Machine Learning, Machine Learning (extended)

9 – Unsupervised Learning: Dimensionality Reduction

Kashif Rajpoot

k.m.rajpoot@cs.bham.ac.uk

School of Computer Science

University of Birmingham

Outline

- High dimensionality
- Curse of dimensionality
- Dimensionality reduction
- Projection
- Preserving ‘interesting’ characteristics in data
- Principal component analysis (PCA)

High dimensionality

- Dimensionality
  - Number of attributes (i.e. features) in the data

- Technological advances lead to increasingly high dimensional data sets
  - Tens to hundreds to thousands…
  - Mass spectrometry, brain functional imaging, genomics, hyperspectral imaging, financial analysis

- High dimensional data => expected to give "more" information (features) about an object?
  - Not always, it could actually cause "curse of dimensionality"

Curse of dimensionality

- Computational burden
  - For example: consider clustering 3000 samples of 2000-dimensional data with k-means algorithm in 10 classes?
- Visualization
  - Typically, we can visualize data only up to 3d/4d
  - What about higher-dimensional data?
- Parameter estimation
  - Higher dimensions need estimation of higher number of parameters (e.g. regression, classification)

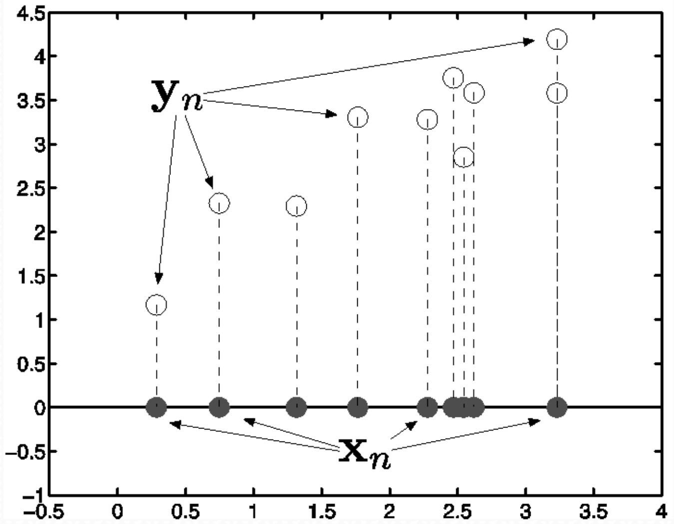
Dimensionality reduction

- Dimensionality reduction aims to avoid the ‘curse of dimensionality’ by reducing the attributes/dimensions/features

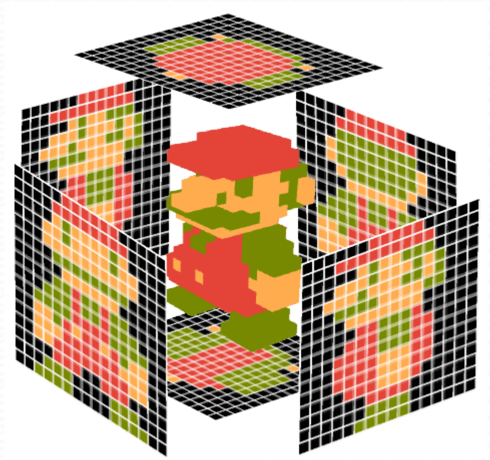
1. Feature selection
   - *Sequential feature selection* (one of the simplest methods for feature selection)
     - Gradually add (remove) a feature to include (exclude)
   - Determine feature scoring/importance by cross-validation
2. **Subspace projection**
   - Make new features by combining (i.e. linearly or non-linearly) the old features
   - Represent the data in fewer number of dimensions but ‘preserving’ the ‘interesting’ characteristics of data

Projection

- Represent the data in fewer number of features but 'preserving' the 'interesting' characteristics of data?

