COMP5310: Principles of Data Science

W8: Clustering and Dimensionality Reduction

Presented by Ali Anaissi

School of Computer Science

THE UNIVERSITY OF SYDNEY

Overview of Week 8

THE UNIVERSITY OF
SYDNEY

Today: Clustering and Dimensionality Reduction

Objective

Learn techniques for unsupervised learning, with tools in Python.

Lecture

– Evaluating clustering
– Principal Component Analysis
– Eigenvalues and Eigenvectors

Readings

– Intro to Data Mining, Ch. 6
http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf
– Intro to Data Mining, Ch. 8
http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf
– Data Science from Scratch, Ch. 11&19

Exercises

– sklearn: clustering and PCA

Unsupervised Learning:

- More unsupervised machine learning techniques
  - ☑ Association *rule mining*
  - **Dimensionality reduction**
  - ***Clustering***
  - Outlier detection
  - Etc.

Clustering

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*:

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

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

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

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

Similarity and Dissimilarity Between Objects (Cont.)

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

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

- 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

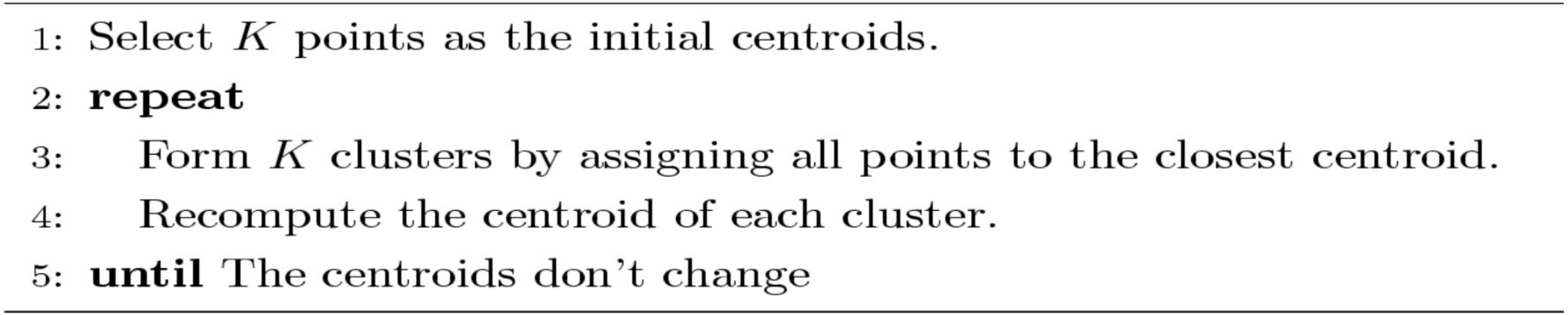
  n-observations with p-attributes (measurements).

- 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

$$\begin{array}{cc} & \text{attributes/dimensions} \\ \text{tuples/objects} & \begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \end{array}$$

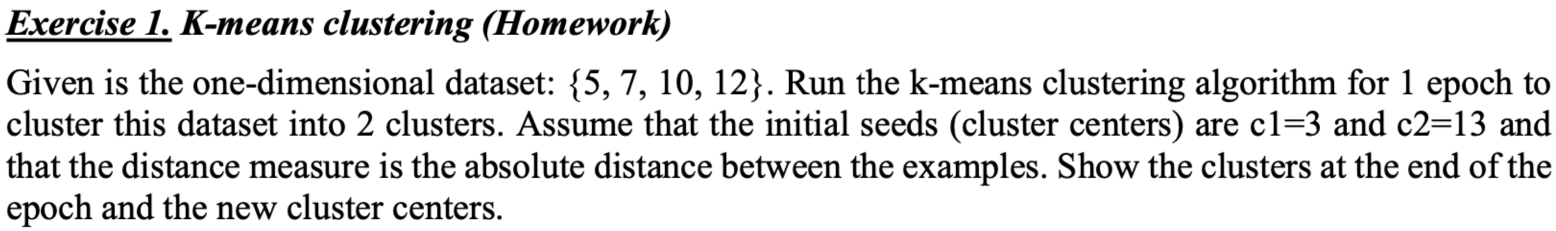
$$\begin{array}{cc} & \text{objects} \\ \text{objects} & \begin{array}{ccccc} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{array} \end{array}$$

