Homework 1 – part 1

Psychophysics - 02458 Cognitive Modeling

For a century, Fechner's log law relating stimulus intensity and perceived intensity was considered valid. Now we know that Steven's power law describe the relationship much better. But how could so many people be mistaken for so long? This issue lies at the heart of cognitive modelling, well, of modelling in general. The problem is that for a part of parameter space, the two models offer very similar predictions.

In the case of stimulus and perceived intensity Fechner might have focused on the relationship between luminance and brightness, which Steven's found to be

$$\textit{Perceived brightness} = 10 \times \textit{Luminance}^{0.33}$$

To see that this relationship might be mistaken for a logarithmic relationship, first calculate the perceived brightness for luminance = 1,2,…,10 using Steven's law. This simulates an observer that rates the brightness. Now, fit Fechner's law to the data:

$$\textit{Perceived brightness} = k \times \log\left(\frac{\textit{Luminance}}{\textit{Luminance threshold}}\right)$$

The parameters that you have to fit are, *k* and the luminance threshold. Although you can du that using an all-purpose optimization routine like Matlab's fminsearch.m function, it is cooler to realize that the problem is linear and can be solved using the pseudoinverse. This is simple and should take no more than 10 lines of code. When you have done the fit, plot the simulated responses and your fit in the same plot. You should be able to see that Fechner's law provide an excellent fit to the data generated using Steven's law. Hence the two models are hard to distinguish in this area of parameter space.

We can now, as Stevens did, compare the two laws in a different area of parameter space. Stevens found that for electric shock

$$\textit{Perceived intensity} = 0.00015 \times \textit{voltage}^{3.3}$$

Now, simulate an observer as before and try and fit Fechner's law to that! (It won't work.)

Of course, we could have realized this analytically by using e.g. Taylor expansions but then we wouldn't have practiced fitting a function to "data", which lies at the heart of mathematical modelling.

02458 Cognitive Modelling – Signal Detection Exercise

Simulate 100 trials from an equal variance observer with *d’* = 1 in three experiments. The criterion of the observer varies between conservative, lax and moderate in the experiments. Estimate *d’* from the simulated data for each of the three experiments. Calculate d’ from the data in each of the experiments. Do you get the correct *d’* for each of the three simulated data sets?

Plot the Receiver Operating Characteristics (ROC) in Gaussian coordinates. Fit the equal variance model to the data and plot that too. Do you get a reasonable fit? Find a single estimate of *d’* from your fit using the ROC. How does this estimate of *d’* compare to the estimates you calculated before?

Simulate 100 trials from an unequal variance observer with μ_S = 2 and σ_S = 1.5 in three experiments in which the criteria ranges from very conservative to very lax. Assume that you did not know that the data came from an unequal variance observer and estimate *d’* from the simulated data for each of the experiments (just as you did in the first part of the exercise). Do you get the same *d’* for each experiment? What are the implications of this result? Fit the unequal variance observer to the simulated data. Can you recover μ_S and σ_S?

Problem Psychometric Function

An observer responds according to a psychometric function shaped like a cumulative Gaussian probability function in a signal detection task. The table lists the number of yes-responses out of 50 trials for five stimulus levels given in arbitrary units.

Stimulus level	0.4	0.9	1.2	1.7	2.3
Number of yes Responses	1	6	13	32	49

Fit the psychometric function to the data using a squared error function.

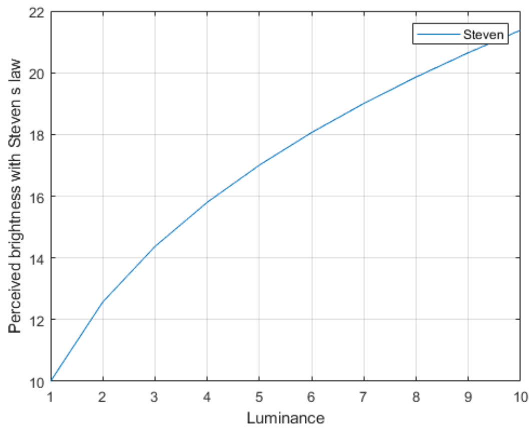
- What are the estimates of the parameters of the psychometric function?
- In a follow-up experiment we use only intensity levels 1 and 2. The task of the observer is to say whether the intensity level is ‘high’ or ‘low’. What value do we expect for the sensitivity (*d*’)?

02458 Cognitive modelling E19
1st Assignment

Adrian Atienza (s192780)
Arianna Taormina (arita@fysik.dtu.dk)
Marie Claire Capolei (macca@elektro.dtu.dk)
Maud Leclerc (s191975)

Part 1: Psychophysics

1) First we calculate the perceived brightness using Steven's law



$$Perceived\ brightness = 10 \times Luminance^{0.33}$$

2) Then we implement Fechner Law in our code, for both Luminance input and electroshock input, and compare it to Steven´s.

$$Perceived\,Brightness = k * log(\frac{luminance}{threshold})$$

3) We fit Fechner's law to the data, getting the two parameters k and the threshold as solutions of linear system in this way:

$$Perceived\,Brightness = k * log(\frac{luminance}{threshold})$$

$$Perceived\,Brightness = k * log(luminance) - k * log(threshold)$$

$$k = x_1$$

$$threshold = x_2$$

$$y = x_1 * log(luminance) - k * log(x_2)$$

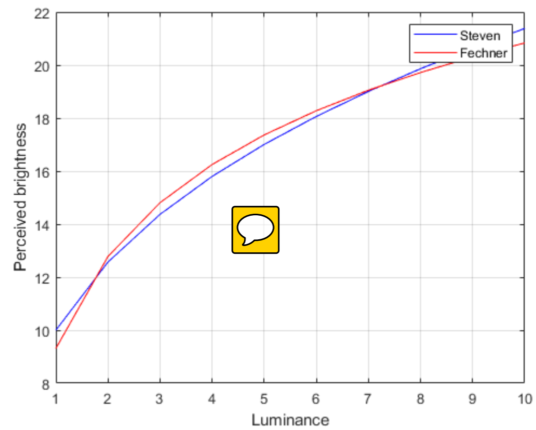
$$k * log(x_2) = x_3$$

$$y = x_1 * log(luminance) - x_3$$

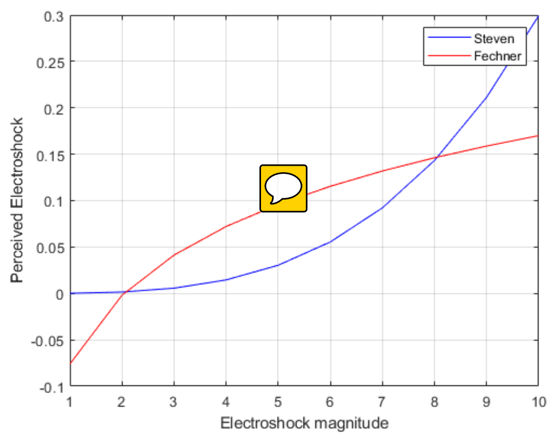
$$y = x_1 * A - x_3$$

$$(x_1, x_3) = pseudoinverse(A) * y^T$$

Comparison Stevens and Fechner for Luminance data



Comparison Stevens and Fechner for electric shock data



As expected, Fechner law can predict the perceived brightness quite well: the results of experiments on luminance conducted for over a century were confirming its validity.

More recent experiments on electric shocks showed that, in all reality, the connection between intensity of the stimulus and its perception is not logarithmic, but exponential (?)/[illegible]er of x (?). As it is shown in the last graph.

Part 2: Signal detection

1) SN and N equal variance

We build our Noise (N) and Signal+Noise (SN) Normal equal variance distributions as follows:

$$N = N(0, 1)$$

$$SN = N(1, 1)$$

Noise is set with mean 0 for convenience. Signal+Noise is therefore set with mean 1, as the distance between the two curves is: d' =1

We run 3 experiments of 100 trials each. From now on, we will refer to SN as "signal" for simplicity.

In the proposed tests, 50% of the trials come from the N noise vector, and 50% come from the SN noisy stimulus vector.

In the experiments, the criterion of the observer varies between lax (the observer is more keen to say ´yes I saw the signal'), moderate (we choose the tradeoff between the N and the SN, aka the point where the curves meet), to conservative (the observer is more keen to say 'no I haven´t perceived the signal').

The change of criterion in real experiments can be accomplished in many ways: from changing the proportion of input/no input in the trials (prior probability of Signal or No Signal, of which the test subjects are informed), to add a payoff or a punishment (were the test subject will have to find the best strategy to optimize their gain).

For each criterion the applied procedure is the following:

1. Count the numbers of Hits (H), Miss (M) on the simulated signal.
2. Count the numbers of False Alarms (FA) and Correct Rejections (CR) on the simulated noise.
3. Get the experimental probabilities and standardize them.
4. Get an estimation of the d' from the standardize probabilities.
5. Compute d' as

$$d' = z_N - z_{SN} = z_{P(hits)} - z_{P(FA)}$$

The number of H Hit, M Miss, FA False alarm, and CR correct rejection for each criterion and the corresponding experimental probabilities are listed in the tables below.