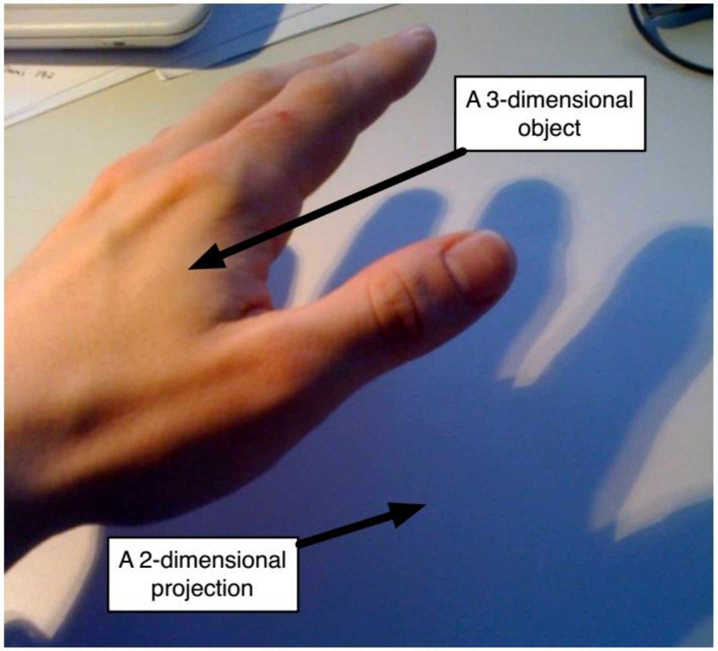


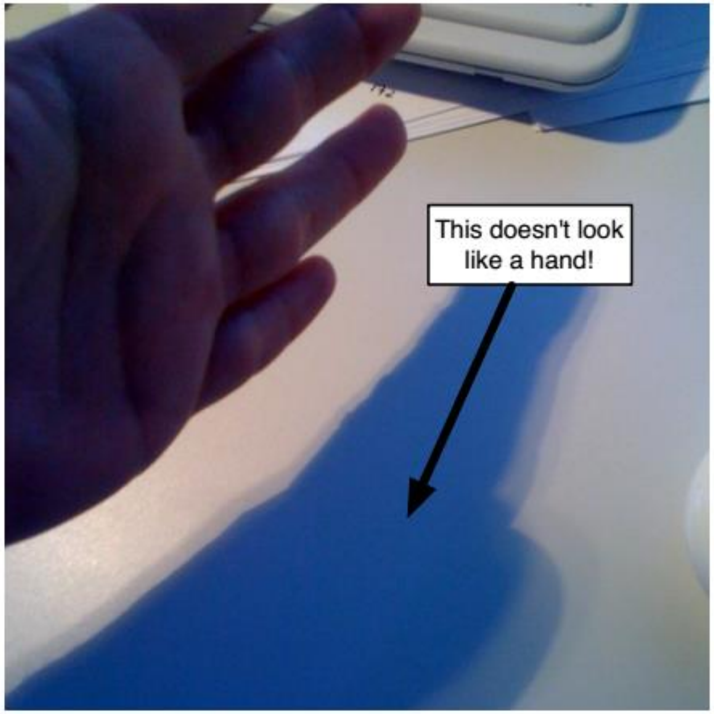
- 3D world to 2D image projection



Projection

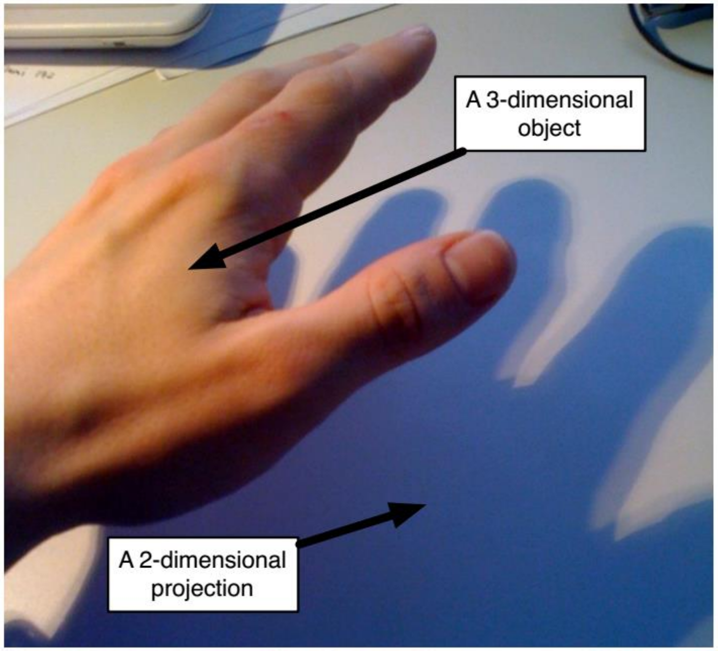
- Project M-dimensional data $\boldsymbol{Y}$ containing N samples into a lower D-dimensional representation $\boldsymbol{X}$ ($D \ll M$)
- $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{W}$
  - $\boldsymbol{Y}$ is NxM
  - $\boldsymbol{W}$ is MxD
  - $\boldsymbol{X}$ is NxD (i.e. D-dimensional)
- What is $\boldsymbol{W}$?
  - Defines the projection
  - Changing $\boldsymbol{W}$ is like changing where the light is coming from or rotating the hand
- $\boldsymbol{Y}$ is hand, $\boldsymbol{X}$ is shadow

