COMP5310: Principles of Data Science

W8: Clustering and Dimensionality Reduction

Presented by

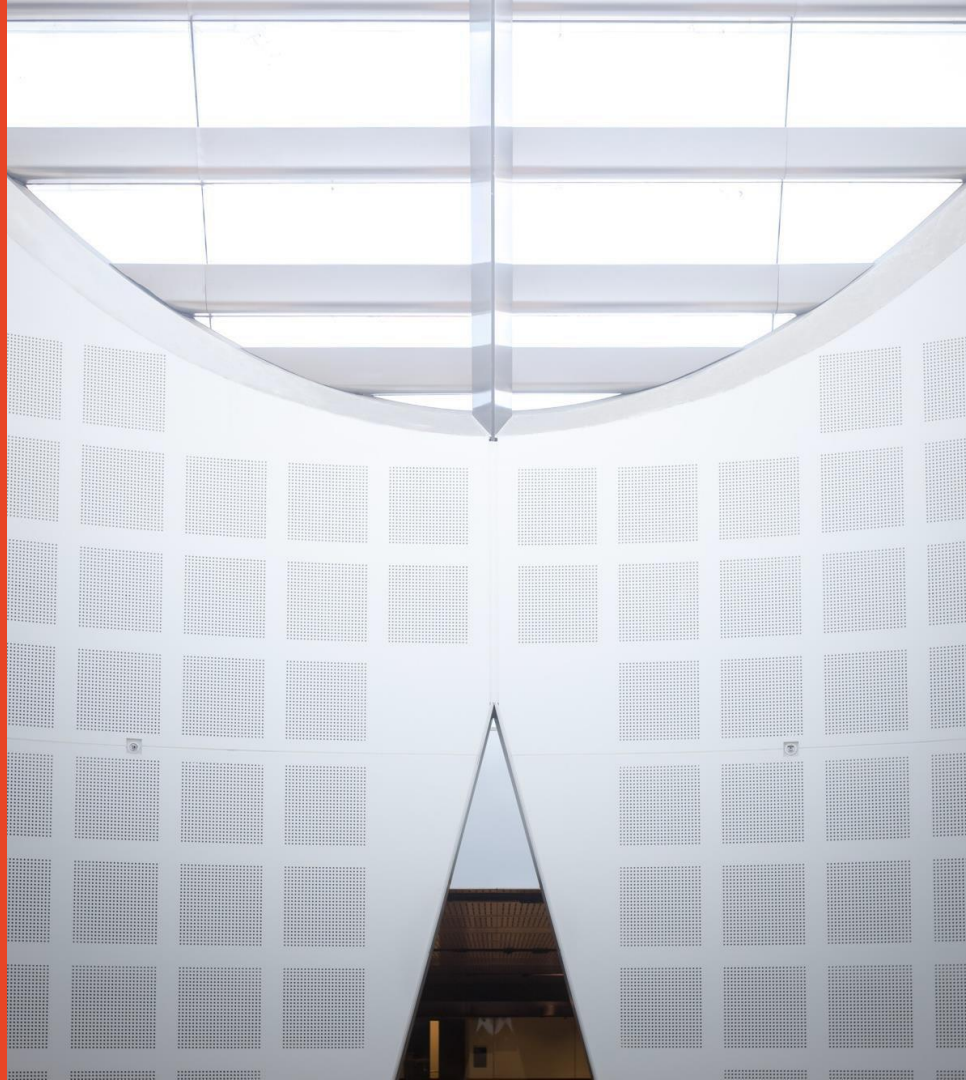
Maryam Khanian

School of Computer Science

Based on slides by previous lecturers of this unit of study

THE UNIVERSITY OF SYDNEY

The University of Sydney



Last week: Association Rule Mining

Objective

Learn techniques for unsupervised learning, with tools in Python.

Lecture

- Association rule mining

Readings

- Intro to Data Mining, Ch. 5
  https://www-users.cse.umn.edu/~kumar001/dmbook/ch5_association_analysis.pdf
- Data Science from Scratch, Ch. 11

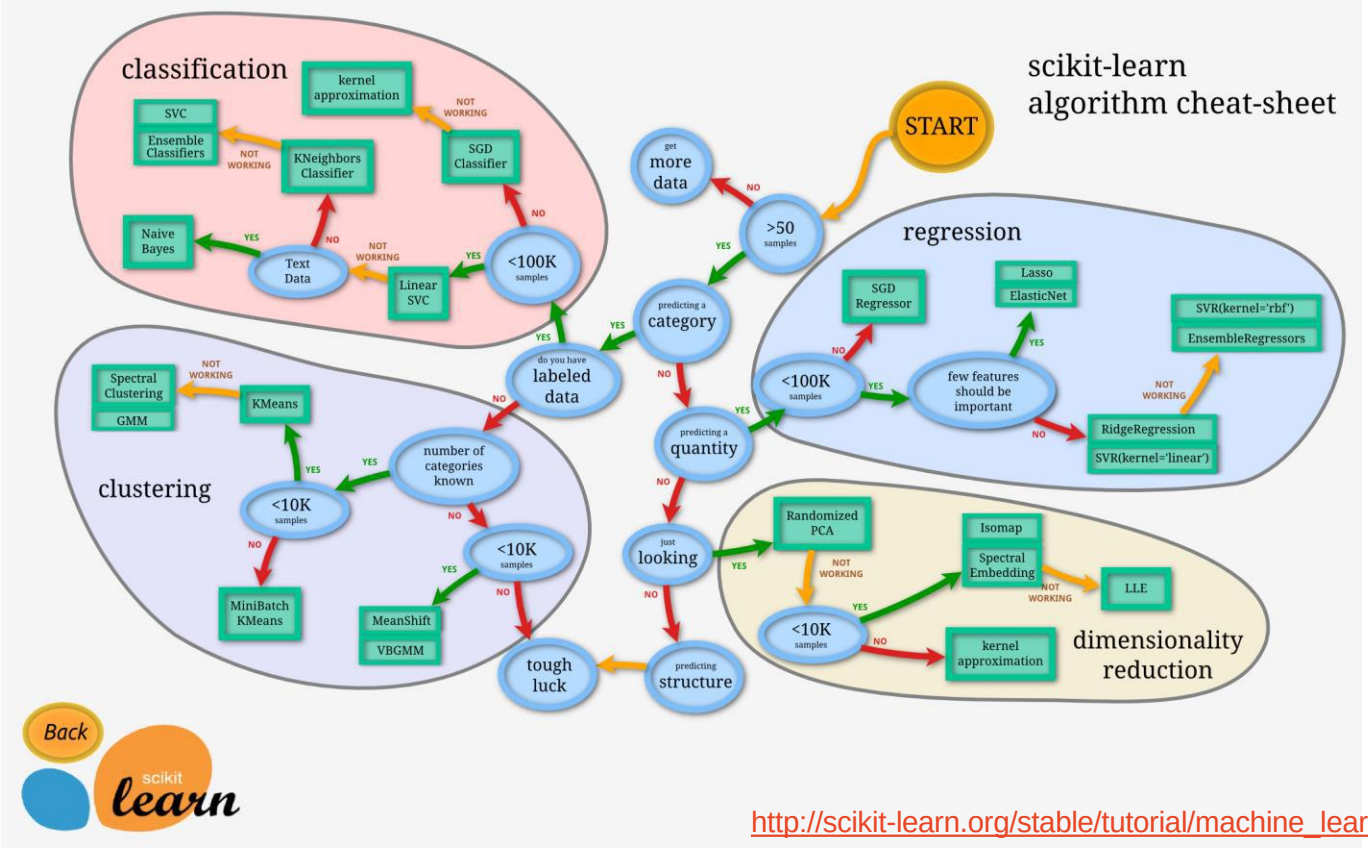
Exercises

- Associations from scratch
- Associations using mlxtend
- Associations using pyfpgrowth

Unsupervised Learning:

- We'll focus on unsupervised machine learning techniques
  - ☑ *Association rule mining*
  - **Clustering**
  - **Dimensionality reduction**
  - Outlier detection
  - Etc.

