Clustering

Instructor: Junghye Lee

Department of Industrial Engineering
junghyelee@unist.ac.kr

Data Mining Tasks (Recap)

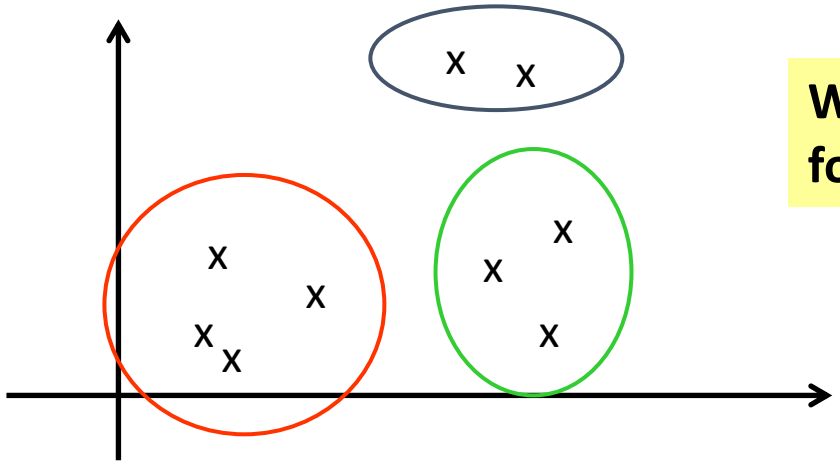
- Prediction Tasks
  - Use some variables to predict unknown or future values of other variables
- Description Tasks
  - Find human-interpretable patterns that describe the data.

- Common data mining tasks
  - Regression [Predictive]
  - Classification [Predictive]
  - **Clustering [Descriptive]**
  - Association Rule Discovery [Descriptive]

Motivation – Why Do Clustering?

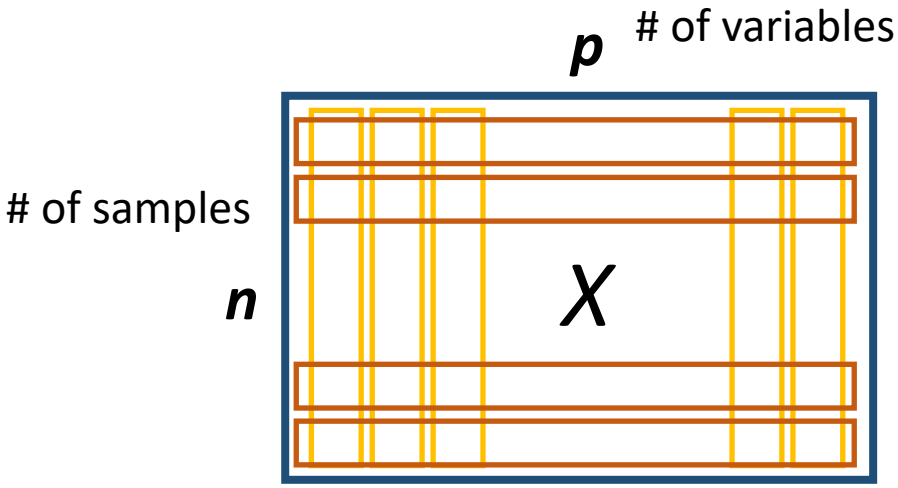
What Is Clustering?

- A form of ***unsupervised learning*** – you generally don't have examples demonstrating how the data *should* be grouped together (i.e. no label)
  - So, it's a method of ***data exploration*** – a way of looking for patterns or structure in the data that are of interest
- A way of grouping together data samples that are ***similar*** in some way - according to some criteria that you pick



What should be the clusters for these data points?

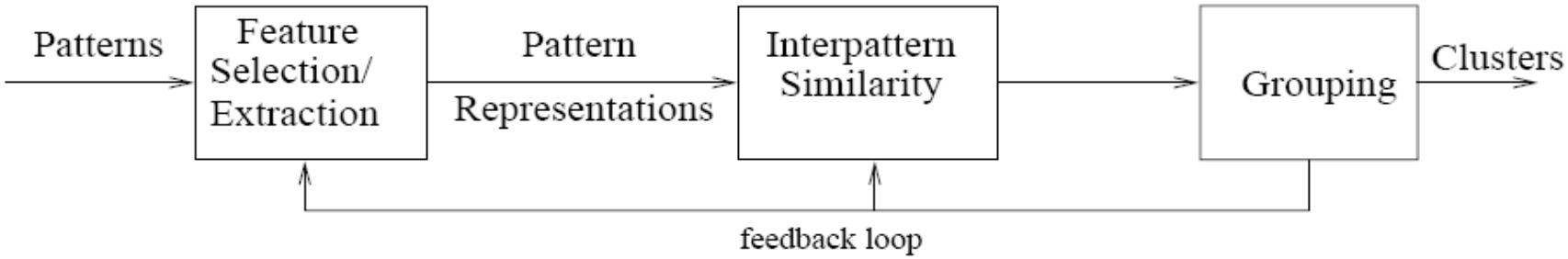
What Kind of Clusters?



A	B	C	D	E	F	G	H	I	J	K	L
ALP_diff	ALP_first	ALP_last	ALP_max	ALP_mean	ALP_min	ALT_diff	ALT_first	ALT_last	ALT_max	ALT_mean	ALT_min
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	233	233	233	233	233	0	1846	1846	1846	1846	1846
8	38	46	56	46.25	38	2	28	30	37	31	28
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	51	51	51	51	51	0	17	17	17	17	17
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	35	35	35	35	35	0	28	28	28	28	28
0	42	42	42	42	42	0	31	31	31	31	31
0	135	135	135	135	135	0	42	42	42	42	42
-31	92	61	92	76.5	61	9	91	100	100	95.5	91
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	85	85	85	85	85	0	13	13	13	13	13
-33	162	129	162	145	129	-3	149	146	165	153.333333	146
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	82	82	82	82	82	0	41	41	41	41	41
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	55	55	55	55	55	0	17	17	17	17	17
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-8	58	50	58	53.6666667	50	-3	17	14	17	15.3333333	14
-40	136	96	136	116	96	19	393	412	412	402.5	393
0	102	102	102	102	102	0	17	17	17	17	17
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	74	74	74	74	74	0	10	10	10	10	10
-43	905	862	905	890	862	1	60	61	61	60.6666667	60
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-15	117	102	117	108	102	-156	304	148	304	228.333333	148
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	83	83	83	83	83	0	14	14	14	14	14
4	36	40	40	38	36	-39	140	101	140	120.5	101
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	52	52	52	52	52	0	16	16	16	16	16
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	38	40	40	39	38	-1555	5845	4290	5845	5067.5	4290
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-50	176	126	176	149	126	-85	164	79	164	120.666667	79
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	54	56	56	55	54	-151	288	137	288	212.5	137
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
57	186	243	243	214.5	186	15	85	100	100	92.5	85

- Cluster genes = columns
  - e.g.) Similar expression patterns may suggest similar functions of genes (gene networks)
- Cluster samples = rows ★
  - e.g.) Similar expression patterns may suggest biological relationship among samples (genotyping)

Overview of clustering



- From the paper “Data clustering: review”
- Feature Selection
  - identifying the most effective subset of the original features to use in clustering
- Feature Extraction
  - transformations of the input features to produce new salient features.
- Inter-pattern Similarity
  - measured by a distance function defined on pairs of patterns.
- Grouping
  - methods to group similar patterns in the same cluster

(Dis)similarity Measures

Data Representations for Clustering

- Input data to algorithm is usually a vector (also called a "tuple" or "record")
- Example: Clinical Sample Data
  - Age (numerical)
  - Weight (numerical)
  - Gender (categorical)
  - Diseased? (binary)
- Types of data
  - Numerical
  - Categorical
  - Boolean
- Must also include a method for computing similarity of or distance between vectors

How do we define “similarity”?

- Recall that the goal is to group together “similar” data – but what does this mean?
- No single answer – it depends on what we want to find or emphasize in the data; this is one reason why clustering is an “art”
- The similarity measure is often more important than the clustering algorithm used – don’t overlook this choice!