In the third table, the initially assumed d' is compare to the experimental d' (d_prime_approx).

The results show that d' approximation (d_approx) is close to the real d' (d_prime). From additional tests, we notice that the quality of the approximation improves with increasing number of trials, for instance for t = 1000 trials there is no large deviation between the approximation of d'.

Number of trials:100

T3 =

3×3 table

Criterions	d_prime	d_approx
-0.5	1	0.91143
0.5	1	0.56035
1.5	1	0.76173

Number of trials:1000

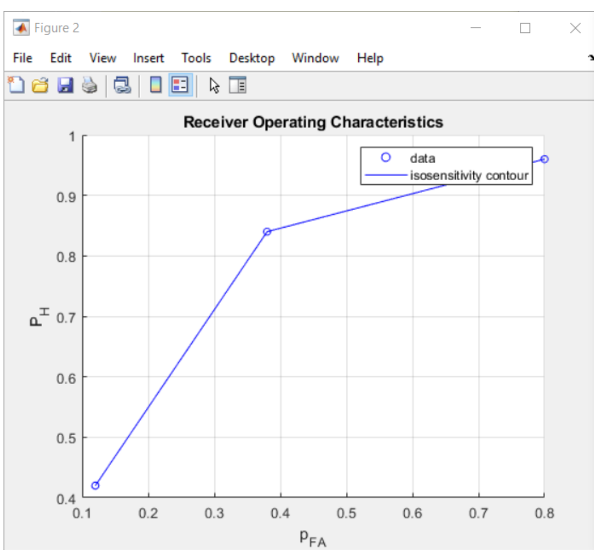
T3 =

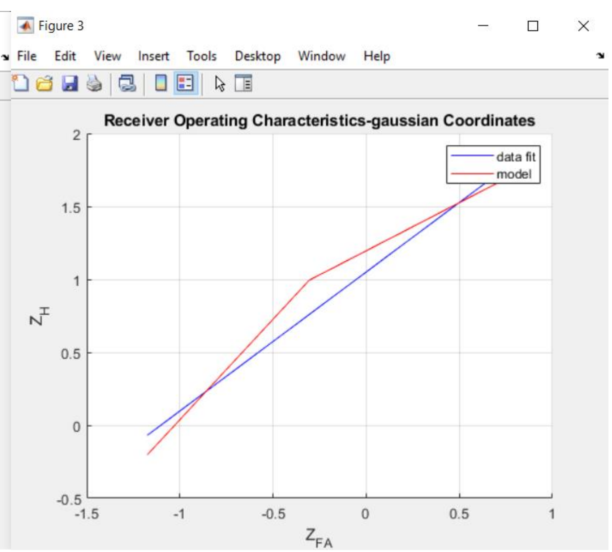
3×3 table

Criterions	d_prime	d_approx
-0.5	1	0.82389
0.5	1	0.93634
1.5	1	1.0386

We deduce that the approximation of d’, the observer capability on distinguishing two different stimuli, is not highly affected by the Criterion. d’ is an index of detectability of the input and it is defined as sensitivity and can be seen as the distance between the mean values of the two normal curves.

We plot the ROC in Gaussian coordinates and fit the equal variance model to the data.





We find a single estimate of *d′* from the fit using the ROC applying

$$d' = z\_H + C$$

Number of trials:100

T4 =

3×4 table

Criterions	d_prime	d_approx	d_eval_ROC
-0.5	1	0.91143	1.0548
0.5	1	0.56035	0.70189
1.5	1	0.76173	0.85665

We can see that the d′ values remain relatively invariant when estimated with different methods and it gets better with higher number of trials.

Number of trials:1000

T3 =

3×4 table

Criterions	d_prime	d_approx	d_eval_ROC
-0.5	1	0.96069	0.99085
0.5	1	1.0379	1.0534
1.5	1	1.0034	0.98134

2) SN and N unequal variance

We build our Noise (N) and Signal+Noise (SN) as Normal distributions. The Noise distribution is kept constant with respect to the previous experiments, while the signal mean and variance are changed.

$$N = N(0, 1)$$

$$SN = N(2, 1.5)$$

We update the criterion to be C = [-0.5 0.8 1.5], again lax, moderate and conservative.

We then estimate *d'* for an unequal variance observer in the 3 experiments with different criterions.

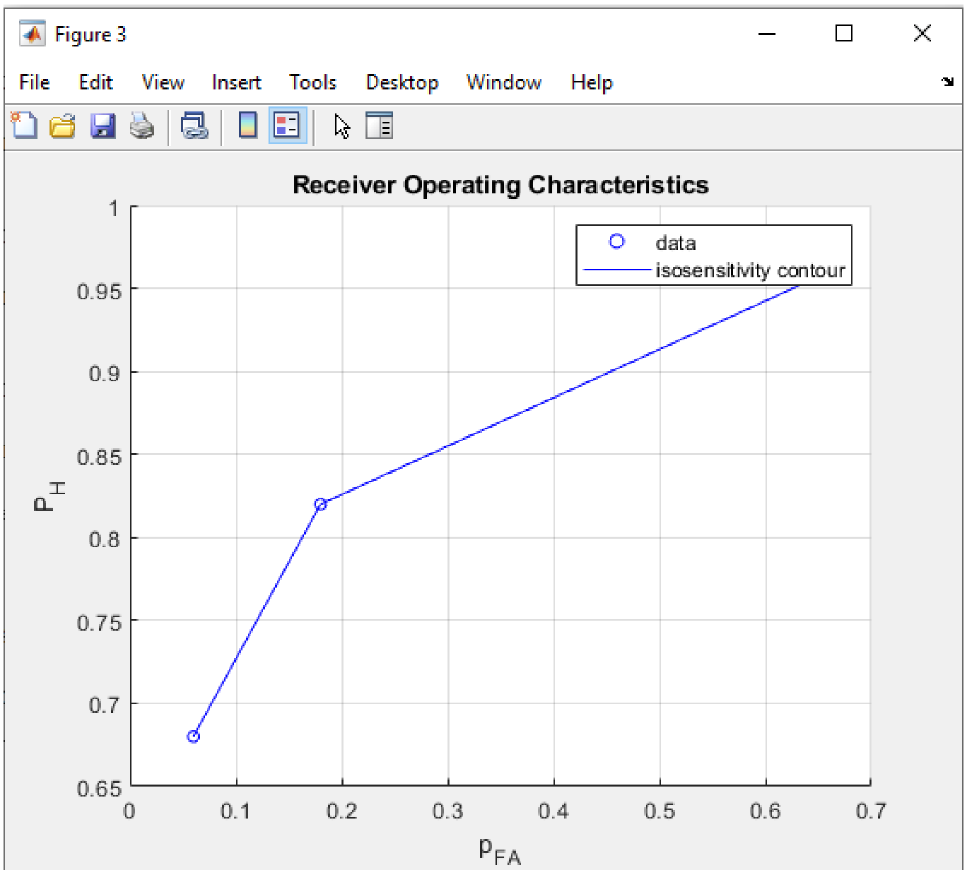
(We keep the d_prime column for reference)

```
Number of trials:100

T3 =

  3×4 table

    Criterions    d_prime    d_approx    d_ROC 
    __________    _______    ________    ______

       -0.5          1        1.3382     1.2507
        0.8          1        1.8307     1.7154
        1.5          1        2.0225     1.9677
```

We can observe that the approximated d' is not constant along the ROC curve, meaning that it varies for different criterion values.

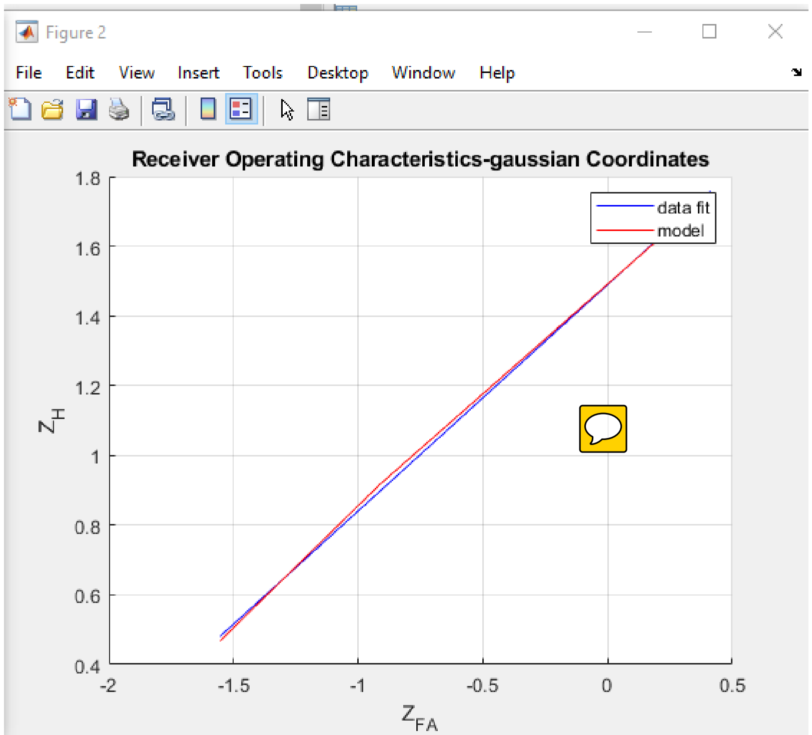
Other values might be used as measure of detectability for normal distributions with unequal variance: delta_m, d_e', d_a or proportion of area under the ROC curve. This last one is particularly handy because its interpretation doesn't depend of the underlying distributions of N and SN and because it doesn't necessarily need curve fitting to be calculated (ROC curves are very fitted to straight lines as can be seen from figure below).