Last week: K-Means



```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

Example

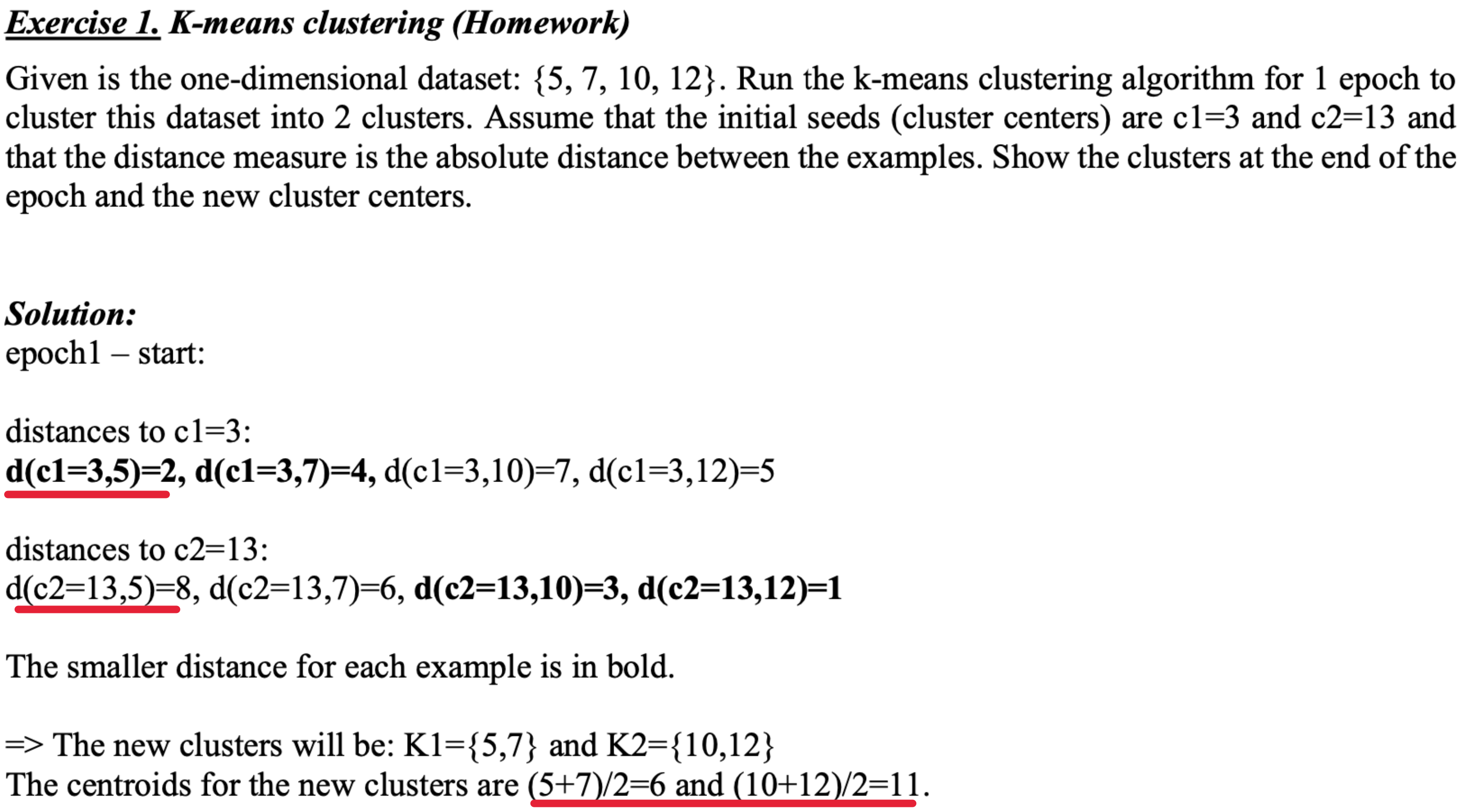


Exercise 1. ***K-means clustering (Homework)***

Given is the one-dimensional dataset: {5, 7, 10, 12}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters. Assume that the initial seeds (cluster centers) are c1=3 and c2=13 and that the distance measure is the absolute distance between the examples. Show the clusters at the end of the epoch and the new cluster centers.

Credit: Irena Koprinska

Example



Exercise 1. ***K-means clustering (Homework)***

Given is the one-dimensional dataset: {5, 7, 10, 12}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters. Assume that the initial seeds (cluster centers) are c1=3 and c2=13 and that the distance measure is the absolute distance between the examples. Show the clusters at the end of the epoch and the new cluster centers.

Solution:
epoch1 – start:

distances to c1=3:
d(c1=3,5)=2, d(c1=3,7)=4, d(c1=3,10)=7, d(c1=3,12)=5

distances to c2=13:
d(c2=13,5)=8, d(c2=13,7)=6, **d(c2=13,10)=3, d(c2=13,12)=1**

The smaller distance for each example is in bold.