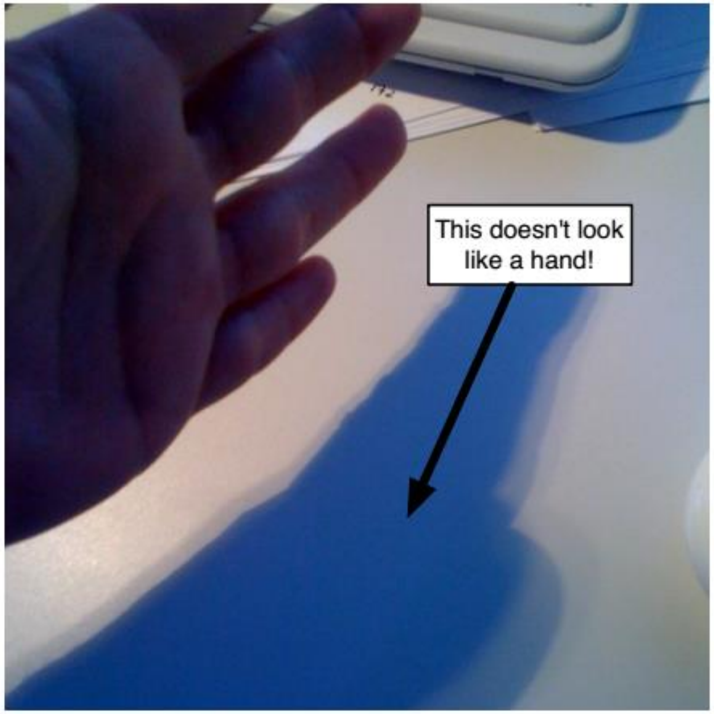




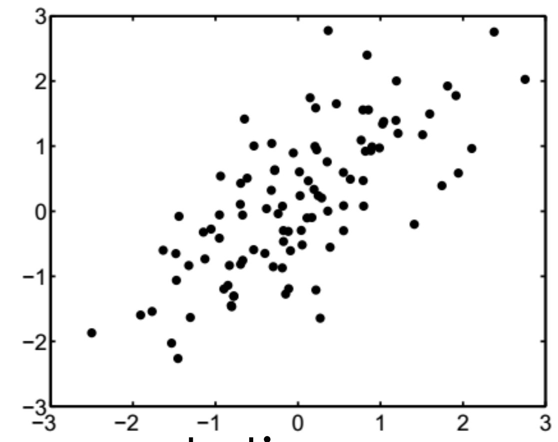
Projection

- Different $\boldsymbol{W}$ will result in different projections
- How to choose $\boldsymbol{W}$?
  - Not all projections will represent our data 'well'
- We should choose a $\boldsymbol{W}$ that preserves the 'interesting' data characteristics
  - such that $D \ll M$
  - M is the actual data's dimensionality
  - D is the projected data's dimensionality

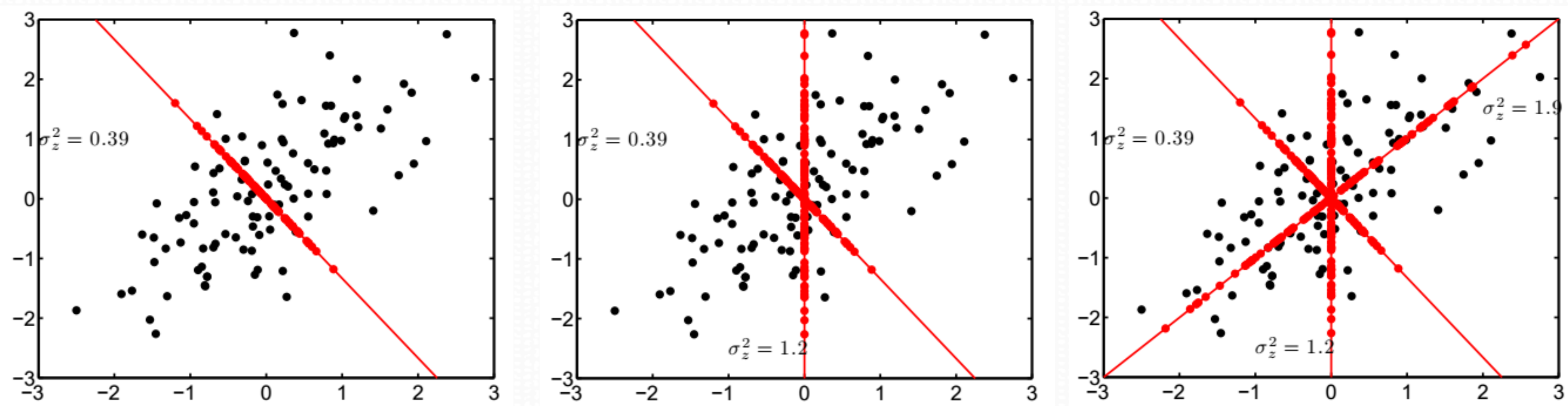




Preserve 'interesting' data characteristics

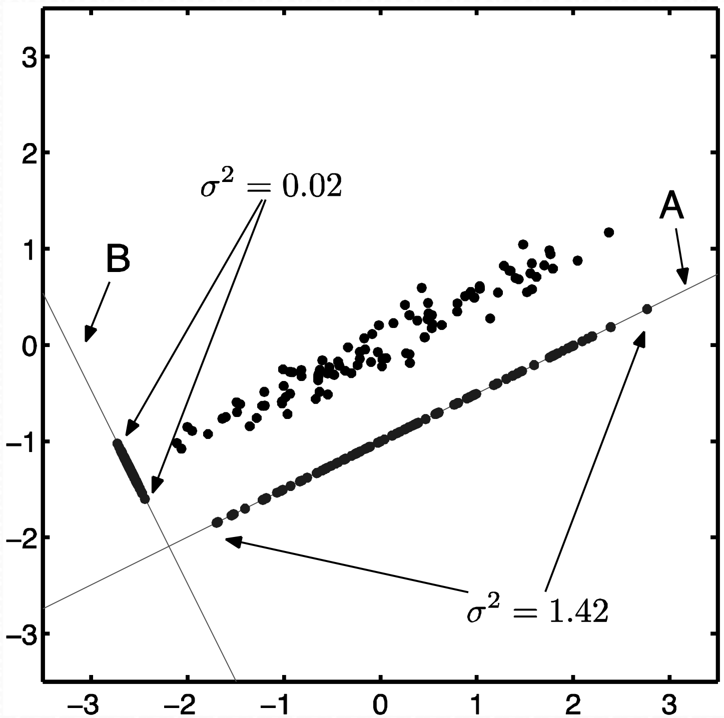


- Project 2-d data into 1-d?
- Pick some arbitrary ***w***
- Project the data onto it
- The position on the line is our 1-d representation
- Compute the variance on the line