Roc curve:



The slope of ROC curves tells us about the variance of SN and N. If the slope= 1, we fall back in the previous case of equal variance, if the slope is <1 then the variance of the SN is greater of the variance of the noise N.

We plot the ROC in Gaussian coordinates and fit the unequal variance model to the data.



As for the equal variance distributions, we can define:

z_H = (1/ std_SN)* z_FA + (µ_SN/std_SN)

Where z_H and z_FA are the z scores of the probabilities P_H and P_FA.

From our fitting we got the parameters y_fit = P(1)*x + P(2)

Running the code for 100 trials we get:

```
Number of trials:100

T4 =

  1×4 table

    real_mu    approx_mu    real_sigma    approx_sigma
    _______    _________    __________    ____________

    2.0347      2.2962        1.3169         1.5416
```

Which gets even better with 1000 trials

```
Number of trials:1000

T3 =

  3×4 table
```

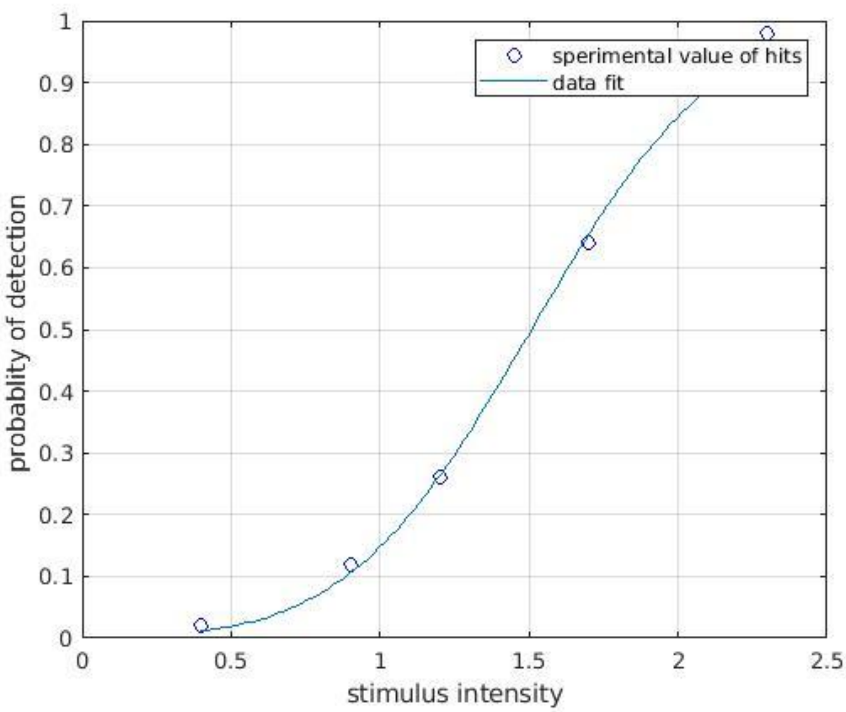
Criterions	d_prime	d_approx	d_ROC
-0.5	1	1.0816	1.0718
0.8	1	1.6549	1.6204
1.5	1	1.8486	1.8266

```
Number of trials:1000

T4 =

  1×4 table
```

real_mu	approx_mu	real_sigma	approx_sigma
1.9838	2.1032	1.51	1.6343

Part 3: Psychometric function

1) Fit the psychometric function to the data using a squared error function. What are the estimates of its parameters?



The above figure illustrates how the estimated model fit the experimental data. The estimated parameters of the cumulative gaussian probability function are:

μ = 1.5075 ; std = 0.4853

In the second part of the experiment inputs are given with intensity 1 or 2. To evaluate the sensitivity of the observer we can calculate the area under the ROC curve, using the cumulative gaussian function that we have defined: Fcb (@ B, x).

We find:

Fcb(B,1) = 0,14 ca

Fcb(B,2) = 0,84 ca

We can say that the observer has a lower sensitivity to the less intense stimulus and a much higher sensitivity to intense stimuli.

Homework 2 – part 1

02458 Cognitive Modelling – Bayes Exercise 1

In this exercise we'll try and solve a perceptual problem using the Bayesian approach. The problem is to infer the 3-dimensional structure of a wireframe from a 2-dimensional projection image. The structure is called S and the image is called I. The Bayesian approach is to maximize the posterior probability P(S|I). For practical reasons we will minimize the negative logarithm of the posterior probability instead.

We'll use Matlab and the first thing you have to do is to get all the Matlab files that comes with the exercise and put them in your active directory.

Question 1

We'll first work with a cube. Open the Matlab function NeckerExercise.m. It is commented so try and read it to see what it does. It can almost solve the problem by numerically maximizing the posterior probability P(S|I). We use Matlab's optimization routine fminunc (*f*unction *min*imization *unc*onstrained). This routine calls the function NeckerError which is nested in the NeckerExercise function. The program is almost ready to run only the NeckerError function is not quite ready. All you have to do is specifying the negative logarithm of the posterior probability in the variable NegLogPost on line 77. Start with a uniform prior probability so that the posterior is simply the likelihood. That won't work, but it'll be illustrative. Then try using some clever prior. If you want you can use the calculation of the angles of the wireframe structure calculated in the variable Angels (a prior for right angles will work well for the cube).

Question 2

Do the same thing with a hexagon cylinder to test whether the prior you came up with for the cube also works for a different structure. Open the Matlab function HexagonalExercise.m. It works just like NeckerExercise.m. Try a flat prior and try the prior you found in 5.1. Why doesn't it work? Can you come up with a prior that works? Is it a realistic model for how the brain works.

Question 3

Evolution gave us two eyes for a reason. So far you've solved the problem using only one eye. Now solve it using two. Do you still need the prior? Explain why/why not.

Bayes Exercise 2

Solve the simple perceptual task facing a monkey in the jungle using a Bayesian approach. Spell out which terms are posteriors, priors, likelihoods, what terms are being discounted and what terms explains away what? You can assume Gaussian distributions throughout.

1. Out in the jungle, 15% of the juju-fruits are ripe. Ripe juju-fruits are orange, on average reflecting light with a wavelength of about 600 nm with some variation (standard deviation of 50 nm). Unripe juju-fruits are green with a wave length of 500 nm (standard deviation 50 nm). What is the probability of a juju-fruit reflecting light with a wavelength between 540-550 nm being ripe?

2. Only 10% of the fruits in the jungle are juju-fruits. 50% are mongo berries. 80% of the mongo berries are ripe. When ripe, mongo berries reflect light with a wavelength of 580 nm (standard deviation 20). When unripe, they reflect light with a wavelength of 520 nm (standard deviation 20). The remaining fruits are all chakavas. Only 10% of the chakavas are ripe. When they are ripe, they reflect light with a wavelength of 400 nm (standard deviation 100). When they are unripe, they reflect light with a wavelength of 550 nm (standard deviation 100). What's the probability that a random fruit reflecting light with a wavelength between 540-550 nm is ripe? How does this result change if you see a monkey enjoying eating the fruit?

3. Simulate fruit picking by drawing a random sample of 1000 fruits using Matlab's random number generator. The probability for *each* fruit being a juju, mongo or chakava; ripe or unripe should be as described in Problem 2. Each fruit should reflect light with specific wavelength. Assume that monkey's can identify the wavelength reflected by a fruit with an accuracy of +/- 5 nm. Simulate a fruit-picking monkey. How good is the monkey at separating ripe from unripe fruit? You can assume that the monkey's visual system has all the information described in Problem 2 and that the monkey uses the maximum posterior decision rule.