=> The new clusters will be: K1={5,7} and K2={10,12}
The centroids for the new clusters are (5+7)/2=6 and (10+12)/2=11.

Credit: Irena Koprinska

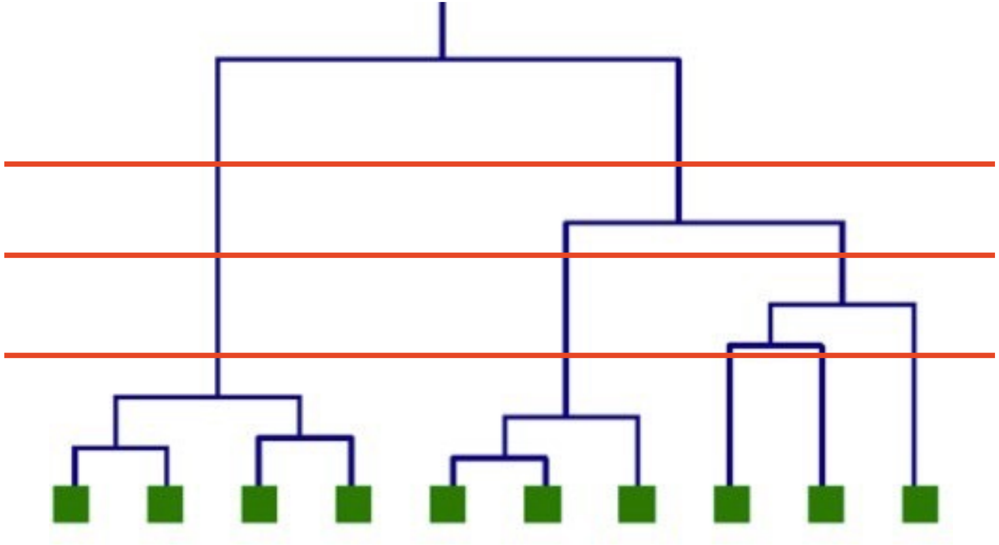
Hierarchical Clustering

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 one 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

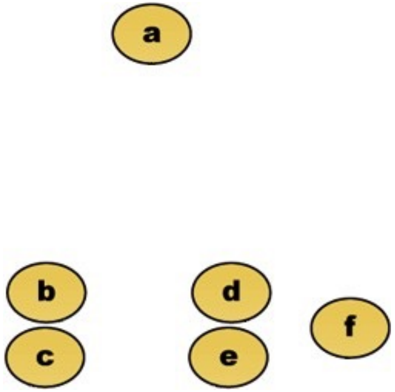


dendrogram

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 hierarchical clusters :

- single linkage:
  - The minimum distance between elements of each cluster
- complete linkage:
  - The maximum distance between elements of each cluster
- average linkage: i.e. mean distance calculation.

Hierarchical Algorithm with Single Linkage

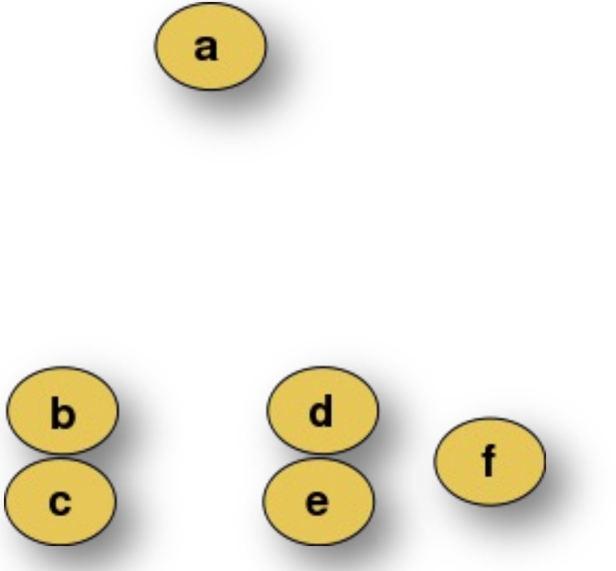
- Repeat the distance calculation process based on single linkage
- Apply merging process based on previous merge results.
  - In this case: {d , e} are merged.
- The final results are: {a}, {b, c} {d, e} → {a}, {b, c}, {d, e, f} → {a}, {b, c, d, e, f} → {a, b, c, d, e, f}

	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

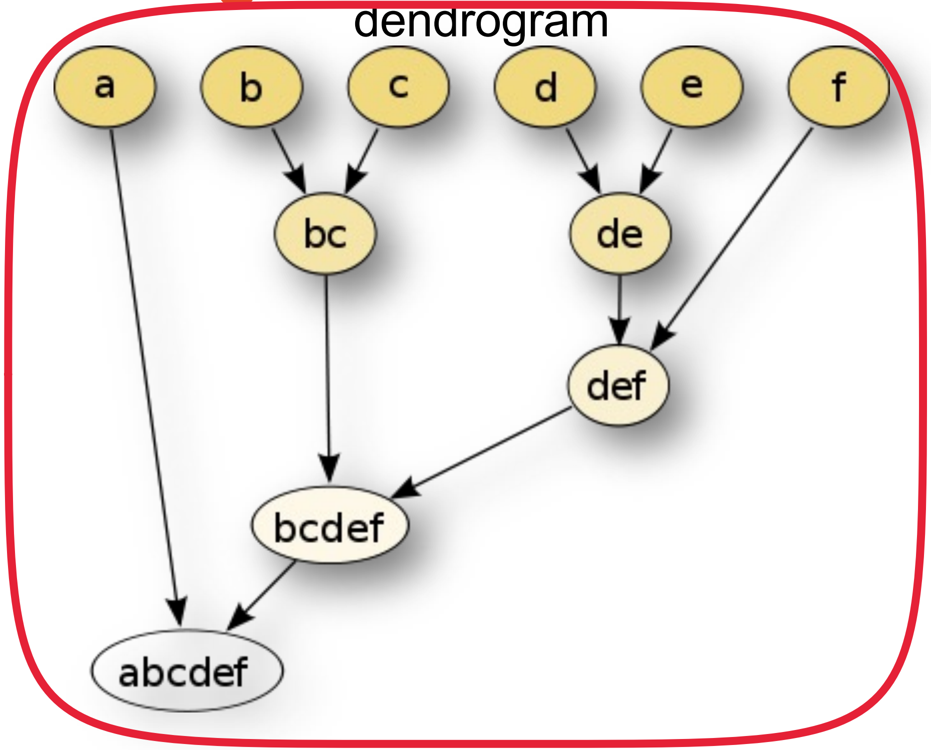
→

