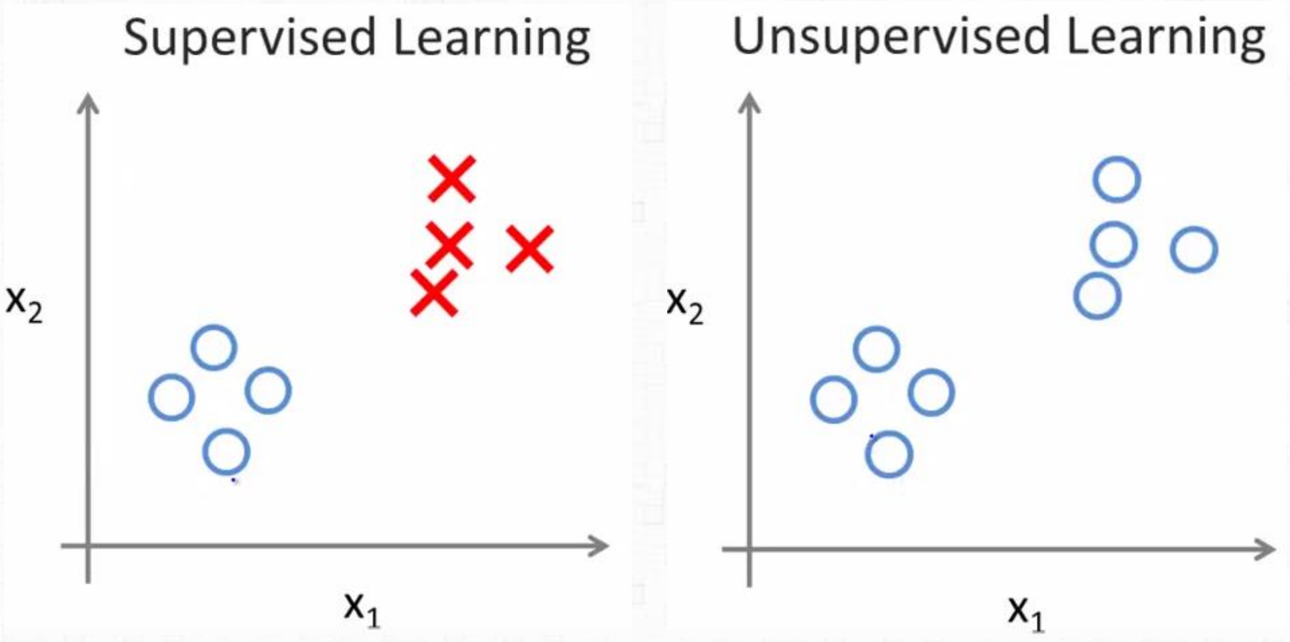
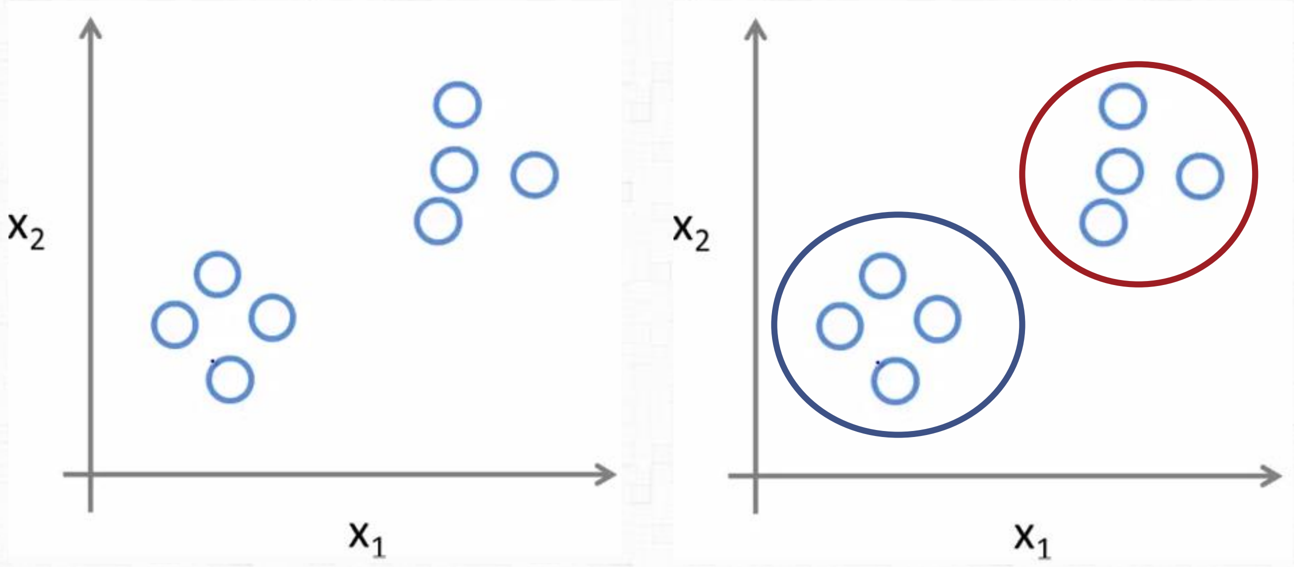
데이터 과학 외전

Day 3 – 클러스터링 & 추천 시스템

Unsupervised Learning

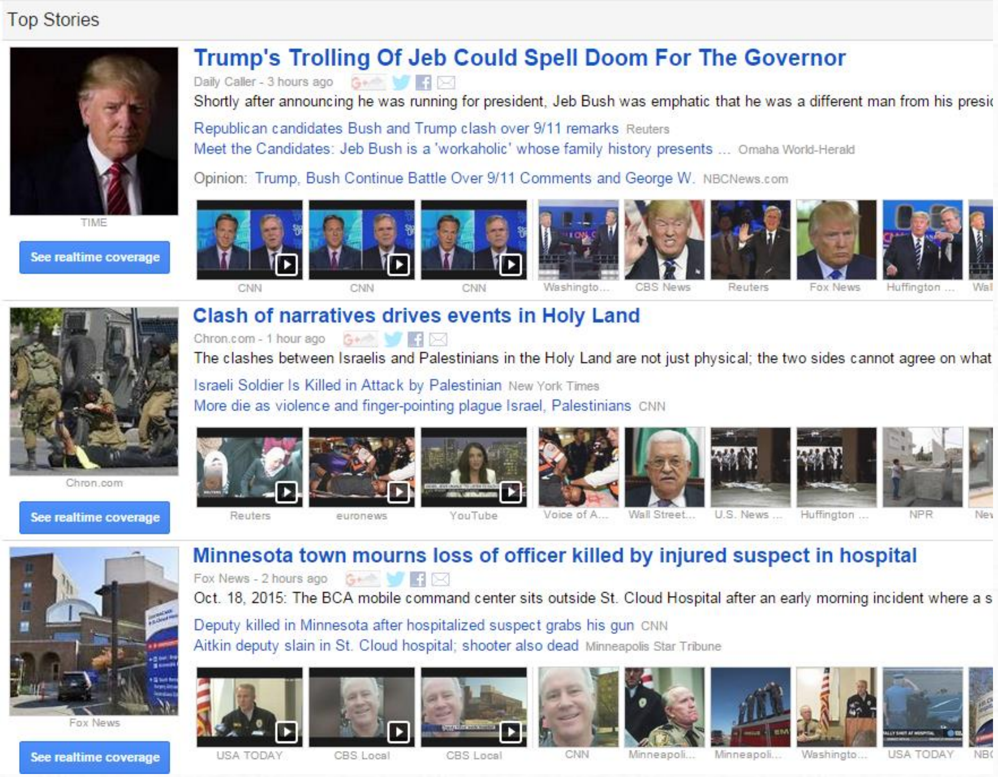


Clustering



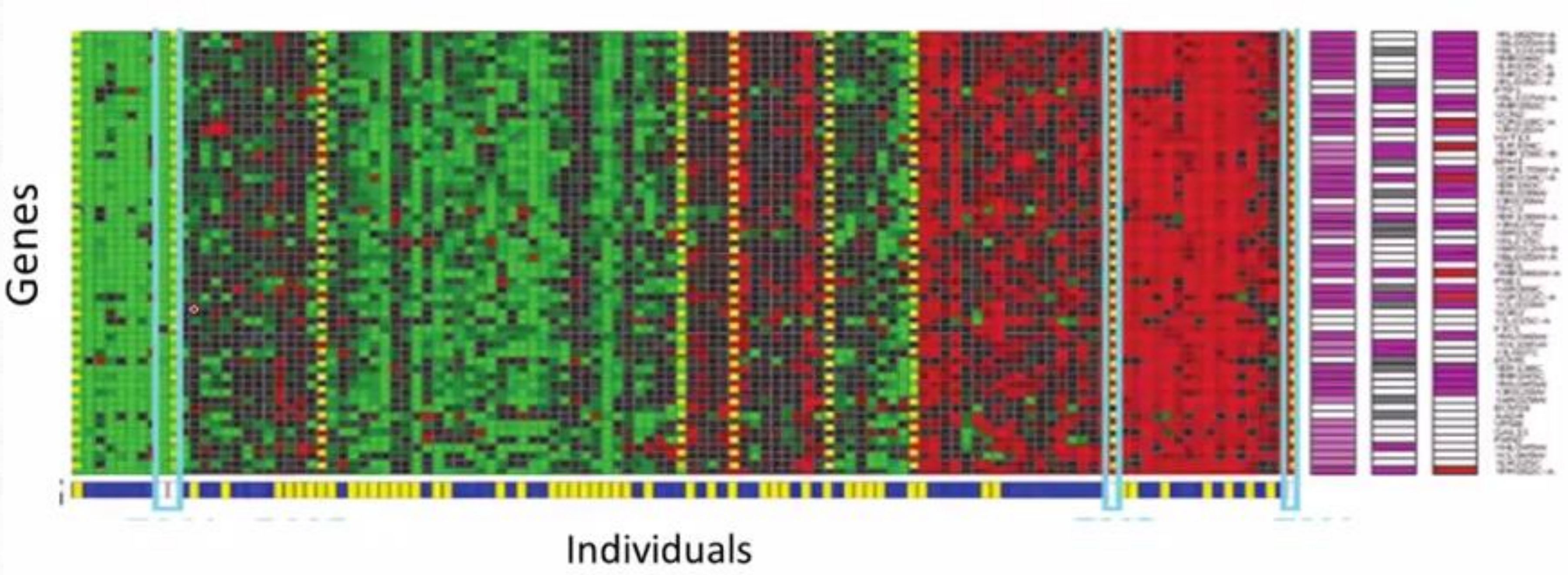
Clustering example

- Google new clustering (news.google.com)