Solution to Bayes Exercise 2

Answer to Problem 1

First we should notice the prior information given,

$$P_{prior}(Ripe) = 0.15 \text{ and } P_{prior}(Unripe) = 0.85$$

Then we should note the likelihood of ripe and unripe fruits reflecting light of specific wavelengths. Let's denote the wavelength, *w*, then:

$$P_{like}(540 < w < 550 \mid \text{ripe}) = \Phi\left(\frac{550-600}{50}\right) - \Phi\left(\frac{540-600}{50}\right) = 0.044$$

and

$$P_{like}(540 < w < 550 \mid \text{unripe}) = \Phi\left(\frac{550-500}{50}\right) - \Phi\left(\frac{540-500}{50}\right) = 0.053$$

where

$$\Phi\left(\frac{w-\mu}{\sigma}\right)$$

is the normal cumulative density function (normcdf.m in Matlab) for the variable, *w*, distributed according to a normal distribution with mean, μ, and standard deviation, σ.

Now we're almost ready to calculate the posterior probability of the fruit with a wavelength between 540 and 550 nm is ripe by using Bayes' rule:

$$P_{post}(ripe \mid 540 < w < 550) = \frac{P_{like}(540 < w < 550 \mid ripe)P_{prior}(ripe)}{P(540 < w < 550)}$$

Note that the denominator is the overall, or unconditional probability of jujus (ripe or unripe) reflecting a wavelength between 540 and 55 nm. It can be calculated as a weighted sum:

$$\sum_{ripe} P_{like}(540 < w < 550 \mid ripe)P_{prior}(ripe) =$$
$$P_{like}(540 < w < 550 \mid ripe)P_{prior}(ripe) + P_{like}(540 < w < 550 \mid unripe)P_{prior}(unripe)$$

so that finally we arrive at

$$P_{post}(ripe \mid 540 < w < 550) =$$

$$\frac{P_{like}(540 < w < 550 \mid ripe) P_{prior}(ripe)}{P_{like}(540 < w < 550 \mid ripe) P_{prior}(ripe) + P_{like}(540 < w < 550 \mid unripe) P_{prior}(unripe)} \approx$$

$$\frac{0.044 \times 0.15}{0.044 \times 0.15 + 0.053 \times 0.85} \approx 0.1263$$

Answer to Problem 2

Again we need to calculate the posterior probability

$$P_{post}(ripe \mid 540 < w < 550)$$

but now we have three kinds of fruit and we only have information about prior probabilities and likelihoods specific to each fruit. The key to the solution is to realize that the type of fruit is irrelevant and needs to be *discounted*:

$$P_{post}(ripe \mid 540 < w < 550) = \sum_{fruit} P_{post}(ripe, fruit \mid 540 < w < 550)$$

We can then insert Bayes' rule

$$P_{post}(ripe, fruit \mid 540 < w < 550) =$$

$$\frac{P_{like}(540 < w < 550 \mid ripe, fruit) P_{prior}(ripe, fruit)}{P(540 < w < 550)} =$$

$$\frac{P_{like}(540 < w < 550 \mid ripe, fruit) P_{prior}(ripe, fruit)}{\sum_{fruit, ripe} P_{like}(540 < w < 550 \mid ripe, fruit) P_{prior}(ripe, fruit)}$$

The joint prior factorize so that $P_{prior}(ripe, fruit) = P_{prior}(ripe \mid \text{fruit}) P_{prior}(fruit)$. Now you can insert everything in one big equation:

$$P_{post}(ripe \mid 540 < w < 550) =$$

$$\sum_{fruit} P_{post}(ripe, fruit \mid 540 < w < 550)$$

$$\sum_{fruit} \frac{P_{like}(540 < w < 550 \mid ripe, fruit) P_{prior}(ripe \mid \text{fruit}) P_{prior}(fruit)}{\sum_{fruit, ripe} P_{like}(540 < w < 550 \mid ripe, fruit) P_{prior}(ripe \mid \text{fruit}) P_{prior}(fruit)} \approx 0.402$$

So, chances are that the fruit is not ripe (<50% chance) and I wouldn't bother to pick it. If, however, I saw a monkey enjoying the fruit, I would think that the fruit was probably ripe. The monkey enjoying the fruit thus *explains away* the possibility of the fruit being unripe.

Answer to Problem 3

Simulating fruiting picking by sampling from a multinomial distribution (see the solution code for one way to do this). The sample gives you the number of jujus, mongos and chakavas as well as the number of them that were ripe. Now that you have the fruits you should generate a wavelength for each and every fruit by sampling from the likelihood function specific to that fruit and its ripeness:

$$P(\mathrm{w} \mid ripe, fruit) = \varphi\left((w - \mu)/\sigma\right)$$

Here, $\varphi\left((w - \mu)/\sigma\right)$, is the normal probability density. Now that you have 1000 wavelengths, you can determine the monkey's posterior probability for each fruit being a juju, mongo or chakava; ripe or unripe. This is exactly what you did in Problem 2. Now you just have to do it for many different wavelength intervals and not just for 540-550 nm. Finally let the monkey make a decision using the *max aposteriori* rule and count the number of times the monkey was right. A typical answer could look like this:

number of ripe jujus: 14
number of ripe jujus picked: 7
number of unripe jujus: 89
number of unripe jujus picked: 15
number of ripe mongos: 373
number of ripe mongos picked: 340
number of unripe mongos: 110
number of unripe mongos picked: 9
number of ripe chakavas: 42
number of ripe chakavas picked: 8
number of unripe chakavas: 372
number of unripe chakavas picked: 119

Note that the monkey makes a lot of mistakes picking unripe chakavas. It's because they look like ripe mongos.

Bayes Exercise 3

An observer with d' = 1.5 does a signal detection task in three different conditions.

- In Condition 1 50% of the trials contain a signal
  In Condition 2 95% of the trials contain a signal
  In Condition 3 15% of the trials contain a signal

Assume equal variance SDT model and maximum a posteriori decision rule. What value for the criterion do we expect in each condition?

We start out with the MAP decision rule

$$P(signal\ |\text{x}) > P(noise\ |\text{x})$$

Insert Bayes to introduce the prior

$$\frac{P(x\ |\text{signal})P(signal)}{P(x)} > \frac{P(x\ |\text{noise})P(noise)}{P(x)}$$

Use $P(signal) = 1 - P(noise)$ and rearrange

$$\frac{P(x\ |\text{signal})}{P(x\ |\text{noise})} > \frac{1 - P(signal)}{P(signal)}$$

Insert Gaussian probability density function, remember that $\sigma = 1$, take natural logarithm and rearrange

$$x^2 - (x - d')^2 > ln\left(\frac{1 - P(signal)}{P(signal)}\right)$$

Isolate x

$$x > \frac{ln\left(\frac{1 - P(signal)}{P(signal)}\right) + \frac{d'^2}{2}}{d'} = \frac{ln\left(\frac{1 - P(signal)}{P(signal)}\right)}{d'} + \frac{d'}{2}$$

02458 Cognitive modelling E19
2nd Assignment

Adrian Atienza (s192780)
Arianna Taormina (arita@fysik.dtu.dk)
Marie Claire Capolei (macca@elektro.dtu.dk)
Maud Leclerc (s191975)

Bayes Exercise 1

Bayesian approach in the study of perception is based on some fundamental concepts as the likelihood function and the prior.

Re-constructing a 3D image with 2D information is an ill-posed problem, we just do not have enough information to get to a unique solution: there are infinite number of possible interpretations. That's the case of Necker Cube, the cube perception is highly influenced by the prospective, e.g. from-above-prospective (fap) and from-below-prospective (fbp), resulting in an unstable perception.

Although the unstable perception scenario, the human visual system experiences only two possible interpretation as compatible with the sensory information, over an infinite range. The visual system does not overlap the two interpretations, but it alternates the two prospective fap and fbp, in a reversal process determined by the observer internal fluctuations.

Likelihood and priors in the Bayesian framework give us a good interpretation of why this possibly happens.

The Likelihood captures how likely a particular scene corresponds to the perceived image, or rather, it represents how likely is an assumption to be true.

Let B be the set of observed outcomes, and A the set of parameters describing the stochastic process. The likelihood is conditioned by the B observation and is function of the A unknown parameters that need to maximize the probability of having the observation,

$L(A|B) = P(B|$

In fact, we proceed by maximizing the likelihood (aka minimizing the minus log likelihood; this is mainly to move the problem from maximizing the result of products to minimizing the result of sums, its more handy).

The other important factor is given by Priors: those are distributions that represent the observer prior believes (of which the observer is not necessarily aware of) before some evidences are taken into account, and provide a "principled way of formulating constraints on possible scenes that lead to unambiguous visual perception".   Priors represent what we think is more plausible in the real word sort of "independently" from the current sensory information (this is not completely true as we will specify soon).

Bayes theorem:

- $P(A|B)*P(B) = P(B|A)*P(A)$
- $P(A|B) = P(B|A)*P(A) * (1/P(B))$

Where

- $P(B|A)$ = Likelihood (Not a probability distributio

- P(A) = Prior, characterized by bias and confidence in the bias. If we model the prior in the form of normal distributions, then bias = mean of the distribution and the confidence in inversely proportional to the std standard deviation. Which mean and which std to use then? The mean can be given by the average perception, but confidence levels estimation, aka std, is trickier. Confidence in fact can be tested and constructed by changing the condition of the stimulus - lightening or orientation, e.g. increasing the contrast in an image decreases the standard deviation of the Prior that encompass the luminance effect, which then is more stable.
  Prior are not fixed entities but are affected by the stimulus characteristics.