	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

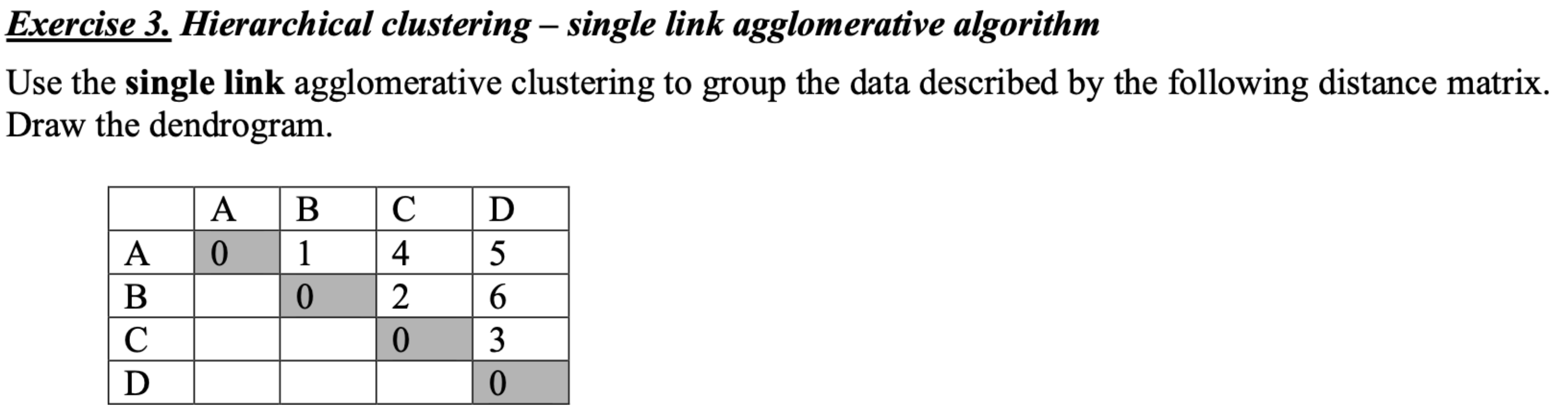


Original Data Items



Hierarchical Data Items

Example



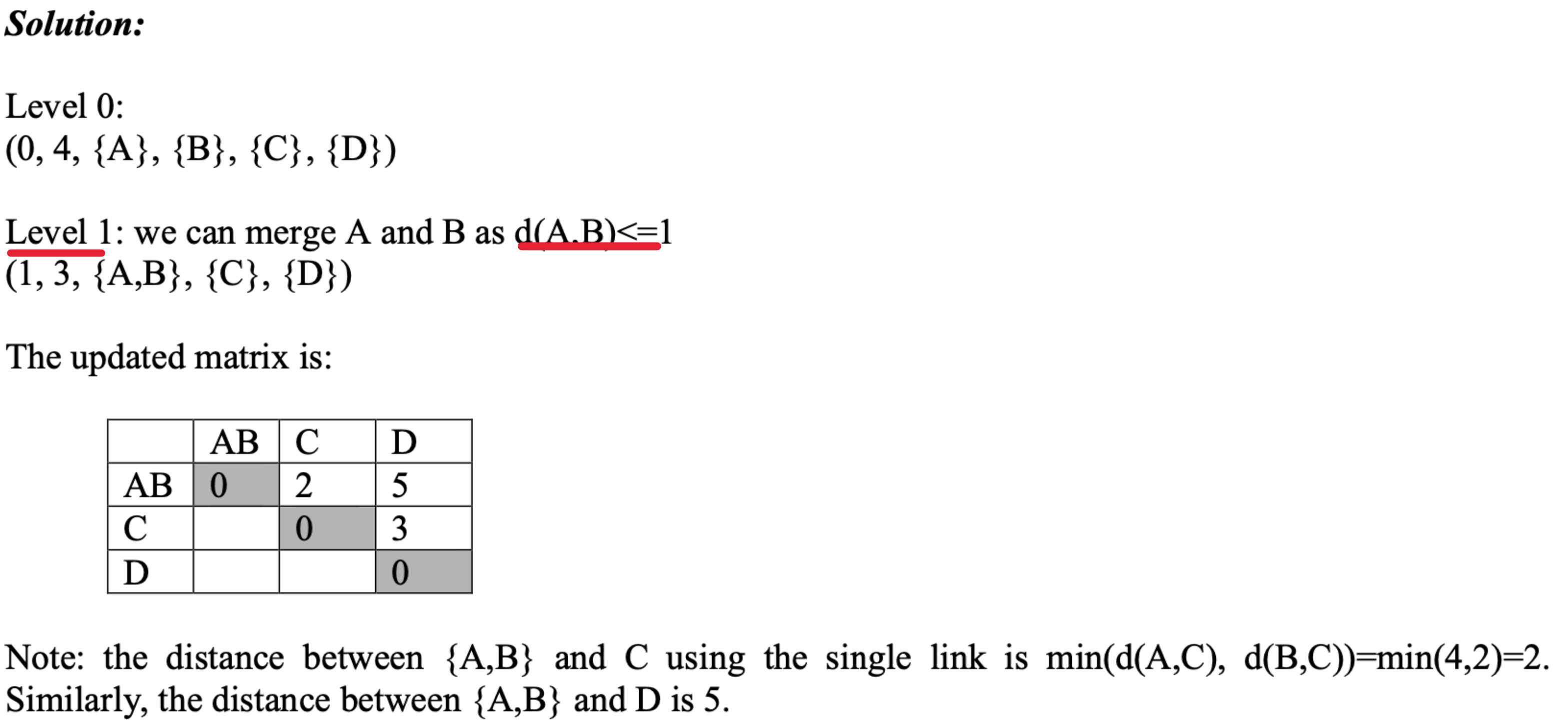
Exercise 3. ***Hierarchical clustering – single link agglomerative algorithm***

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Credit: Irena Koprinska

Example



Solution:

Level 0:
(0, 4, {A}, {B}, {C}, {D})

Level 1: we can merge A and B as d(A,B)<=1
(1, 3, {A,B}, {C}, {D})

The updated matrix is:

	AB	C	D
AB	0	2	5
C		0	3
D			0

Note: the distance between {A,B} and C using the single link is min(d(A,C), d(B,C))=min(4,2)=2. Similarly, the distance between {A,B} and D is 5.

Credit: Irena Koprinska

Example

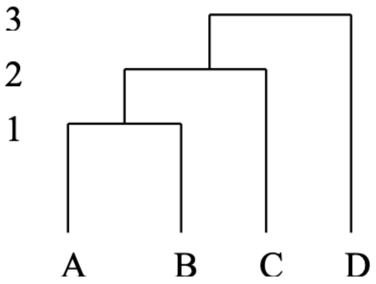
Level 2: we can merge {A,B} and C as the distance between them<=2
(2, 2, {A,B,C}, {D})

The updated matrix is:

	ABC	D
ABC	0	3
D		0

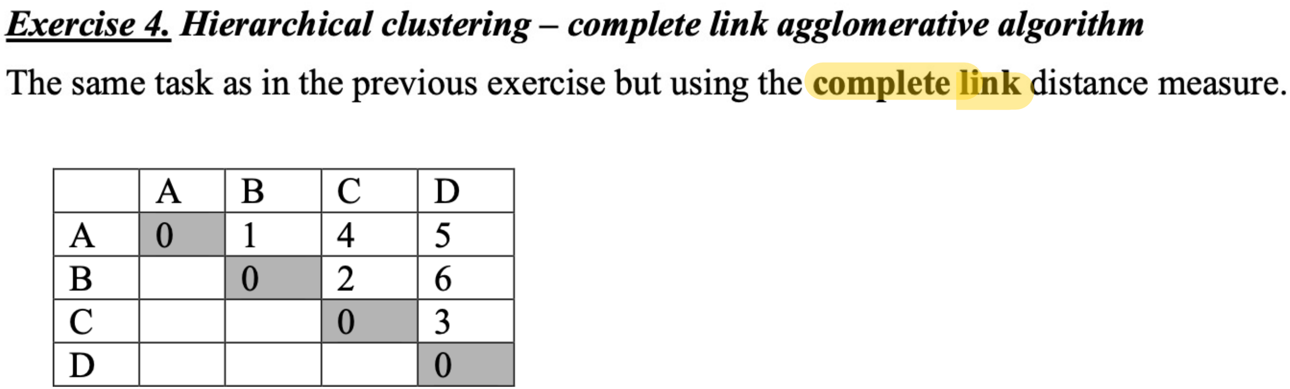
Level 3: we can merge {A,B,C} with D as the distance between them is <=3
(3, 1, {A,B,C,D})
Stop: all items are in 1 cluster.

Dendrogram:



Credit: Irena Koprinska

Example



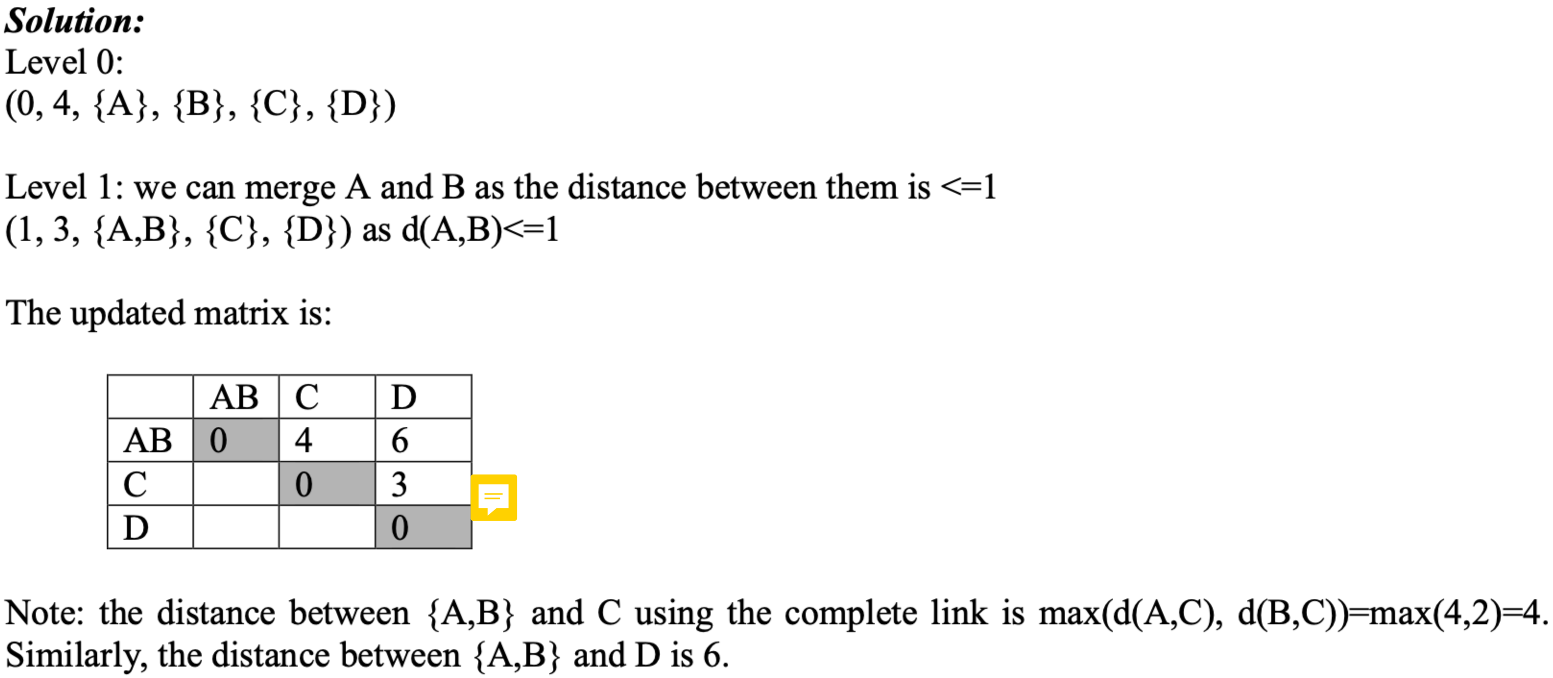
