Advanced Statistical Inference
Projection

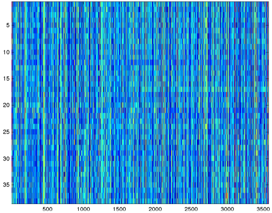
Maurizio Filippone
Maurizio.Filippone@eurecom.fr

Department of Data Science
EURECOM

Part 1: Feature selection and PCA

A problem - too many features

- Aim: To build a classifier that can diagnose leukaemia using Gene expression data.
- Data: 27 healthy samples,11 leukaemia samples ($N = 38$). Each sample is the expression (activity) level for 3751 genes. (Also have an independent test set)



- In general, the number of parameters will increase with the number of **features** – $D = 3751$.
  - e.g. Logistic regression – $\mathbf{w}$ would have length 3751!
- Fitting lots of parameters is hard – imagine Metropolis-Hastings in 3751 dimensions rather than 2!

Features

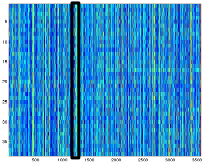
- For visualisation, most examples we've seen have had only 2 features $\mathbf{x} = [x_1, x_2]^\mathsf{T}$.
- We sometimes **created** more: $\mathbf{x} = [1, x_1 x_1^2, x_1^3, \ldots]^\mathsf{T}$.
- Now, we've been given lots (3751) to start with.
- We need to reduce this number.

Features

- For visualisation, most examples we've seen have had only 2 features $\mathbf{x} = [x_1, x_2]^\mathsf{T}$.
- We sometimes **created** more: $\mathbf{x} = [1, x_1 x_1^2, x_1^3, \ldots]^\mathsf{T}$.
- Now, we've been given lots (3751) to start with.
- We need to reduce this number.
- 2 general schemes:
  - Use a **subset** of the originals.
  - Make new ones by **combining** the originals.

Finding a subset – example

- Take one feature – N values.

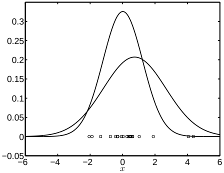


- Some values from objects in class 1, some from class 0.
- Split them based on class and compute μ and σ^2 for each class.
- Compute s for each feature:

$$s = \frac{|\mu_1 - \mu_0|}{\sigma_0^2 + \sigma_1^2}$$

- Keep features with high s.

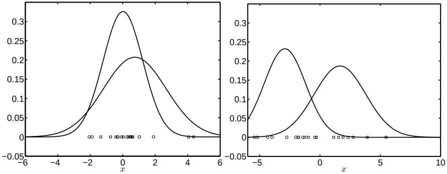
Examples



Features get better (higher s) from left to right...

$$s = \frac{|\mu_1 - \mu_0|}{\sigma_0^2 + \sigma_1^2}$$

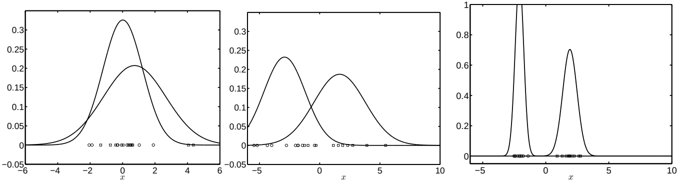
Examples



Features get better (higher s) from left to right...

$$s = \frac{|\mu_1 - \mu_0|}{\sigma_0^2 + \sigma_1^2}$$

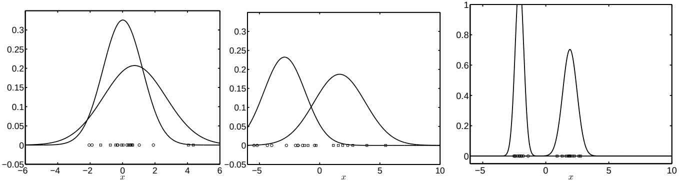
Examples



Features get better (higher *s*) from left to right...

$$s = \frac{|\mu_1 - \mu_0|}{\sigma_0^2 + \sigma_1^2}$$

Examples



Features get better (higher s) from left to right...

$$s = \frac{|\mu_1 - \mu_0|}{\sigma_0^2 + \sigma_1^2}$$

- Each feature has an s-score. The higher the better.
- Use the S features with the highest scores.
- How to choose S?