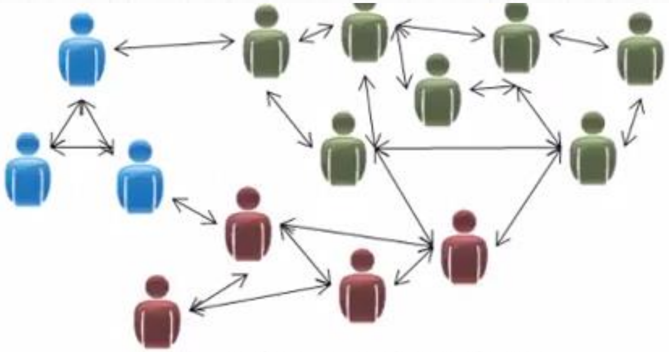
Clustering example

✓ Genome micro-array



Unsupervised Learning Examples

Organize computing clusters



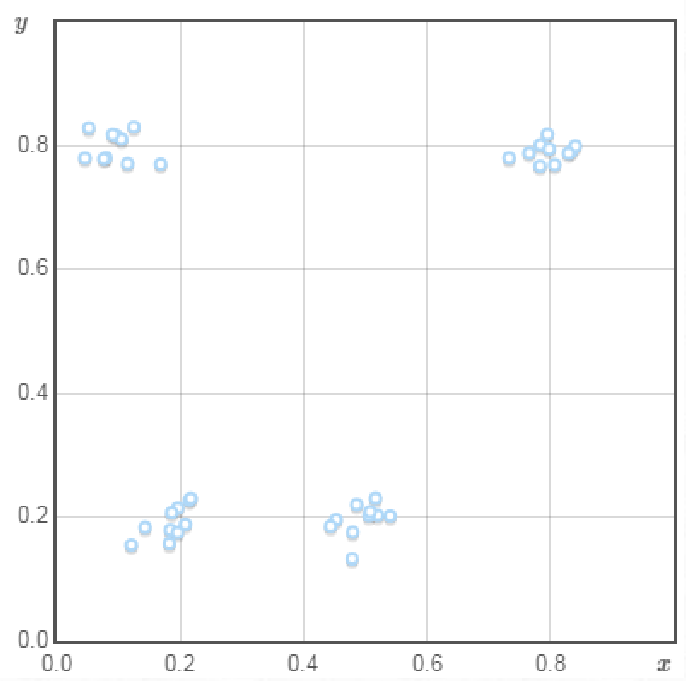
Social network analysis

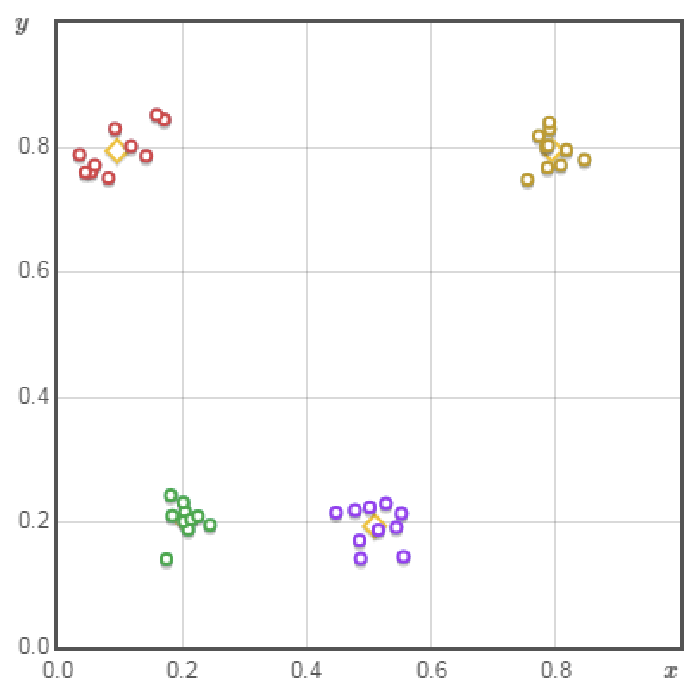
Market segmentation

Astronomical data analysis

k-means clustering by MacQueen, J 1967

- Goal: given n data points, group the data points into k cluster s.t. data points in a cluster are close each other with respect to predefined similarity measure
  - e.g. Euclidean distance





Procedure

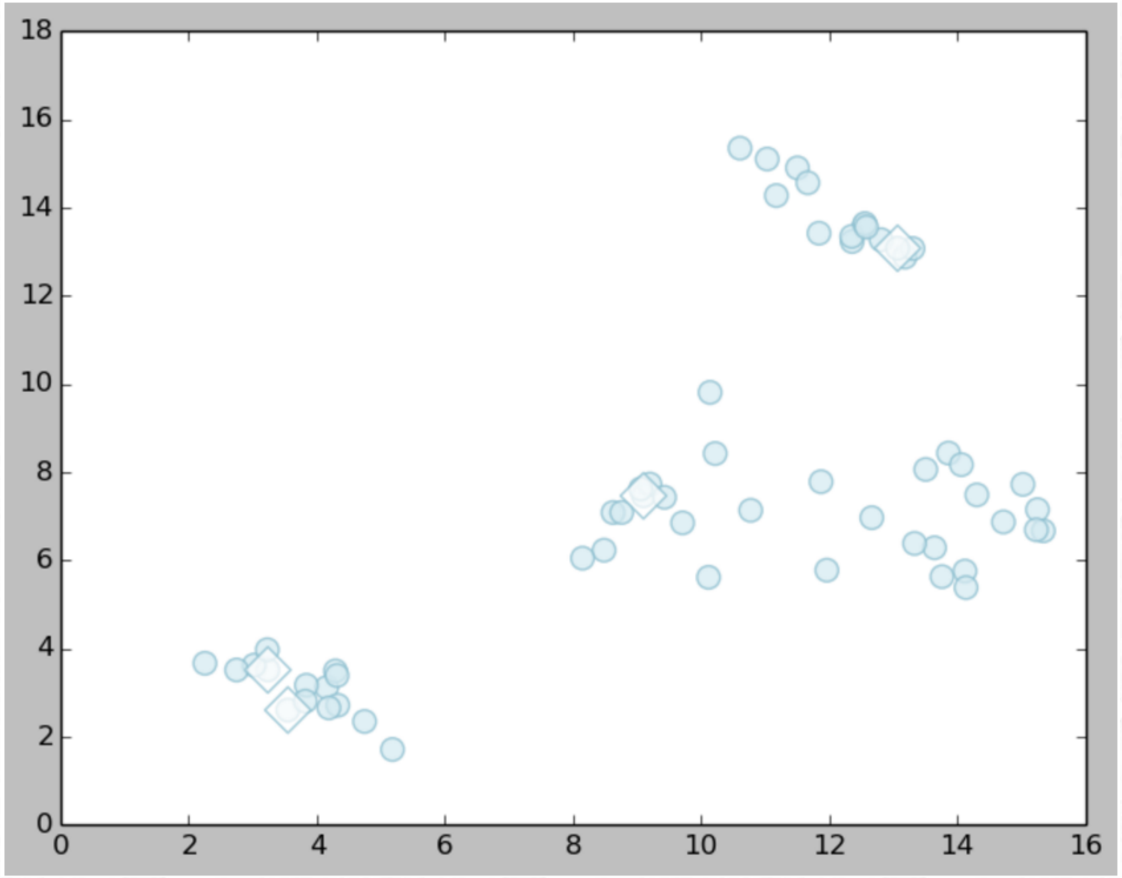
1. Initialize the center of the clusters	$\mu_i =$ some value $, i = 1, \ldots, k$
2. Attribute the closest cluster to each data point	$\mathbf{c}_i = \{j : d(\mathbf{x}_j, \mu_i) \leq d(\mathbf{x}_j, \mu_l), l \neq i, j = 1, \ldots, n\}$
3. Set the position of each cluster to the mean of all data points belonging to that cluster	$\mu_i = \frac{1}{\lvert c_i \rvert} \sum_{j \in c_i} \mathbf{x}_j, \forall i$
4. Repeat steps 2-3 until convergence	
Notation	$\lvert \mathbf{c} \rvert =$ number of elements in $\mathbf{c}$

$$d(\mathbf{x}, \mu_i) = \|\mathbf{x} - \mu_i\|_2^2$$

- Initialization of centroid of clusters: Up to designer's choice
- Forgy: set the positions of the k clusters to k observations chosen randomly from the dataset.
- Random partition: assign a cluster randomly to each observation and compute means of each cluster and set them to centroid.