Exercise 4. ***Hierarchical clustering – complete link agglomerative algorithm***

The same task as in the previous exercise but using the **complete link** distance measure.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Credit: Irena Koprinska

Example



Solution:

Level 0:

(0, 4, {A}, {B}, {C}, {D})

Level 1: we can merge A and B as the distance between them is <=1

(1, 3, {A,B}, {C}, {D}) as d(A,B)<=1

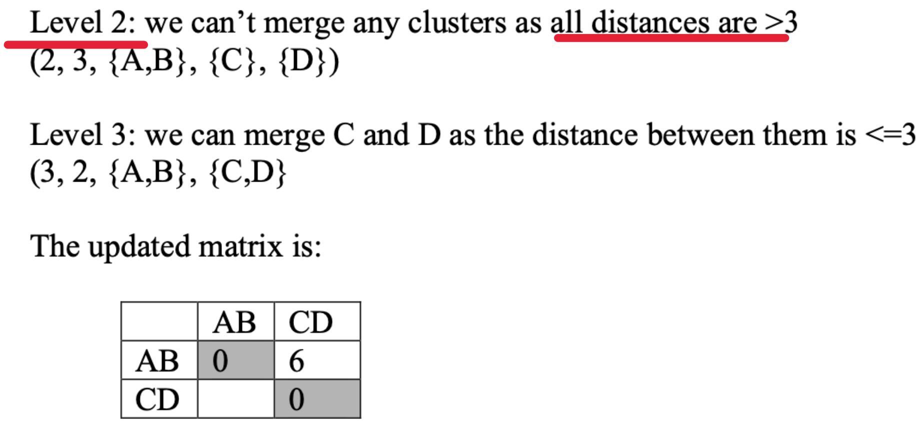
The updated matrix is:

	AB	C	D
AB	0	4	6
C		0	3
D			0

Note: the distance between {A,B} and C using the complete link is max(d(A,C), d(B,C))=max(4,2)=4. Similarly, the distance between {A,B} and D is 6.

Credit: Irena Koprinska

Example



Level 2: we can’t merge any clusters as all distances are >3
(2, 3, {A,B}, {C}, {D})

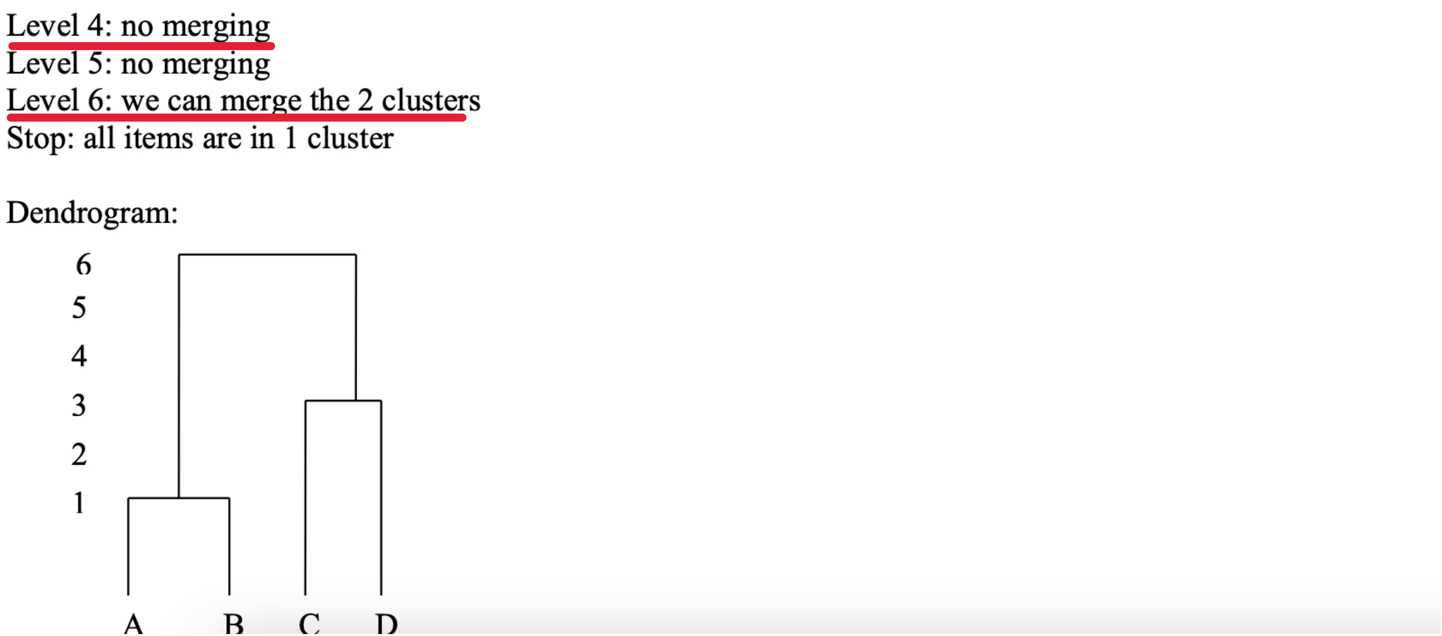
Level 3: we can merge C and D as the distance between them is <=3
(3, 2, {A,B}, {C,D}

The updated matrix is:

	AB	CD
AB	0	6
CD		0

Credit: Irena Koprinska

Example



Level 4: no merging

Level 5: no merging

Level 6: we can merge the 2 clusters

Stop: all items are in 1 cluster

Dendrogram:

Credit: Irena Koprinska

Evaluating Clustering

Internal: Sum of Squared Error (SSE, or inertia)

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

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

- *x* is a data point in cluster *C*$_i$ and m_i is the centroid point (mean) for cluster *C*$_i$

SSE Example

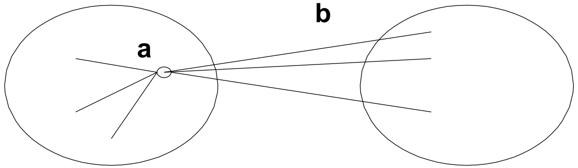
- Suppose we have 3 clusters:
  - Cluster 1: [2, 4] with centroid at 3
  - Cluster 2: [5, 6, 7] with centroid at 6
  - Cluster 3: [8, 10, 12] with centroid at 10
- Squared error for each cluster:
  - SE1 = $(2\text{-}3)^2 + (4\text{-}3)^2 = 1 + 1 = 2$
  - SE2 = $(5\text{-}6)^2 + (7\text{-}6)^2 = 1 +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

Internal: Silhouette Coefficient

- For an individual point *i*
  - Calculate ***a*** = average distance of *i* to points in its cluster
  - Calculate ***b*** = average distance of *i* 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 ≥ b, not the usual case)

  - The closer to 1 the better



- Silhouette coefficient for dataset is average across all *i*

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 it's cluster, i.e. cluster 1