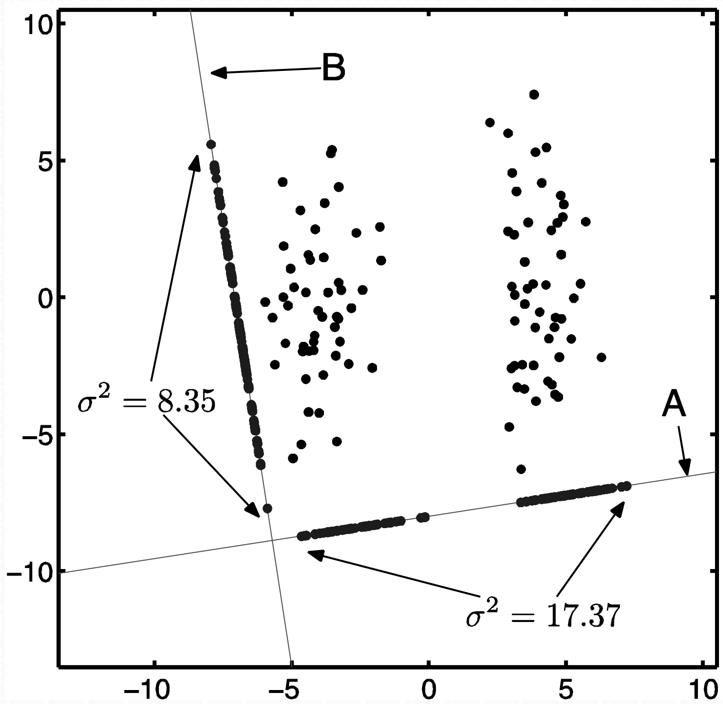


Preserve 'interesting' data characteristics

- Represent the data in fewer number of features but 'preserving' the 'interesting' characteristics of data?
  - High variance = highly "interesting" characteristics



(a) Data from a single, elongated Gaussian



(b) Data from two Gaussians

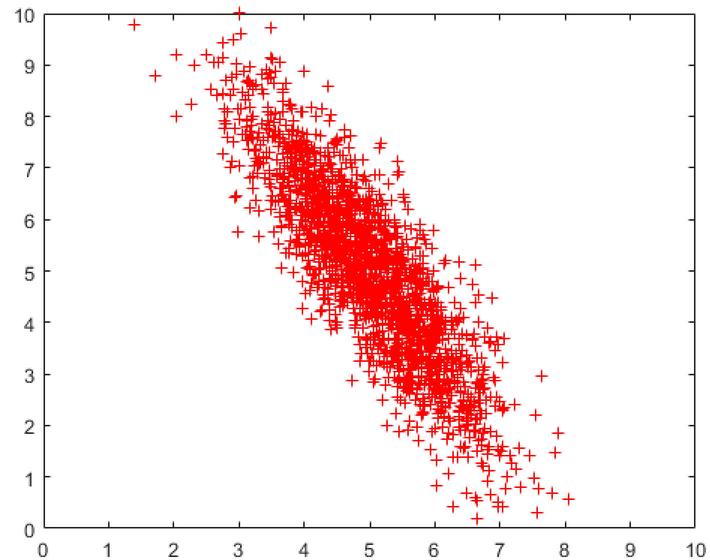
Principal component analysis (PCA)

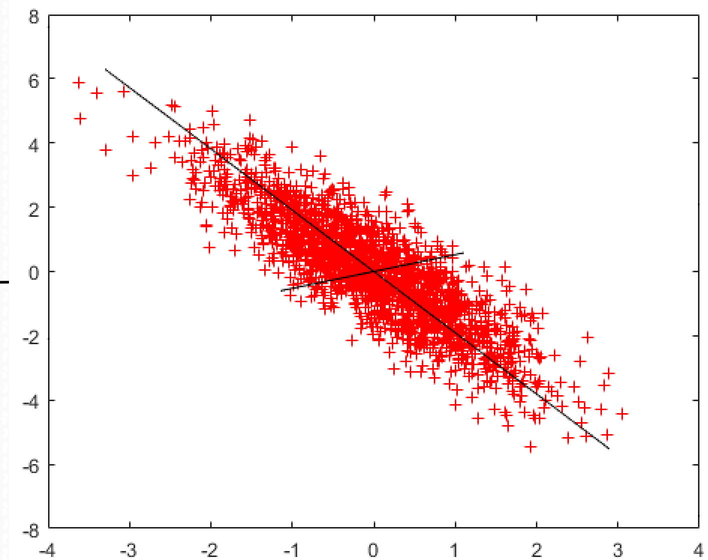
- PCA chooses $\boldsymbol{W}$ such that it transforms M-dimensional data $\boldsymbol{Y}$ into D-dimensional representation $\boldsymbol{X}$ with $D \ll M$
  - It preserves dimensions with high variance ('interesting' characteristics)
  - It removes dimensions which are redundant (i.e. not much 'interesting' – having low variance)