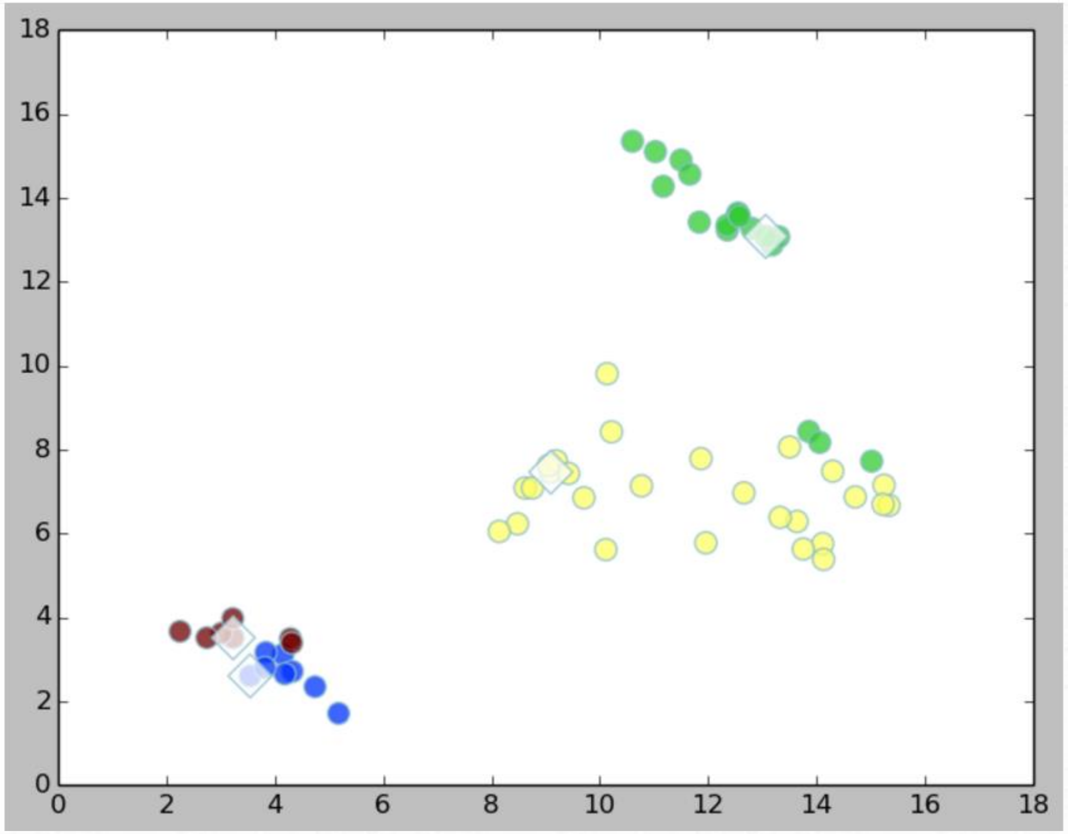
Example

- Select initial centroids: given n data points, select k points randomly



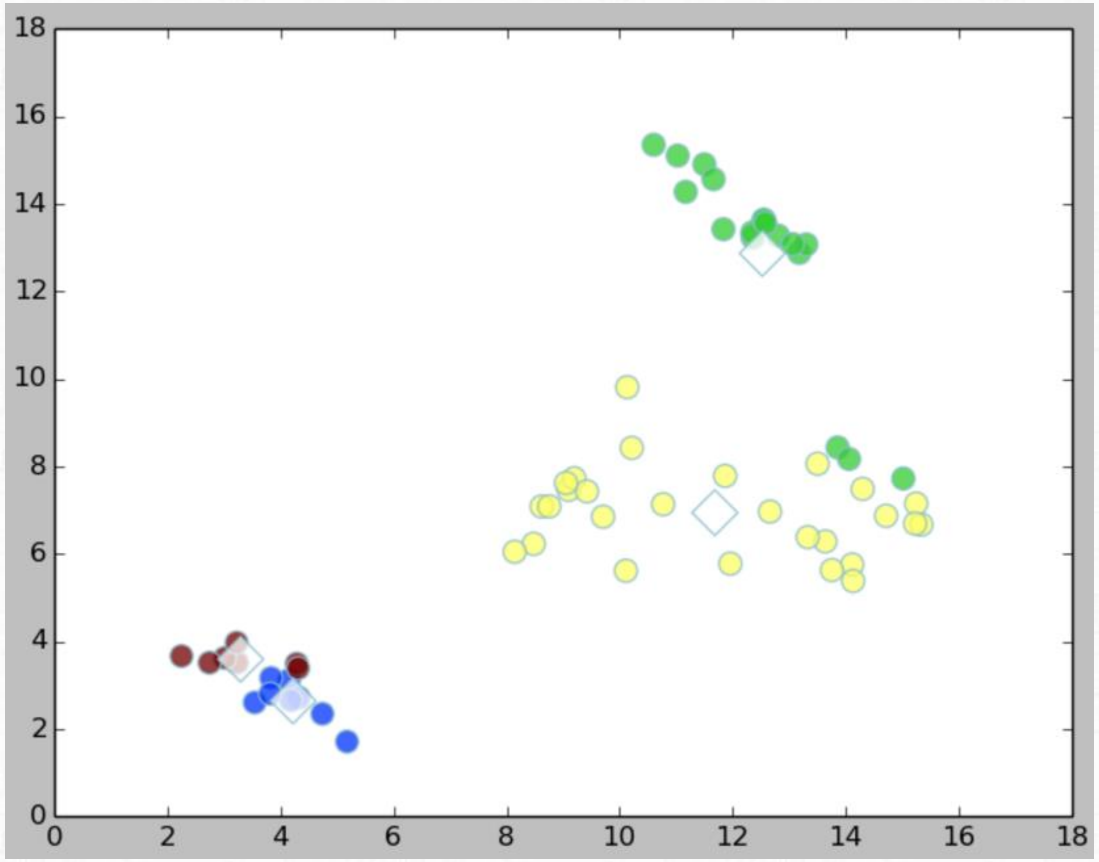
Example

- Assign data points to their closest centroid



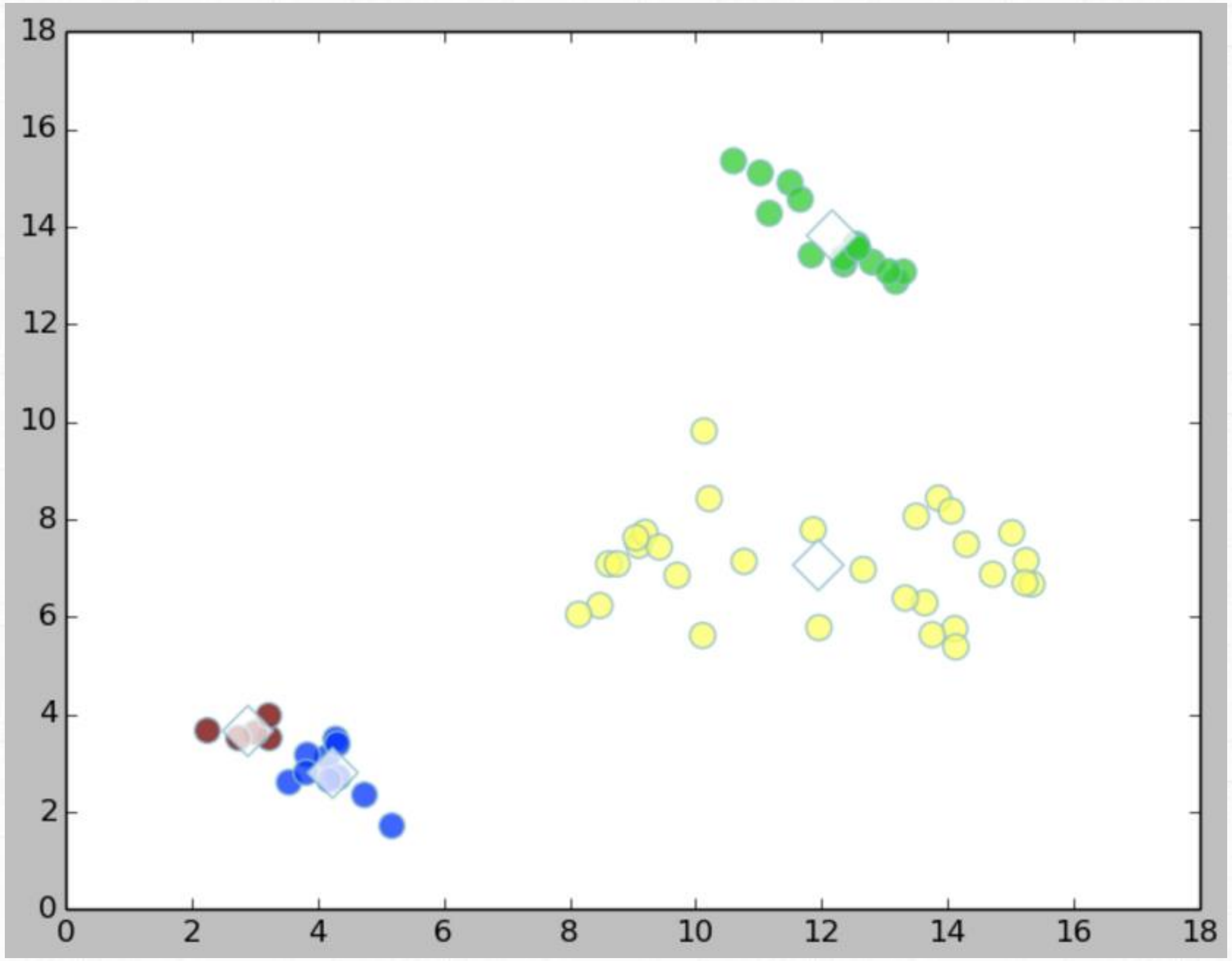
Example

- Re-calculate the centroids as mean of data point in cluster



Example

- Repeat steps above until there is no change to clusters



Hierarchical Clustering

data loading and preparation

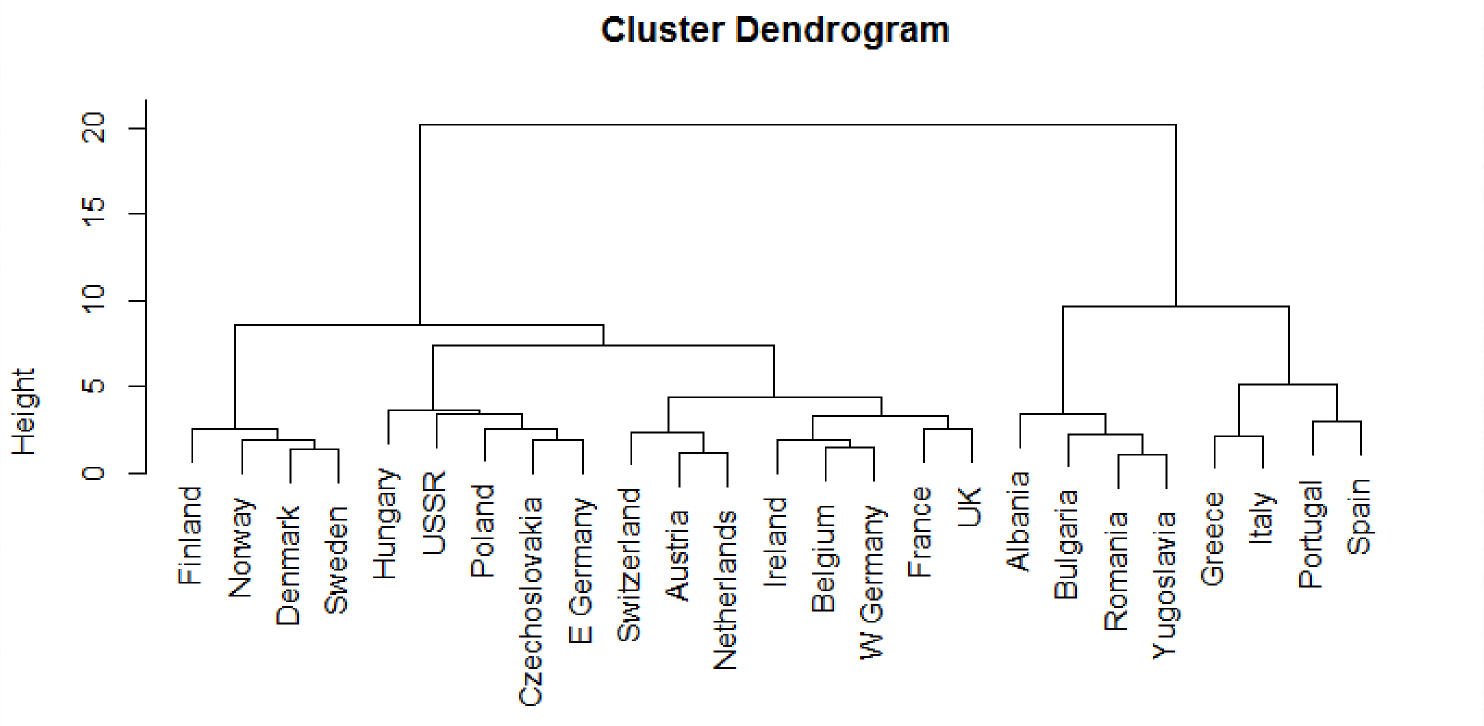
```
protein <- read.table("protein.txt", sep="\t", header=TRUE)
summary(protein)
##             Country      RedMeat         WhiteMeat           Eggs
##  Albania       : 1   Min.   : 4.400   Min.   : 1.400   Min.   :0.500
##  Austria       : 1   1st Qu.: 7.800   1st Qu.: 4.900   1st Qu.:2.700
##  Belgium       : 1   Median : 9.500   Median : 7.800   Median :2.900
##  Bulgaria      : 1   Mean   : 9.828   Mean   : 7.896   Mean   :2.936
##  Czechoslovakia: 1   3rd Qu.:10.600   3rd Qu.:10.800   3rd Qu.:3.700
##  Denmark       : 1   Max.   :18.000   Max.   :14.000   Max.   :4.700
##  (Other)       :19
##       Milk            Fish           Cereals          Starch
##  Min.   : 4.90   Min.   : 0.200   Min.   :18.60   Min.   :0.600
##  1st Qu.:11.10   1st Qu.: 2.100   1st Qu.:24.30   1st Qu.:3.100
##  Median :17.60   Median : 3.400   Median :28.00   Median :4.700
##  Mean   :17.11   Mean   : 4.284   Mean   :32.25   Mean   :4.276
##  3rd Qu.:23.30   3rd Qu.: 5.800   3rd Qu.:40.10   3rd Qu.:5.700
##  Max.   :33.70   Max.   :14.200   Max.   :56.70   Max.   :6.500
##
##       Nuts           Fr.Veg
##  Min.   :0.700   Min.   :1.400
##  1st Qu.:1.500   1st Qu.:2.900
##  Median :2.400   Median :3.800


vars.to.use <- colnames(protein)[-1]
pmatrix <- scale(protein[,vars.to.use])
pcenter <- attr(pmatrix, "scaled:center")
pscale <- attr(pmatrix, "scaled:scale")
```

protein dataset from 1973 on protein consumption from nine different food groups in 25 countries in Europe.

hierachical clustering

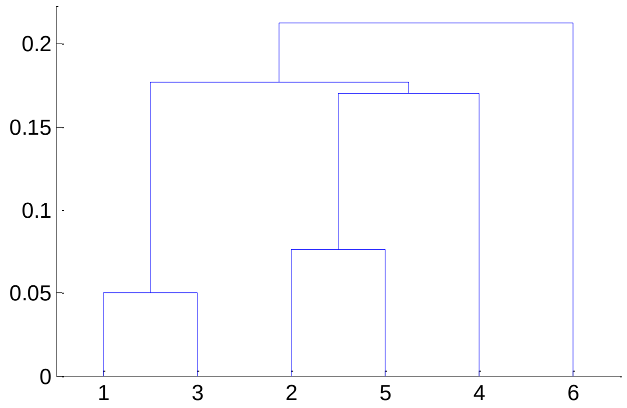
```
d <- dist(pmatrix, method="euclidean")
pfit <- hclust(d, method="ward.D")
plot(pfit, labels=protein$Country)
```

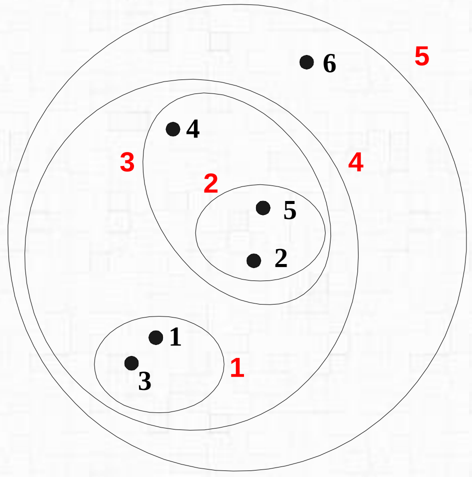
ward:
For each data point as an individual cluster,
merges clusters iteratively so as to minimize the *total within sum of squares (WSS)* of the clustering
http://rfriend.tistory.com/227



Hierarchical Clustering

- Produces a set of ***nested clusters*** organized as a hierarchical tree
- Can be visualized as a **dendrogram**
  - A tree-like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