- (1/P(B)) = corrective term that turns the Likelihood into a probability distribution, it is not necessary, and it can be omitted.

- P(A|B) = is the posterior distribution that considers both geometrical knowledge and prior weights. If we don't have a strong prior, then our final output will be mainly influenced by the likelihood, or rather, the observations.

1) Question 1:

Let $\mathbf{S}_{3x8}$ be the matrix describing the 3-dimensional structure of the real cube wireframe, and $\mathbf{I}_{3x8}$ the matrix describing the projection of **S** on the 2D plane given the $\mathbf{M}_{4x4}$ orthographic transformation matrix. **M** is function of the observer's α_{el} elevation and α_{az} azimuth angles.
We first estimate the $\mathbf{S}_{hat}$ observer's perception of the cube by minimizing the likelihood error of the estimate image,

$$\text{Argmin}[\text{ WSSE}] = \text{Argmin}[\text{ std}_N^{-2} \bullet (\mathbf{I} - \mathbf{I}_{hat})^2],$$

where WSSE stand for weighted sum of squared errors, although we are here considering only the likelihood error. $\mathbf{I}_{hat}$ is the projection of the $\mathbf{S}_{hat}$ observer's intuitive estimate on the 2D plane,

$$\mathbf{I}_{hat} = \mathbf{M}\ \mathbf{S}_{hat}\ ,$$

And std_N is the standard deviation of the noise affecting the visual perception.
Considering the observer's position respect to the cube is,

$$\alpha_{el} = 25°\ ,\ \ \alpha_{az} = -32°\ ,$$

And the noise is normally distributed with standard deviation,

$$\text{std}_N = 2,$$

The result is shown in **Fig.1**, from the observer's position (left cube) the perceived image (in blue) tend to resemble a cube, but if we change prospective the wireframe results mashed. The Likelihood only is not enough to predict the 3D scene. It gives in fact only a good prediction of the 2D image projected on our field of vision, which inevitably breaks the moment we move around the viewpoint. Priors on the consistency of volumes, lengths, angles or luminosity can bridge the gap of information. When two Priors are in discordance, we have the switching from one interpretation to the other.

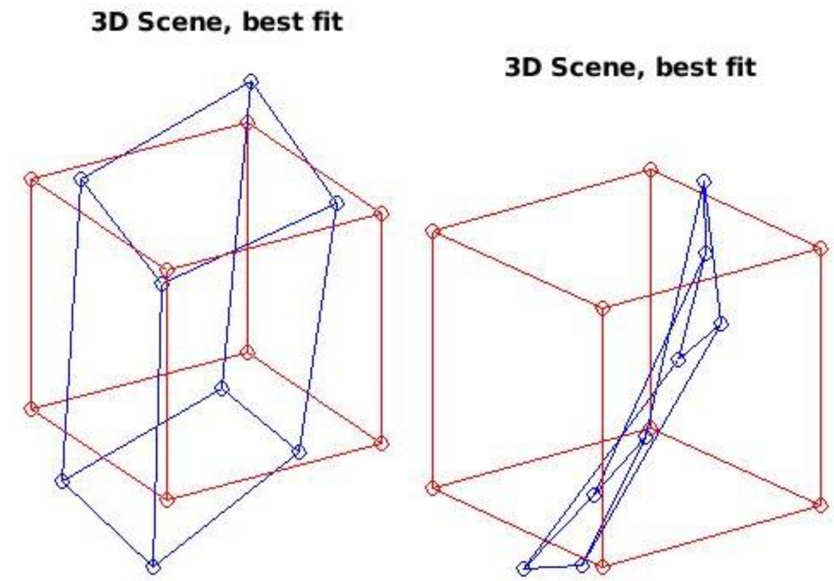


Figure 1. Necker cube perception from different point of view considering only the likelihood error. In red the real cube, in blue the perceived.

We extend the WSSE weighted sum of squared errors used for the optimization of the estimate, including first the information regarding the angles,

$$\text{Argmin}[\ \text{WSSE}] = \text{Argmin}[\ std_N^{-2} \bullet (\mathbf{I} - \mathbf{I}_{hat})^2 + std_{s1}^{-2} \bullet (\mathbf{S}_1 - \mathbf{S}_{1,hat})^2],$$

Where $\mathbf{S}_1$ is a matrix of rectangular angles, and $\mathbf{S}_{1,hat}$ is the matrix describing the estimated angle, computed from the $\mathbf{S}_{hat}$ matrix.

Given the same prospective as in the previous test case, and perception noise,

$$std_{s1} = 2,$$

The outcome is apparently improved (**Fig.2**), however it results not robust to changes in the observation angles (right cube).

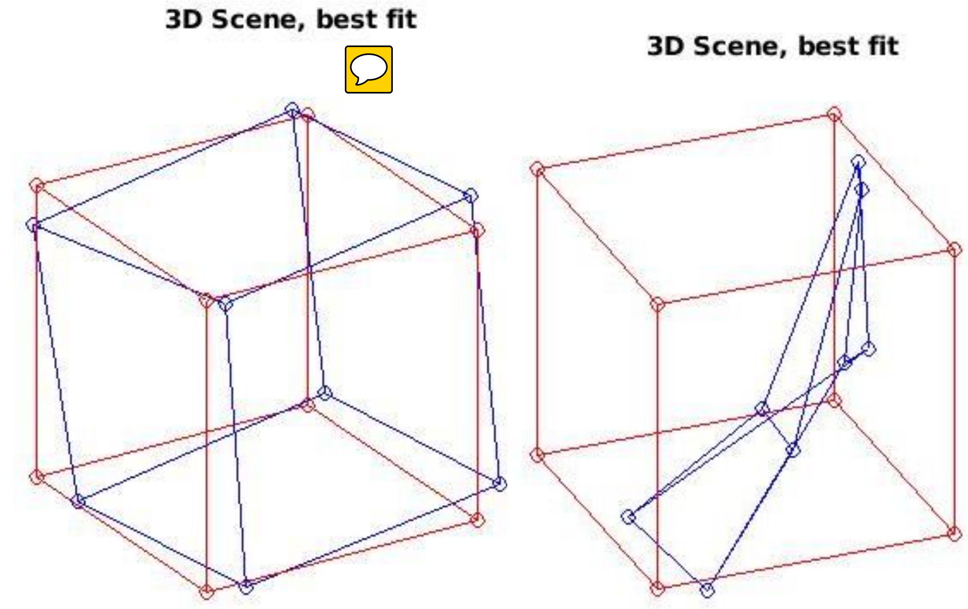


Figure 2. Necker cube perception from different point of view considering the likelihood and the perceived angles error. In red the real cube, in blue the perceived.

We consider an extra features in the estimation: the lengths of the cube,

$$\mathrm{Argmin}[\ \mathrm{WSSE}] = \mathrm{Argmin}[\ \mathrm{std}_N^{-2} \bullet (\mathbf{I} - \mathbf{I}_{hat})^2 + \mathrm{std}_{s1}^{-2} \bullet (\mathbf{S}_1 - \mathbf{S}_{1,hat})^2 + \mathrm{std}_{s2}^{-2} \bullet (\mathbf{S}_2 - \mathbf{S}_{2,hat})^2\ ],$$

Where $\mathbf{S}_2$ is a matrix of real lengths, and $\mathbf{S}_{2,hat}$ is the matrix describing the estimated lengths, computed from the $\mathbf{S}_{hat}$ matrix. The perception results highly robust to varying observer's position thanks to the larger amount of prior information exploited during the perception process (**Fig.3**).

