LING 506 - TOPICS IN COMPUTATIONAL LINGUISTICS

Introductory Machine Learning

Yan Tang

Department of Linguistics, UIUC

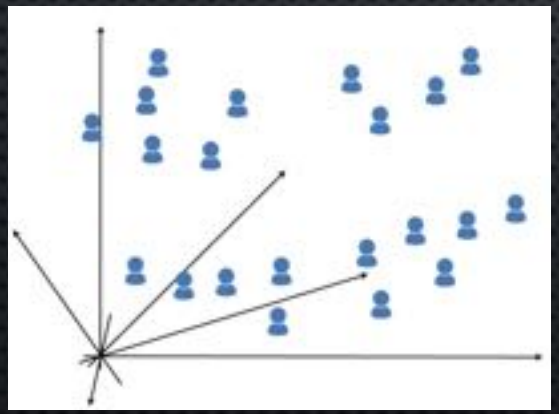
Week 2

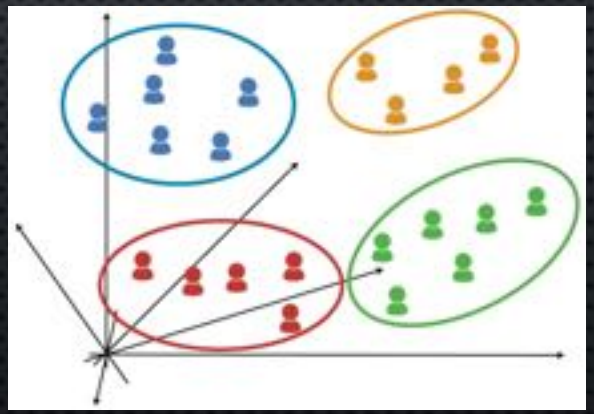
Last class…

- Applications of ML
- Brief history of ML
- Relationships: AI, ML and DL
- Types of ML: unsupervised, supervised and reinforcement
- Some challenges of ML

Clustering

- The simplest problem in ML
- To identify any natural patterns of groupings in data based on their "similarity"
- No criteria for a good or bad clustering
  - Clustering interpretation





Clustering

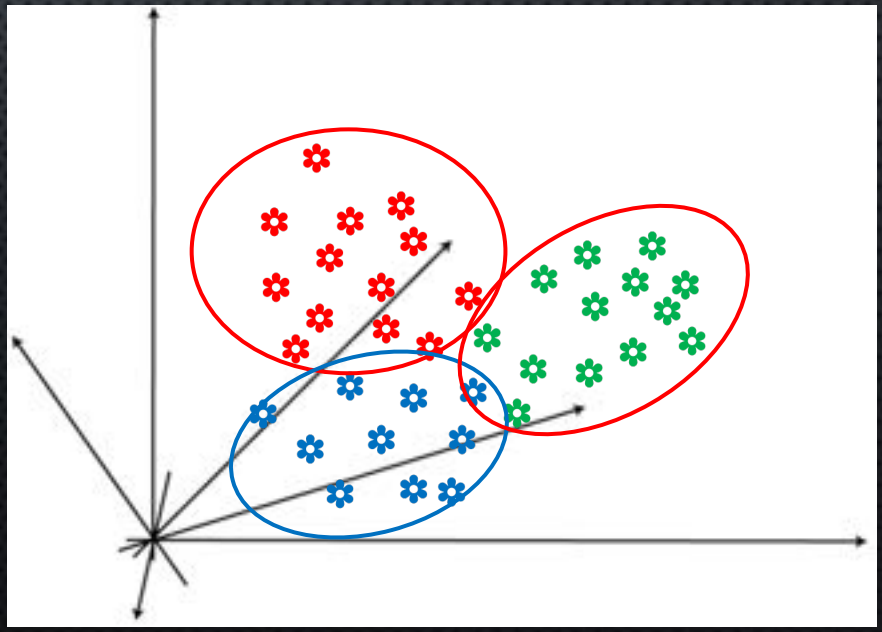


- Considering the following measurements for a large number of iris flowers
  - How many categories do the samples belong to?
  - Can the implicit pattern be revealed by the machine?