Machine Learning Map from Scikit-learn



http://scikit-learn.org/stable/tutorial/machine_learning_map/

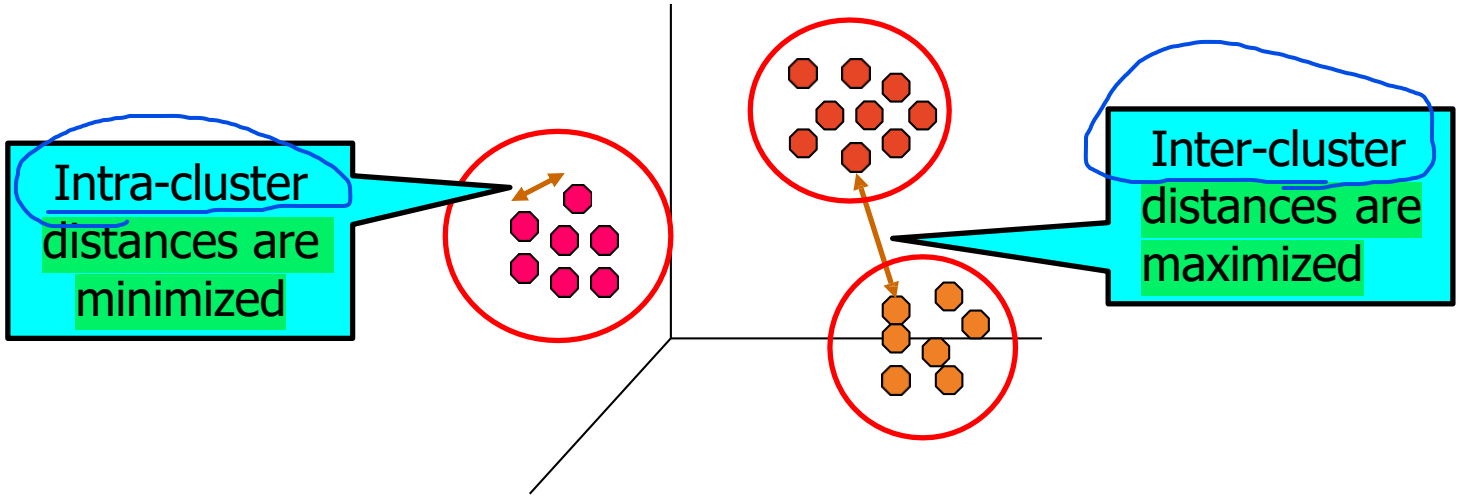
Clustering

Clustering for Understanding

- Group related documents for browsing
- Group genes, proteins, or cells that have similar functionality
- Group stocks with similar price fluctuations
- etc

Clustering: Group Similar Objects

- Group data points into clusters such that
  - Data points in one cluster are more similar to one another.
  - Data points in separate clusters are less similar to one another.
  - Distance function specifies the "closeness" of two objects.



Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i,j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q}$$

where $x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ and $x_j = (x_{j1}, x_{j2}, \cdots, x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

Similarity and Dissimilarity Between Objects (Cont.)

- *If* $q = 2$, d is Euclidean distance: L2 norm

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2}$$

- Properties
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
  - $d(i,j) \leq d(i,k) + d(k,j)$

Data Structures

- Data matrix

  n-observations with p-attributes (measurements).

- Dissimilarity matrix $d(i, j)$ is the dissimilarity between objects i and j
  - expresses the pairwise dissimilarities (distances) between observations in the data set
  - the desired data input to some clustering algorithm

attributes/dimensions

tuples/objects

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

objects

objects

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3, & 0 & \\ : & (& (& \\ ,2) & \dots \dots & & 0 \end{bmatrix}$$

Types of Clusterings

Important distinction between hierarchical and partitional sets of clusters

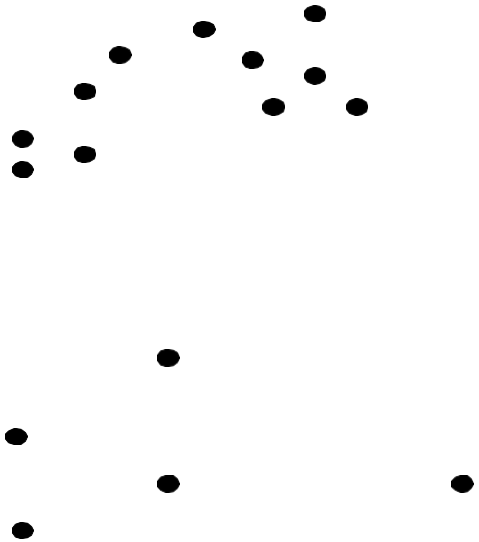
- ***Partitional clustering***

  A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

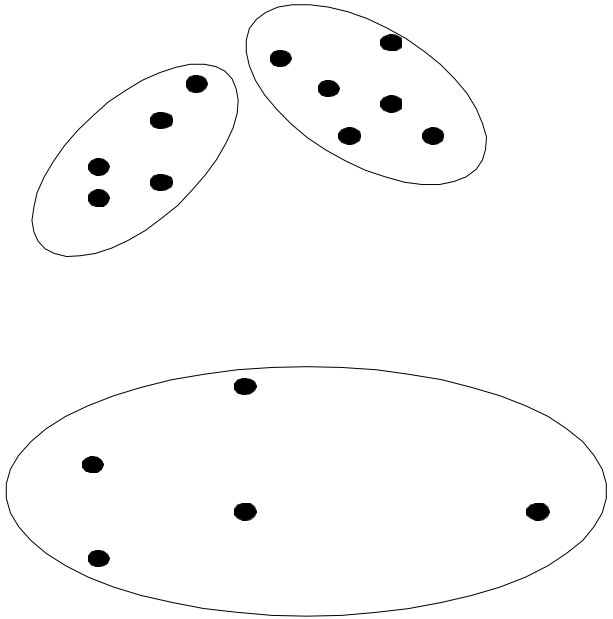
- ***Hierarchical clustering***

  A method of cluster analysis which seeks to build a hierarchy of clusters. It produces a set of nested clusters organized as a hierarchical tree

Partitional Clustering



Original Points



A Partitional Clustering

Partitional Clustering: K-Means Clustering

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

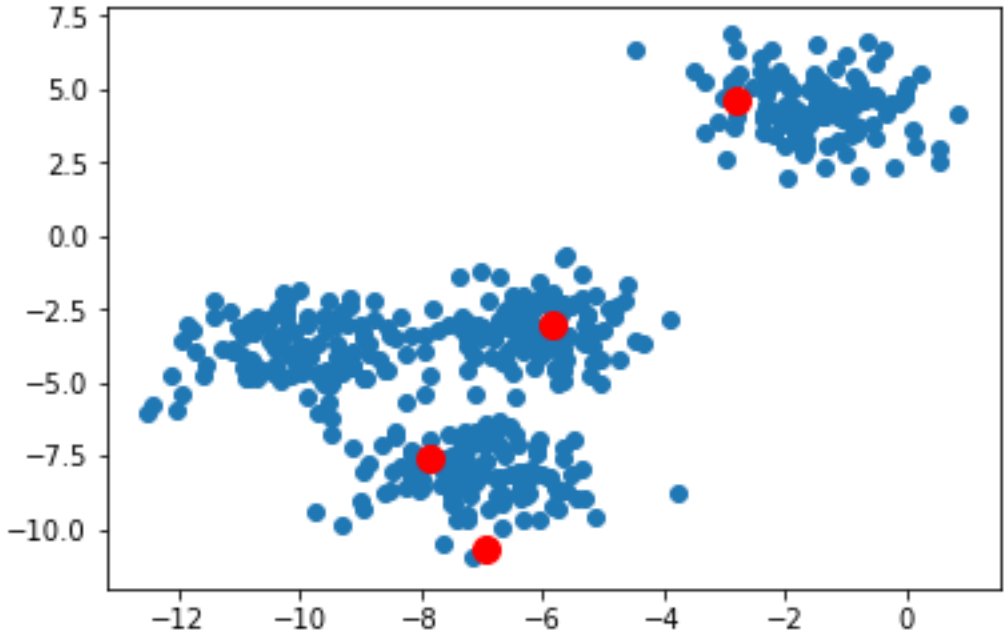
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

K-Means Clustering – Details

- Initial centroids are often chosen randomly
- Clusters produced vary from one run to another
- K-means will converge for common similarity measures
- Most of the convergence happens in the first few iterations
- Complexity is $O(n \times k \times i \times d)$
  - n = number of points, k = number of clusters, i = number of iterations, d = number of attributes (or dimensions)