Data structures

- Data matrix
  - (two modes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{bmatrix}$$

(Dis)similarity measures

- Instead of talking about similarity measures, we often equivalently refer to dissimilarity measures.
- Jagota defines a dissimilarity measure as a function $f(x, y)$ such that $f(x, y) > f(w, z)$ if and only if x is less similar to y than w is to z.
- This is always a *pair-wise* measure.

Continuous Variable

- Standardize data
  - Calculate the mean absolute deviation:

$$s_f = \frac{1}{n}\left(\left|x_{1f} - m_f\right| + \left|x_{2f} - m_f\right| + \cdots + \left|x_{nf} - m_f\right|\right)$$

  where

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \cdots + x_{nf}\right)$$

  - Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation

Distance Measure

- Euclidean distance

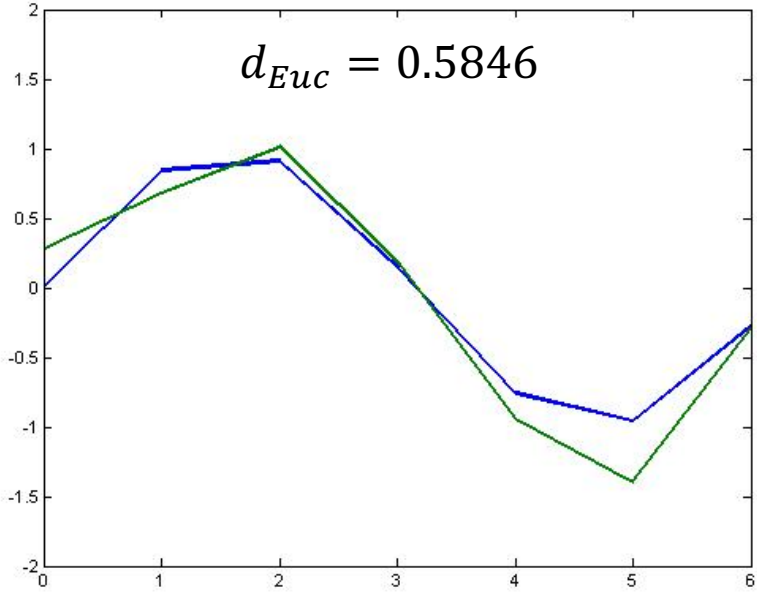
$$d(g_1, g_2) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

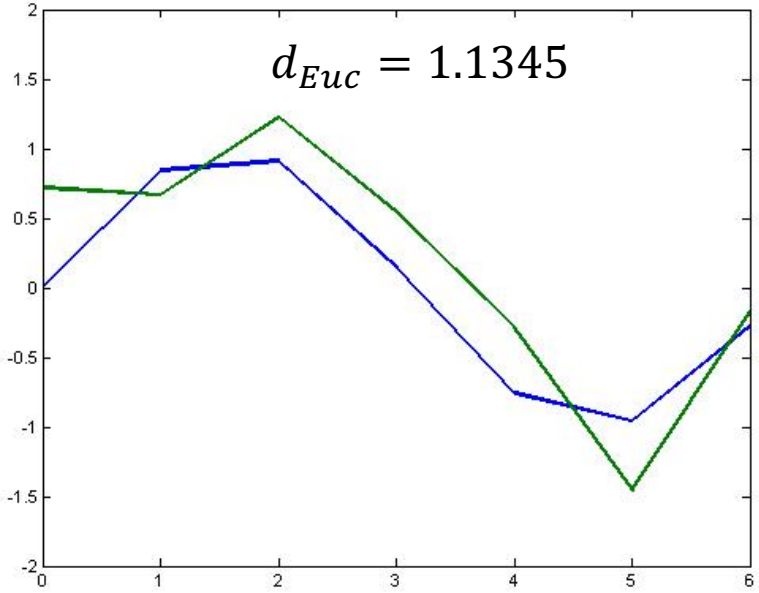
- Manhattan distance

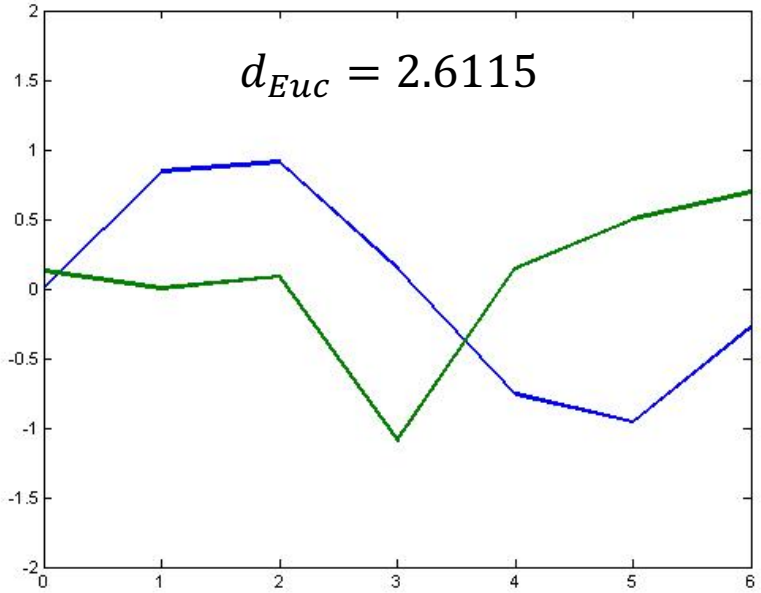
$$d(g_1, g_2) = \sum_{i=1}^{n} |x_i - y_i|$$

- Minkowski distance

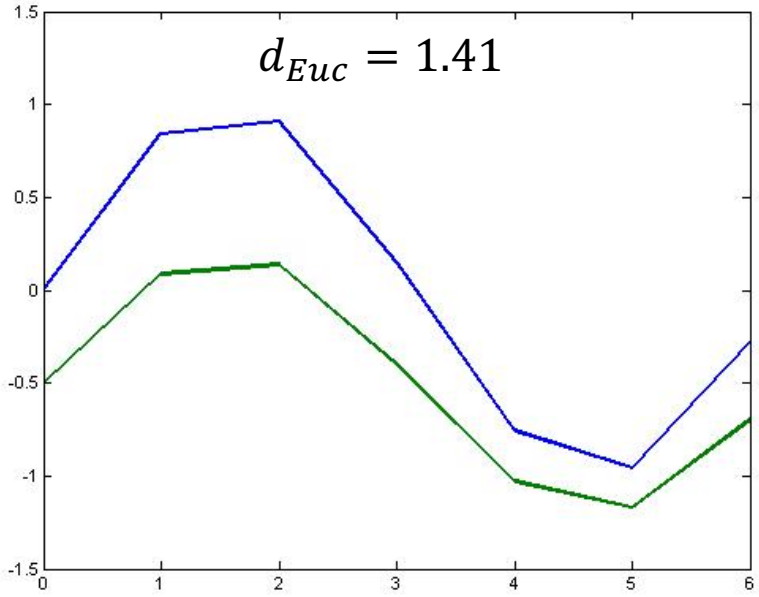
$$d(g_1, g_2) = \sqrt[m]{\sum_{i=1}^{n}(x_i - y_i)^m}$$

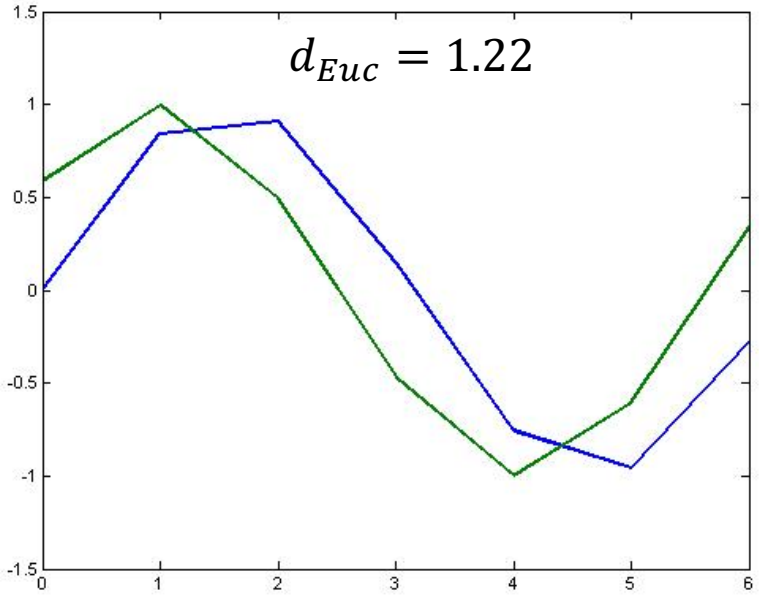






These examples of Euclidean distance match the intuition of dissimilarity pretty well.