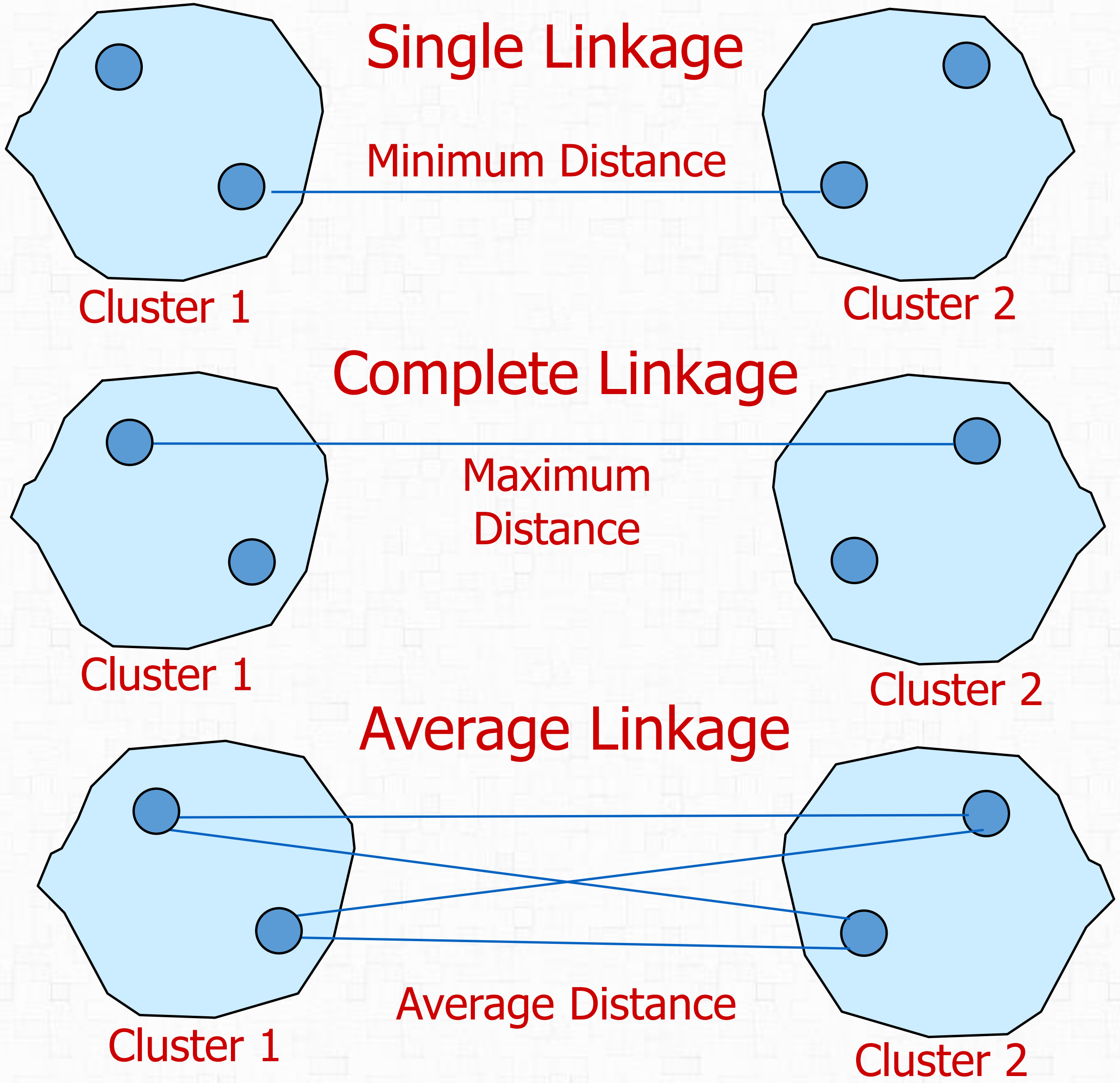
- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

- Hierarchical clusterings may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc

Hierarchical Agglomerative Clustering-Linkage Method

- The **single linkage** method is based on minimum distance, or the nearest neighbor rule.
- The **complete linkage** method is based on the maximum distance or the furthest neighbor approach.
- The **average linkage** method the distance between two clusters is defined as the average of the distances between all pairs of objects

Linkage Methods of Clustering

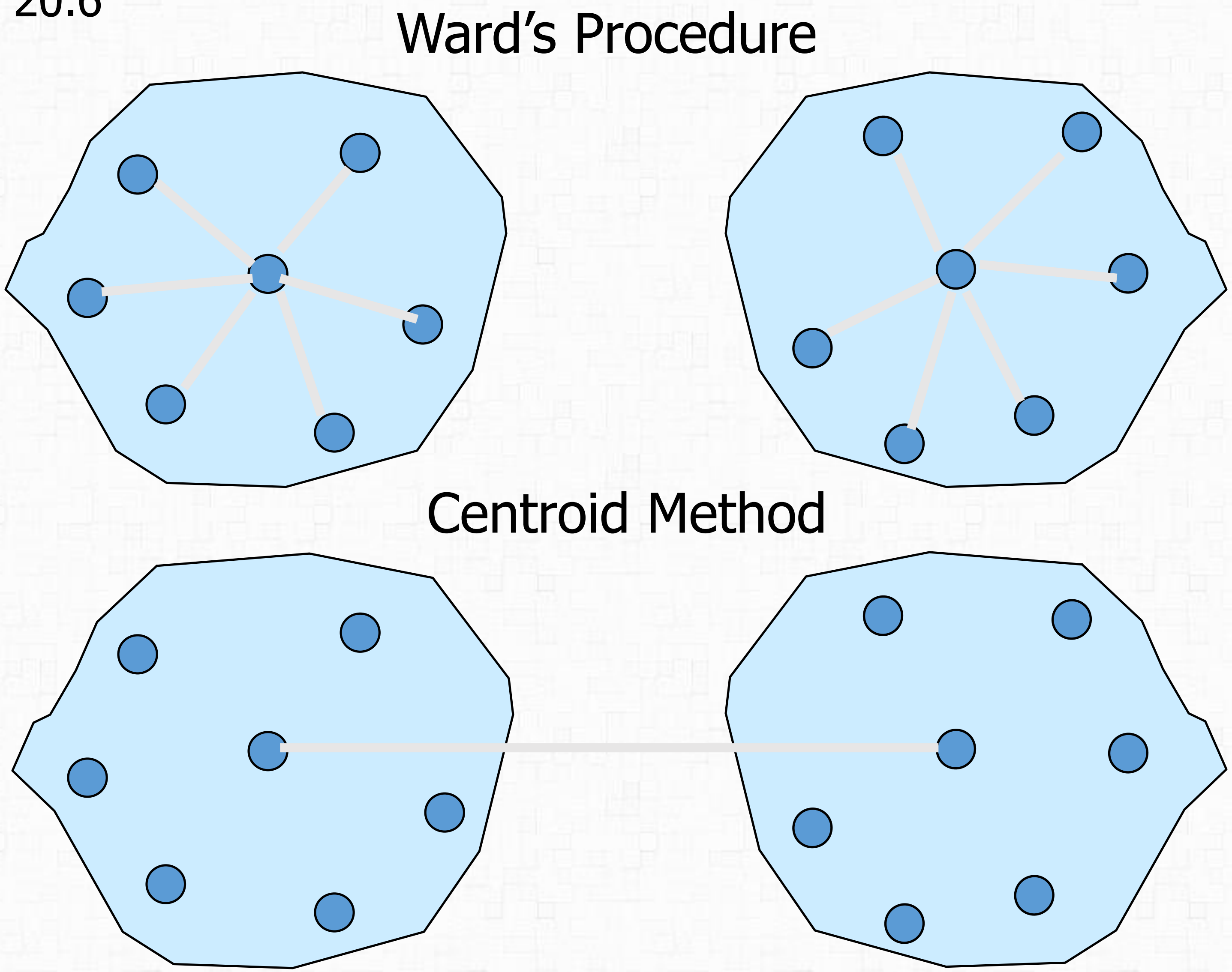


Hierarchical Agglomerative Clustering-Variance and Centroid Method

- **Variance methods** generate clusters to minimize the within-cluster variance.
- **Ward's procedure** is commonly used. For each cluster, the sum of squares is calculated. The two clusters with the smallest increase in the overall sum of squares within cluster distances are combined.
- In the **centroid methods,** the distance between two clusters is the distance between their centroids (means for all the variables),
- Of the hierarchical methods, average linkage and Ward's methods have been shown to perform better than the other procedures.