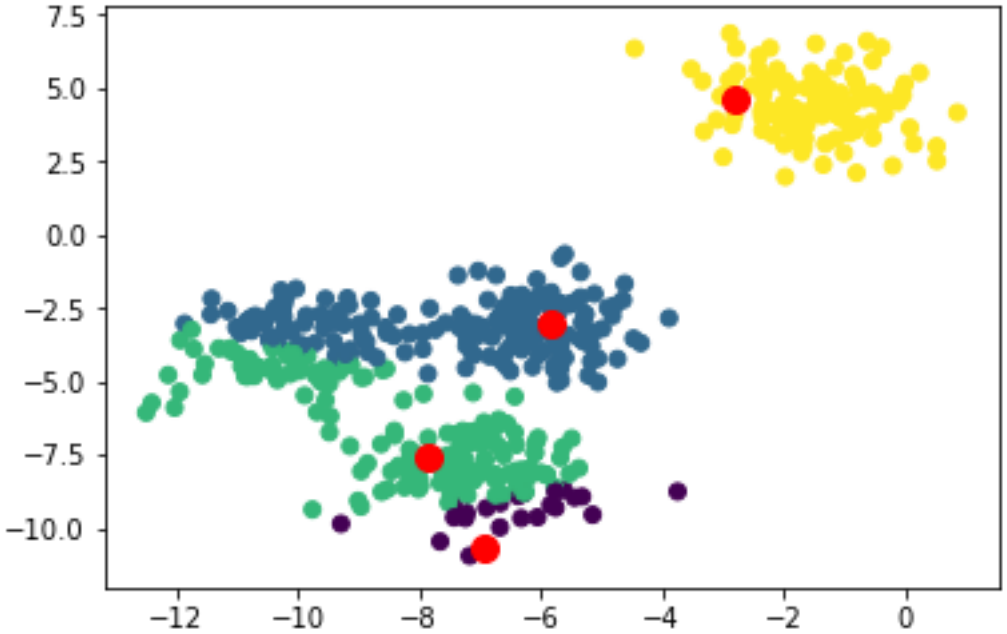
K-Means Example

- Number of clusters: K = 4
- Initialize 4 centroids randomly



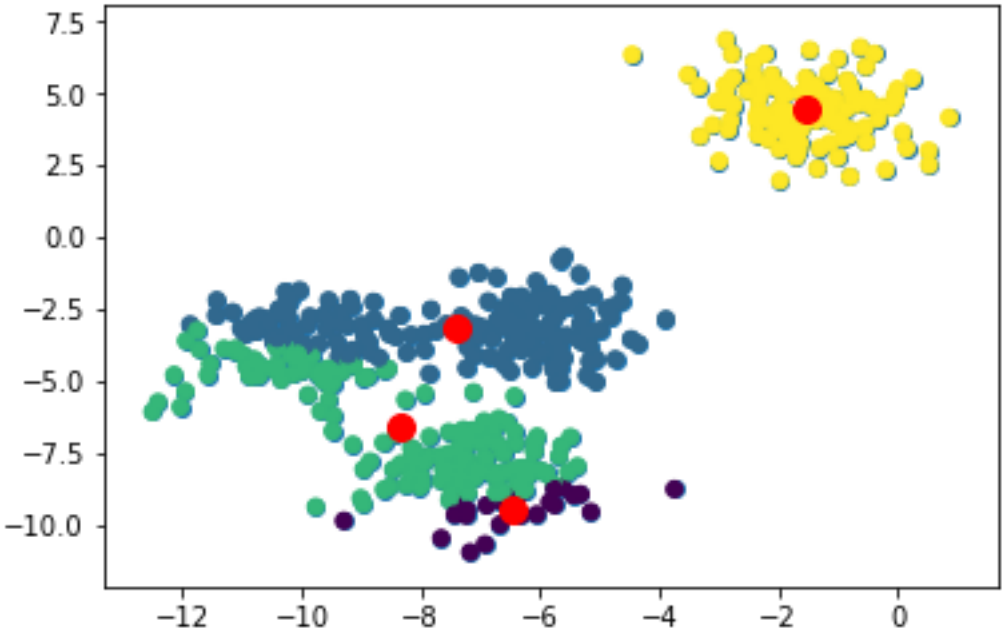
K-Means Example (Cont.)

– Assigned each point to the cluster with the closest centroid



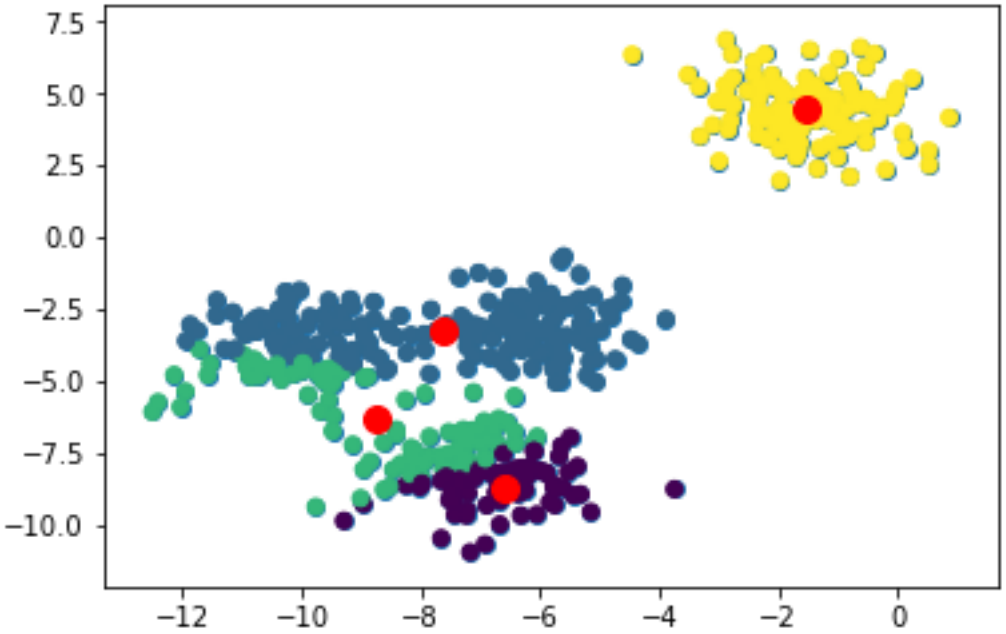
K-Means Example (Cont.)

– Re-compute each cluster center to be the mean of the points previously assigned.



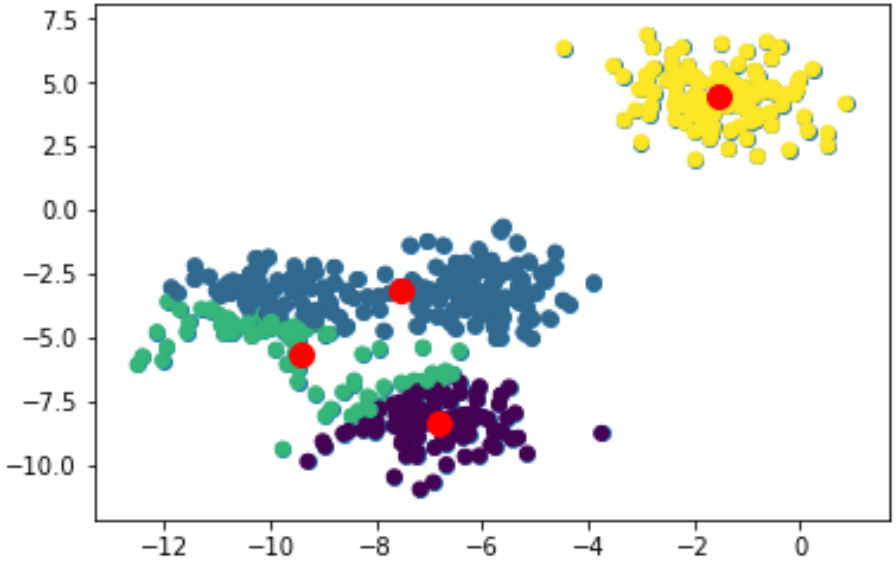
K-Means Example (Cont.)

