

Examples

- Target marketing of a niche product for business without large marketing budget
- Gene expression clustering where large quantities of genes may exhibit similar behavior
- As a preliminary step in data mining, to reduce the search space for downstream algorithms

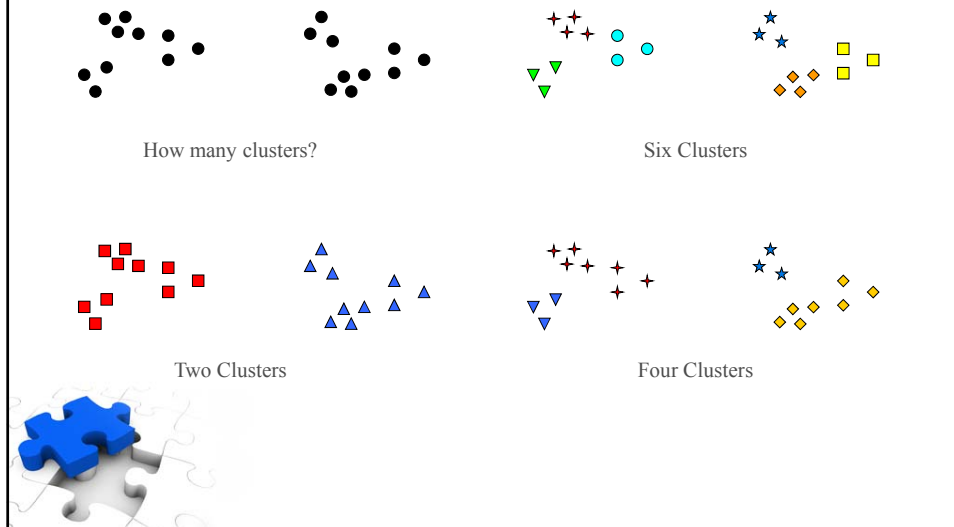


Clustering Issues

- **Need to Determine How...**
 - to recode categorical variables
 - to standardize or normalize numerical variables
 - many clusters we expect to uncover
 - to measure similarity or distance

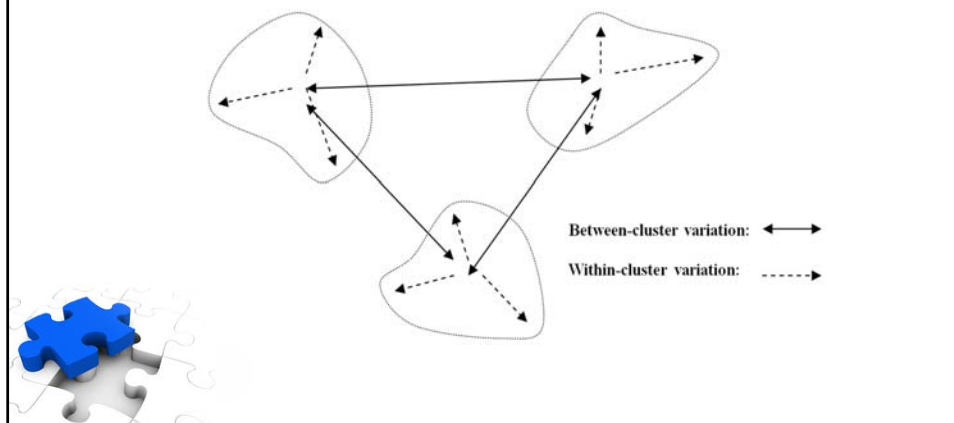


Notion of a Cluster can be Ambiguous



Goal of Clustering

- To construct clusters of records such that the between-cluster variation is large compared to the within-cluster variation



Similarity

- Dissimilarity measure
 - Euclidean distance

$$x_i = (x_{1i}, x_{2i}, \dots, x_{pi})^T$$

$$d(x_i, x_j) = \sqrt{\sum_{a=1}^p (x_{ai} - x_{aj})^2} = \sqrt{(x_i - x_j)^T (x_i - x_j)}$$