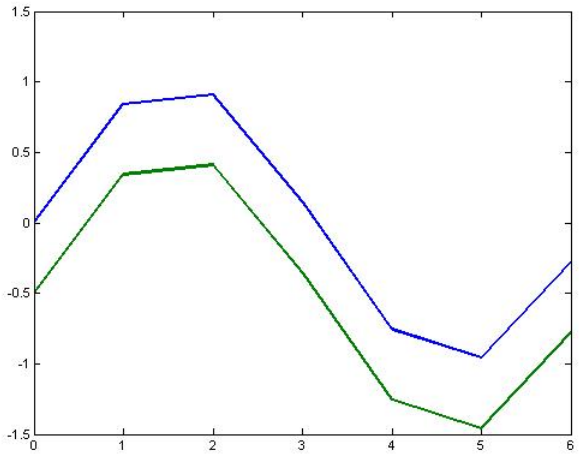


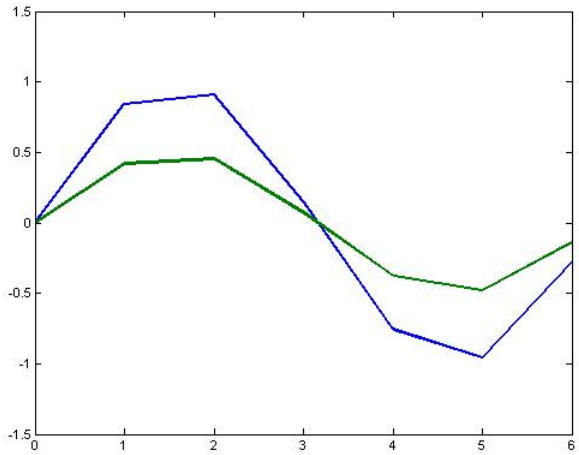
What about these?

What might be going on with the expression profiles on the left? On the right?

Correlation

- We might care more about the overall shape of expression profiles rather than the actual magnitudes
- That is, we might want to consider genes similar when they are “up” and “down” together
- When might we want this kind of measure? What experimental issues might make this appropriate?





Pearson Linear Correlation

$$\rho(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

- We're shifting the expression profiles down (subtracting the means) and scaling by the standard deviations (i.e., making the data have mean = 0 and std = 1)

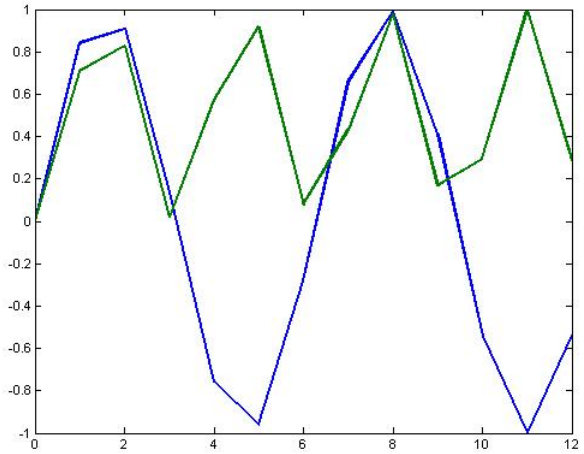
Pearson Linear Correlation

- Pearson linear correlation (PLC) is a measure that is invariant to scaling and shifting (vertically) of the expression values
- Always between –1 and +1 (perfectly anti-correlated and perfectly correlated)
- This is a similarity measure, but we can easily make it into a dissimilarity measure:

$$d_p = \frac{1 - \rho(x, y)}{2}$$

Pearson Linear Correlation

- PLC only measures the degree of a *linear* relationship between two expression profiles!
- If you want to measure other relationships, there are many other possible measures.



$\rho = 0.0249$ so $d_p = 0.4876$

The green curve is the square of the blue curve – this relationship is not captured with PLC

Binary Variable

- A contingency table for binary data

		Object *j*		
		1	0	*sum*
Object *i*	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- Simple matching coefficient (invariant, if the binary variable is *symmetric*):

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

- Jaccard coefficient (non-invariant if the binary variable is *asymmetric*):

$$d(i,j) = \frac{b+c}{a+b+c}$$

Dissimilarity of Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Gender is a symmetric attribute.
- The remaining attributes are asymmetric binary.
- Let the values Y and P be set to 1, and the value N be set to 0.

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.33$$

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i,j) = \frac{p - m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
  - replacing x_{if} by their rank $r_{if} \in \{1, \dots, M_f\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - compute the dissimilarity using methods for interval-scaled variables

Clustering Algorithms – 1. Hierarchical Clustering

Hierarchical Clustering

- There are two styles of hierarchical clustering algorithms to build a tree from the input set *S*:
  - **Agglomerative (bottom-up)**:
    - Beginning with singletons (sets with 1 element)
    - Merging them until *S* is achieved as the root.
    - It is the most common approach.
  - **Divisive (top-down)**:
    - Recursively partitioning *S* until singleton sets are reached.

Linkage in Hierarchical Clustering

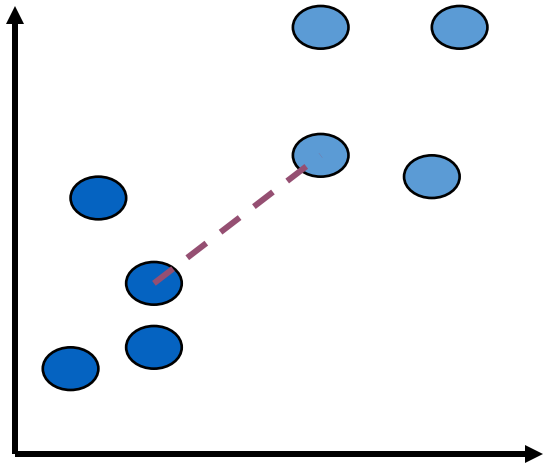
- We already know about distance measures between data items, but what about between a data item and a cluster or between two clusters?
- We just treat a data point as a cluster with a single item, so our only problem is to define a ***linkage*** method between clusters
- As usual, there are lots of choices…

Average Linkage (Two Styles)

- Eisen's cluster program defines average linkage as follows:
  - Each cluster c_i is associated with a mean vector μ_i which is the mean of all the data items in the cluster.
  - The distance between two clusters c_i and c_j is then just $d(\mu_i, \mu_j)$.
- This is somewhat non-standard.
  - This method is usually referred to as centroid linkage.
- The real average linkage is defined as the average of all pairwise distances between points in the two clusters.