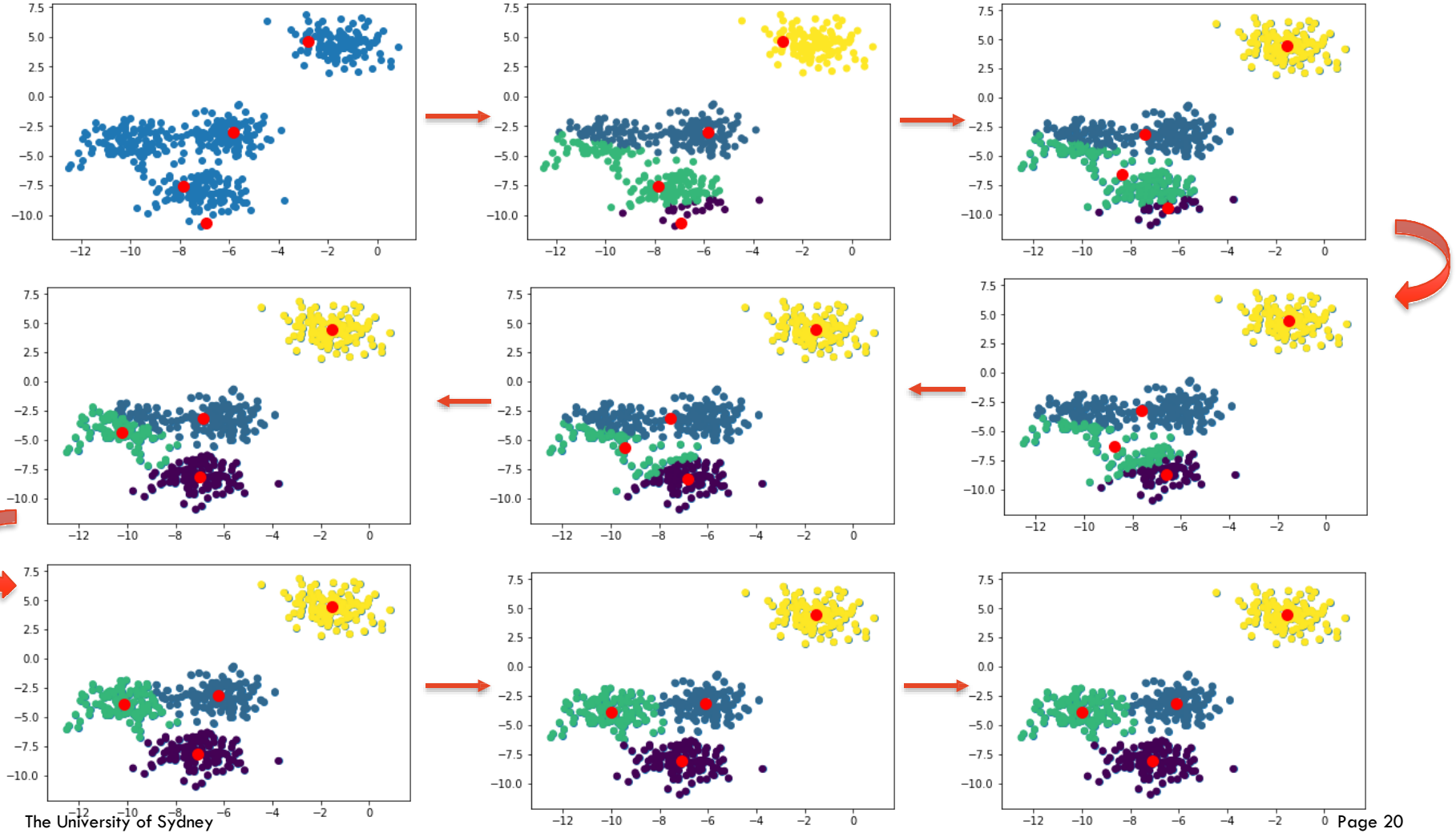
– Assigned each point to the cluster with the closest centroid



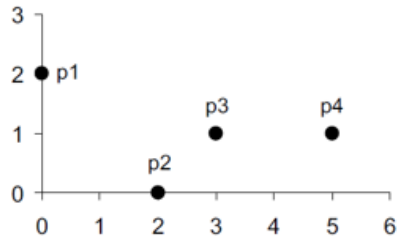
K-Means Example (Cont.)

- Re-compute each cluster center
- Centroid changed?
  - Assigned each point to the cluster with the closest centroid
  - Re-compute each cluster center





point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1



Pairwise Euclidean Distances:

1. p1-p2
   $\sqrt{(2-0)^2 + (0-2)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$

2. p1-p3
   $\sqrt{(3-0)^2 + (1-2)^2} = \sqrt{9+1} = \sqrt{10} \approx 3.16$

3. p1-p4
   $\sqrt{(5-0)^2 + (1-2)^2} = \sqrt{25+1} = \sqrt{26} \approx 5.10$

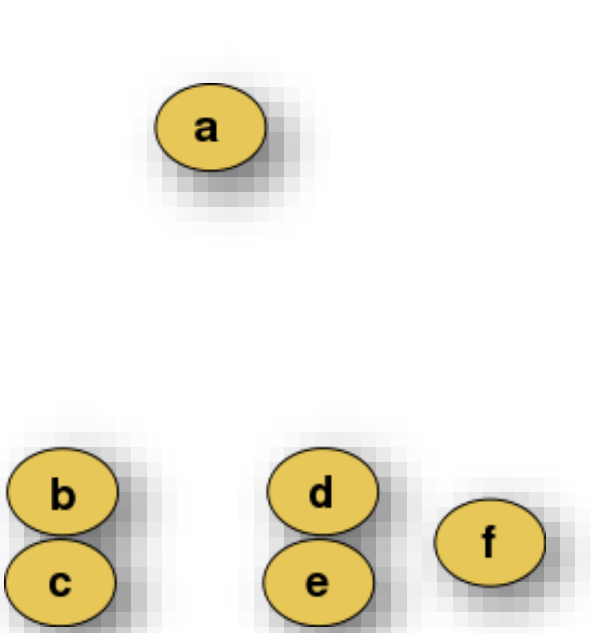
4. p2-p3
   $\sqrt{(3-2)^2 + (1-0)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$

5. p2-p4
   $\sqrt{(5-2)^2 + (1-0)^2} = \sqrt{9+1} = \sqrt{10} \approx 3.16$

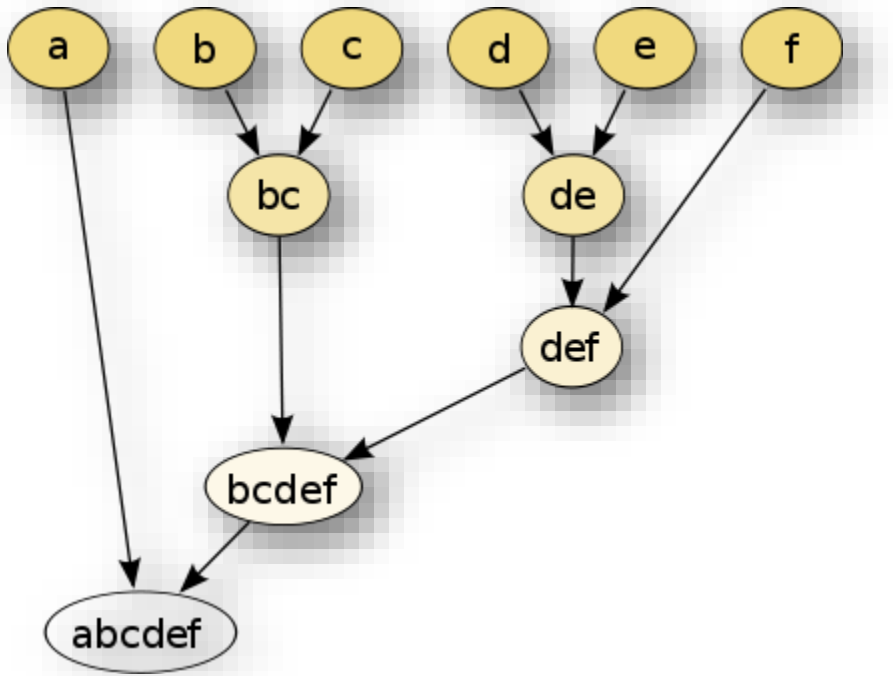
6. p3-p4
   $\sqrt{(5-3)^2 + (1-1)^2} = \sqrt{4+0} = \sqrt{4} = 2.00$