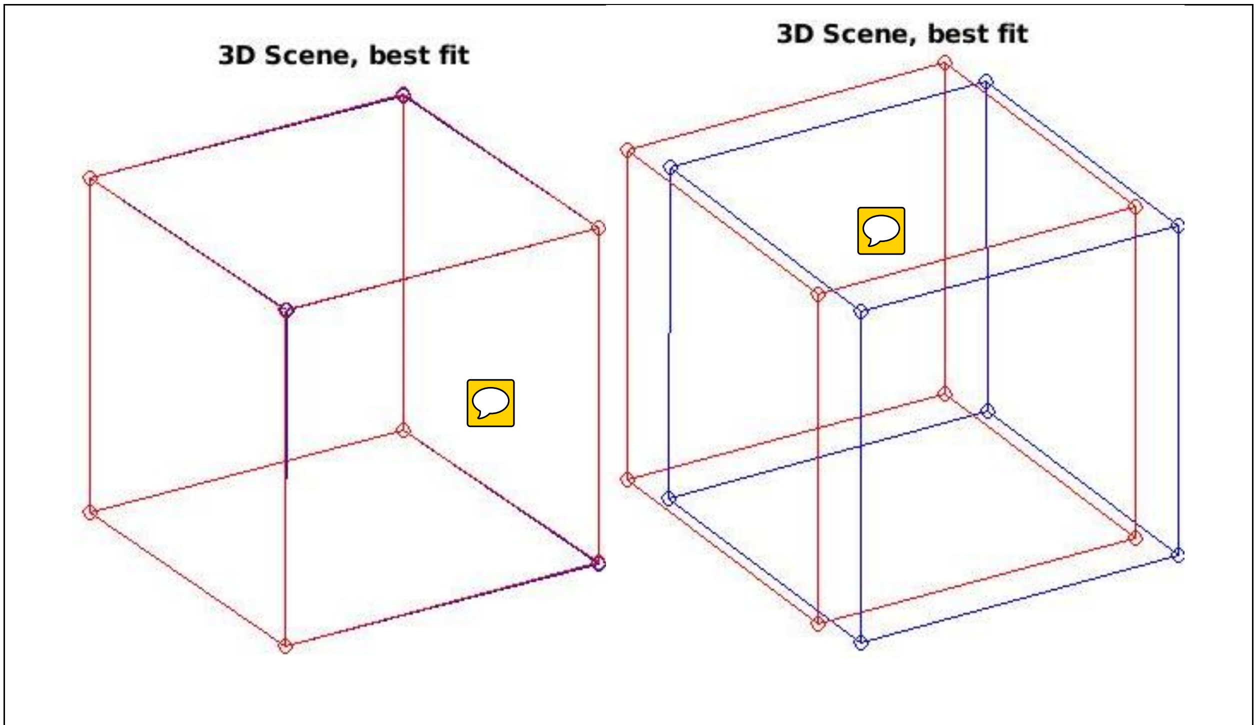


Figure 3. Necker cube perception from different point of view considering the likelihood, the perceived angles error, and the lenght. In red the real cube, in blue the perceived.

other comments?

2) Question 2:

With only 8 vertexes out of 12 , don't get enough information from the Priors to have a good prediction. Modifying the code with the number of angles and lengths to 12 solved the issue.

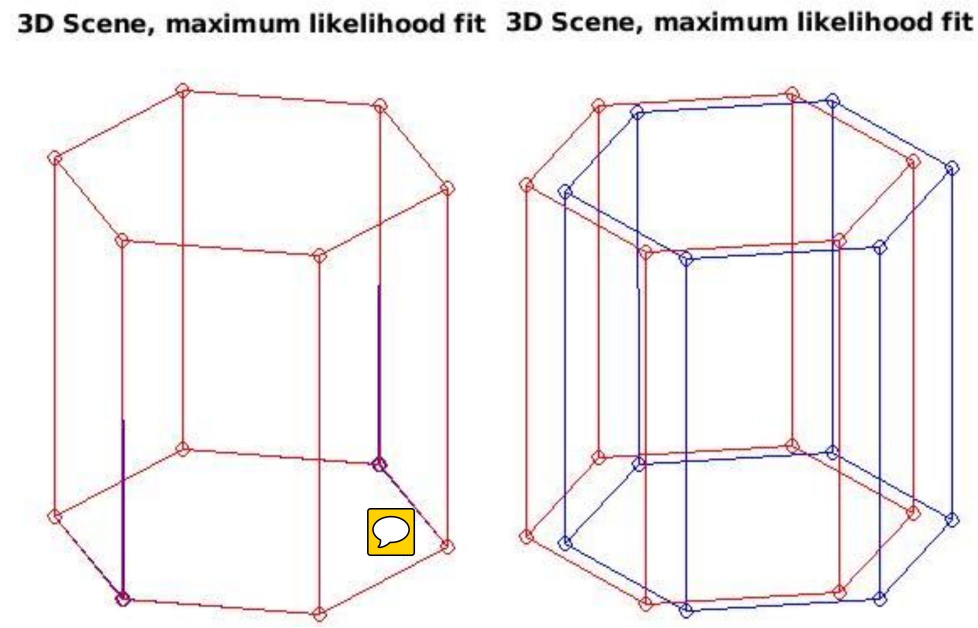


Figure 3. Hexagonal prism perception from different point of view considering the likelihood, the perceived angles error, and the lenght. In red the real prism, in blue the prism.

3) Question 3:

Binocular vision allows us to perceive a single 3D image of our surroundings, gives a wide field of vision (ca 190 of which 120 for both eyes) and, very important, the different position of the eyes in the head (defined as parallax or binocular disparity) allows our brain to get to a precise depth perception via a process called stereopsis. In the last model we have once again the real 3D Scene (S), with the real projection on 2D planes of vision of each eye (I1 and I2). Here we have hp same elevation (EL1 = EL2) for both and a different azimuth angle (AZ1 = AZ2-60 degrees). We calculate the minimum of NegLogLikelihood given by the sum of MSE for each eye (S_guess as a contribution from both viewpoints and therefore the projections on the two different planes I_guess_1 and I_guess_2).

We see that, also without any Prior on volume, lengths, angles o light, we get a perfect estimate of the original 3D scene.

Bayes Exercise 2

4) Question 1:

Ripe juju-fruits are orange with a wavelength of

$$Wave_{Juju,ripe} = 600 \pm 50 \text{ nm},$$

while not ripe juju fruits are in a range of

$$Wave_{Juju,\sim ripe} = 500 \pm 50 \text{ nm},$$

We can describe this information with two normal distribution respectively,

$$N_{w,Juju,ripe}(600,50^2),$$
$$N_{w,Juju,\sim ripe}(500,50^2).$$

The P(Juju,ripe) prior probability of juju-fruits to be ripe is,

$$P(Juju,ripe) = 0.15,$$

accordingly, the P(Juju,~ripe) prior probability of juju-fruits to not be ripe is,

$$P(Juju,\sim ripe) = 1 - P(Juju,ripe) = 0.85.$$

The posterior probability of a juju-fruit to be ripe given a wavelength between 540-550 nm is low,

$$P(Juju,ripe\ |540\text{-}550) = \beta^{-1} \bullet P(540\text{-}550|\ Juju,ripe) \bullet P(Juju,ripe) = \mathbf{0.12}$$

Where β is the ratio,

$$\beta = (P(540\text{-}550|\ Juju,ripe) \bullet P(Juju,ripe) + P(540\text{-}550|\ Juju,\sim ripe) \bullet P(Juju,\sim ripe)$$

and P(540-550| Juju,ripe) is the likelihood of having that wavelength interval if the juju-fruit is ripe, or rather,

$$P(540\text{-}550|\ N_{w,Juju,ripe}(600,50^2)) = \mathbf{L}\,(N_{w,Juju,ripe}(600,50^2)\ |550) - \mathbf{L}\,(N_{w,Juju,ripe}(600,50^2)\ |540),$$

Same argument is valid for the P(540-550| Juju,~ripe) likelihood of not being ripe given that wavelength interval.

This low probability is highly compromised by the low prior P(Juju,ripe) to have ripe juju-fruit, for instance, if we change the prior to,

$$P(Juju,ripe) = 0.85,$$

The posterior probability of having a ripe juju-fruit given that wavelength range drastically increase,

$$P(Juju,ripe\ |540\text{-}550) = 0.8228.$$

5) Question 2:

The jungle offers to the monkeys several type of fruits, statistics show that among all the edible fruits only a small percentage are juju-fruits, or rather the prior of being a juju-fruit is,

$$P(\text{Juju}) = 0.10,$$

While half of the fruits are mongo berries,

$$P(\text{mongo berries}) = 0.50,$$

and the remaining are chakavas,

$$P(\text{chakavas}) = 1 - P(\text{Juju}) - P(\text{mongo berries}) = 0.40.$$

The reflected light of ripe and not ripe mongo berries and chakavas can describe as normal distribution,

$$N_{w,mongo,ripe}\,(580,20^2),$$
$$N_{w,mongo,\sim ripe}\,(520,20^2),$$

for the mongo berries, and

$$N_{w,chakavas,ripe}\,(400,100^2),$$
$$N_{w,\,chakavas,\sim ripe}\,(550,100^2),$$

for the chakavas respectively. The P(mongos,ripe) and P(chakavas,ripe) prior probability of mongos berries and chakavas to be ripe are,

$$P(\text{mongos,ripe}) = 0.80,$$

$$P(\text{chakavas,ripe}) = 0.10,$$

accordingly, the P(mongos,~ripe), P(chakavas,~ripe) prior probability of mongos berries and chakavas to not be ripe is,

$$P(\text{mongos,}\sim\text{ripe}) = 1 - P(\text{mongos,ripe}) = 0.20,$$

$$P(\text{chakavas,}\sim\text{ripe}) = 1 - P(\text{chakavas,ripe}) = 0.90,$$

The probability that a random fruit is ripe given a reflecting wavelength between 540-550 is,