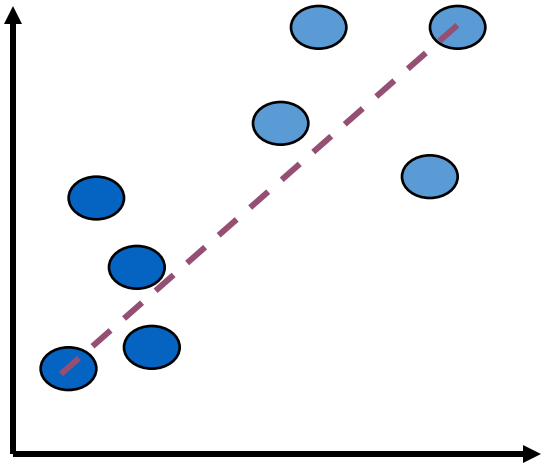
Single Linkage

- The minimum of all pairwise distances between points in the two clusters
- Tends to produce long, **“loose”** clusters



Complete Linkage

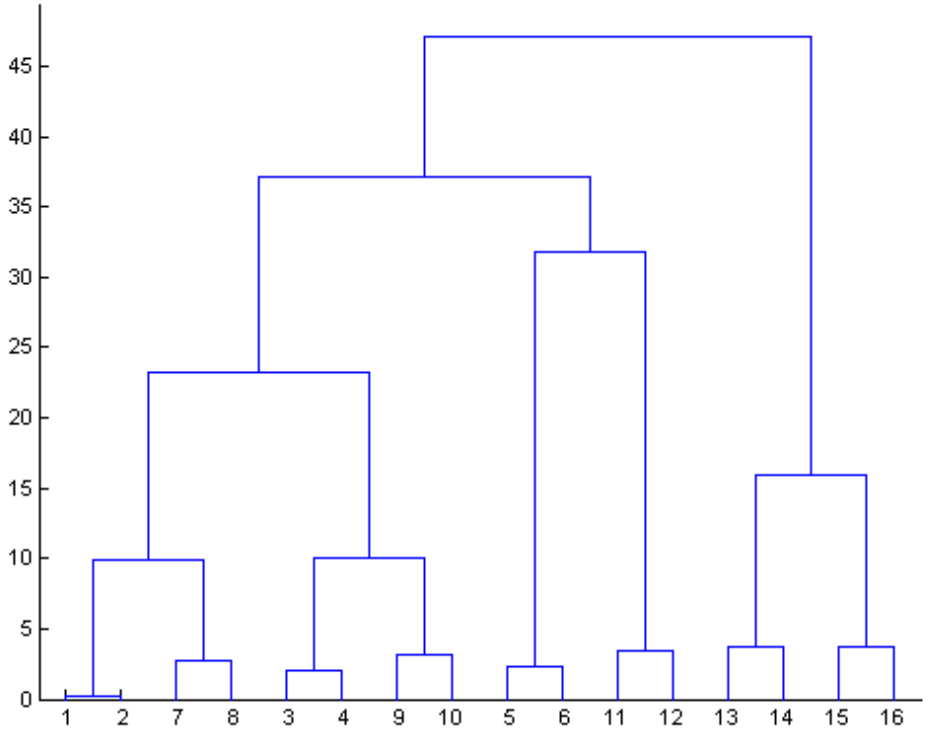
- The maximum of all pairwise distances between points in the two clusters
- Tends to produce very **“tight”** clusters



Hierarchical Agglomerative Clustering

- We start with every data point in a separate cluster
- We keep merging the most similar pairs of data points/clusters until we have one big cluster left
- This is called a bottom-up or agglomerative method

Hierarchical Agglomerative Clustering

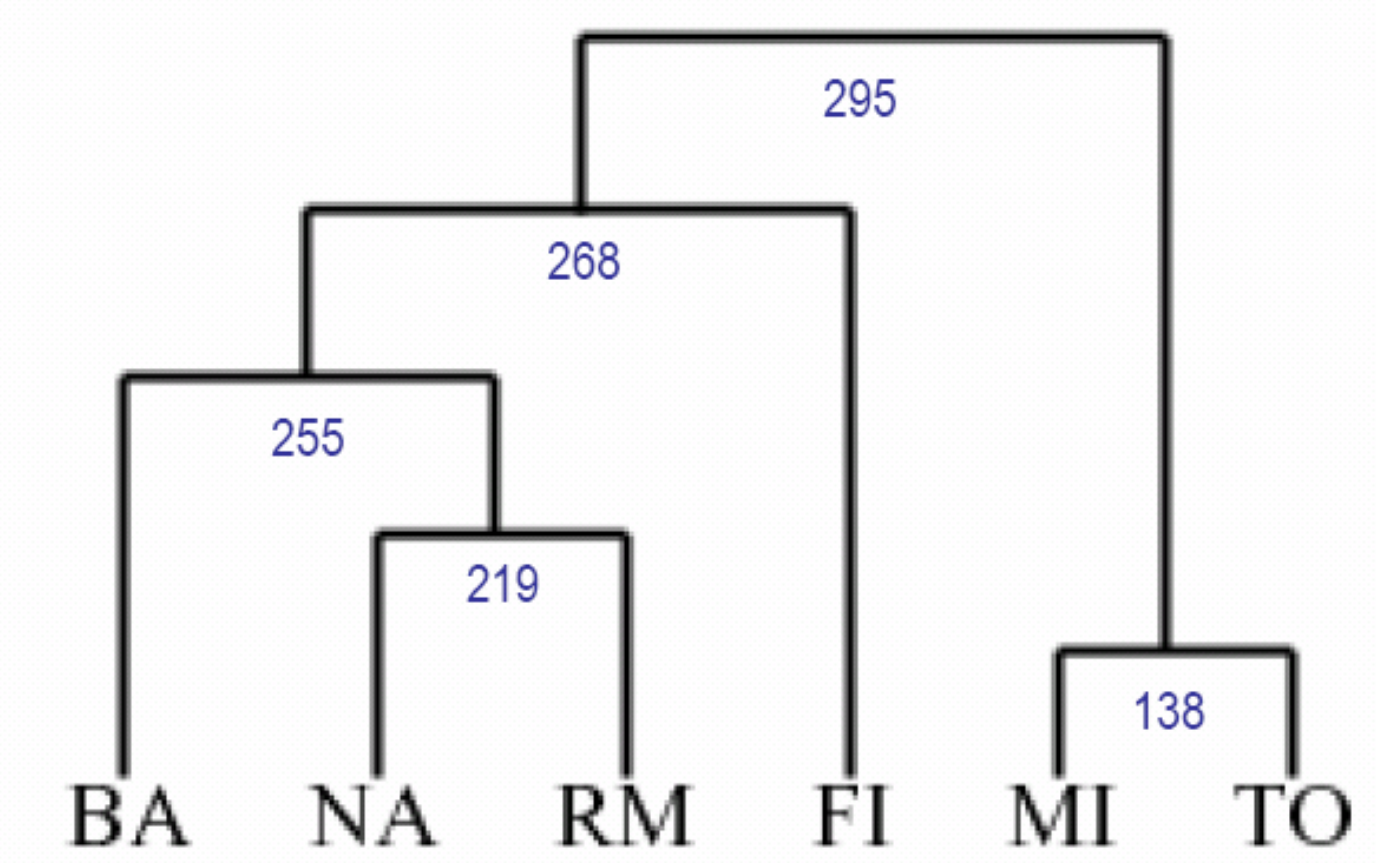


- This produces a binary tree or ***dendrogram***
- The final cluster is the root and each data item is a leaf
- The height of the bars indicate how close the items are