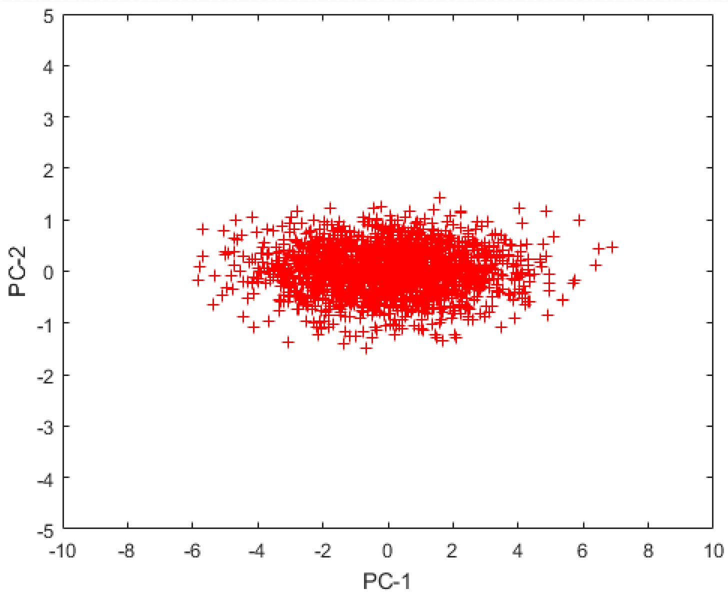
- Find the columns of $\boldsymbol{W}$ one at a time
  - $\boldsymbol{w}_d$ as the d^{th} column of $\boldsymbol{W}$
  - Each Mx1 column defines a new dimension

PCA

- PCA determines the dominant modes of variation from within the data and then projects data onto this ‘natural’ coordinate system
- Matches the coordinate system to the shape of the data







PCA

- Consider $\boldsymbol{w}_d$ as a new dimension, then the data projection in this dimension is computed as:

$$\boldsymbol{x}_d = \boldsymbol{Y}\boldsymbol{w}_d$$

- PCA chooses $\boldsymbol{w}_d$ that maximizes the variance of $\boldsymbol{x}_d$
  - $\sigma_d^2 = \frac{1}{N}\sum_{n=1}^{N}\left(x_d^{(n)} - \mu_d\right)^2$ where $\mu_d = \frac{1}{N}\sum_{n=1}^{N} x_d^{(n)}$

  and $x_d^{(n)}$ denotes the n^{th} sample value for d^{th} attribute
- Each new column of $\boldsymbol{W}$ is found such that it maximizes variance and is orthogonal (perpendicular) to the previous columns

PCA: process

- Search for $\boldsymbol{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_D]$?
- Fortunately, analytical solution exists
- $\boldsymbol{W}$ are the *eigenvectors* of the *covariance matrix* $\boldsymbol{\Sigma}$ of $\boldsymbol{Y}$
  - The covariance matrix $\boldsymbol{\Sigma}$ describes the way different attributes (i.e. features or variables) co-vary

PCA: process

- Covariance matrix $\boldsymbol{\Sigma}$ describes the way attributes co-vary
  - $\boldsymbol{\Sigma}_{i,j}$ denotes the covariance of ith and jth attributes of $\boldsymbol{Y}$

$$\boldsymbol{\Sigma}_{i,j} = \frac{1}{N}\sum_{n=1}^{N}\left(y_i^{(n)} - \mu_i\right)\left(y_j^{(n)} - \mu_j\right)$$

  where μ_i and μ_j denote the mean of ith and jth attributes, respectively, while $y_i^{(n)}$ and $y_j^{(n)}$ denote the nth sample value for ith and jth attributes, respectively.

- The data $\boldsymbol{Y}$ is mean subtracted (along each dimension) to translate the data to the centre of coordinate system
  - Each row is an object, each column is a dimension
  - $\widetilde{\boldsymbol{Y}} = \boldsymbol{Y} - \overline{\boldsymbol{Y}}$, where each column of $\overline{\boldsymbol{Y}}$ is mean value μ across that particular attribute
- The covariance matrix can then be computed as:

$$\boldsymbol{\Sigma} = \frac{1}{N}(\boldsymbol{Y} - \overline{\boldsymbol{Y}})^T(\boldsymbol{Y} - \overline{\boldsymbol{Y}}) = \frac{1}{N}\widetilde{\boldsymbol{Y}}^T\widetilde{\boldsymbol{Y}}$$

PCA: process

- PCA finds new coordinate vectors

$$\boldsymbol{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \ \dots\ \boldsymbol{w}_D]$$

  that align with data shape
  - $\boldsymbol{w}_i^T \boldsymbol{w}_j = 0,\ \forall\ i \neq j$ new dimensions are orthogonal
- PCA looks to find the direction $\boldsymbol{w}$ that maximizes variance in that direction

$$\mathcal{F} = \underset{\boldsymbol{w}}{argmax}\ \ var(\widetilde{\boldsymbol{Y}}\boldsymbol{w})$$