$$P(\text{ripe}|540\text{-}550) = \beta^{-1} \bullet \alpha = \mathbf{0.4018},$$

Where α is the sum of the likelihoods multiplied the priors of being ripe for each fruit, times the probability of existence of that type of fruit,

$$\alpha = \alpha_1 \bullet P(\text{Juju}) + \alpha_2 \bullet P(\text{mongos}) + \alpha_3 \bullet P(\text{chakavas})$$

$$\alpha_1 = P(540\text{-}550|\ \text{Juju,ripe}) \bullet P(\text{Juju,ripe}),$$

$$\alpha_2 = P(540\text{-}550|\ \text{mongos,ripe}) \bullet P(\text{mongos,ripe}),$$

$$\alpha_3 = P(540\text{-}550|\ \text{chakavas,ripe}) \bullet P(\text{chakavas,ripe}),$$

and β is the ratio considering both the probability of being or not ripe,

$$\beta = \beta_1 \bullet P(Juju) + \beta_2 \bullet P(mongos) + \beta_3 \bullet P(chakavas)$$

$$\beta_1 = P(540\text{-}550 |\ Juju,ripe) \bullet P(Juju,ripe) + P(540\text{-}550 |\ Juju,\sim ripe) \bullet P(Juju,\sim ripe),$$

$$\beta_2 = P(540\text{-}550 |\ mongos,ripe) \bullet P(mongos,ripe) + P(540\text{-}550 |\ mongos,\sim ripe) \bullet P(mongos,\sim ripe),$$

$$\beta_3 = P(540\text{-}550 |\ chakavas,ripe) \bullet P(chakavas,ripe) + P(540\text{-}550 |\ chakavas,\sim ripe) \bullet P(chakavas,\sim ripe).$$

The overall probability of having a ripe fruit is largely influenced by the juju-fruits and chakavas prior of being ripe, together they represent the 50% of the fruit available in the jungle and their prior probability of being ripe is incredibly low, moreover the considered range can be out of the ripe wavelength for the mongos berries.
The event considering that the monkey is enjoying eating the fruit could be seeing as a signal that the fruit is ripe increasing the prior of that specific fruit, and consequentially influencing the future observations.

Should we write something else?

6) Question 3:

We tested the picking capabilities of the monkey per 1000 trials.

We first initialized a multinomial probability distribution including all the possible scenario of P(fruit type, ripe) and P(fruit type, ~ripe),

$$\mathbf{Probabilities}_{fruit} = [P(Juju,ripe) \bullet P(Juju)\ ,\ P(Juju,\sim ripe) \bullet P(Juju)\ ,\ P(mongos,ripe) \bullet P(mongos)\ ,\ P(mongos,\sim ripe) \bullet P(mongos),\ P(chakavas,ripe) \bullet P(chakavas)\ ,\ P(chakavas,\sim ripe) \bullet P(chakavas)\ ]$$

We used this distribution to create a vector $\mathbf{Fruits}_{1xtrials}$ of trials = 1000 random numbers corresponding to the ripe or not ripe fruits.

Per 1000 times we picked a value from the vector (fruit,(not) ripe) and the corresponding m_i mean and std_i standard deviation of the *i-th* wavelengths distribution, we used this value to calculate the random interval,

$$\mathbf{interval}_i = [x-5,x+5]$$

that will be used in to compute the likelihoods,

$$P(\mathbf{interval}_i \mid N_{w,fruit,ripe}(m_i, std_i^{\ 2})),$$

$$P(\mathbf{interval}_i \mid N_{w,fruit,\sim ripe}(m_i, std_i^{\ 2})).$$

As in question 2, for the I-th trial the probability that the I-th fruit is ripe given a reflecting wavelength between the interval $\mathbf{interval}_i$ is,

$$P(ripe |\ \mathbf{interval}_i) = \beta^{-1} \bullet \alpha\ ,$$

Where α is the sum of the likelihoods multiplied the priors of being ripe for each fruit, times the probability of existence of that type of fruit,

$$\alpha = \alpha_1 \bullet P(\text{Juju}) + \alpha_2 \bullet P(\text{mongos}) + \alpha_3 \bullet P(\text{chakavas})$$

$$\alpha_1 = P(\textbf{interval}_i \mid \text{Juju,ripe}) \bullet P(\text{Juju,ripe}),$$

$$\alpha_2 = P(\textbf{interval}_i \mid \text{mongos,ripe}) \bullet P(\text{mongos,ripe}),$$

$$\alpha_3 = P(\textbf{interval}_i \mid \text{chakavas,ripe}) \bullet P(\text{chakavas,ripe}),$$

and β is the ratio considering both the probability of being or not ripe,

$$\beta = \beta_1 \bullet P(\text{Juju}) + \beta_2 \bullet P(\text{mongos}) + \beta_3 \bullet P(\text{chakavas})$$

$$\beta_1 = P(\textbf{interval}_i \mid \text{Juju,ripe}) \bullet P(\text{Juju,ripe}) + P(\textbf{interval}_i \mid \text{Juju,\~{}ripe}) \bullet P(\text{Juju,\~{}ripe}),$$

$$\beta_2 = P(\textbf{interval}_i \mid \text{mongos,ripe}) \bullet P(\text{mongos,ripe}) + P(\textbf{interval}_i \mid \text{mongos,\~{}ripe}) \bullet P(\text{mongos,\~{}ripe}),$$

$$\beta_3 = P(\textbf{interval}_i \mid \text{chakavas,ripe}) \bullet P(\text{chakavas,ripe}) + P(\textbf{interval}_i \mid \text{chakavas,\~{}ripe}) \bullet P(\text{chakavas,\~{}ripe}).$$

If the I-th $P(\text{ripe} \mid \textbf{interval}_i)$ probability is higher than 0.5, so is ripe, and the picked fruit is actually ripe than we increase a counter. The accuracy of the monkey will be,

Accuracy = sum of correct response / trials = 0.8010

Exercise 3

See code

02458 Cognitive Modeling

PCA/ICA – Homework 3.1

Load an image into Matlab using the imread.m function. It can be the image of Mona Lisa that you find in the exercise folder or it can be any image that you choose. It is important that the image has thousands of pixels and that it is black and white. (You can transform it to gray scale either by using the rgb2gray.m function in Matlab's image processing toolbox or by simply extracting one of the three color channels in the image). We'll call this image matrix ***I***.

Now transform the image to a set of patches each 10 by 10 pixels. Transform the patches to 100-dimensional vectors (e.g. using reshape.m) and catenate them to form a matrix, ***S***, where each row is a patch and each column is a pixel.

Perform PCA on the transformed matrix using.

It can return the principal components (also called the *loadings* or *coefficients*) in a 100-by-100 matrix, ***X***. The principal components are the columns of ***X***. It can also return the transformed data, ***W*** (also known as *scores*), which has the same size as ***S***.

Now, XW^T, is a reconstruction of ***S***, except for the scaling. Test this by transforming XW^T back into image-space and use imagesc.m to display the image. For comparison display the original image matrix, ***I***, in another figure. The two images should look identical. Also check that the difference between the original image and the reconstruction is negligible.

The princomp.m function can also return the amount of variance each principal component contains (also known as the *eigenvalues*). Have a look at the eigenvalues. How many eigenvalues does it take to explain 95% of the total variance?

The columns of ***X*** are the principal components ordered according to how much variance they contain. Take some (like 6) of the first ones and transform them back to image space. Scale the resulting images on the *same scale* and display them using the image.m (not imagesc.m) function. What do they look like?

Now reconstruct ***S*** from XW^T using only the first six principal components. Transform ***S*** back to image-space and display the image using imagesc.m. What does it look like? How does it compare to the original image?

02458 Modelling Cognition / ICA exercise

PCA/ICA – Homework 3.2

Last time we saw how we could analyze the natural statistics of images using Principal Component Analysis to obtain a more efficient internal representation. This was entirely based on Gaussian statistics and you may have wondered if that was a reasonable assumption. In fact, when we look at natural statistics, we see that the distributions are characterized by higher moments than just the variance, which means that the Gaussian assumption is not a good one.

1. Let's have a look at what happens when do PCA on a bivariate distribution with excess kurtosis. First, pull 2 times 10000 samples from a Laplacian distribution using the randlpl.m function that comes with the exercise. Mix the two distributions, i.e. create two linear combinations of them. Plot the linear combinations against each other in a 2D plot.

2. Now perform PCA on the joint distribution. Plot the score in a 2D plot. Did PCA separate the two distributions?

3. Now get the fastICA toolbox (http://research.ics.tkk.fi/ica/fastica/) and perform ICA (score=fastica(data'))on the data. Be careful to flip your matrix the right way. Plot the ICA score in a 2D plot to see that the distribution now appear independent.

4. Compare the ICA score with the original samples. Are they identical? Why not? What can go wrong in recovering the original data?