Hierarchical Clustering Example

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

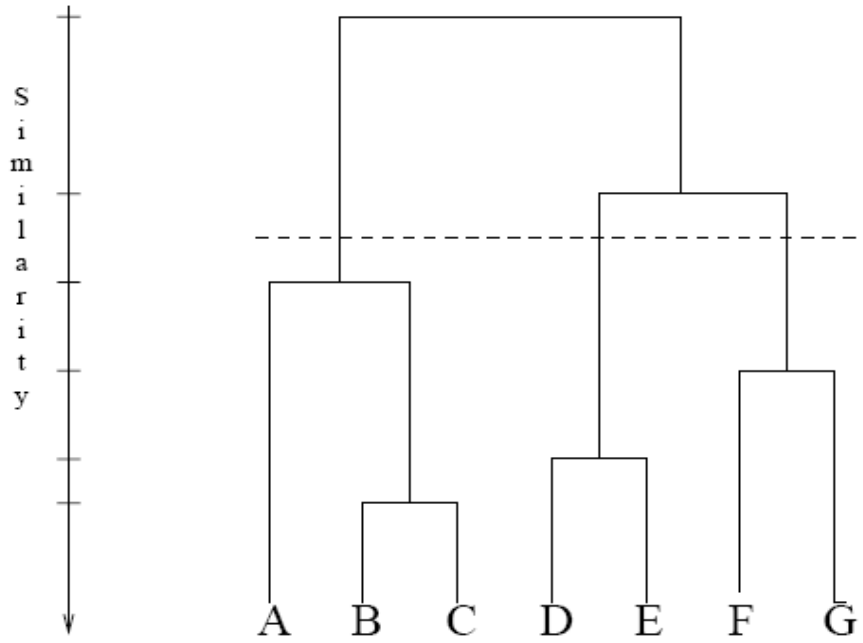
Hierarchical Clustering Example

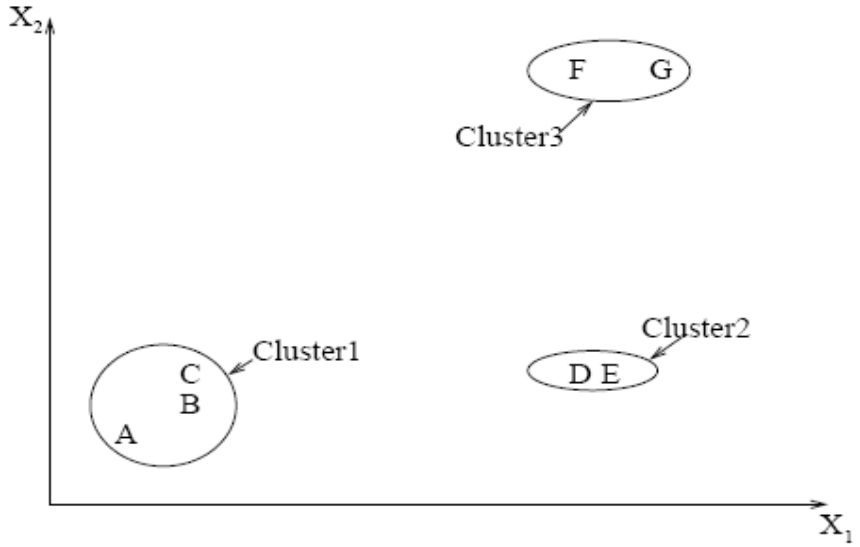


Which linkage method did they use?

Formation of Clusters

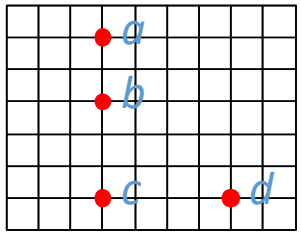
- Forming clusters from dendrograms

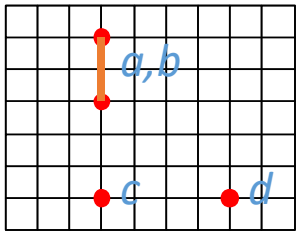




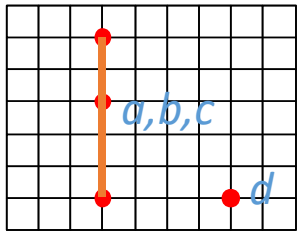
Single-Link Method

Euclidean Distance

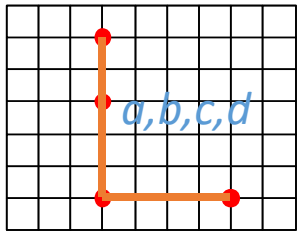




(1)



(2)



(3)

	b	*c*	*d*
a	2	5	6
b		3	5
c			4

	b	*c*	*d*
a	2	5	6
b		3	5
c			4

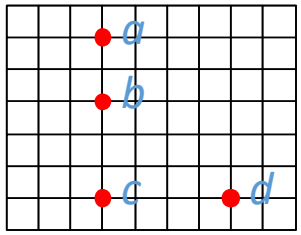
	c	*d*
a*,*b	3	5
c		4

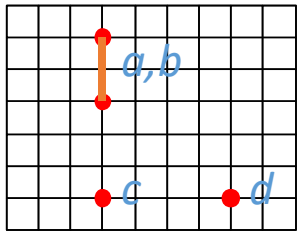
	d
a*,*b*,*c	4

Distance Matrix

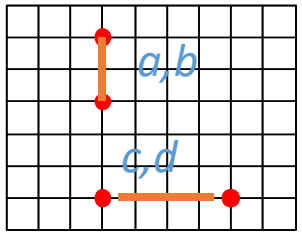
Complete-Link Method

Euclidean Distance

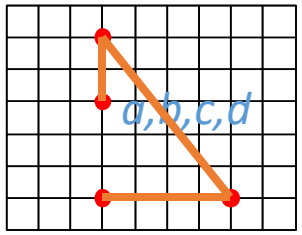




(1)



(2)



(3)

	b	c	d
a	2	5	6
b		3	5
c			4

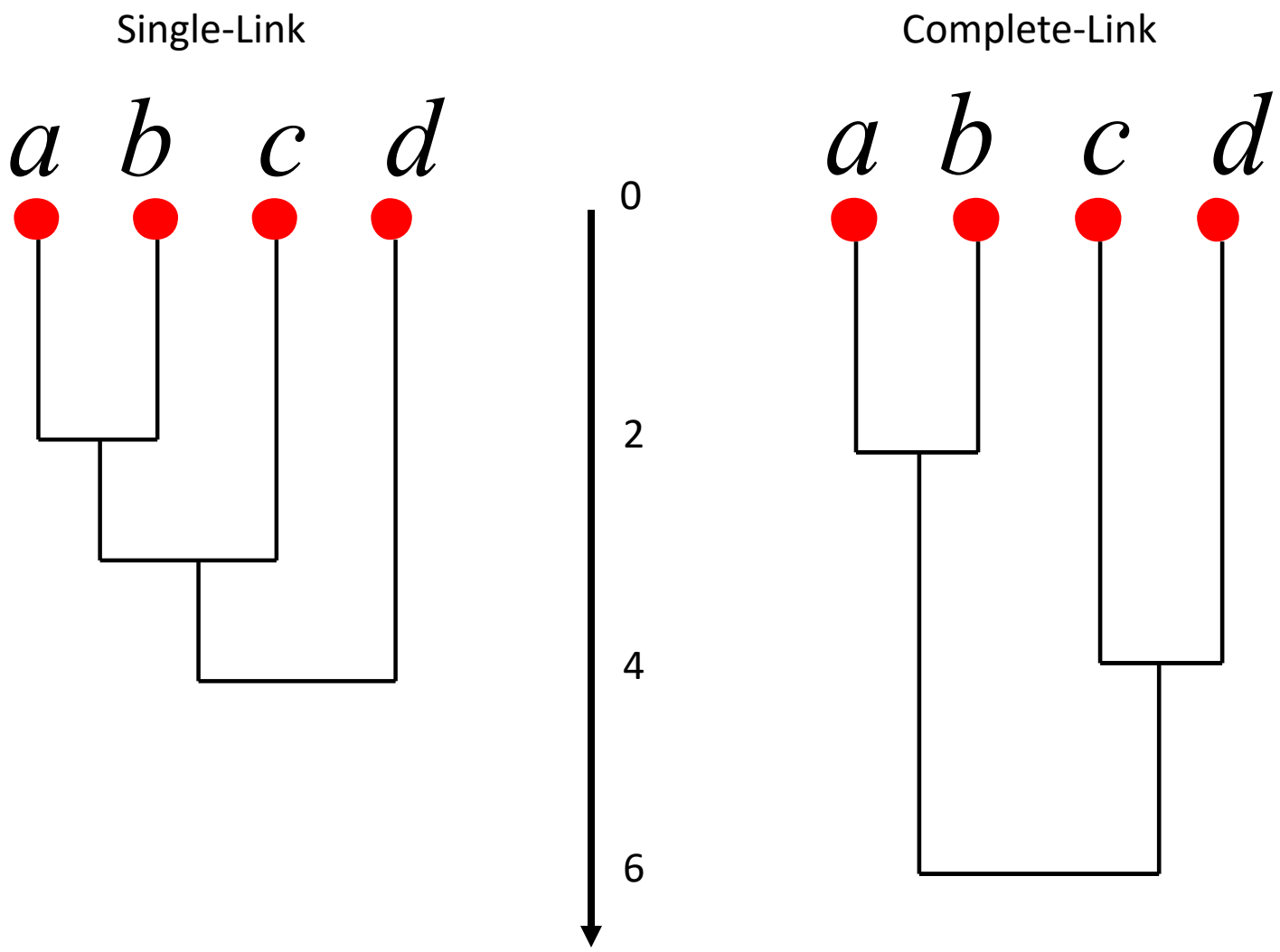
	b	c	d
a	2	5	6
b		3	5
c			4

	c	d
a,b	5	6
c		4

	c,d
a,b	6

Distance Matrix

Compare Dendrograms



Hierarchical Clustering Issues

- Distinct clusters are not produced – sometimes this can be good, if the data has a hierarchical structure without clear boundaries
- There are methods for producing distinct clusters, but these usually involve specifying somewhat arbitrary cutoff values
- What if data doesn't have a hierarchical structure? Is HC appropriate?