- Minkowski distance

$$d(x_i, x_j) = \left(\sum_{a=1}^p |x_{ai} - x_{aj}|^m \right)^{1/m}$$



Similarity

- standardized Euclidean distance

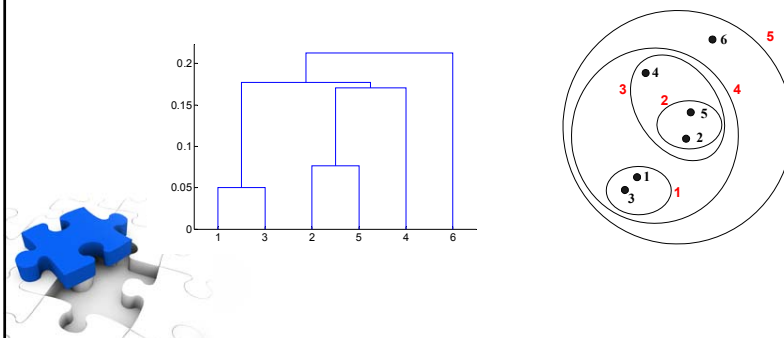
$$d(x_i, x_j) = \sqrt{\sum_{a=1}^p \left(\frac{x_{ai} - x_{aj}}{s_a} \right)^2}$$

$$\text{, where } s_a = \sqrt{\frac{\sum_{a=1}^p (x_{ai} - x_{aj})^2}{(n-1)}}$$



Hierarchical Clustering

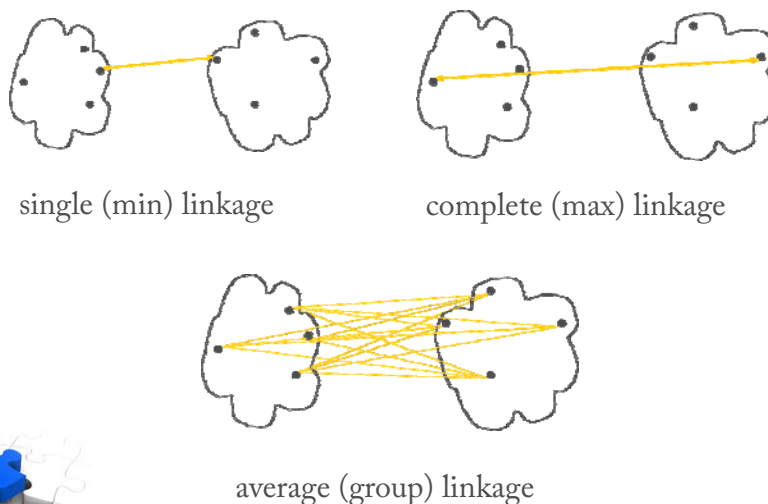
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
 - A tree like diagram that records the sequences of merges or splits



Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
 - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level
- They may correspond to meaningful taxonomies
 - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)

Determining the Distance Between Clusters



Single Linkage

- Two clusters, C_1 and C_2

$$d(C_1, C_2) = \min\{d(x, y) \mid x \in C_1, y \in C_2\}$$

– (UV) vs W

$$d\{(uv), w\} = \min\{d_{uw}, d_{vw}\}$$

– Example

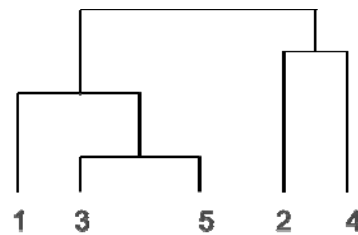
	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

$\Rightarrow \min\{d_{ik}\} = d_{53} = 2$

Single Linkage

- Example
 - Calculate distance (UV) vs W
- $$d\{(35)1\} = \min\{d_{31}, d_{51}\} = 3$$
- $$d\{(35)2\} = \min\{d_{32}, d_{52}\} = 7$$
- $$d\{(35)4\} = \min\{d_{34}, d_{54}\} = 8$$

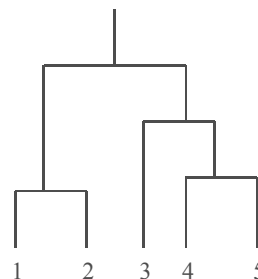
(35) 1 2 4

$$D = \begin{array}{c|cccc} & 0 & & & \\ \hline 3 & 0 & & & \\ 7 & 9 & 0 & & \\ 8 & 6 & 5 & 0 & \end{array}$$


MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
 - Determined by all pairs of points in the two clusters

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00




Average Linkage

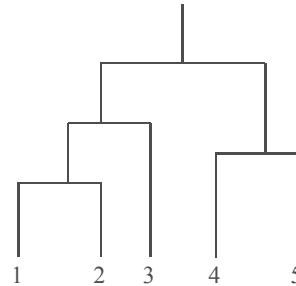
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$d(C_i, C_j) = \frac{1}{n_i n_j} \sum_i \sum_j d_{ij}$$

- Need to use average connectivity for scalability since total proximity favors large clusters



	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Ward Linkage

- Ward (1963)
 - Error sum of squares

$$ESS_A = \sum_{j=1}^{n_A} (X_{Aj} - \bar{X}_A)' (X_{Aj} - \bar{X}_A) = \sum_{j=1}^{n_A} \sum_{t=1}^p (X_{Ajt} - \bar{X}_{At})^2$$

$$ESS_B = \sum_{j=1}^{n_B} (X_{Bj} - \bar{X}_B)' (X_{Bj} - \bar{X}_B) = \sum_{j=1}^{n_B} \sum_{t=1}^p (X_{Bjt} - \bar{X}_{Bt})^2$$

$$ESS_{AB} = \sum_{j=1}^{n_{AB}} (X_{ABj} - \bar{X}_{AB})' (X_{ABj} - \bar{X}_{AB}) = \sum_{j=1}^{n_{AB}} \sum_{t=1}^p (X_{ABjt} - \bar{X}_{ABt})^2$$



Example

– Crime rate in 1975

ID	City	Murder	Rape
1	Atlanta	16.5	24.8
2	Boston	4.2	13.3
3	Chicago	11.6	24.7
4	Dallas	18.9	34.2
5	Denver	6.9	41.5
6	Detroit	13	35.7
7	Hartford	2.5	8.8
8	Honolulu	3.6	12.7
9	Houston	16.8	26.6
10	Kansas City	10.8	43.2
11	Los Angeles	9.7	51.8
12	New Orleans	10.3	39.7
13	New York	9.4	19.4
14	Portland	5.9	23
15	Tucson	5.1	22.9
16	Washington	12.5	27.6



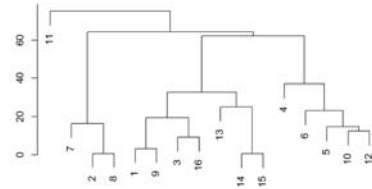
Example

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	16.84														
3	4.90	13.59													
4	9.70	25.55	11.98												
5	19.26	28.33	17.45	14.05											
6	11.45	24.07	11.09	6.09	8.42										
7	21.26	4.81	18.32	30.23	32.99	28.88									
8	17.69	0.85	14.42	26.39	28.99	24.85	4.05								
9	1.82	18.32	5.54	7.88	17.89	9.86	22.83	19.17							
10	19.26	30.62	18.52	12.11	4.25	7.82	35.39	31.34	17.65						
11	27.84	38.89	27.17	19.86	10.67	16.43	43.60	39.57	26.18	8.67					
12	16.14	27.10	15.06	10.21	3.85	4.83	31.87	27.82	14.62	3.54	12.11				
13	8.92	8.02	5.74	17.59	22.24	16.69	12.65	8.86	10.32	23.84	32.40	20.32			
14	10.75	9.85	5.95	17.16	18.53	14.55	14.60	10.55	11.48	20.79	29.05	17.27	5.02		
15	11.56	9.64	6.74	17.84	18.69	15.04	14.34	10.31	12.27	21.09	29.26	17.59	5.54	0.81	
16	4.88	16.53	3.04	9.19	14.99	8.12	21.29	17.36	4.41	15.69	24.36	12.30	8.77	8.04	8.77