A feature selection scheme (CV)

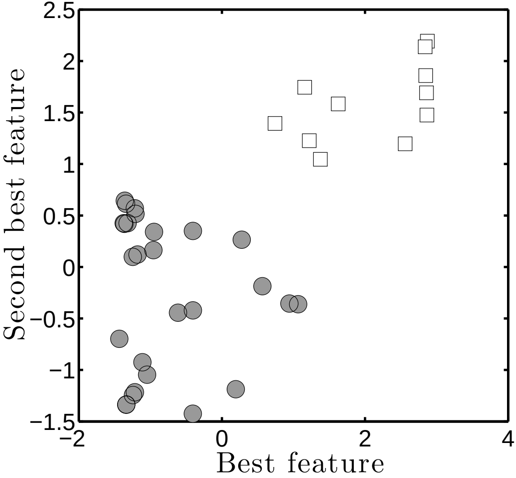
- For each candidate S value:
- Split the data into C folds (just as in CV)
- For each fold...
  1. Find the feature scores on the **training** data.
  2. Train the classifier (whichever we choose).
  3. Record the performance.

A feature selection scheme (CV)

- For each candidate S value:
- Split the data into C folds (just as in CV)
- For each fold...
  1. Find the feature scores on the **training** data.
  2. Train the classifier (whichever we choose).
  3. Record the performance.

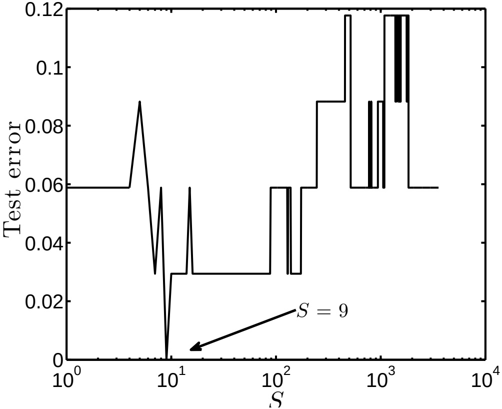
- Important: Must only compute scores on training data. Otherwise we are implicitly using the test labels for training – biased.

Example



Best two features in our leukaemia data (points labeled by class).

Example



Performance as S increases.

Making new features

- An alternative to choosing features is making new ones.

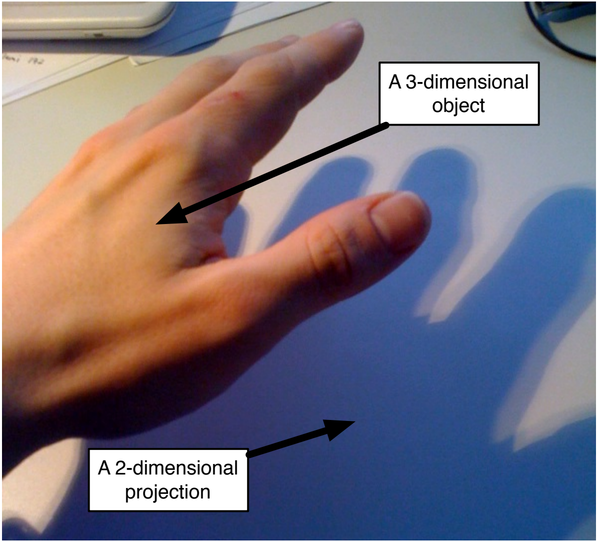
Making new features

- An alternative to choosing features is making new ones.
- Cluster:
  - Cluster the features (turn our clustering problem around)
  - If we use say K-means, our new features will be the K mean vectors.

Making new features

- An alternative to choosing features is making new ones.
- Cluster:
  - Cluster the features (turn our clustering problem around)
  - If we use say K-means, our new features will be the K mean vectors.
- Projection/combination
  - Reduce the number of features by projecting into a lower dimensional space.
  - Do this by making new features that are combinations (linear) of the old ones.

Projection



A 3-dimensional object

A 2-dimensional projection

Projection

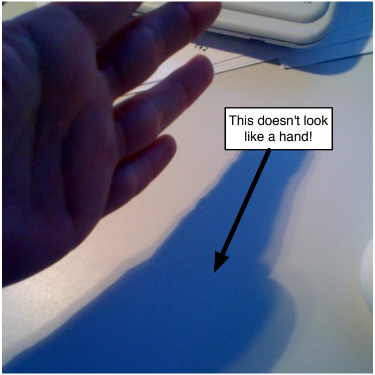
- We can project data (D dimensions) into a lower number of dimensions (M).
- $\mathbf{Z} = \mathbf{X}\mathbf{W}$
  - $\mathbf{X}$ is $N \times D$
  - $\mathbf{W}$ is $D \times M$
- $\mathbf{Z}$ is $N \times M$ – an M-dimensional representation of our N objects.
- $\mathbf{W}$ defines the projection
  - Changing $\mathbf{W}$ is like changing where the light is coming from for the shadow (or rotating the hand).
  - ($\mathbf{X}$ is the hand, $\mathbf{Z}$ is the shadow)
- Once we've chosen $\mathbf{W}$ we can project test data into this new space too: $\mathbf{Z}_{\text{new}} = \mathbf{X}_{\text{new}}\mathbf{W}$

Choosing **W**

- Different **W** will give us different projections (imagine moving the light).
- Which should we use?