	Sepal length	Sepal width	Petal length	Petal width
Sample_1				
Sample_2				
Sample_3				
…				
Sample_n				

Clustering

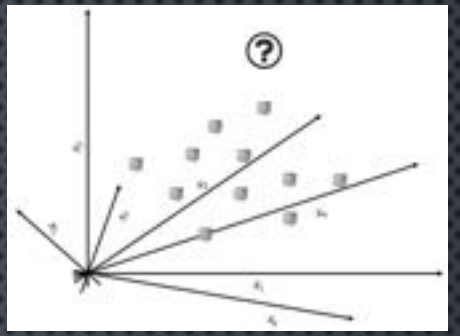
- Without prior knowledge, apply unsupervised ML to the measurements
  - Works on unlabelled data
- Useful data analysis method
  - very few assumptions required



Clustering: algorithms

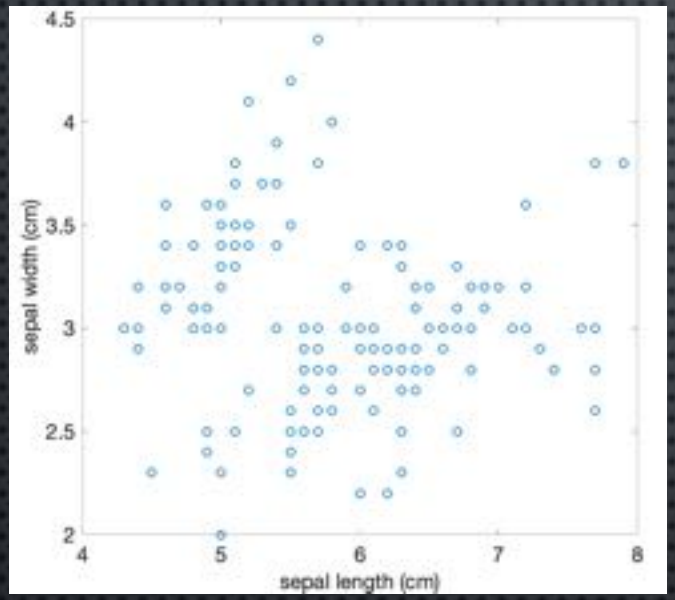
- Centroid models
  - k-means: each cluster identified as data surrounding the nearest centre
- Distribution models
  - Gaussian Mixture Models: each cluster built surrounding a centre of normal distribution
- Connectivity models
  - Hierarchical clustering: based on distance connectivity