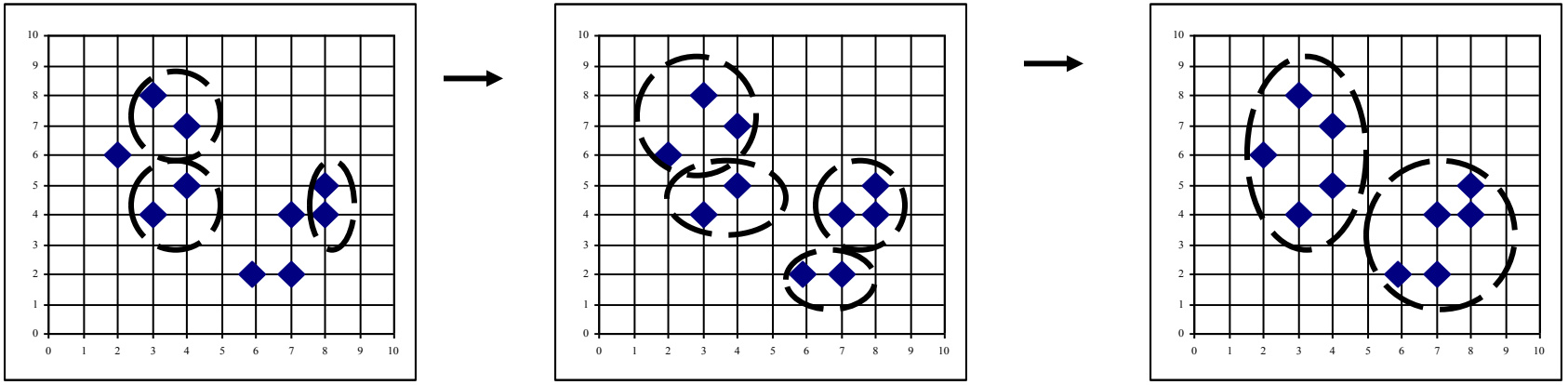
Hierarchical Clustering

- Advantages
  - Dendrograms are great for visualization
  - Provides hierarchical relations between clusters
  - Shown to be able to capture concentric clusters

- Disadvantages
  - Not easy to define levels for clusters
  - Experiments showed that other clustering techniques outperform hierarchical clustering

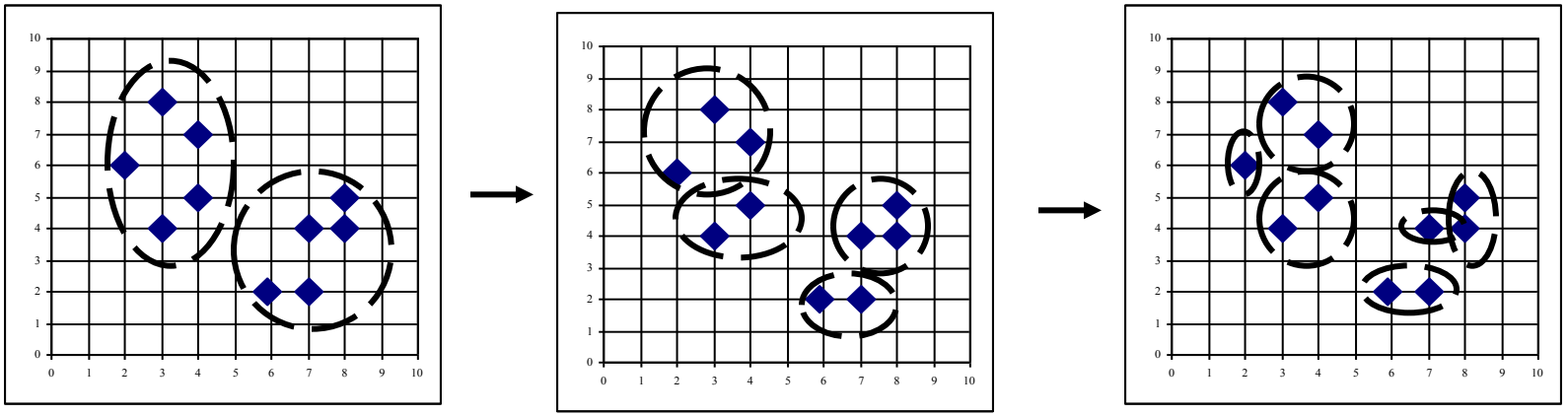
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



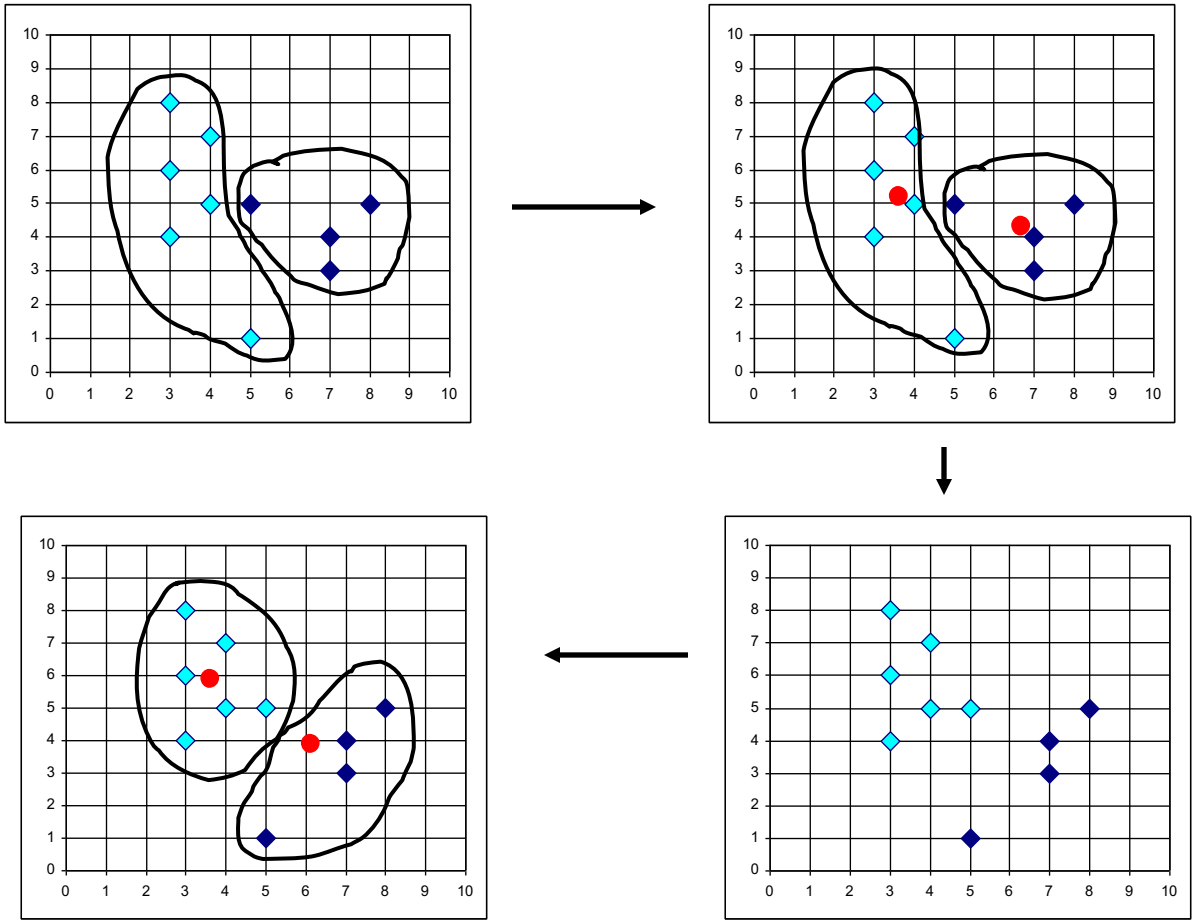
Clustering Algorithms – 2. *k*-means Clustering

k-means Clustering

1. Choose a number of clusters *k*
2. Initialize cluster centers $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$
   - Could pick *k* data points and set cluster centers to these points
   - Or could randomly assign points to clusters and take means of clusters
3. For each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster
4. Re-compute cluster centers (mean of data points in cluster)
5. Stop when there are no new re-assignments

k-means Clustering

- Example



k-means Clustering

- Stopping criteria:
  - No change in the members of all clusters
  - when the squared error is less than some small threshold value α
    - Squared error *se*

$$se = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in c_i} \|\boldsymbol{x} - \boldsymbol{m_i}\|^2$$