Choosing **W**

- Different **W** will give us different projections (imagine moving the light).
- Which should we use?
- Not all will represent our data well...



This doesn't look like a hand!

Principal Components Analysis

- Principal Components Analysis (PCA) is a method for choosing $\mathbf{W}$.
- It finds the columns of $\mathbf{W}$ one at a time (define the mth column as $\mathbf{w}_m$).
  - Each $D \times 1$ column defines one new dimension.

Principal Components Analysis

- Principal Components Analysis (PCA) is a method for choosing $\mathbf{W}$.
- It finds the columns of $\mathbf{W}$ one at a time (define the mth column as $\mathbf{w}_m$).
  - Each $D \times 1$ column defines one new dimension.
- Consider one of the new dimensions (columns of $\mathbf{Z}$):

$$\mathbf{z}_m = \mathbf{X}\mathbf{w}_m$$

Principal Components Analysis

- Principal Components Analysis (PCA) is a method for choosing $\mathbf{W}$.
- It finds the columns of $\mathbf{W}$ one at a time (define the mth column as $\mathbf{w}_m$).
  - Each $D \times 1$ column defines one new dimension.
- Consider one of the new dimensions (columns of $\mathbf{Z}$):

$$\mathbf{z}_m = \mathbf{X}\mathbf{w}_m$$

- PCA chooses $\mathbf{w}_m$ to maximise the variance of $\mathbf{z}_m$

$$\frac{1}{N}\sum_{n=1}^{N}(z_{mn} - \mu_m)^2, \quad \mu_m = \frac{1}{N}\sum_{n=1}^{N} z_{mn}$$

Principal Components Analysis

- Principal Components Analysis (PCA) is a method for choosing $\mathbf{W}$.
- It finds the columns of $\mathbf{W}$ one at a time (define the mth column as $\mathbf{w}_m$).
  - Each $D \times 1$ column defines one new dimension.
- Consider one of the new dimensions (columns of $\mathbf{Z}$):

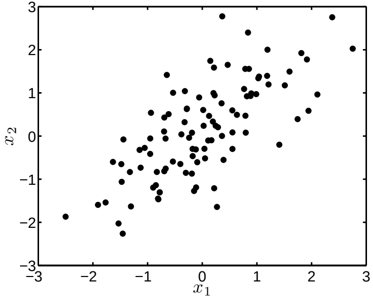
$$\mathbf{z}_m = \mathbf{X}\mathbf{w}_m$$

- PCA chooses $\mathbf{w}_m$ to maximise the variance of $\mathbf{z}_m$

$$\frac{1}{N}\sum_{n=1}^{N}(z_{mn} - \mu_m)^2, \quad \mu_m = \frac{1}{N}\sum_{n=1}^{N} z_{mn}$$

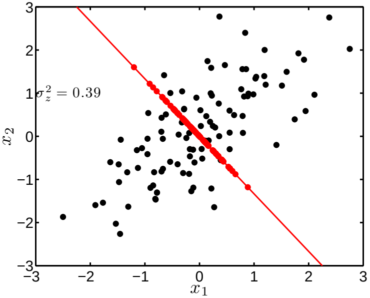
- Once the first one has been found, the $\mathbf{w}_2$ is found that maximises the variance and is **orthogonal** to the first one etc etc.

PCA – a visualisation



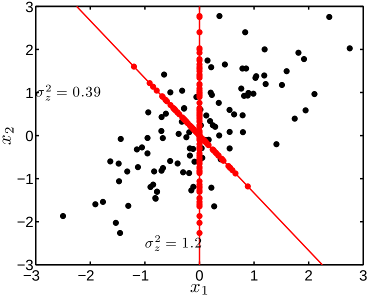
- Original data in 2-dimensions.
- We'd like a 1-dimensional projection.