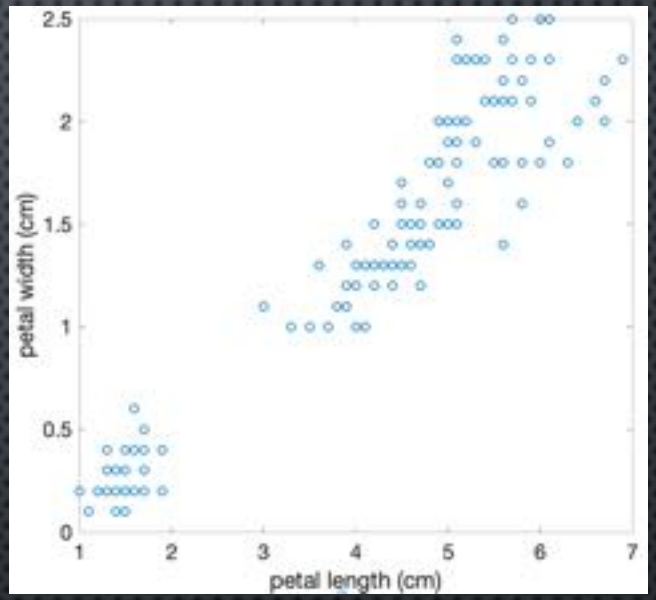
Data visualisation



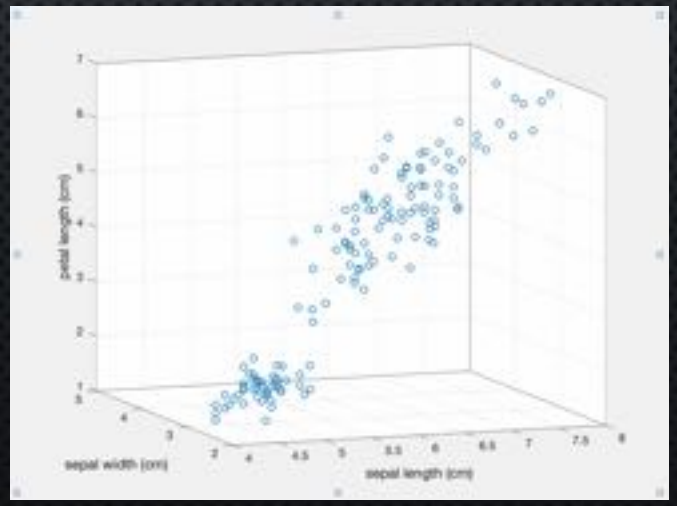
- All comes from data; all is about data
- Data visualisation
  - A quick way to find out explicit groups or patterns
  - 2-variable data: 2-D plot
  - 3-variable data: 3-D plot
  - How about data with more than three variables
- Recall the Iris data set:
  - Four variables: sepal length and width, petal length and width

Data visualisation: Iris data set



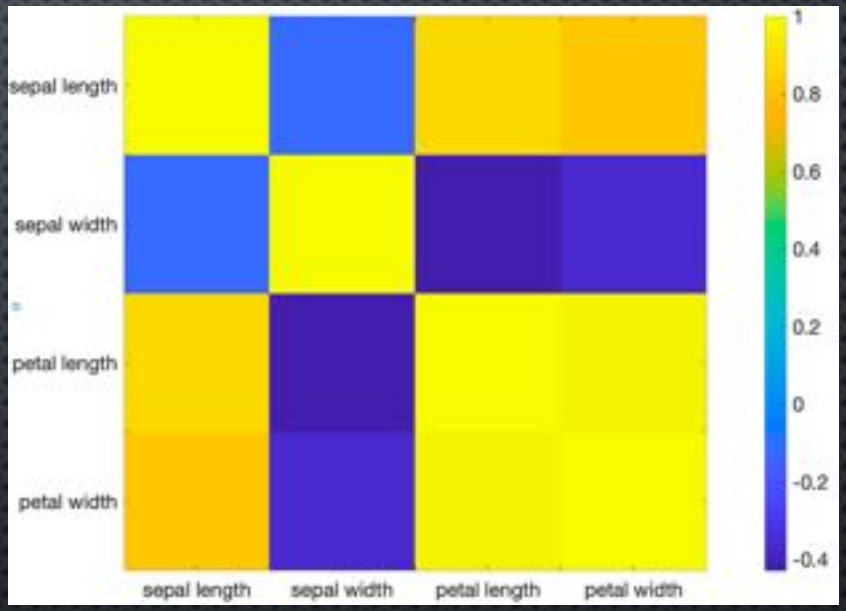


For a clearer picture, more variables usually need to be visualised



Data visualisation: Iris data set

- Some variables may have high correlation
- However, this may still not be enough to cluster flower samples with similar measurements.



TIP: To visualise data having over three variables, ***dimensionality reduction*** techniques can be used.

Dimensionality Reduction

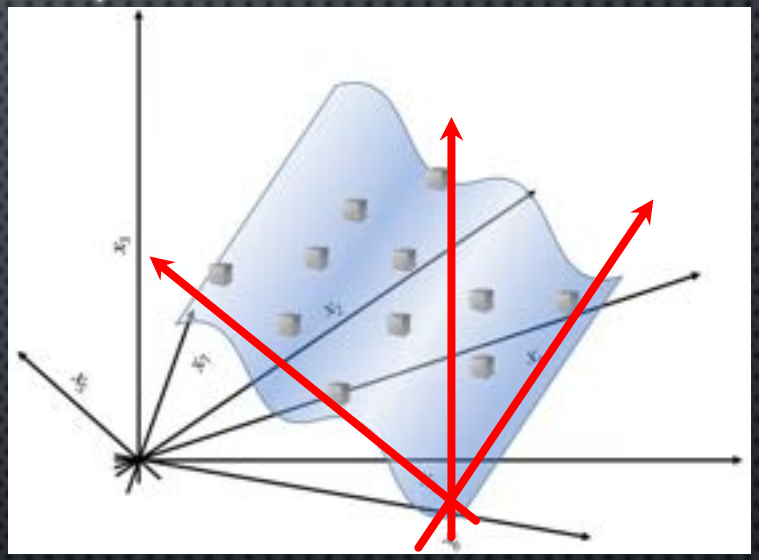
Why reduce data dimensionality?