- So a1 = $\sqrt{(\,(1-1)\text{^}2 + (0-1)\text{^}2)} = \sqrt{(0+1)} = 1$

Silhouette Coefficient Example (Cont.)

- Now for the point [1,0] in cluster 1 calculate its average distance from all the objects 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

Exercise: Evaluation

- Evaluating with respect to a gold partition
  - ⏭ code cell after “Evaluating clustering”
  - ⏭ code cell after “Comparing initialisations”
  - TODO Discuss evaluation output

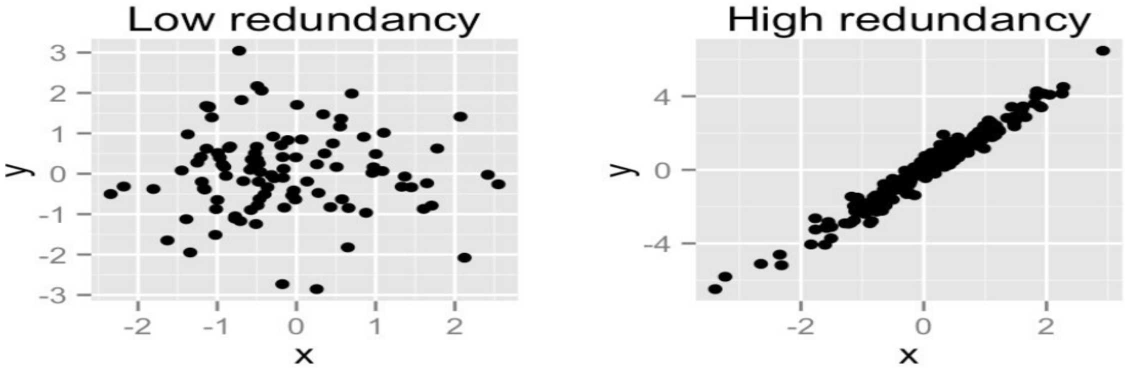
Principal Component Analysis

Principal Components Analysis

- It aims transforming 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):

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

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

- The diagonal elements are the **variances** of the different variables.
- 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

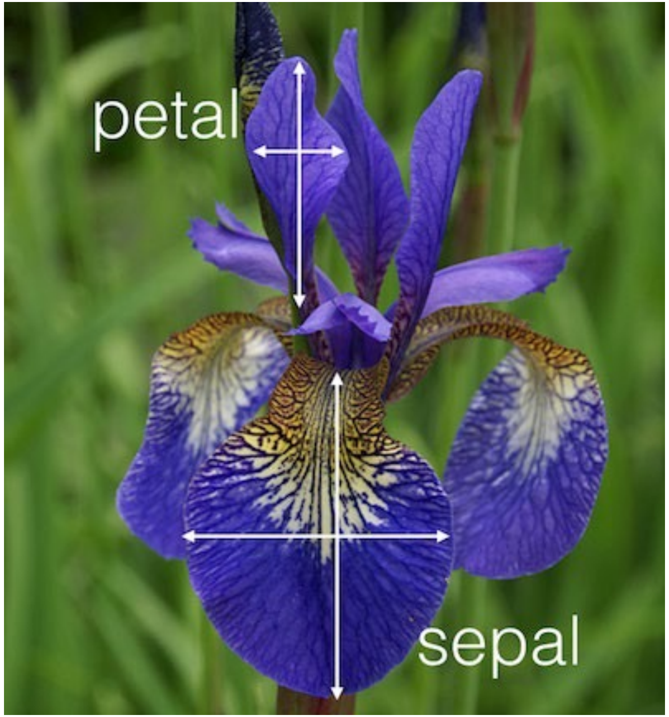
PCA on Iris Dataset

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

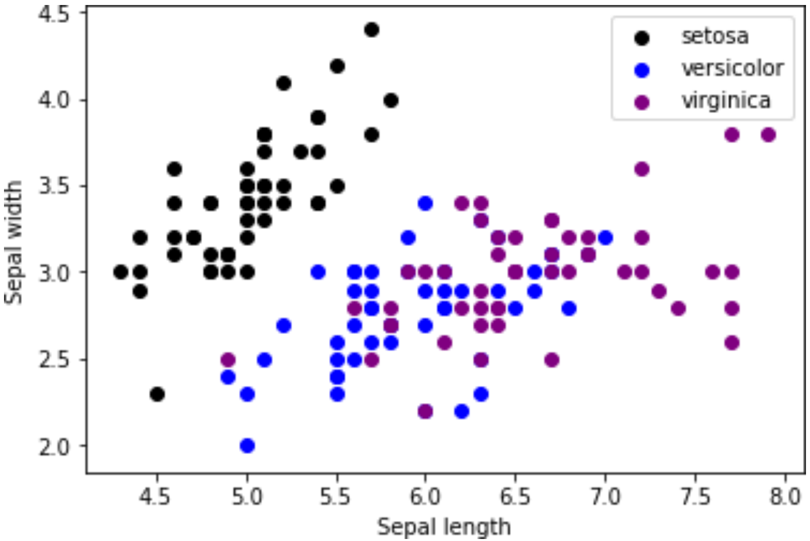
  Setosa, Versicolor and Verginica.

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