PCA – a visualisation



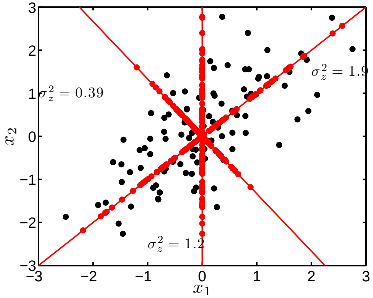
- Pick some arbitrary $\mathbf{w}$.
- Project the data onto it.
- Compute the variance (on the line).
- The position on the line is our 1 dimensional representation.

PCA – a visualisation



- Pick some arbitrary **w**.
- Project the data onto it.
- Compute the variance (on the line).
- The position on the line is our 1 dimensional representation.

PCA – a visualisation

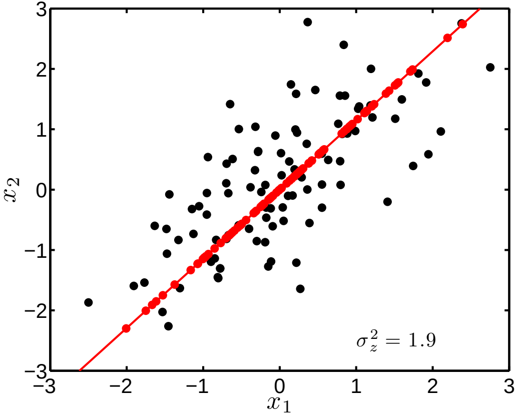


- Pick some arbitrary **w**.
- Project the data onto it.
- Compute the variance (on the line).
- The position on the line is our 1 dimensional representation.

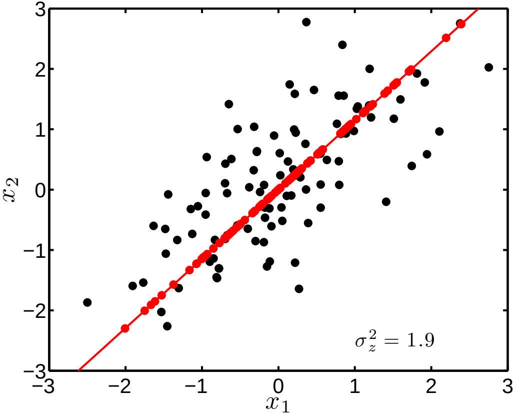
PCA – analytic solution

- Could search for $\mathbf{w}_1, \ldots, \mathbf{w}_M$
- But, analytic solution is available.
- $\mathbf{w}$ are the **eignvectors** of the covariance matrix of $\mathbf{X}$.
  - You don't need to know this!
- Matlab: `princomp(x)`

PCA – analytic solution

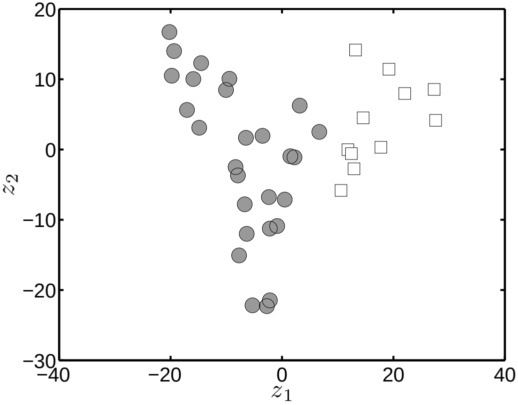


PCA – analytic solution



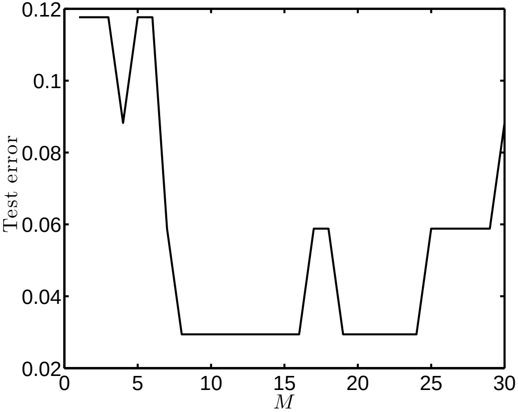
- What would be the second component?

PCA – leukaemia data



First two principal components in our leukaemia data (points labeled by class).

PCA – leukaemia data



Test error as more and more components are used.