      - where m_i is the mean of all instances in cluster c_i
    - $se^{(t)} < \alpha$(after *t* iterations)
- Properties of *k*-means
  - Guaranteed to converge
  - Guaranteed to achieve local optimal, not necessarily global optimal.

k-means Clustering

- Pros:
  - Low complexity

- Cons:
  - Necessity of specifying *k*
  - Sensitive to noise and outlier data points
    - Outliers: a small number of such data can substantially influence the mean value)
  - Clusters are sensitive to initial assignment of centroids
    - *k*-means is not a deterministic algorithm
    - Clusters can be inconsistent from one run to another

k-means Clustering Issues

- Random initialization means that you may get different clusters each time
- Data points are assigned to only one cluster (hard assignment)
- Implicit assumptions about the “shapes” of clusters
- You have to pick the number of clusters

Determining # of Clusters

- We'd like to have a measure of cluster quality *Q* and then try different values of *k* until we get an optimal value for *Q*
- But, since clustering is an unsupervised learning method, we can't really expect to find a "correct" measure *Q*.
- So, once again there are different choices of *Q* and our decision will depend on what dissimilarity measure we're using and what types of clusters we want

Cluster Quality Measures

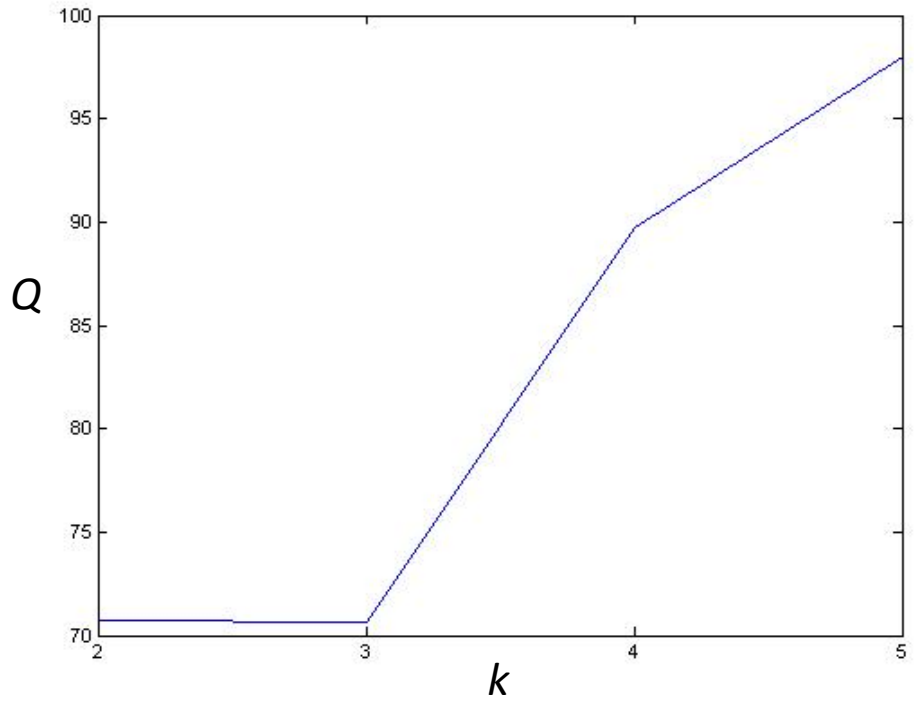
- Jagota suggests a measure that emphasizes cluster tightness or homogeneity:

$$Q = \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x} \in c_i} d(\boldsymbol{x}, \boldsymbol{m}_i)$$

- $|c_i|$ is the number of data points in cluster i
- Q will be small if (on average) the data points in each cluster are close

Cluster Quality

- This is a plot of the *Q* measure for *k*-means clustering on the data shown earlier.
- How many clusters do you think there actually are?



Cluster Quality

- The *Q* measure takes into account homogeneity within clusters, but not separation between clusters
- Other measures try to combine these two characteristics (i.e., Davies-Bouldin measure, Silhouette)

- An alternate approach is to look at cluster stability:
  - Add random noise to the data many times and count how many pairs of data points no longer cluster together
  - How much noise to add?
    - Should reflect estimated variance in the data

Davies-Bouldin Measure

$$DB \equiv \frac{1}{k}\sum_{i=1}^{k} D_i$$

where $D_i = \max_{j \neq i} R_{i,j}$, $R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$, k is the number of clusters, and

$$S_i = \left(\frac{1}{n_i}\sum_{j=1}^{n_i} \left|\boldsymbol{x}_j - \boldsymbol{m}_i\right|^p\right)^{1/p}$$ and $$M_{i,j} = \left\|\boldsymbol{m}_i - \boldsymbol{m}_j\right\|_p,$$

Within **Between**

where n_i is the number of samples in the *i*-th cluster.

Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average distance between i and all other data within the same cluster, and $b(i)$ is **the lowest average distance of i to all points in the other clusters, where i is not a member.**

Same as "neighboring cluster"

$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i) \end{cases}$$

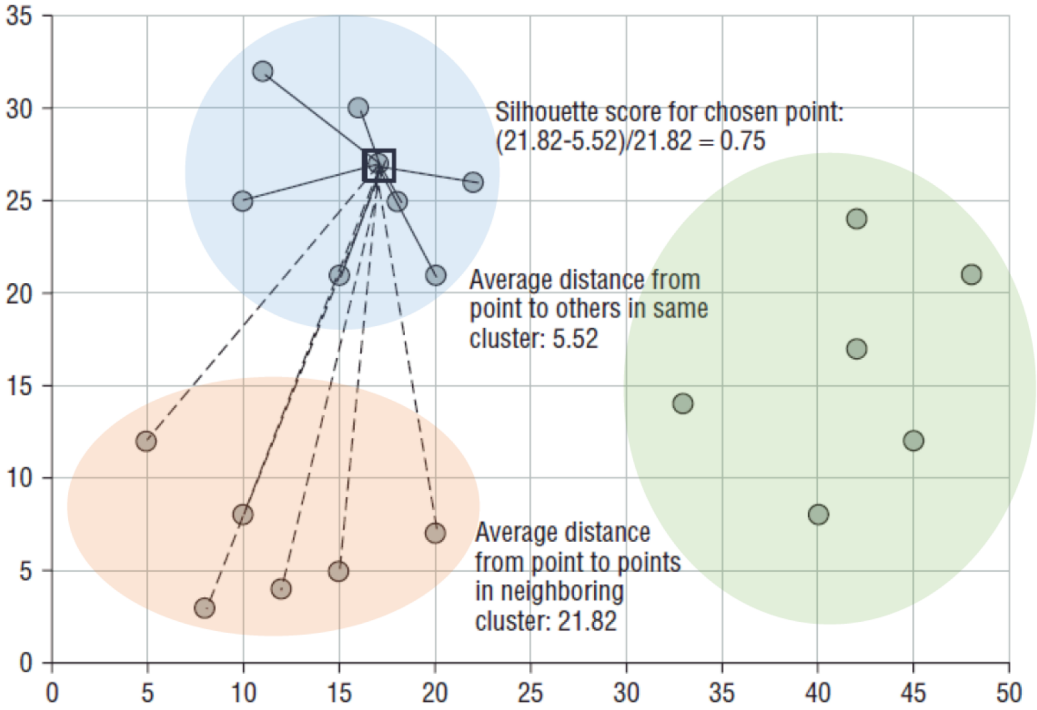
What is the range of this measure?