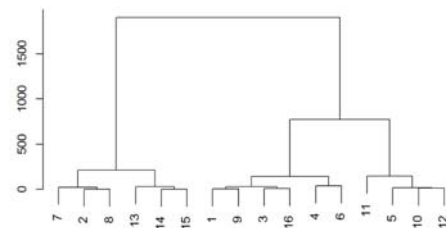


Example

dendrogram: single

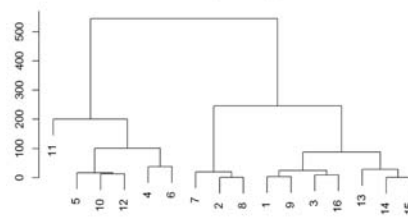


complete linkage

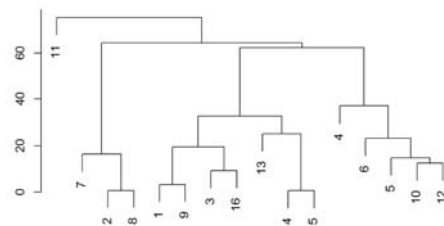


Example

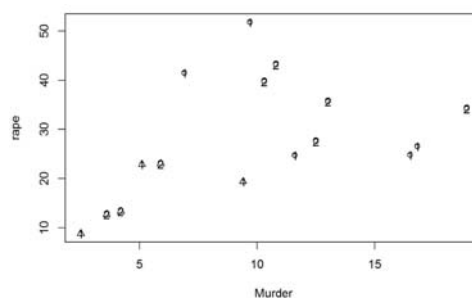
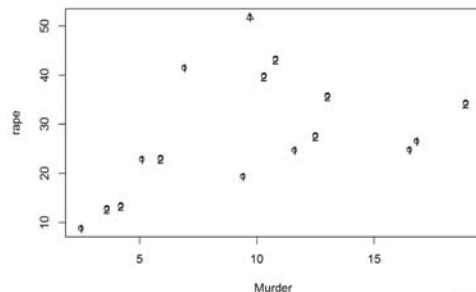
Average Linkage



Ward Method



Example



K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

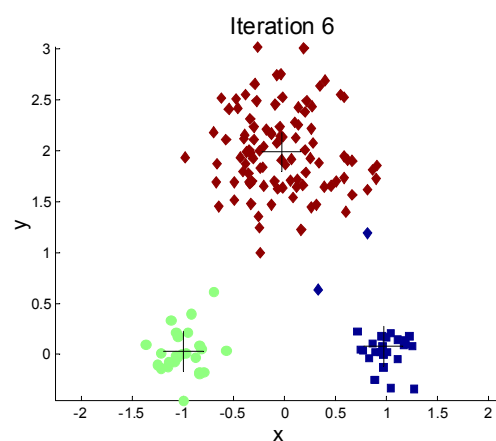
-
- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

K-means Clustering – Details

- **Step 1:** Ask how many clusters k
- **Step 2:** Randomly assign k records to be the initial centers
- **Step 3:** For each record, find the *nearest* center where nearest is usually Euclidean distance
- **Step 4:** For each of the k clusters, find the cluster *centroid*
 C_1, C_2, \dots, C_k
- **Step 5:** Repeat steps 3 to 5 until convergence or termination



Importance of Choosing Initial Centroids



K-Mean Clustering Centroid

- Suppose we have n data points
 $(a_1, b_1, c_1), (a_2, b_2, c_2) \dots (a_n, b_n, c_n)$
- The centroid of these points is located at

$$(\sum a_i/n, \sum b_i/n, \sum c_i/n)$$

For example points (1,1,1), (1,2,1), (1,3,1), and (2,1,1)

Have Centroid $\left(\frac{1+1+1+2}{4}, \frac{1+2+3+1}{4}, \frac{1+1+1+1}{4} \right) = (1.25, 1.75, 1.00)$



Evaluating K-means Clusters

- Stopping criteria
 - Algorithm terminates when the centroid no longer changes
 - Some convergence criterion is met
 - Mean Squared Error (MSE)
 - no significant shrinkages in the MSE

$$MSE = \frac{SSE}{N-k} = \frac{\sum_{i=1}^k \sum_{p \in C_i} d(p, m_i)^2}{N-k}$$

, where SSE ~ Sum of squares error, $p \in C_i$ ~ each data point in cluster i , m_i represents the centroid (cluster center) of cluster i , N is the total sample size, and k is the number of cluster.