- Important and effective feature(s) to ML may not be known beforehand
- To discovery the internal structure of data
- Most useful information may be contained in fewer dimensions

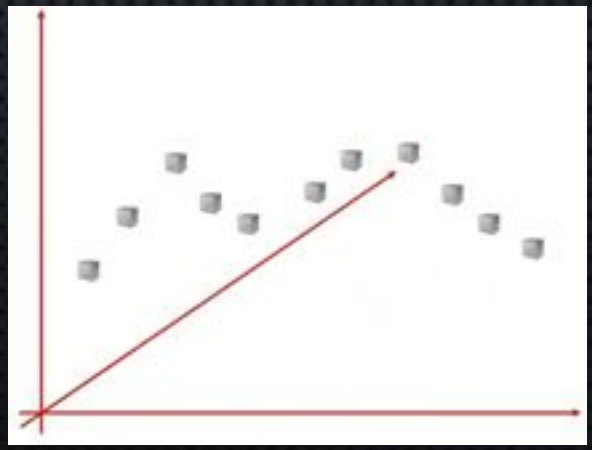
Dimensionality Reduction

How to reduce data dimensionality?

- Transfer variables to a lower dimensional space without losing much information

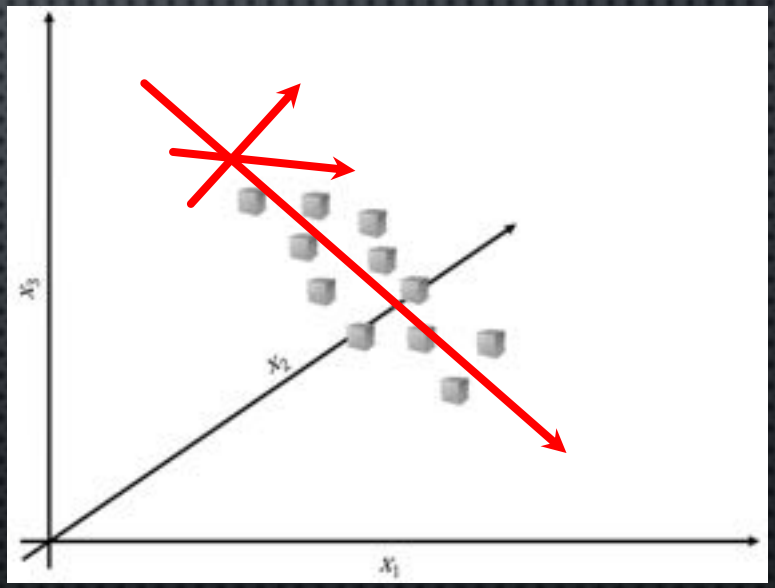


- Techniques:
  - Principal Component Analysis (PCA)
  - Multi-dimensional Scaling (MDS) or Principal Coordinate Analysis (PCoA)



Principle Component Analysis (PCA)

- Can be thought to build a new orthogonal axes for data
- Coordinates are in order how much variance in data they can explain
- Transform data from one N-D space to another N-D space
- Widely used in statistics and other areas beyond ML



Principle Component Analysis (PCA): calculation

Give an input N×P matrix, X:

1) Subtracting the mean
2) Computing the covariance matrix
3) Computing the eigenvalue decomposition of the covariance matrix.