Other Agglomerative Clustering Methods

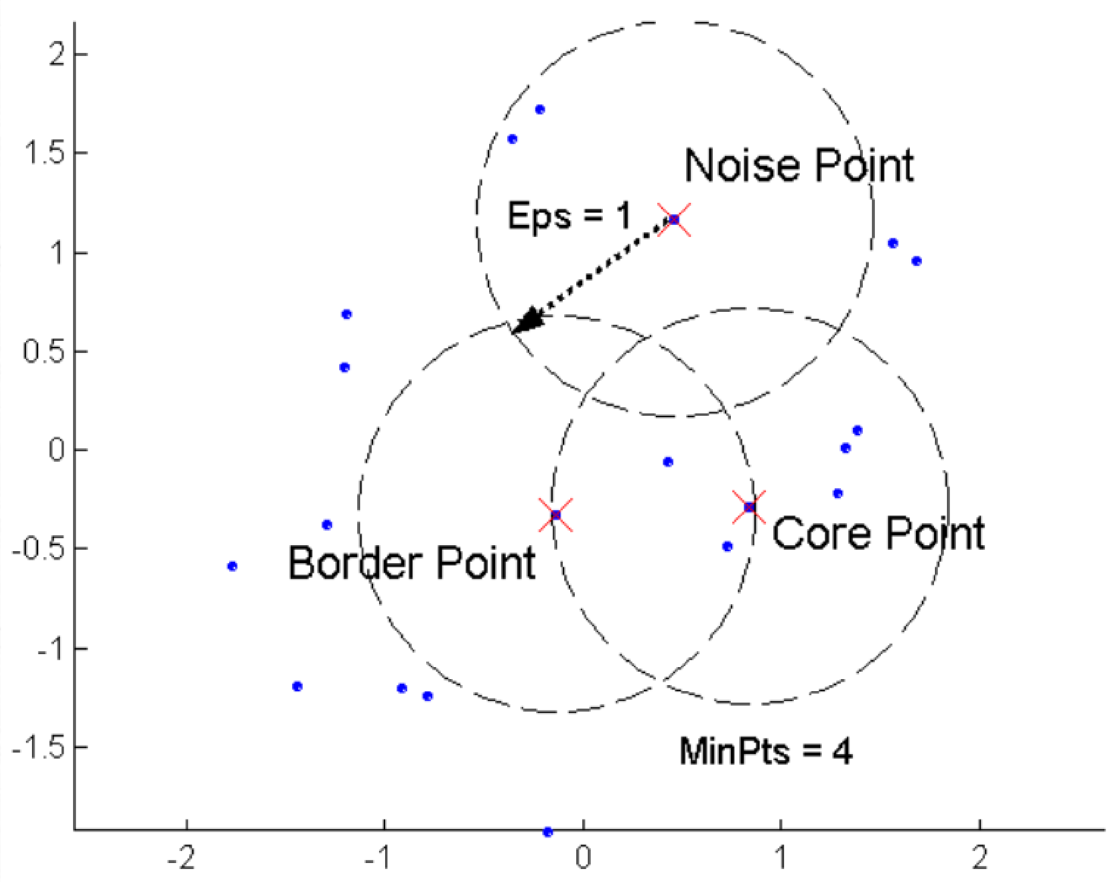
Fig. 20.6



Density-Based Spatial Clustering of Applications with Noise – DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius e (Epsilon)
  - A point is a **core point** if it has more than a specified number of points (MinPts) within Eps
- These are points that are at the interior of a cluster
  - A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A **noise point** is any point that is not a core point or a border point.

DBSCAN

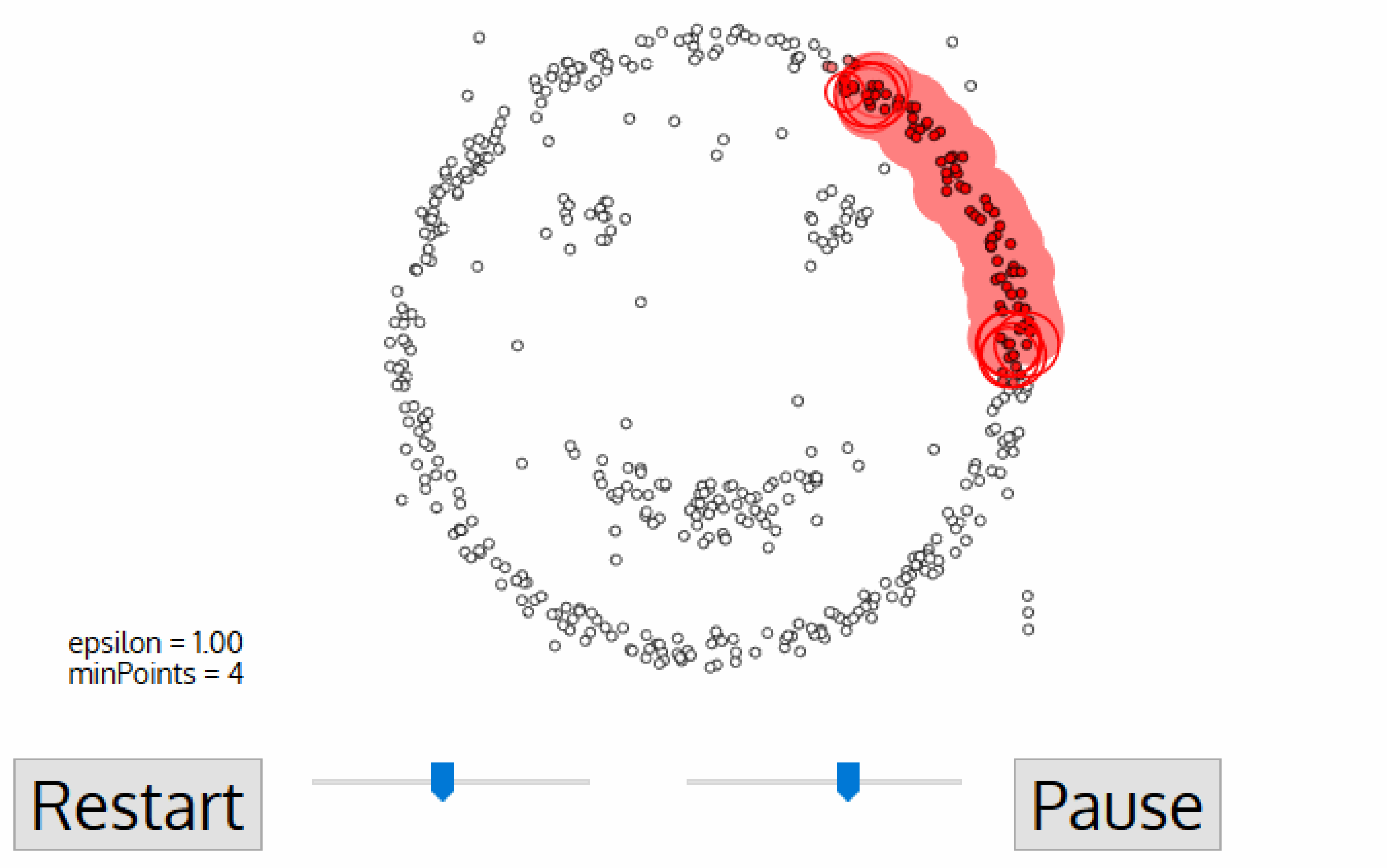

Noise Point
Eps = 1
Border Point
Core Point
MinPts = 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
0
1
2

DBSCAN: Algorithm

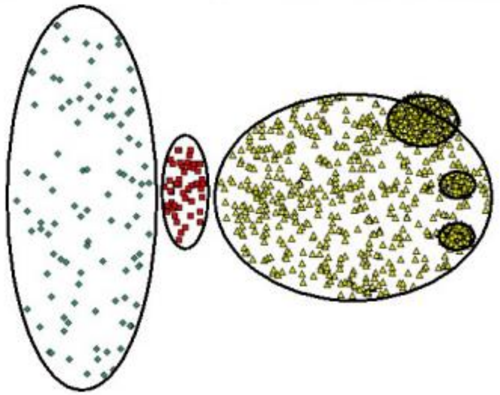
- Let ClusterCount=0. For every point p:

1. If p it is not a core point, assign a null label to it [e.g., zero]
2. If p is a core point, a new cluster is formed
   - [with label ClusterCount:= ClusterCount+1]
   - Then find all points density-reachable from p and classify them in the cluster.

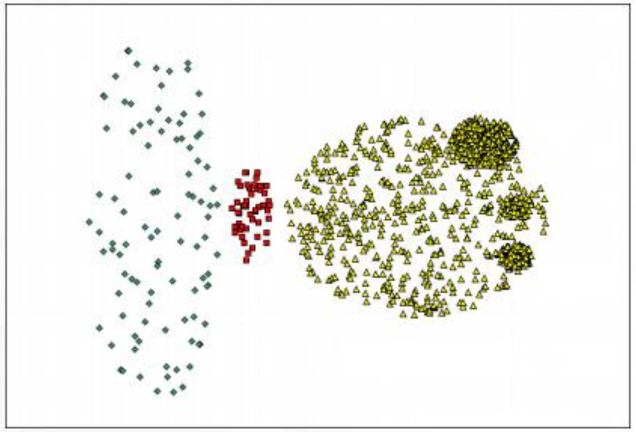
- Repeat this process until all of the points have been visited.
  - Since all the zero labels of border points have been reassigned in 2, the remaining points with zero label are noise.


epsilon = 1.00
minPoints = 4
Restart
Pause

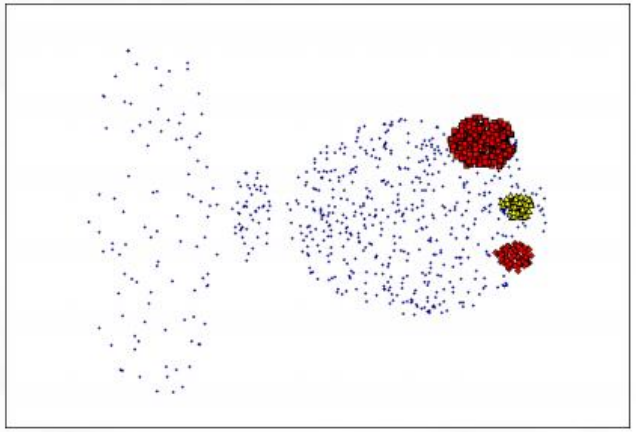
DBSCAN: Flaws



Original Points



(MinPts=4, Eps=large value).