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Hierarchical Clustering



Original Data Items



Hierarchical Clustering

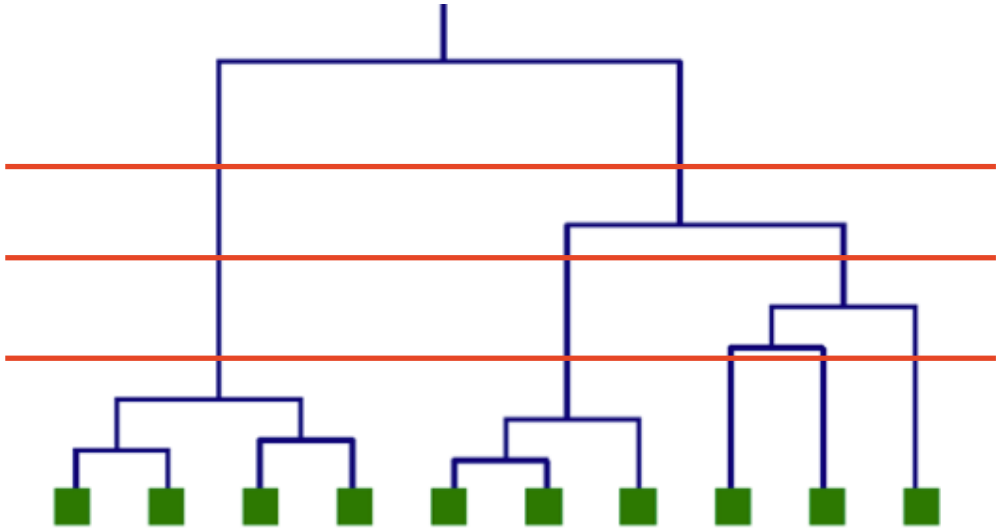
Hierarchical Clustering

Strategies for hierarchical clustering generally fall into two types:

- **Agglomerative**: This is a "bottom up" approach: each object starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- **Divisive**: This is a "top down" approach: all objects start in the same cluster, and splits are performed recursively as one moves down the hierarchy.

Hierarchical Clustering: e.g. Agglomerative

- Initial
  - Each point in its own cluster
- Repeat
  - Find closest pair of clusters
    - Min-distance between any two points
  - Merge them into one cluster
  - Recompute distances between new cluster and others
- Until
  - Desired number of clusters remaining e.g. single cluster

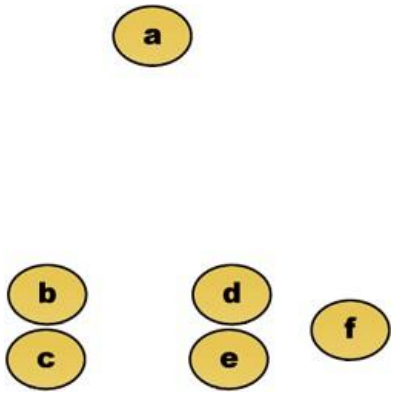


dendrogram

Hierarchical Algorithm

Steps in Hierarchical Algorithm:

- The first step generates the distance calculation matrix for each data item as shown in table below, in this case: {a}, {b}, {c}, {d}, {e}, {f}.



	a	b	c	d	e	f
a	0	184	222	177	216	231
b	184	0	45	123	128	200
c	222	45	0	129	121	203
d	177	123	129	0	46	83
e	216	128	121	46	0	83
f	231	200	203	83	83	0

Hierarchical Algorithm

- Next step is to merge the closest data items.
  - In this case: {b , c} are merged.
  - Therefore, the first clustering process generates: {a}, {b, c}, {d},{e},{f}.

	a	b	c	d	e	f
a	0	184	222	177	216	231
b	184	0	45	123	128	200
c	222	45	0	129	121	203
d	177	123	129	0	46	83
e	216	128	121	46	0	83
f	231	200	203	83	83	0

	a	b,c	d	e	f
a	0	?	177	216	231
b,c	?	0	?	?	?
d	177	?	0	46	83
e	216	?	46	0	83
f	231	?	83	83	0

Hierarchical Algorithm

Distance Calculation between two clusters:

- single linkage:
  - The minimum distance between elements of the two clusters
- complete linkage:
  - The maximum distance between elements of the two clusters
- average linkage: i.e. mean distance calculation.

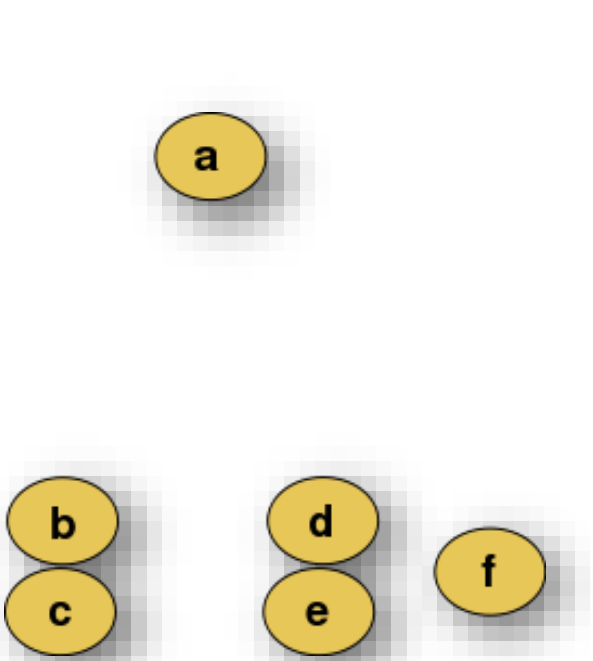
Hierarchical Algorithm with Single Linkage

- Repeat the distance calculation process based on single linkage
- Apply merging process based on previous merged results.
  - In this case: {d, e} are merged.
- The final results are: {a}, {b, c} {d, e} {f} → {a}, {b, c}, {d, e, f} → {a}, {b, c, d, e, f} → {a, b, c, d, e, f}

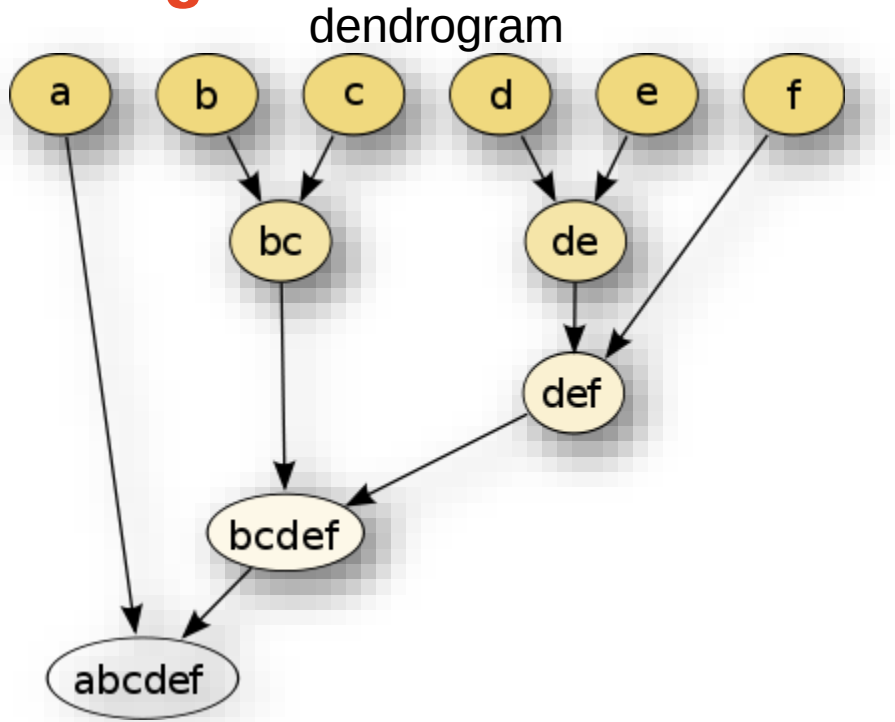
	a	b	c	d	e	f
a	0	184	222	177	216	231
b	184	0	45	123	128	200
c	222	45	0	129	121	203
d	177	123	129	0	46	83
e	216	128	121	46	0	83
f	231	200	203	83	83	0

	a	b, c	d	e	f
a	0	184	177	216	231
b, c	184	0	123	121	200
d	177	123	0	46	83
e	216	121	46	0	83
f	231	200	83	83	0

Resultant Hierarchical Clustering



Original Data Items



Hierarchical Clustering

Evaluating Clustering

Cluster Validity

- For supervised classification, we have a variety of measures to evaluate how good our model is (accuracy, precision, recall) –
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters.
- While clusters are in the eye of the beholder, we may still want to evaluate them…
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
- https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

Measures of Cluster Validity

Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.