Evaluating K-means Clusters

- *pseudo-F statistics*

$$F_{k-1, N-k} = \frac{MSB}{MSE} = \frac{SSB / k - 1}{SSE / N - k}$$

, where MSB ~ the mean square between,
SSB ~ sum of square between clusters

$$SSB = \sum_{i=1}^k n_i d(m_i, M)^2$$

, where M is the mean of all data

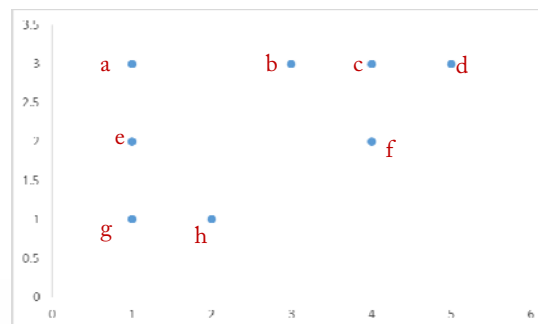
- Good clusters have *large pseudo-F Statistic*



K-Means Clustering

- Data

a	b	c	d	e	f	g	h
(1,3)	(3,3)	(4,3)	(5,3)	(1,2)	(4,2)	(1,1)	(2,1)



K-Means Clustering

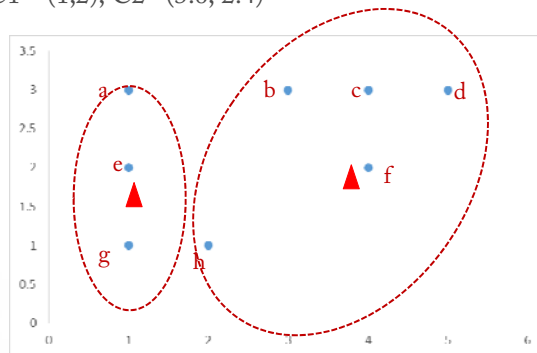
- Iteration
 - Step 1: $k=2$
 - Step 2: Randomly assign k records to be the initial cluster center locations
 - $m_1 = (1,1)$, $m_2 = (2,1)$
 - Step 3 (first pass): Find the nearest cluster center
 - $c1 \leftarrow \{a,e,g\}$, $c2 \leftarrow \{b,c,d,f,h\}$

Point	Distance from m1	Distance from m2	Cluster Membership
A	2.00	2.24	C1
B	2.83	2.24	C2
C	3.61	2.83	C2
D	4.47	3.61	C2
E	1.00	1.41	C1
F	3.16	2.24	C2
G	0.00	1.00	C1
h	1.00	0.00	C2



K-Means Clustering

- Step 4 (first pass):
 - for each of the k cluster, find the cluster centroid and update the location of each cluster center to the new value of the centroid
 - $C1 = (1,2)$, $C2 = (3.6, 2.4)$



K-Means Clustering

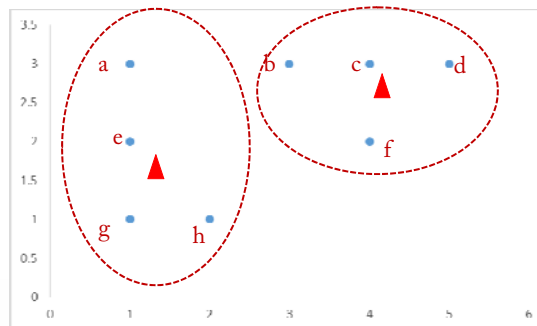
- Step 3 (Second pass)
 - for each record, find the nearest cluster center
 - $c1 \leftarrow \{a,e,g,h\}$, $c2 \leftarrow \{b,c,d,f\}$

Point	Distance from m1	Distance from m2	Cluster Membership
A	1.00	2.67	C1
B	2.24	0.85	C2
C	3.16	0.72	C2
D	4.12	1.52	C2
E	0.00	2.63	C1
F	3.00	0.57	C2
G	1.00	2.95	C1
h	1.41	2.13	C1



K-Means Clustering

- Step 4 (second pass)
 - find the cluster centroid
 - $c1 = (1.25, 1.75)$, $c2 = (4, 2.75)$



K-Means Clustering

- Step 3 (third pass);
 - find the nearest cluster

Point	Distance from m1	Distance from m2	Cluster Membership
A	1.27	3.01	C1
B	2.15	1.03	C2
C	3.02	0.25	C2
D	3.95	1.03	C2
E	0.35	3.09	C1
F	2.76	0.75	C2
G	0.79	3.47	C1
h	1.06	2.66	C1