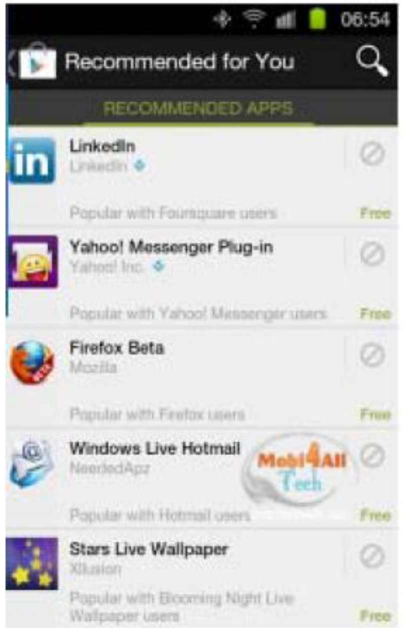


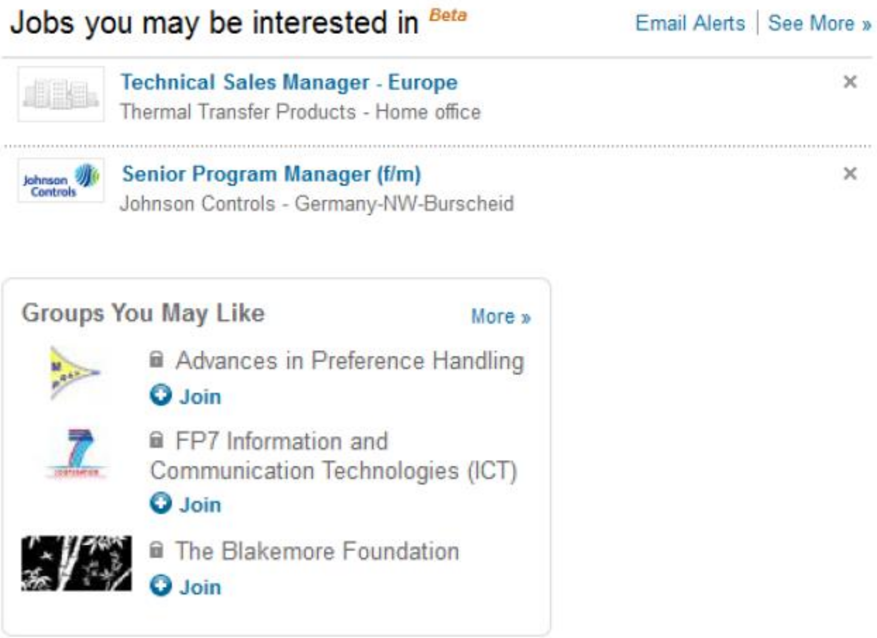
(MinPts=4, Eps=small value; min density increases)

- Varying densities
- High-dimensional data

Recommender system

Recommender system





Recommender system

- Recommender systems apply statistical and knowledge discovery techniques to the problem of making product recommendations (Sarwar et al., 2000).

- Advantages of recommender systems (Schafer et al., 2001):

✓ Improve conversion rate: Help customers find a product she/he wants to buy.

✓ Cross-selling: Suggest additional products.

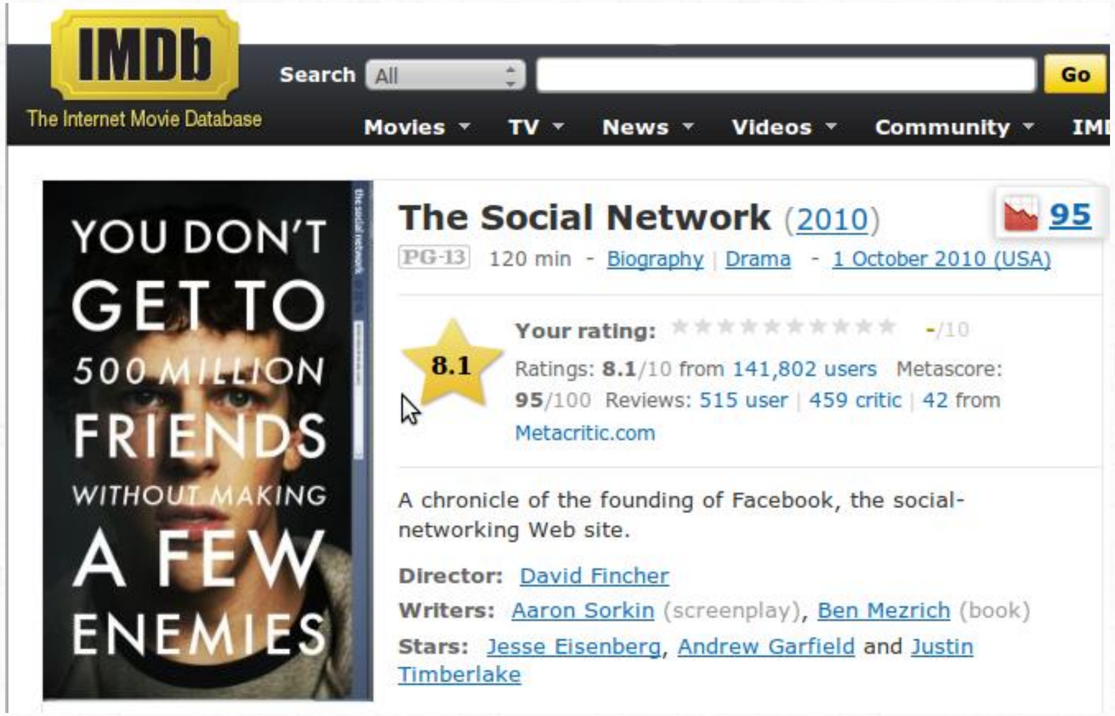
✓ Improve customer loyalty: Create a value-added relationship.

✓ Improve usability of software!

Types of Recommender Systems

- Content-based filtering: Consumer preferences for product attributes.
- Collaborative filtering: Mimics word-of-mouth based on analysis of

Content-based Approach



1. Analyze the objects (documents, video, music, etc.) and extract attributes/features (e.g., words, phrases, actors, genre).
2. Recommend objects with similar attributes to an object the user likes.

Music Genome Project

Musical Attributes	Low =====>=====>=====> High										
Level of vibrato in Lead Vocal	0	1	2	3	4	5	6	7	8	9	10
Lead Vocal sound: Nasal	0	1	2	3	4	5	6	7	8	9	10
Lead Vocal sound: Thickness	0	1	2	3	4	5	6	7	8	9	10
Prominence of Percussion	0	1	2	3	4	5	6	7	8	9	10
Prominence of Horn Section	0	1	2	3	4	5	6	7	8	9	10
Use of Woodwinds (Saxes etc..)	0	1	2	3	4	5	6	7	8	9	10
Prominence of vocal harmony	0	1	2	3	4	5	6	7	8	9	10
Vocal Backups gender male -to- female		1	2	3	4	5	6	7	8	9	10
Use of Vocal call-and-response harmony	0	1	2	3	4	5	6	7	8	9	10
Amount of distortion on the electric guitar	0	1	2	3	4	5	6	7	8	9	10
Prominence of Electric Piano	0	1	2	3	4	5	6	7	8	9	10
Song form: Number of distinct sections	0	1	2	3	4	5	6	7	8	9	10
Amount of rhythmic syncopation	0	1	2	3	4	5	6	7	8	9	10

- “The Music Genome Project is an effort to capture the essence of music at the fundamental level using almost 400 attributes to describe songs and a complex mathematical algorithm to organize them."

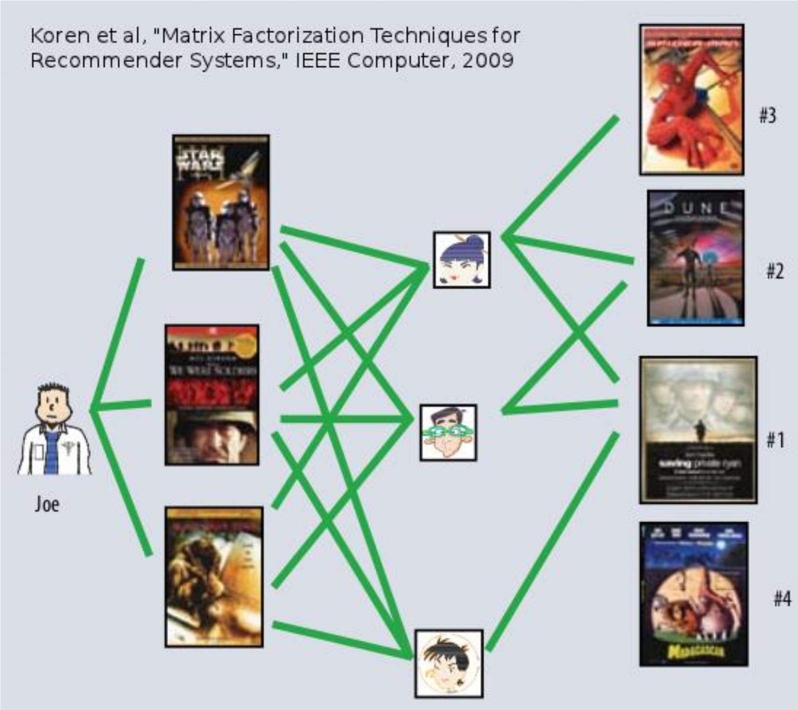
http://en.wikipedia.org/wiki/Music_Genome_Project

Limitation

- Need to encode contents into some meaningful features
  - Which represent user’s taste
- Quality judgement
  - Content is not the only reason to prefer certain item other others
- Limit the chance to expose new diverse item to users
  - No surprises

Collaborative Filterinf (CF)