- $var(\widetilde{\boldsymbol{Y}}\boldsymbol{w}) = (\widetilde{\boldsymbol{Y}}\boldsymbol{w})^T \widetilde{\boldsymbol{Y}}\boldsymbol{w} = \boldsymbol{w}^T \widetilde{\boldsymbol{Y}}^{\boldsymbol{T}} \widetilde{\boldsymbol{Y}}\boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}$
- So

$$\mathcal{F} = \underset{\boldsymbol{w}}{argmax}\, \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}$$

subject to $\boldsymbol{w}^T \boldsymbol{w} = \mathbf{1}$

(we could keep increasing $\boldsymbol{w}$ to maximise $\mathcal{F}$, hence the need to constrain $\boldsymbol{w}^T \boldsymbol{w} = \mathbf{1}$)

PCA: process

$$\mathcal{F} = \underset{\boldsymbol{w}}{argmax}\, \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}$$

subject to $\boldsymbol{w}^T \boldsymbol{w} = \mathbf{1}$

- Using the Lagrange multiplier "trick" (beyond our module scope):

$$\mathcal{F} = \underset{\boldsymbol{w}}{argmax}\, \left(\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w} - \lambda(\boldsymbol{w}^T \boldsymbol{w} - \mathbf{1})\right)$$

- To maximize, differentiate with respect to $\boldsymbol{w}$ and set to 0:

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{w}} = 2\boldsymbol{\Sigma}\boldsymbol{w} - 2\lambda\boldsymbol{w} = 0$$

$$\boldsymbol{\Sigma}\boldsymbol{w} = \lambda\boldsymbol{w}$$

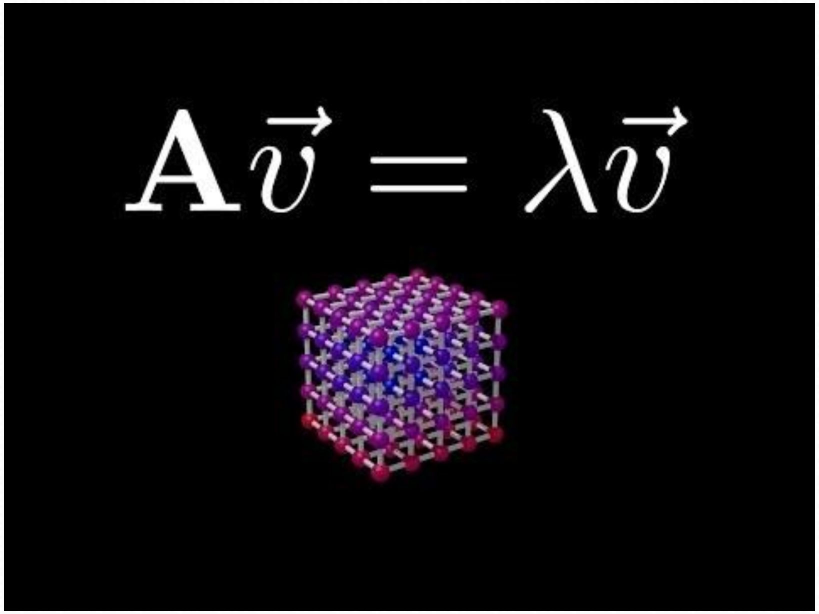
- This can be solved with eigenvalue/eigenvector method (beyond our module scope)

Aside: Eigenvector

- We obtained solution for $\boldsymbol{w}$:
$$\boldsymbol{\Sigma w} = \lambda \boldsymbol{w}$$
- Note that the multiplication of matrix $\boldsymbol{\Sigma}$ with vector $\boldsymbol{w}$ changes it only by a scalar factor
- Such a vector $\boldsymbol{w}$ is called eigenvector
  - λ is the eigenvalue, the scale by which eigenvector $\boldsymbol{w}$ changes
- The eigenvector is a ‘special’ vector which has nice properties to become a transformation vector

Aside: Eigenvector

What is eigenvector?
https://www.youtube.com/watch?v=ue3yoeZvt8E



Introduction to eigenvalues and eigenvectors
https://www.youtube.com/watch?v=PhfbEr2btGQ

PCA: process

- The eigenvector and eigenvalue solution of covariance matrix $\boldsymbol{\Sigma}$ will provide M eigenvectors $[\boldsymbol{w}_1\ \boldsymbol{w}_2\ \dots\ \boldsymbol{w}_M]$ and M eigenvalues $[\lambda_1\ \lambda_2\ \dots\ \lambda_M]$:
$$\boldsymbol{\Sigma}\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$$
- Which $\boldsymbol{w}$ has highest variance?
- Multiplying both sides of above equation by $\boldsymbol{w}_i^T$:
$$\boldsymbol{w}_i^T \boldsymbol{\Sigma}\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i^T \boldsymbol{w}_i = \lambda_i$$

i.e. λ_i denotes variance along $\boldsymbol{w}_i$ dimension since $\boldsymbol{w}_i^T \boldsymbol{\Sigma}\boldsymbol{w}_i = var(\widetilde{\boldsymbol{Y}}\boldsymbol{w}_i)$

- The principal components of data $\boldsymbol{Y}$ are the eigenvectors $[\boldsymbol{w}_1\ \boldsymbol{w}_2\ \dots\ \boldsymbol{w}_M]$ of covariance matrix $\boldsymbol{\Sigma}$, ordered by eigenvalues $[\lambda_1\ \lambda_2\ \dots\ \lambda_M]$