- External Index: Measure the extent to which cluster labels match externally supplied class labels (e.g., accuracy, precision, recall, F1-score)
- Internal Index: Measure the goodness of a clustering structure *without* respect to external information (e.g., Sum of Squared Error)
- Relative Index: Compare two different clusterings or clusters (often an external or internal index is used)

External Evaluation Measures

- Homogeneity ranges from 0 to 1, preferring each cluster contains only members of a single class.
  (analogous to precision, $P = TP / (TP+FP)$)
- Completeness ranges from 0 to 1, preferring all members of a given class are assigned to the same cluster
  (analogous to recall, $R = TP / (TP+FN)$)
- V-measure is the harmonic mean of homogeneity and completeness
  (analogous to F1 score = 2PR / (P+R))
- http://scikit-learn.org/stable/modules/clustering.html#homogeneity-completeness-and-v-measure

Internal: Sum of Squared Error (SSE)

- For each point, the error is the distance to the nearest cluster
- To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(x, m_i)$$

- x is a data point in cluster C_i and m_i is the centroid point (mean) for cluster C_i

SSE Example

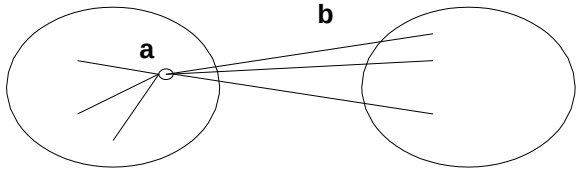
- Suppose we have 3 clusters:
  - Cluster 1: [2, 4] with centroid at 3
  - Cluster 2: [5, 6, 7] with centroid at 6
  - Cluster 3: [8, 10, 12] with centroid at 10
- Squared error for each cluster:
  - $SE1 = (2-3)^2 + (4-3)^2 = 1 + 1 = 2$
  - $SE2 = (5-6)^2 + (7-6)^2 = 1 + 1 = 2$
  - $SE3 = (8-10)^2 + (12-10)^2 = 4 + 4 = 8$
- $SSE = SE1 + SE2 + SE3 = 12$

Internal: Silhouette Coefficient

- For an individual point *x*
  - Calculate ***a*** = average distance of *x* to points in its cluster
  - Calculate ***b*** = average distance of *x* to points in the next nearest cluster
  - The silhouette coefficient for a point is then given by

    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)

  - The closer to 1 the better



- Silhouette coefficient for dataset is average across all *data points*

Silhouette Coefficient Example

- Suppose we have 3 clusters:
  - Cluster 1 = [[1,0], [1,1]]
  - Cluster 2 = [[1,2], [2,3], [2,2], [1,2]],
  - Cluster 3 = [[3,1], [3,3], [2,1]]
- Take a point [1,0] in cluster 1
- Calculate its average distance to all other points in its cluster, i.e. cluster 1
- So a1 = $\sqrt{(1-1)^2 + (0-1)^2} = \sqrt{0+1} = 1$

Silhouette Coefficient Example (Cont.)

- Now for the point [1,0] in cluster 1 calculate its average distance from all the points in cluster 2 and cluster 3.
- Of these take the minimum average distance.
- So for cluster 2:
  - [1,0] → [1,2], distance = $\sqrt{(1-1)^2 + (0-2)^2} = \sqrt{0+4} = 2$
  - [1,0] → [2,3], distance = $\sqrt{(1-2)^2 + (0-3)^2} = \sqrt{1+9} = 3.16$
  - [1,0] → [2,2], distance = $\sqrt{(1-2)^2 + (0-2)^2} = \sqrt{1+4} = 2.24$
  - [1,0] → [1,2], distance = $\sqrt{(1-1)^2 + (0-2)^2} = \sqrt{0+4} = 2$
- Therefore, the average distance of point [1,0] in cluster 1 to all the points in cluster 2 = $(2+3.16+2.24+2)/4 = 2.35$

Silhouette Coefficient Example (Cont.)

- Similarly, for cluster 3.
  - [1,0] → [3,1], distance = $\sqrt{(1-3)^2 + (0-1)^2} = \sqrt{4+1} = 2.24$
  - [1,0] → [3,3], distance = $\sqrt{(1-3)^2 + (0-3)^2} = \sqrt{4+9} = 3.61$
  - [1,0] → [2,1], distance = $\sqrt{(1-2)^2 + (0-1)^2} = \sqrt{1+1} = 1.41$
- Therefore, the average distance of point [1,0] in cluster 1 to all the points in cluster 3 = $(2.24+3.61+1.41)/3 = 2.42$
- Now, the minimum average distance of the point [1,0] in cluster 1 to the other clusters 2 and 3 is,

$b1 = 2.35$ $(2.35 < 2.42)$

Silhouette Coefficient Example (Cont.)

- So the silhouette coefficient of point [1,0] in cluster 1

$$s1 = 1-(a1/b1) = 1- (1/2.35) = 1-0.43 = 0.57$$

- In a similar fashion you need to calculate the silhouette coefficient for each data point in each cluster
- Then we average them to calculate the overall silhouette coefficient to evaluate the resultant clusters
- The closer to 1 the better

Choosing *k*

Notion of a Cluster can be Ambiguous

How many clusters..?

Maybe six clusters

Two clusters?

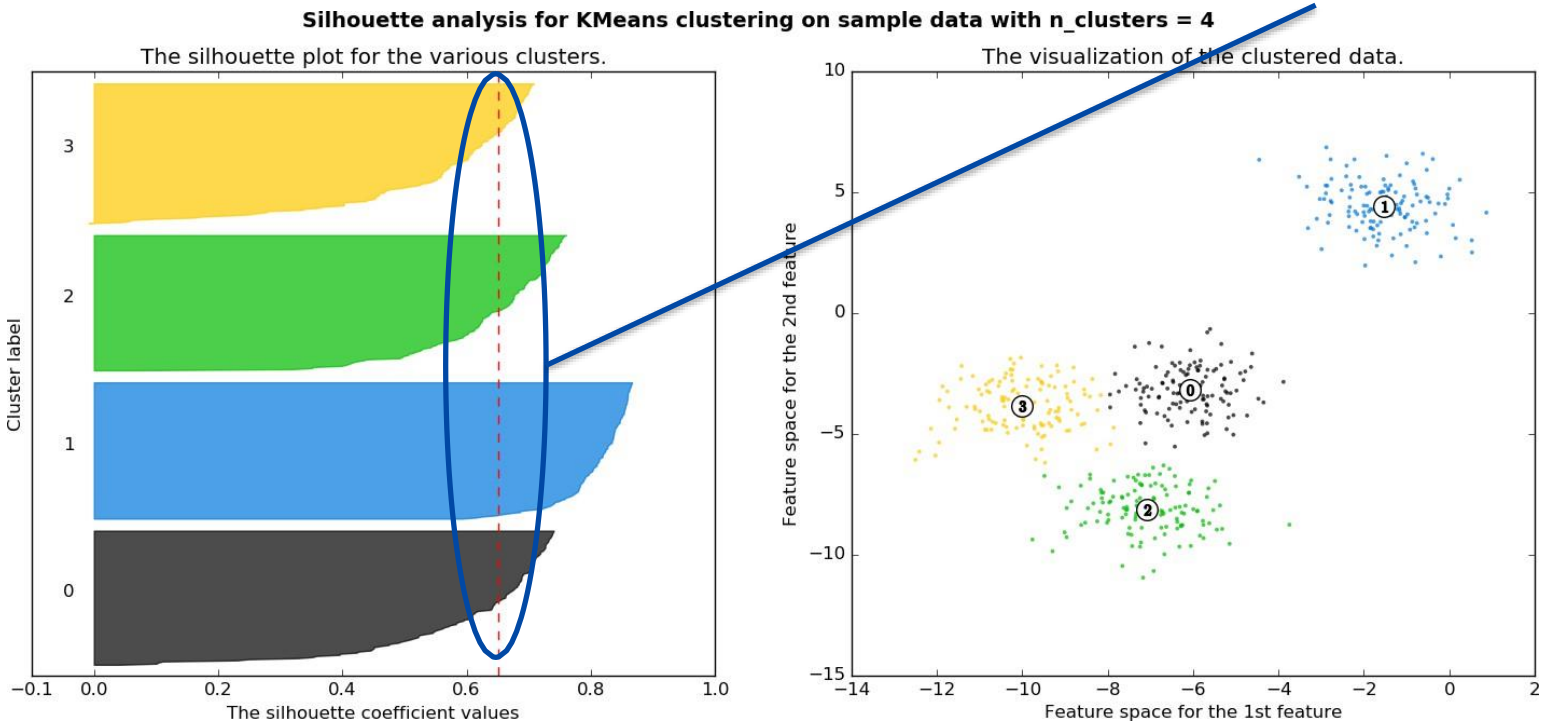
Or four clusters

Choosing *k*

- Often we don't know the number of clusters in our data
- Selecting *k* is generally an interactive process
- There are some approaches to aid interactive selection, and sometimes automate it