Principle Component Analysis (PCA): outputs of PCA

k: principal component coefficients (loadings) for X

V: principal component variances

- the eigenvalues of the covariance matrix of X

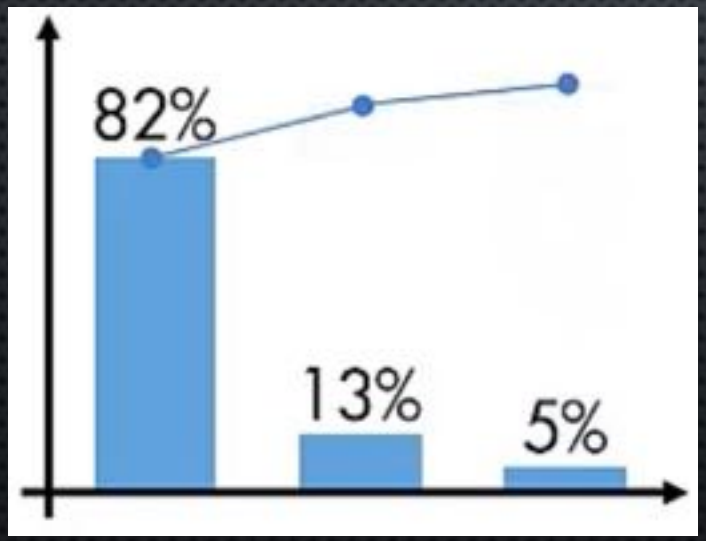
r: % of the total variance explained by each PC

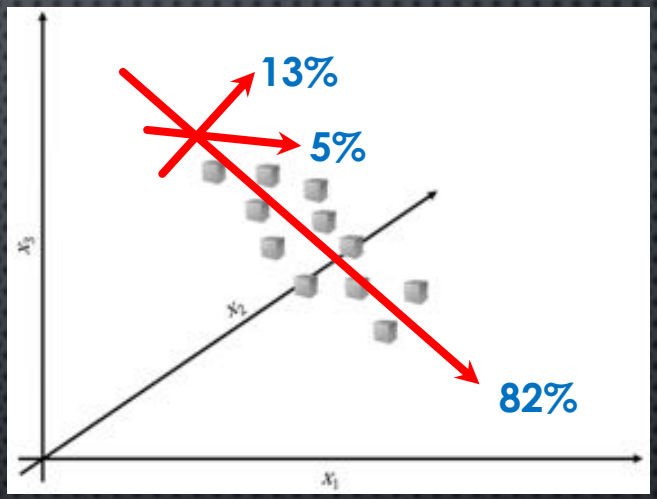
mu: estimated means of the variables in X

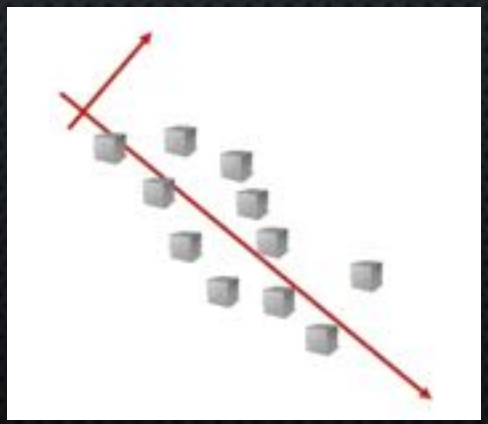
$$PC^i = k_1^i X_1 + k_2^i X_2 + \ldots + k_n^i X_n \longrightarrow \text{Transformed data!}$$

Principle Component Analysis (PCA): data visualisation

Pareto chart







Principle Component Analysis (PCA): Iris data

Raw data

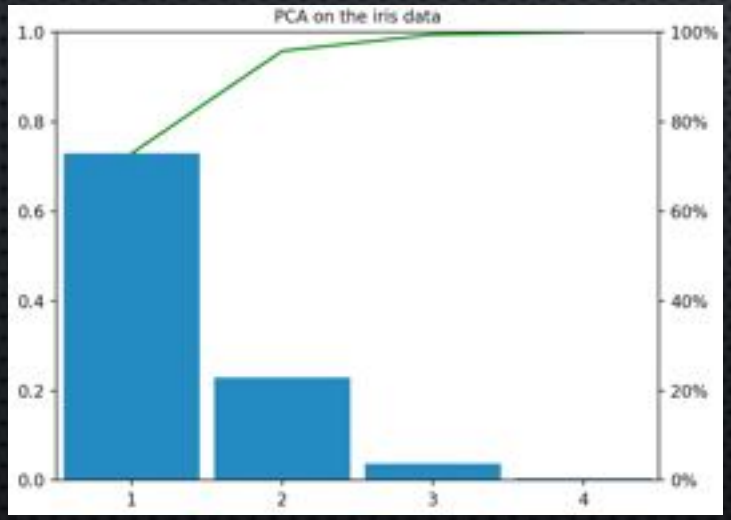
	1	2	3	4
1	5.1000	3.5000	1.4000	0.2000
2	4.9000	3	1.4000	0.2000
3	4.7000	3.2000	1.3000	0.2000
⋮				
148	6.5000	3	5.2000	2
149	6.2000	3.4000	5.4000	2.3000
150	5.9000	3	5.1000	1.8000