PCA: process

- Having found the principal components (PCs)
$$\boldsymbol{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \dots\ \boldsymbol{w}_M]$$
data $\boldsymbol{Y}$ can now be transformed to these new dimensions:
$$\boldsymbol{X} = \widetilde{\boldsymbol{Y}}\boldsymbol{W}$$

where $\widetilde{\boldsymbol{Y}}$ is the mean-subtracted data

- $\boldsymbol{W}$ is the projection (aka *loadings*)
- $\boldsymbol{X}$ is the projected data (aka *scores*)

PCA: algorithmic workflow

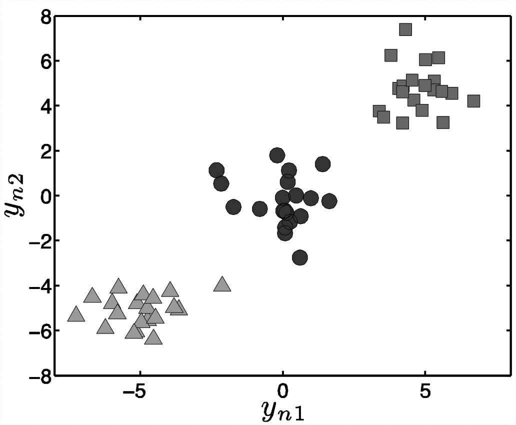
1. Form the NxM zero-mean matrix, by subtracting mean
   - $\widetilde{\boldsymbol{Y}} = \boldsymbol{Y} - \overline{\boldsymbol{Y}}$, where each column of $\overline{\boldsymbol{Y}}$ is mean value across that particular attribute
2. Calculate the MxM covariance matrix $\boldsymbol{\Sigma}$
$$\boldsymbol{\Sigma} = \frac{1}{N}\widetilde{\boldsymbol{Y}}^T\widetilde{\boldsymbol{Y}}$$
3. Calculate the M eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_M)$ and eigenvectors $(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_M)$ of $\boldsymbol{\Sigma}$
4. Sort the eigenvalues and eigenvectors from largest to smallest by eigenvalue
5. Choose D eigenvectors corresponding to highest eigenvalues
6. Compute the scores $\boldsymbol{X}$ (i.e. projections) by projecting data $\widetilde{\boldsymbol{Y}}$ on to new coordinates $\boldsymbol{W}$ (i.e. PCs or eigenvectors)
$$\boldsymbol{X} = \widetilde{\boldsymbol{Y}}\boldsymbol{W}$$

How to choose D?

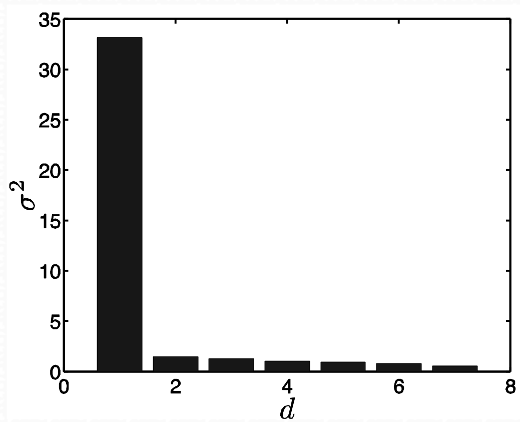
- We get M eigenvectors from the MxM covariance matrix
  - i.e. M new dimensions
- How to choose D dimensions (such that $D \ll M$)?
  - Application domain knowledge
  - Visualization
  - Computational burden
  - 'interesting' structure (i.e. defined by variance)
  - Post-processing results
- Total variation ($\sum_{d=1}^{M} \lambda_d$)
  - Percentage variation preserved by D eigenvectors can be estimated as: $\left[\sum_{d=1}^{D} \lambda_d \,/ \sum_{d=1}^{M} \lambda_d\right] * 100$

PCA: example

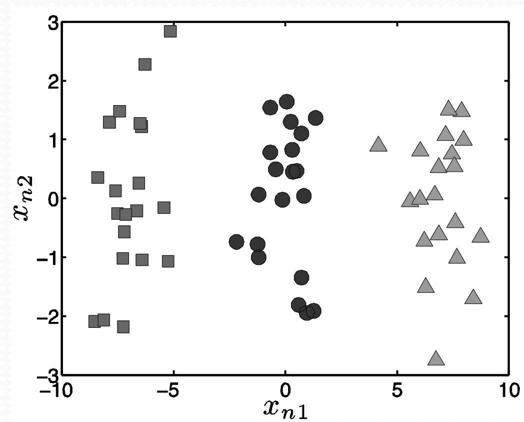
- Consider this 2D data for 3-classes
  - Five additional dimensions were added with random values $\mathcal{N}(0,1)$
  - Perform PCA and dimensionality reduction



(a) First two dimensions of the data objects $\mathbf{y}_n$



(b) The seven eigenvalues (variances of the projected dimensions)