Using Silhouettes to choose *k*

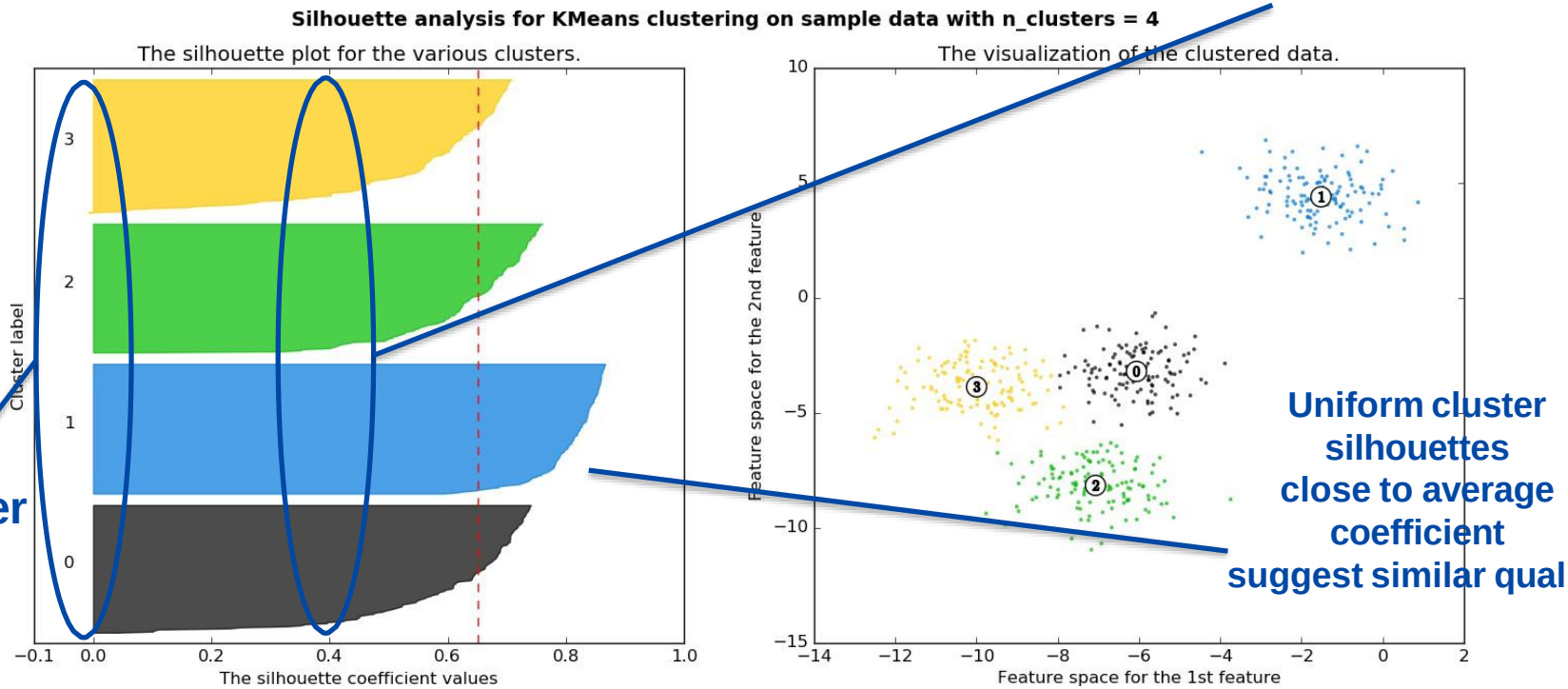
High average silhouette indicates points far away from neighbouring clusters



http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Using Silhouettes to choose *k*

Bar chart showing silhouette values for each item grouped by cluster



Cluster IDs

Uniform cluster silhouettes close to average coefficient suggest similar quality

http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Pre-Processing for Clustering

We only looked at choosing *k*, but more pre-processing needed

- Data Cleansing
- Data Transformation
- **Data Normalisation**
- **Dimensionality Reduction** / choice or projection of dimensions
  - closely related to choice of **distance metric**
  - we used Euclidean Metric so far (L_2-Norm)
  - but others possible too, eg. Manhatten Distance (L_1-Norm)
  - "Curse of high dimensionality"; cf Aggarwal et al.: "On the Surprising Behavior of Distance Metrics in High Dimensional Space", 2001.

Principal Component Analysis

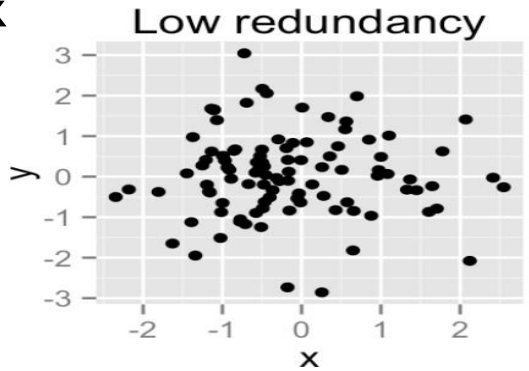
Principal Components Analysis

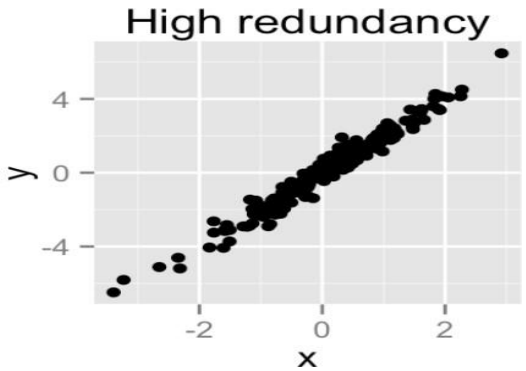
- Aims to transform the original data from high dimensional space into lower dimensional space.
- The new variables in the lower dimensional space corresponds to a linear combination of the originals and are called principal components (PC)
- PCA helps in
  - **Visualization**. Using the right variables to plot items will give more insights.
  - **Uncovering Clusters**. With good visualizations, hidden categories or clusters could be identified.
  - **Dimensionality reduction.** Reduce number of dimensions in data

Principal Components Analysis

- PCA method is particularly useful when the variables within the data set are highly correlated.
  - Correlation indicates that there is redundancy in the data.
  - Correlation is captured by the covariance matrix[1].
- PCA is traditionally performed on covariance matrix or correlation matrix

[1]covariance matrix is a square **matrix** that contains the variances and **covariances** associated with several variables.





Covariance Matrix

- Representing Covariance between dimensions as a matrix e.g for three attributes (x,y,z):

$$C = \begin{pmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \mathrm{cov}(\\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \mathrm{cov}(\\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \mathrm{cov}(\end{pmatrix}$$

Variances

- The covariance between one dimension and itself is the variance
  - Diagonal is the variances of x, y and z
- cov(x,y) = cov(y,x) hence matrix is symmetrical about the diagonal
- N-dimensional data will result in NxN covariance matrix

Covariance Matrix Example

- Below is the covariance matrix of some 3 variables.
- Their variances are on the diagonal, and the sum of the 3 values (3.448) is the overall variability

1.343730	-.1601522	.1864702
-.1601522	.61920562	-.1266842
.1864702	-.1266842	1.485549

- In the covariance table above, the off-diagonal values are different from zero. This indicates the presence of redundancy in the data.
- In other words, there is a certain amount of correlation between variables.