Summary

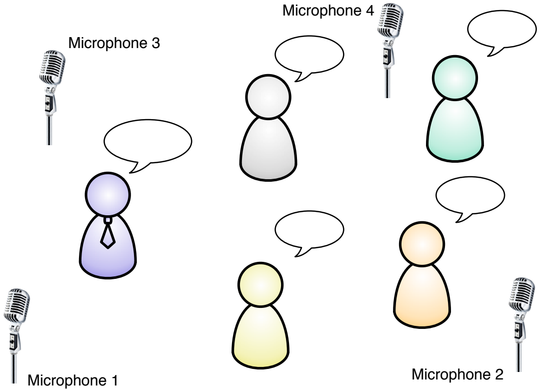
- Sometimes we have too much data (too many dimensions).
- Need to select features.
- Features can be dimensions that already exist.
- Or we can make new ones.
- We've seen one example of each.

Summary

- Sometimes we have too much data (too many dimensions).
- Need to select features.
- Features can be dimensions that already exist.
- Or we can make new ones.
- We've seen one example of each.
- To think about during the break: Why might PCA do worse than the scoring method?

Part 2: ICA
(the cocktail party problem)

The cocktail party problem



- Each microphone will record a combination of all speakers.
- Can we separate them back out again?

Demo

- Online:
- `http://www.cis.hut.fi/projects/ica/cocktail/cocktail_en.cgi`

- Matlab:
  - Available on course webpage
  - To run:
    - load `ica_demo.mat`
    - `ica_image`

Independent components analysis – how it works...

- Corrupted data (images/sounds) is a vector of D numbers. i.e. nth image:

$$\mathbf{x}_n$$

- We have **N** images – stack them up into an $N \times D$ matrix:

$$\mathbf{X}$$

- Assume that this is the result of the following corrupting process:

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{E}$$

- **A** is mixing matrix. **E** is noise. (**S** is $N \times D$).

$$e_{nd} \sim \mathcal{N}(0, \sigma^2)$$

Inference

- From Bayes' (look back...)

$$p(\mathbf{S}|\mathbf{X}, \mathbf{A}, \sigma^2) \propto p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \sigma^2)p(\mathbf{S})$$

Inference

- From Bayes' (look back...)

$$p(\mathbf{S}|\mathbf{X}, \mathbf{A}, \sigma^2) \propto p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \sigma^2) p(\mathbf{S})$$

- In our demo, we found values of **S**, **A** and σ^2 that maximised the log posterior.
- MAP solution...

Inference

- From Bayes' (look back...)

$$p(\mathbf{S}|\mathbf{X},\mathbf{A},\sigma^2) \propto p(\mathbf{X}|\mathbf{S},\mathbf{A},\sigma^2)p(\mathbf{S})$$

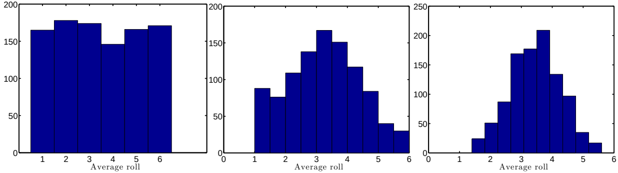
- In our demo, we found values of $\mathbf{S}$, $\mathbf{A}$ and σ^2 that maximised the log posterior.
- MAP solution...
- There is some further reading on the webpage if you want to know more...

Aside – ICA and the central limit theorem

- Central limit theorem (paraphrased):
  - If we keep adding the outcomes of independent random variables together, we eventually get something that looks Gaussian.

Aside – ICA and the central limit theorem

- Central limit theorem (paraphrased):
  - If we keep adding the outcomes of independent random variables together, we eventually get something that looks Gaussian.
- Example: Roll a die m times and take the average. (Repeat this lots of times to get histogram)



- From left to right: $m = 1$, $m = 2$, $m = 5$. Looking more Gaussian as m increases.

Aside – ICA and the central limit theorem

- Sometimes ICA is performed by **reversing** this theorem:

$$\mathbf{X} = \mathbf{AS} + \mathbf{E}$$

- **X** is some random variables added together.
- It will be more 'Gaussian' than **S**
- Find **S** that is as non-Gaussian as possible.

- More resource:
  - `http://www.cis.hut.fi/projects/ica/icademo/`
  - `http://www.cis.hut.fi/projects/ica/`

Summary

- PCA and ICA are both examples of projection techniques.
- Both assume a linear transformation
  - ICA: $\mathbf{X} = \mathbf{AS} + \mathbf{E}$
  - PCA: $\mathbf{Z} = \mathbf{XW}$
- PCA can be used for Data pre-processing or visualisation.
- ICA can be used to separate sources that have been mixed together.
- Also looked at PCA as a feature selection method.