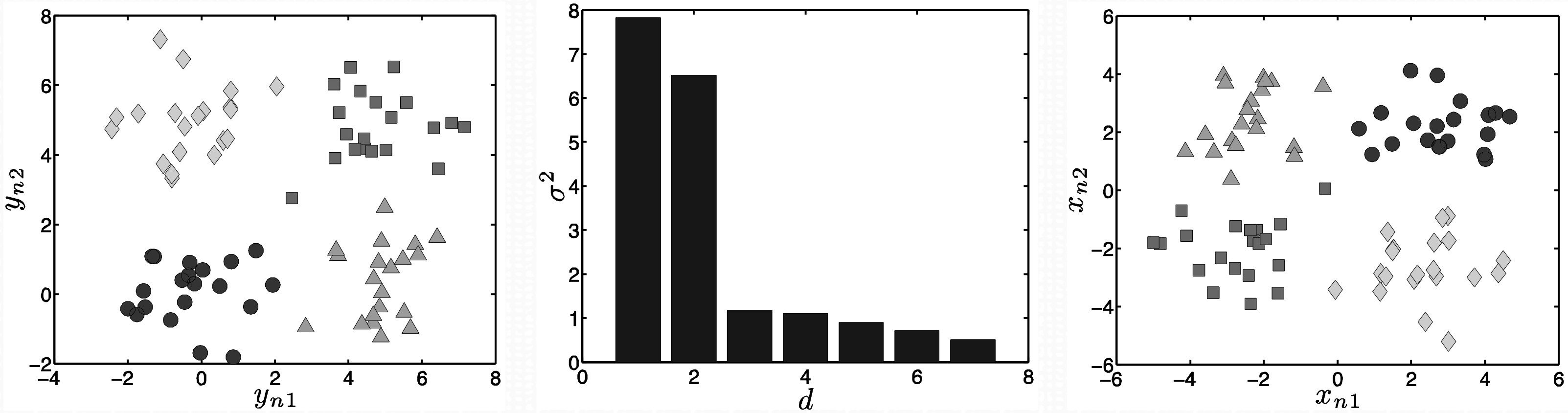


(c) The data projected onto the first two principal components

- It determines the dominant modes of variation from within the data and then projects data onto this 'natural' coordinate system

PCA: example

- Consider this 2D data for 4-classes
  - Five additional dimensions were added with random values $\mathcal{N}(0,1)$
  - Perform PCA and dimensionality reduction



(a) First two dimensions of the data objects $\mathbf{y}_n$

(b) The seven eigenvalues (variances of the projected dimensions)

(c) The data projected onto the first two principal components

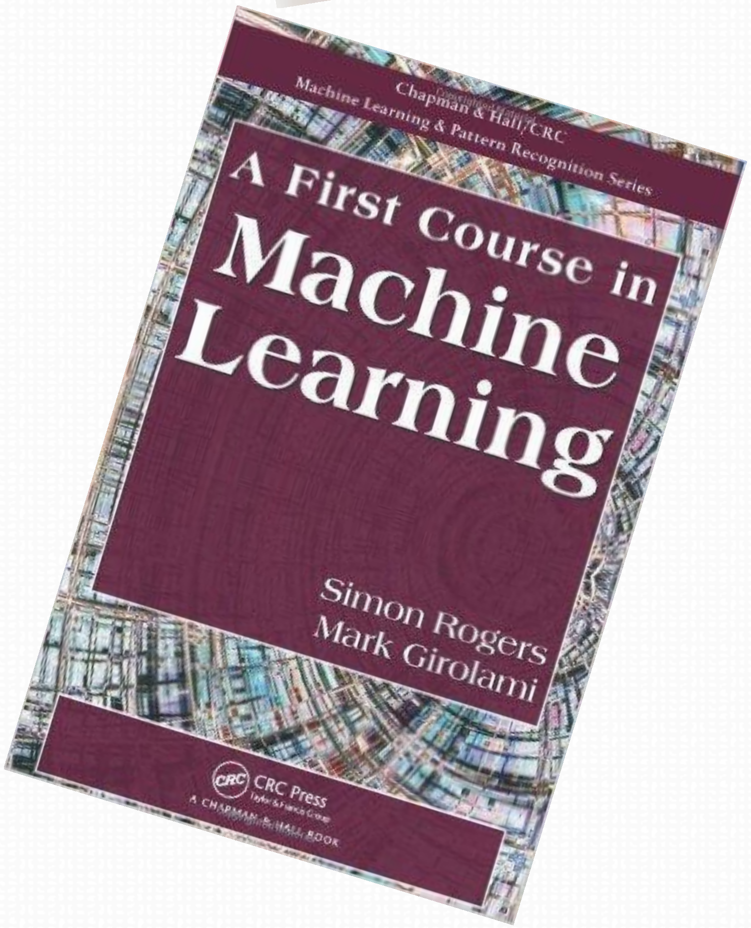
- It determines the dominant modes of variation from within the data and then projects data onto this ‘natural’ coordinate system

Summary

- "curse of dimensionality" with high-dimensional data can pose various problems
- Data projection to preserve 'interesting' characteristics within data
  - Avoid "curse of dimensionality"
- Dimensionality reduction is a form of unsupervised learning
- Dimensionality reduction is often used as a pre-processing step before classification or clustering
  - The 'success' of dimensionality reduction can be evaluated by subsequent processing operation

Exercise (ungraded)

- Experiment with MATLAB code – pcaexample.m (from FCML book website)
- Experiment with MATLAB code – pcaexample2.m (from FCML book website)
- Experiment with MATLAB code – rgbeyepcaexample.m (from Canvas)



Author's material
(Simon Rogers)

Ata Kaban

Iain Styles

That's All

Thank You