PCA Example

- PCA creates uncorrelated PC variables (called eigenvectors) having zero covariations and variances (called eigenvalues) sorted in decreasing order.
- The first PC captures the greatest variance, the second greatest variance is the second PC, and so on.
- By eliminating the later PCs we can achieve dimensionality reduction.
  - The 1st PC accounts for or "explains" 1.651/3.448 = 47.9% of the overall variability;
  - the 2nd one explains 35.4% of it; the 3rd one explains 16.7% of it.

1.65135	.000000	.000000
.000000	1.220288	.000000
.000000	.0000000	.576843

The covariance matrix between the principal components

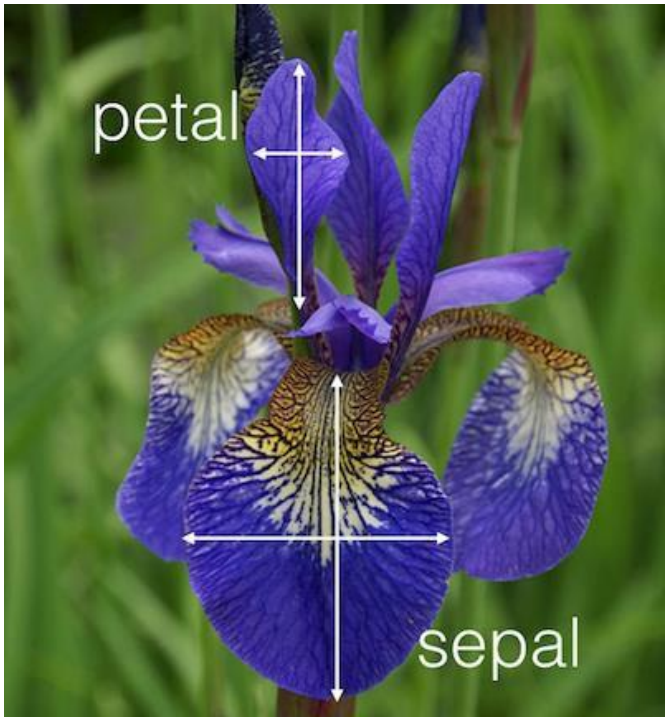
PCA on Iris Dataset

- Iris data has 150 observations equally distributed among three species:

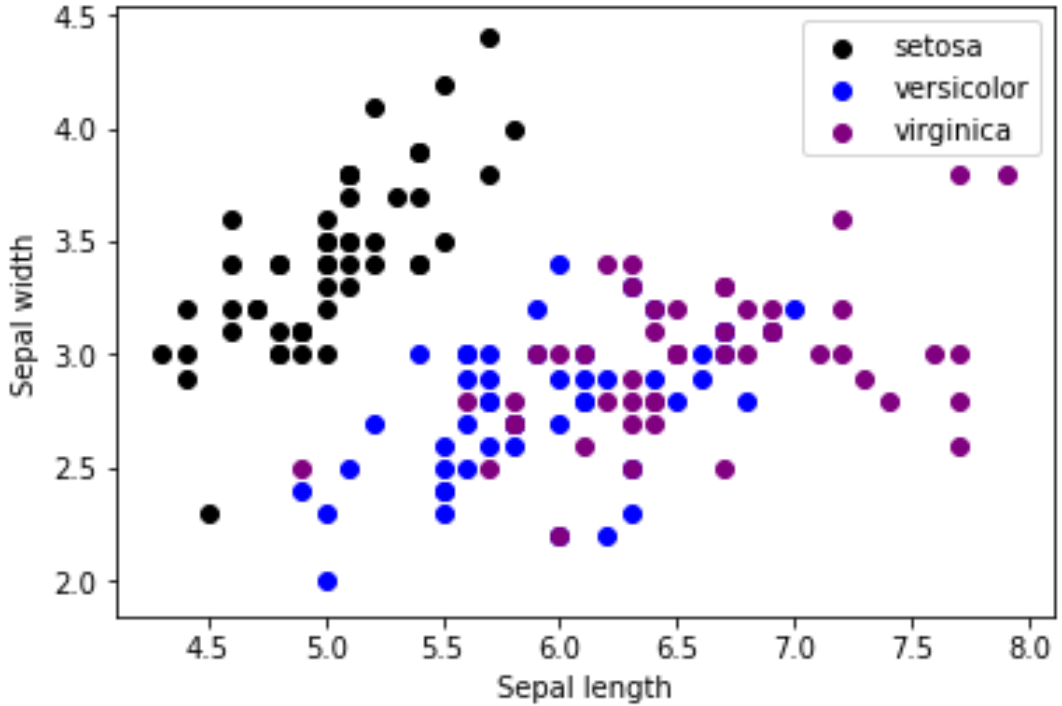
  Setosa, Versicolor and Verginica.

- It has four variables:
  - Sepal length and width
  - Petal length and width
- Which variables can I use to plot the data in two dimensional space?
- Let's try using the two features:

  Sepal length VS. Sepal width



Plotting the data points using Sepal Length vs Sepal Width

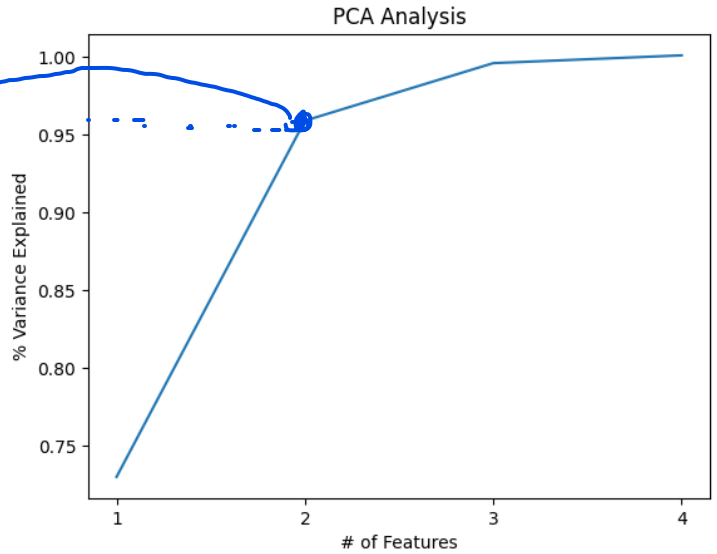


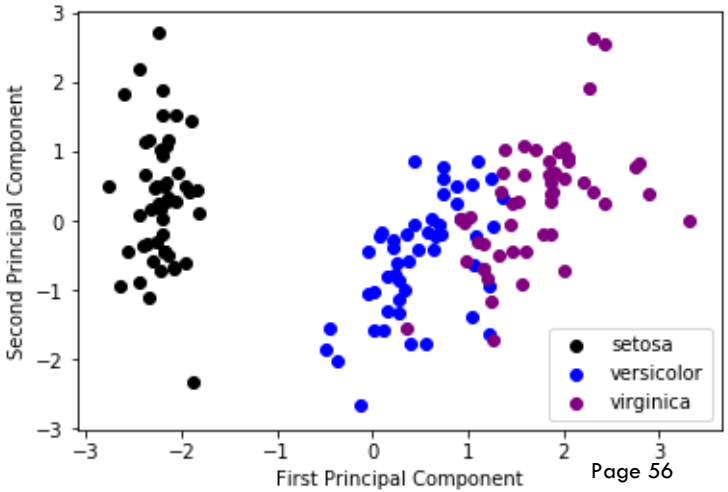
PCA on IRIS Dataset

- Let's now choose the best variables using PCA and then plot the data
- The eigenvalues are:

[0.728 0.230 0.037 0.005]

- The first two PCs represent 95.8% of the variance of the data
- Which means we can reduce the data into two dimensional spaces by eliminating PC3 and PC4





Review

W8 review: Clustering and Dimensionality Reduction

Objective

Learn techniques for unsupervised learning, with tools in Python.

Lecture

- Clustering algorithms
- Evaluating clustering
- Choosing k
- Principal Component Analysis

Readings

- Intro to Data Mining, Ch. 7 https://www-users.cse.umn.edu/~kumar001/dmbook/ch7_clustering.pdf
- Data Science from Scratch, Ch. 20

Exercises

- sklearn: clustering and PCA