  Sepal length VS. Sepal width



Plotting the data points using Sepal Length vs Sepal Width

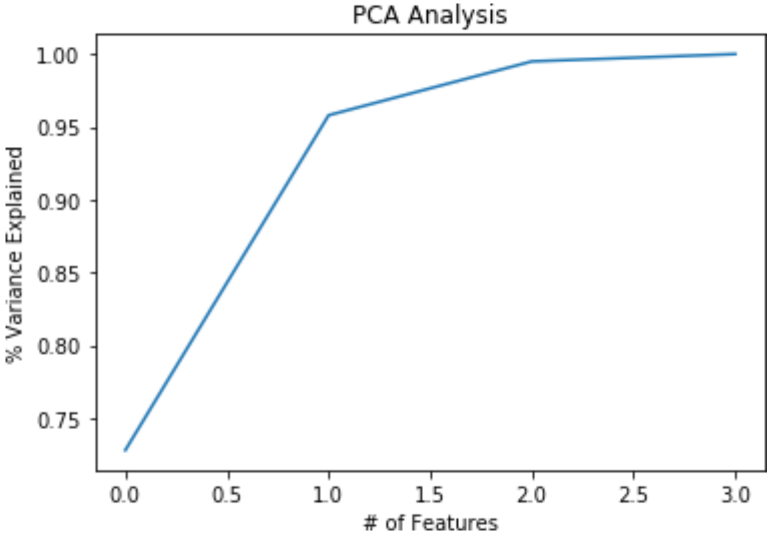


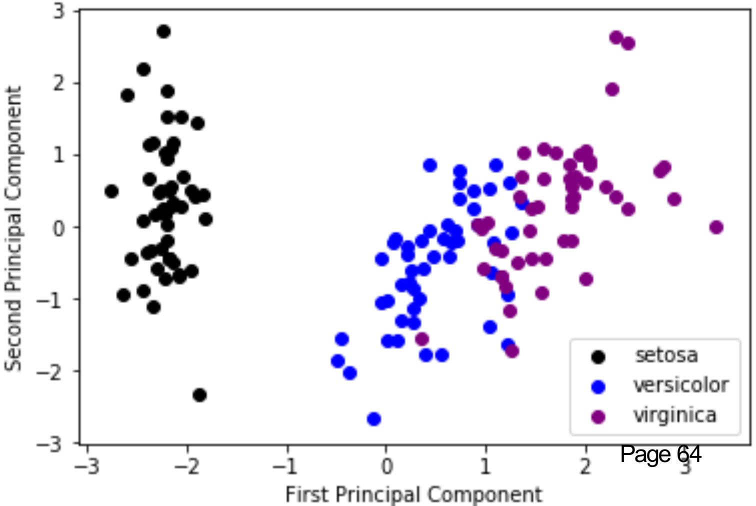
PCA on IRIS Dataset

- Lets 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





Exercise: Dimensionality Reduction

- Selecting the number of clusters
  - ⏭ code cell after “Dimensionality Reduction”
  - ⏭ code cell after “Deciding how many componenets”
  - TODO PCA on digits dataset

Review

Additional Reading (not examinable)

– Tan et al. Introduction to data mining.
  https://goo.gl/hWwuZb
– Aggarwal. Data mining: the textbook.
  https://goo.gl/IQqLwT
– Han. Data mining: concepts and techniques.
  https://goo.gl/CFIMMs
– Scikit-learn user guide, § 2 (Unsupervised learning).
  http://scikit-learn.org/stable/unsupervised_learning.html

Other tools and Techniques (not examinable)

- Scikit-learn user guide, §4.4 (Dimensionality reduction).
  http://scikit-learn.org/stable/modules/unsupervised_reduction.html
- Scikit-learn user guide, §2.7 (Outlier detection).
  http://scikit-learn.org/stable/modules/outlier_detection.html
- Etc