- Since no records have shifted cluster membership, the cluster centroids also remain unchanged



K-Means Clustering

- Evaluation (First pass)
 - $SSB = \sum_{i=1}^k n_i d(m_i, M)^2 = 12.975$
 - $MSB = SSB/k-1 = 12.975$
 - $SSE = \sum_{i=1}^k \sum_{p \in C_i} d(p, m_i)^2 = 36$
 - $MSE = SSE/N-k = 6$
 - $F = MSB/MSE = 2.1625$
- Second Pass
 - $MSB = 17.125$, $MSE = 1.313333$, $F = 13.03934$
- Third pass
 - $MSB = 17.125$, $MSE = 1.041667$, $F = 16.44$



Problems with Selecting Initial Points

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
 - Chance is relatively small when K is large
 - If clusters are the same size, n, then
- Solutions
 - Multiple runs
 - Helps, but probability is not on your side
 - Sample and use hierarchical clustering to determine initial centroids
 - Select more than k initial centroids and then select among these initial centroids
 - Select most widely separated



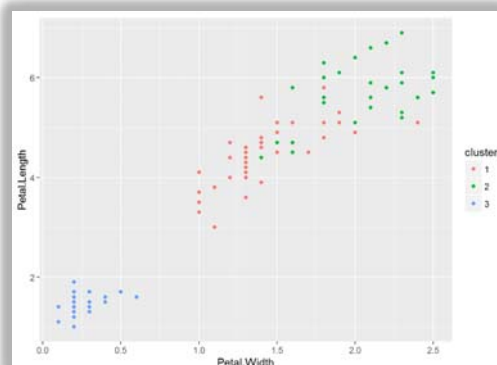
Example

- k-means

```
iris.kmeans <- kmeans(training.data[,1:4], centers = 3)
iris.kmeans$centers
```

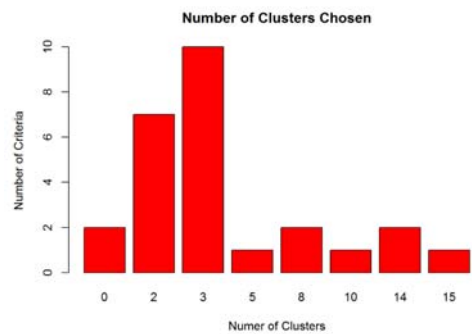
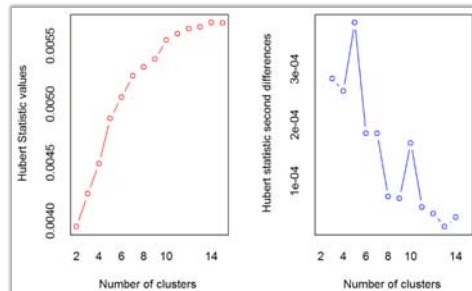
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
0.01167492	-0.8925851	0.3508791	0.2763463
1.14228028	0.2738255	1.0159839	1.0529085
-1.02474316	0.7520636	-1.2908510	-1.2405049

	1	2	3
setosa	0	0	35
versicolor	29	6	0
virginica	10	25	0



Example

- k ?
 - NbCluster library



Fuzzy K-means

- The clusters of k-means procedure
 - called "hard" or "crisp" clusters
 - any feature vector x either is or is not a member of a particular cluster.
- "fuzzy" K-means
 - this is in contrast to K-means
 - "soft" clusters
 - a feature vector x can have a degree of membership in each cluster



Fuzzy K-means

- The fuzzy-k-means procedure:
 - Dunn and Bezdek allows each feature vector x to have a degree of membership in Cluster i
 - Make initial guesses for the means m_1, m_2, \dots, m_k

- centroid
$$c_j = \frac{\sum_{i=1}^n p_{ij}^m x_i}{\sum_{i=1}^n p_{ij}^m}$$



Model Based Clustering

- Scott and Symons (1971)
- Yeung et al. (2001)
 - equal volume spherical model
 - unequal volume spherical model
 - unconstraint model
 - elliptical model
 - diagonal model