- ➢ Memory-based CF
- ➢ Model-based CF



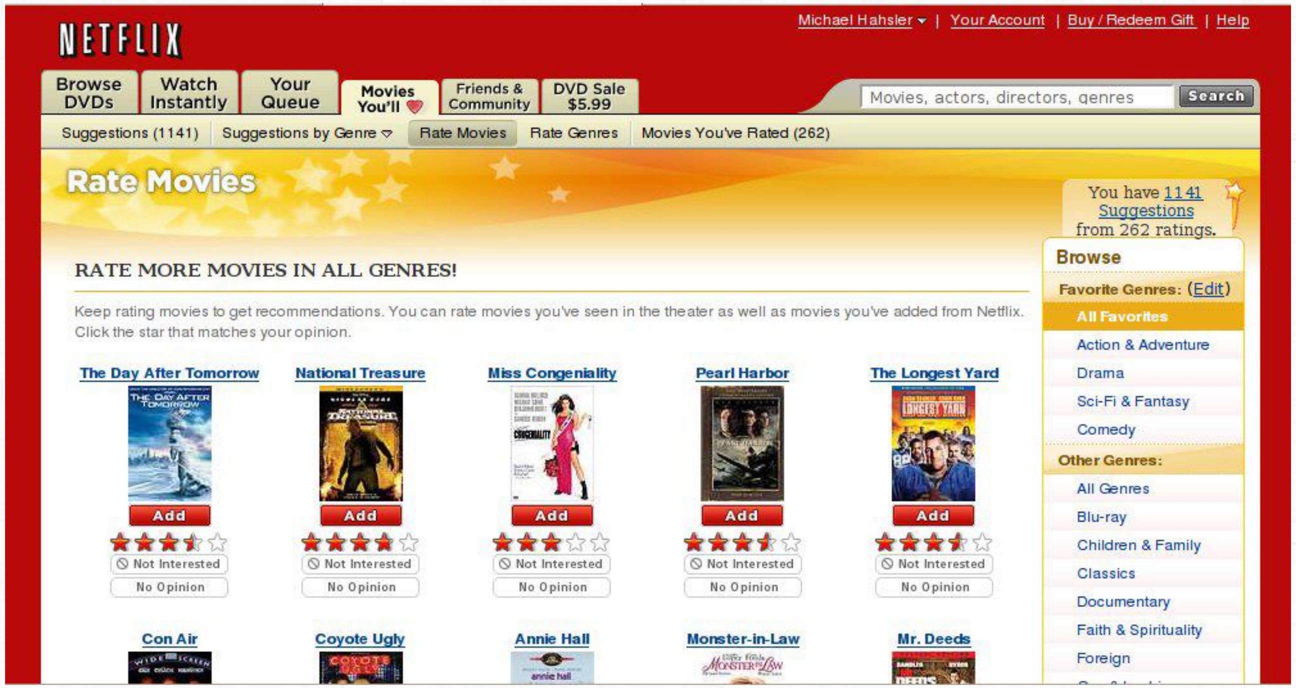
- Make automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many other users (collaboration).
- Assumption: those who agreed in the past tend to agree again in the future.

Collaborative Filtering

- Goal: predict users' movie ratings based on past ratings of other movies

$$
\begin{array}{cc}
 & \xleftrightarrow{\quad\text{Movies}\quad} \\
\textit{Ratings} = & \begin{bmatrix} 1 & ? & ? & 4 & 5 & ? & 3 \\ ? & ? & 3 & 5 & ? & ? & 3 \\ 5 & ? & 5 & ? & ? & ? & 1 \\ 4 & ? & ? & ? & ? & 2 & ? \end{bmatrix}
\end{array}
\begin{array}{c} \big\updownarrow \\ \text{Users} \end{array}
$$

Data Collection



- Data sources:

✓ Explicit: ask the user for ratings, rankings, list of favorites, etc.

✓ Observed behavior: clicks, page impressions, purchase, uses, downloads, posts, tweets, etc.

- What is the incentive structure?

Example of User-rating Matrix

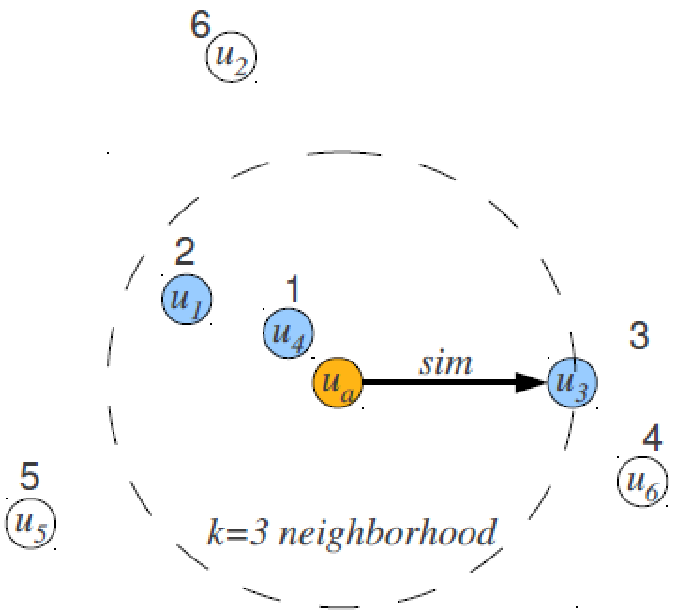
	The Avengers	Sherlock	Transformers	Matrix	Titanic	Me Before You
A	2		2	4	5	
B	5		4			1
C			5		2	
D		1		5		4
E			4			2
F	4	5		1		

User-based CF (UCBF)

- Produce recommendations based on the preferences of similar users (Goldberg et al., 1992; Resnick et al., 1994; Mild and Reutterer, 2001).

	i_1	i_2	i_3	i_4	i_5	i_6
u_a	?	?	4.0	3.0	?	1.0
u_1	?	4.0	4.0	2.0	1.0	2.0
u_2	3.0	?	?	?	5.0	1.0
u_3	3.0	?	?	3.0	2.0	2.0
u_4	4.0	?	?	2.0	1.0	1.0
u_5	1.0	1.0	?	?	?	?
u_6	?	1.0	?	?	1.0	1.0
	3.5	4.0			1.3	

Recommendations: i_2, i_1



1. Find k nearest neighbors for the user in the user-item matrix.
2. Generate recommendation based on the items liked by the k nearest neighbors. E.g., average ratings or use a weighting scheme.

User–based CF (UCBF)

- Pearson correlation coefficient:

$$\mathrm{sim}_{\mathrm{Pearson}}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i \in I} x_i y_i - I\bar{\mathbf{x}}\bar{\mathbf{y}}}{(I-1) s_x s_y}$$

- Cosine similarity:

$$\mathrm{sim}_{\mathrm{Cosine}}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}$$

- Jaccard index (only binary data):

$$\mathrm{sim}_{\mathrm{Jaccard}}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

where $\mathbf{x} = b_{u_x,\cdot}$ and $\mathbf{y} = b_{u_y,\cdot}$ represent the user's profile vectors and X and Y are the sets of the items with a 1 in the respective profile.

Problem

Memory-based. Expensive online similarity computation.

Item-Based CF (ICBF)

- Produce recommendations based on the relationship between items in the user-item matrix (Kitts et al., 2000; Sarwar et al., 2001)

S	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
i_1	-	0.1	0	**0.3**	**0.2**	**0.4**	0	0.1
i_2	0.1	-	**0.8**	**0.9**	0	**0.2**	0.1	0
i_3	0	**0.8**	-	0	**0.4**	0.1	0.3	**0.5**
i_4	**0.3**	**0.9**	0	-	0	**0.3**	0	0.1
i_5	**0.2**	0	**0.7**	0	-	**0.2**	0.1	0
i_6	**0.4**	**0.2**	0.1	**0.3**	0.1	-	0	0.1
i_7	0	**0.1**	**0.3**	0	0	0	-	0
i_8	**0.1**	0	**0.9**	**0.1**	0	0.1	0	-
	-	0	4.56	2.75	-	2.67	0	-

k=3

$u_a = \{i_1, i_5, i_8\}$

$r_{ua} = \{2, ?, ?, ?, 4, ?, ?, 5\}$

Recommendation: i_3

1. Calculate similarities between items and keep for each item only the values for the k most similar items.
2. Use the similarities to calculate a weighted sum of the user's ratings for related items.

$$\hat{r}_{ui} = \sum_{j \in s_i} s_{ij} r_{uj} / \sum_{j \in s_i} |s_{ij}|$$

Regression can also be used to create the prediction.

Item–Based CF (ICBF)

Similarity measures:

- Pearson correlation coefficient, cosine similarity, jaccard index
- Conditional probability-based similarity (Deshpande and Karypis, 2004):

$$\mathrm{sim}_{\mathrm{Conditional}}(x, y) = \frac{\mathrm{Freq}(xy)}{\mathrm{Freq}(x)} = \hat{P}(y|x)$$

  where x and y are two items, $\mathrm{Freq}(\cdot)$ is the number of users with the given item in their profile.

Properties

- Model (reduced similarity matrix) is relatively small ($N \times k$) and can be fully precomputed.
- Item-based CF was reported to only produce slightly inferior results compared to user-based CF (Deshpande and Karypis, 2004).
- Higher order models which take the joint distribution of sets of items into account are possible (Deshpande and Karypis, 2004).
- Successful application in large scale systems (e.g., Amazon.com)

Mean Normalization:

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix}$$

$$\mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \rightarrow Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

Cold Start Problem

- What happens with new users where we have no ratings yet?
    - ✓ Recommend popular items
    - ✓ Have some start-up questions (e.g., "tell me 10 movies you love")

- What do we do with new items?
    - ✓ Content-based filtering techniques.
    - ✓ Pay a focus group to rate them.