Standardised data

	1	2	3	4
1	-0.9007	1.0190	-1.3402	-1.3154
2	-1.1430	-0.1320	-1.3402	-1.3154
3	-1.3854	0.3284	-1.3971	-1.3154
⋮				
148	0.7957	-0.1320	0.8196	1.0539
149	0.4322	0.7888	0.9333	1.4488
150	0.0687	-0.1320	0.7628	0.7907

Transformed data

	1	2	3	4
1	-2.2647	0.4800	0.1277	-0.0242
2	-2.0810	-0.6741	0.2346	-0.1030
3	-2.3642	-0.3419	-0.0442	-0.0284
⋮				
148	1.5212	0.2691	-0.1802	-0.1192
149	1.3728	1.0113	-0.9334	-0.0261
150	0.9607	-0.0243	-0.5282	0.1631



PCA

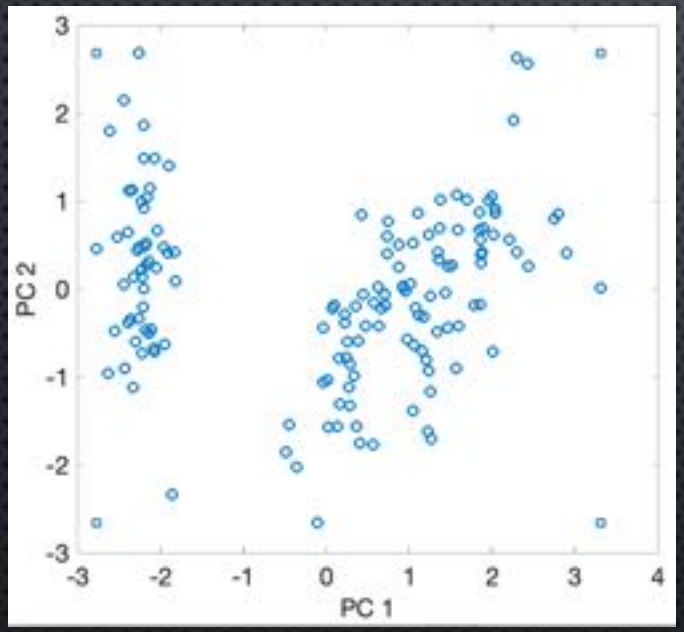
PC coefficients, k:

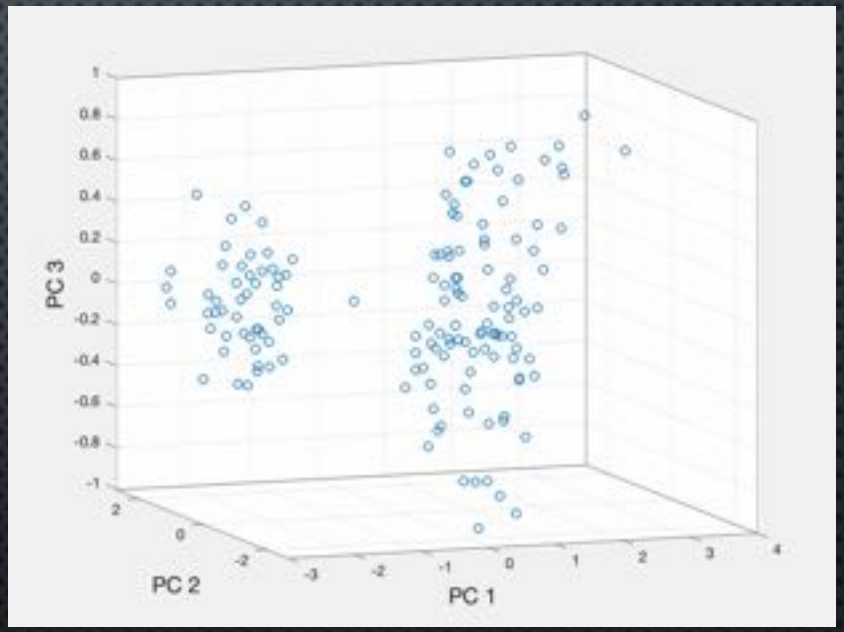
```
[[ 0.52106591 -0.26934744  0.5804131   0.56485654]
 [ 0.37741762  0.92329566  0.02449161  0.06694199]
 [-0.71956635  0.24438178  0.14212637  0.63427274]
 [-0.26128628  0.12350962  0.80144925 -0.52359713]]
```

% Total variance explained, r:

```
[0.72962445 0.22850762 0.03668922 0.00517871]
```

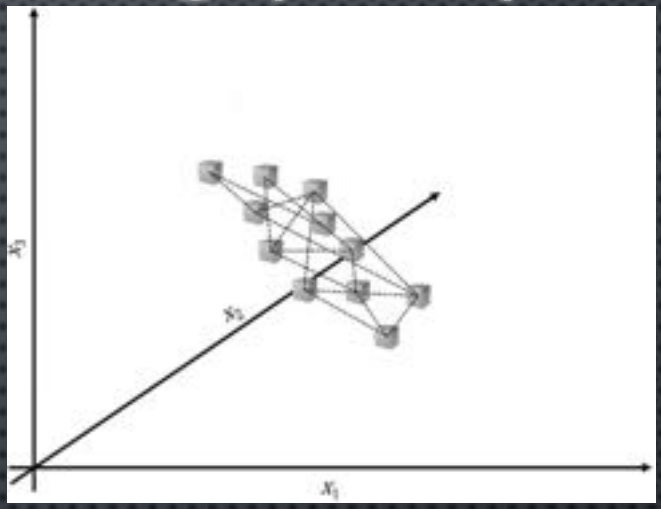
Principle Component Analysis (PCA): Iris data

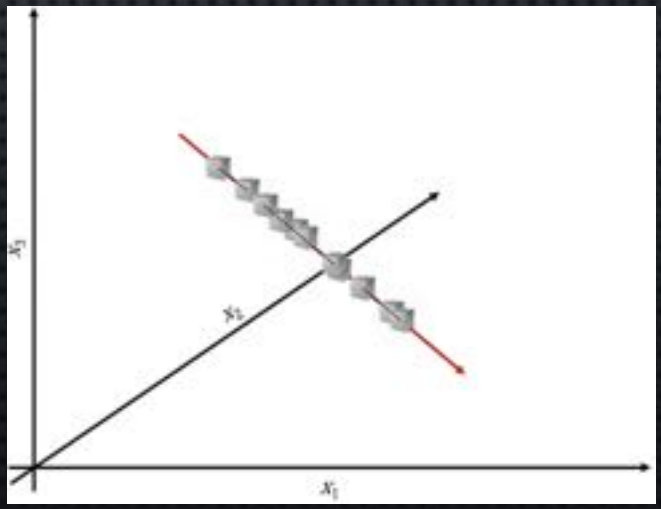




Multi-dimensional Scaling (MDS)

- Seeks a low-dimensional representation of data
- Coordinates are in order how closely pair-wise distances between observations
- Transform data from one N-D space into smallest space
- Can be used with any distance metric
  - E.g. Euclidean, Chebychev and standardised Euclidean





Multi-dimensional Scaling (MDS): calculation

Give an input N×P matrix, X:

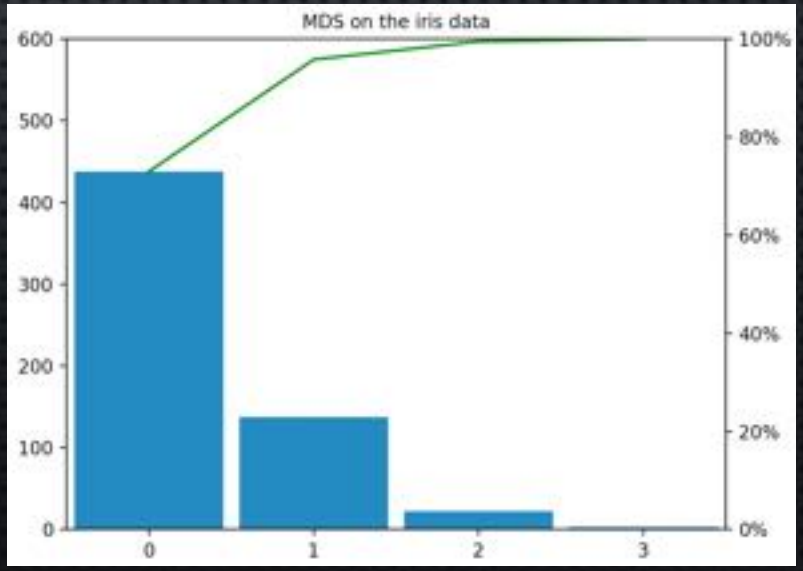
1) Computing pair-wise distance between observations
2) Centering the squared distance matrix
3) Finding the greatest eigenvalues and corresponding eigenvectors
4) Computing the product of the eigenvectors and the diagonal matrix of squared eigenvalues

Principle Component Analysis (PCA): outputs of PCA

E: the position of X in the embedding space

v: the eigenvalues of the centered squared distance matrix

Transformed data!



The percentage of each dimension is equal to that of variances each PC can explain when Euclidean distance is used as the distance metric, hence the transformed data!

Multi-dimensional Scaling (MDS): Iris data

Raw data

	1	2	3	4
1	5.1000	3.5000	1.4000	0.2000
2	4.9000	3	1.4000	0.2000
3	4.7000	3.2000	1.3000	0.2000
⋮				
148	6.5000	3	5.2000	2
149	6.2000	3.4000	5.4000	2.3000
150	5.9000	3	5.1000	1.8000

Standardised data

	1	2	3	4
1	-0.9007	1.0190	-1.3402	-1.3154
2	-1.1430	-0.1320	-1.3402	-1.3154
3	-1.3854	0.3284	-1.3971	-1.3154
⋮				
148	0.7957	-0.1320	0.8196	1.0539
149	0.4322	0.7888	0.9333	1.4488
150	0.0687	-0.1320	0.7628	0.7907

Distance matrix

	1	2	3	4	5	6
1	0	1.1762	0.8456	1.1037	0.2601	1.0384
2	1.1762	0	0.5234	0.4340	1.3865	2.1812
3	0.8456	0.5234	0	0.2839	0.9916	1.8539
4	1.1037	0.4340	0.2839	0	1.2502	2.1008
5	0.2601	1.3865	0.9916	1.2502	0	0.9001
6	1.0384	2.1812	1.8539	2.1008	0.9001	0

MDS

Transformed data

	1	2	3	4
1	-2.2647	0.4800	0.1277	-0.0242
2	-2.0810	-0.6741	0.2346	-0.1030
3	-2.3642	-0.3419	-0.0442	-0.0284
⋮				
148	1.5212	0.2691	-0.1802	-0.1192
149	1.3728	1.0113	-0.9334	-0.0261
150	0.9607	-0.0243	-0.5282	0.1631

Eigenvalues:

```
[ 4.37774672e+02  1.37104571e+02  2.20135313e+01  3.10722546e+00
  8.57403560e-14  5.38239455e-14  5.04407555e-14  4.57653076e-14
  4.56961346e-14  4.20456526e-14  3.55021750e-14  3.51871272e-14
```

PCA vs MDS

- Both reduce dimensionality of data
- Classical MDS using Euclidean distances is equivalent to PCA
- However, different criteria are used
  - PCA: covariance
  - MDS: pair-wise distance
- For data mapping and factor analysis
  - PCA is both
  - MDS is only mapping