$$-1 \leq s(i) \leq 1$$

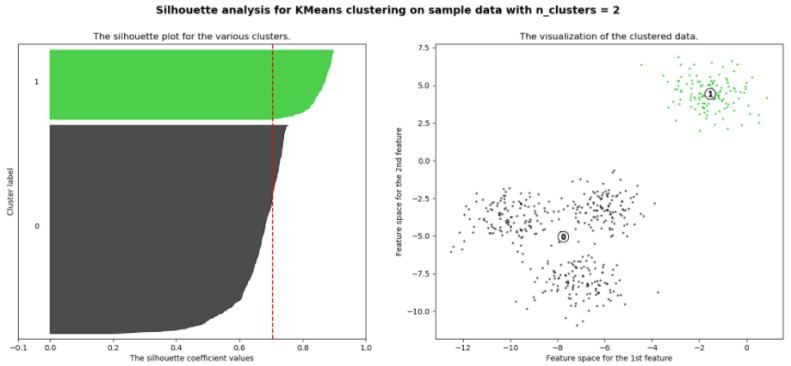
Silhouette

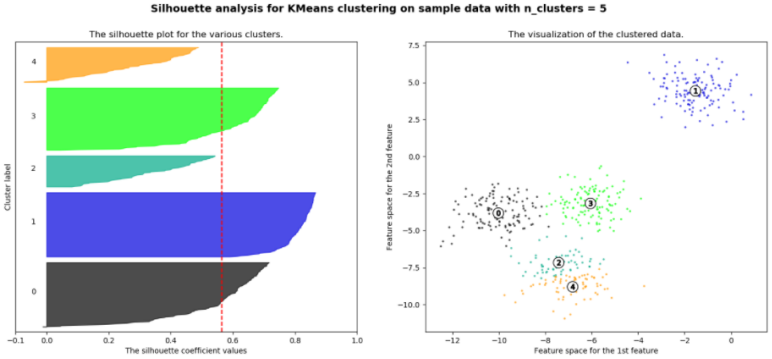
Best case $a(i) = 0$, worse case $b(i) = 0$

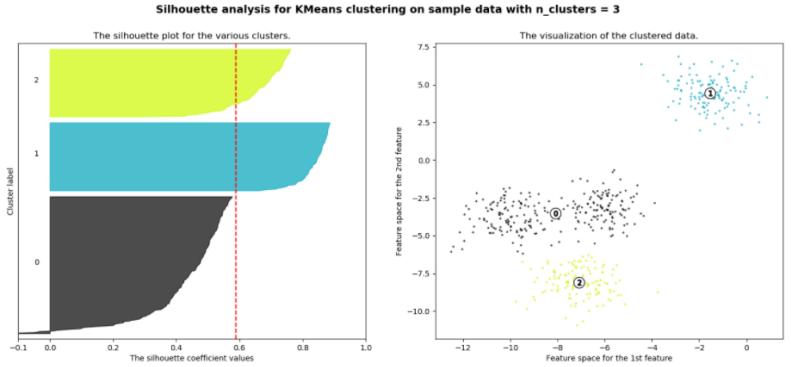
If $1/n \sum_{i=1} s(i) > 0.5$, then we can say the clustering result is reasonable.

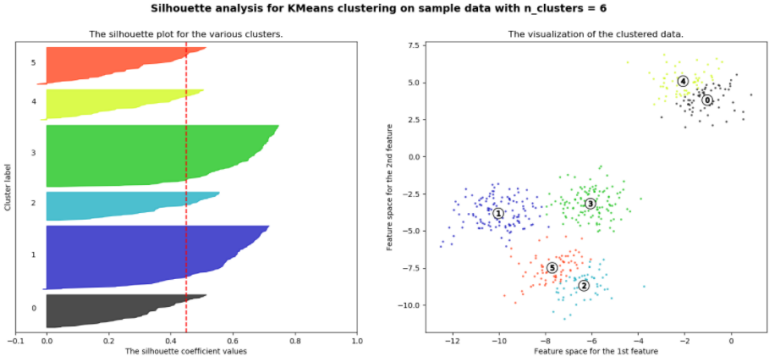


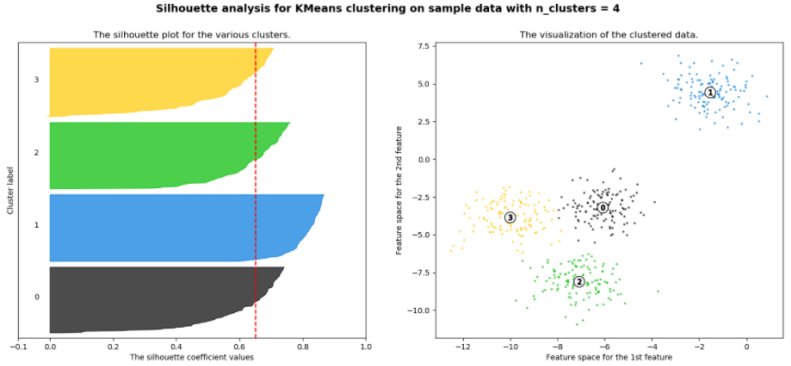
Silhouette












```
For n_clusters = 2 The average silhouette_score is : 0.704978749608
For n_clusters = 3 The average silhouette_score is : 0.588200401213
For n_clusters = 4 The average silhouette_score is : 0.650518663273
For n_clusters = 5 The average silhouette_score is : 0.563764690262
For n_clusters = 6 The average silhouette_score is : 0.450466629437
```

Fuzzy *c*-means

- An extension of *k*-means
- Hierarchical, *k*-means generates partitions
  - each data point can only be assigned in one cluster
- Fuzzy *c*-means allows data points to be assigned into more than one cluster
  - each data point has a degree of membership (or probability) of belonging to each cluster

Fuzzy *c*-means Algorithm

- Let x_i be a data point.

1. Initialize membership $U^{(0)} = [u_{ij}]$ for a data point of cluster cl_j by random
2. At the t-th step, compute the fuzzy centroid $C^{(t)} = [\boldsymbol{c_j}]$ for j = 1, .., k, where k is the number of clusters, using

$$\boldsymbol{c_j} = \frac{\sum_{i=1}^{n} (u_{ij})^m \boldsymbol{x_i}}{\sum_{i=1}^{n} (u_{ij})^m}$$

where m is the fuzzy parameter and n is the number of data points.

Fuzzy *c*-means Algorithm

3. Update the fuzzy membership $U^{(t)} = [u_{ij}]$, using

$$u_{ij} = \frac{\left(\frac{1}{\left\|x_i - c_j\right\|}\right)^{\frac{2}{(m-1)}}}{\sum_{j=1}^{k}\left(\frac{1}{\left\|x_i - c_j\right\|}\right)^{\frac{2}{(m-1)}}}$$

4. If $\left\|U^{(t)} - U^{(t-1)}\right\| < \epsilon$, then STOP, else return to step 2.
5. Determine membership cutoff
   - For each data point, assign to cluster cl_j if u_{ij} of $U^{(t)} > \alpha$

Fuzzy *c*-means Algorithm

- *k*-means algorithm: $se = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in c_i} \|\boldsymbol{x} - \boldsymbol{m_i}\|^2$
- Fuzzy *c*-means algorithm:

$$se = \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij} \left\|\boldsymbol{x_i} - \boldsymbol{c_j}\right\|^2$$

- The recommended fuzzy parameter $m \in [1.5,4]$

Fuzzy *c*-means

- Pros:
  - Allows a data point to be in multiple clusters
  - A more natural representation of the behavior of genes
    - genes usually are involved in multiple functions
- Cons:
  - Need to define *k*, the number of clusters
  - Need to determine membership cutoff value
  - Clusters are sensitive to initial assignment of centroids
    - Fuzzy *c*-means is not a deterministic algorithm

Other Clustering Algorithms

- Clustering is a very popular method of microarray analysis and also a well established statistical technique – huge literature out there
- Many variations on *k*-means, including algorithms in which clusters can be split and merged or that allow for soft assignments (multiple clusters can contribute)
  - *k*-medoids
- *Semi-supervised* clustering methods, in which some examples are assigned by hand to clusters and then other membership information is inferred