5. Take two mono sound files with an equal amount of samples. You can use the guitar sound files in the file-sharing folder or you can use your own. Create two mixes of the sound files. Separate them using ICA.

    a. Plot (part of) the time series of the original sounds and of the reconstructed sounds. Estimate the error. Evaluate how well it worked.
    b. Listen to the sounds. Evaluate how well it worked based on your listening experience.
    c. Compare the two ways of evaluating whether ICA managed to separate the sounds. Why might the two methods arrive at different results?

02458 Cognitive Modelling

Bayesian Multisensory integration – Homework 3.3

Introduction

We are generally better at a perceptual task if we can use more than one sensory modality to perform it. One example is spatial localisation. We are much better at localising a sound if it is accompanied by a visual stimulus. This is due to a process of cross-modal perceptual integration, which is illustrated by the ventriloquist effect: it appears that a voice comes out of the dummy. The voice obviously comes out of the ventriloquist but due to the coordination of the dummy's mouth and the voice we experience an illusory displacement of the origin of the sound.

The strong fusion model has been very influential in modelling this and other audio-visual illusions. The model is based on Bayes' rule and some simplifying assumptions.

Let's denote the true azimuthal angle of the stimulus as S. As in signal detection theory and psychophysics we assume that the azimuthal angle is represented by an internal representation value, x_A. The internal representation is influenced by Gaussian noise centred on the true value, $x_A \sim \varphi(x_A|\mu_A, \sigma_A)$. When there is no visual stimulus and the observer performs the task based on hearing only. In terms of Bayes' rule the observer's response, R, reflects an estimate of the azimuthal angle, S, based on the internal representation value, x_A, which in the case of an uninformative prior can be written as:

$$P(R) = P(S|x_A) = \frac{P(x_A|S)P(S)}{P(x_A)} = \varphi(x_A|\mu_A, \sigma_A)$$

Similarly, when there is no auditory stimulus and the observer performs the task based only on a visual stimulus, the observer's estimate of S is based on an internal representation value, x_V.

When the observer is presented with both an auditory stimulus and a visual stimulus strong fusion asserts that the observer makes only a single estimate of the stimulus based on the auditory and visual internal representation values

$$P(R) = P(S|x_A, x_V) = \frac{P(x_A, x_V|S)P(S)}{P(x_A, x_V)}$$

A reasonable assumption often used in strong fusion models is that the Gaussian noise in the auditory system is independent from the Gaussian noise in the visual system. If we again assume that the observer as no prior knowledge of the location of the stimulus, the strong fusion model becomes

$$P(R) = P(S|x_A, x_V) = \frac{P(x_A|S)P(x_V|S)}{P(x_A, x_V)} = \frac{\varphi(x_A|\mu_A, \sigma_A)\varphi(x_V|\mu_V, \sigma_V)}{P(x_A, x_V)}$$

The next assumption is that the internal representation values are really amodal, meaning that the observer cannot tell if an internal representation value was caused by a visual stimulus or an auditory stimulus; x_A and x_V are just two values of x, hence

$$\frac{\varphi(x_A|\mu_A, \sigma_A)\varphi(x_V|\mu_V, \sigma_V)}{P(x_A, x_V)} = \frac{\varphi(x|\mu_A, \sigma_A)\varphi(x|\mu_V, \sigma_V)}{P(x)}$$

It turns out that the normalised product of two Gaussians is a Gaussian so that

$$\frac{\varphi(x|\mu_A, \sigma_A)\varphi(x|\mu_V, \sigma_V)}{P(x)} = \varphi(x|\mu_{AV}, \sigma_{AV})$$

where

$$\mu_{AV} = w\mu_A + (1 - w)\mu_V$$

$$w = \frac{\sigma_V^2}{\sigma_A^2 + \sigma_V^2}$$

$$\sigma_{AV}^2 = \frac{\sigma_A^2\sigma_V^2}{\sigma_A^2 + \sigma_V^2}$$

Problems

An observer has completed 50 trials in an auditory, a visual and an audio-visual spatial localisation task. The data is in files xA, xV and xAV. Fit the strong fusion model to the data. Remember to use all the data to estimate the parameters.

What are your estimates of the free parameters?

What is the negative log likelihood of your fit?

Plot the estimated auditory, visual and audio-visual probability densities and the normalised histograms. Compare the probability densities and the histograms to assess the goodness-of-fit.

Another observer has completed the same experiment. This mind of this observer tends to wander. In 4% of the trials the observer zones out, forgets the task and guess more or less at random. In the remaining trials the observer behaves strictly according to the strong fusion model. How does the zoning out affect the goodness-of-fit?

02458 Cognitive modelling E19
3rd Assignment

Adrian Atienza (s192780)

Arianna Taormina (arita@fysik.dtu.dk)

Marie Claire Capolei (macca@elektro.dtu.dk)

¶

Homework 1 – Principal components analysis PCA

We are asked to verify how applying PCA is useful to reduce multidimensional data (in this case an image) to lower dimensions while retaining most of the information (variance) of the data.

In Figure 1.1 A we can see that the original data Covariance is linear and positive, which means that there is probably a strong relation between variables.

We use PCA to try convert observations of possibly correlated variables into a set of values of linearly uncorrelated variables, the principal components. The result applying PCA is visible in Figure 1.1 B

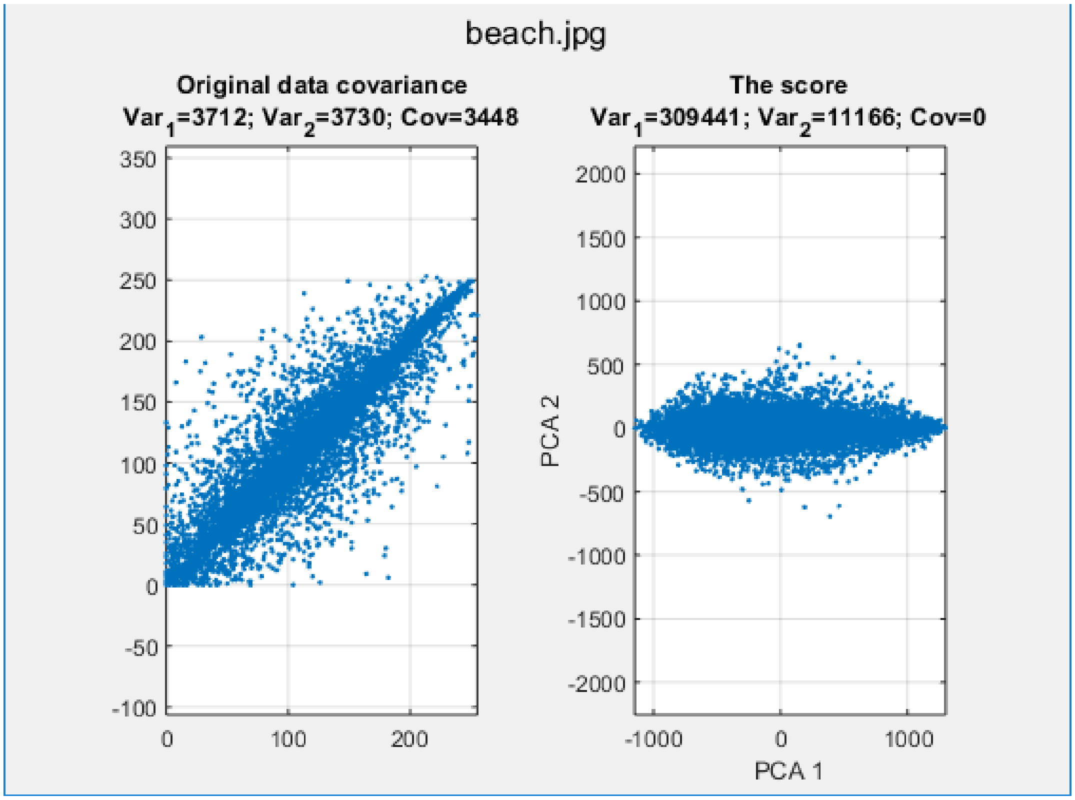


Figure 1.1. A and B : Covariance of the data without and with PCA

We start by fixing the dimensions of the chosen image,

I = image matrix, of shape (1000,1000),

We divide the image in 10000 patches of shape (10x10) and transform each patch in a vector (1x100) Each one of the 10000 vectors in stacked in a transformed matrix S,

S = transformed matrix, of shape (10000,100),

We apply PCA on the transformed Matrix,

[X,W,latent]= pca(S);

Getting:

- X = coefficients matrix, each column contains the coefficients for one principal components, in descending order of component variance. Eigenvectors of covariance matrix.
- W = scores, representations of the data rotated to the new basis of the principal component space.
- Latent = eigenvalues of covariance matrix in descending order of variance.

Exercise 1 – Verify that X*W' is a reconstruction of S

Keeping all the components, we try to reconstruct the original image.

```
n_pca = 100;
I_pca = W(:,1:n_pca)*X(:,1:n_pca)';
```

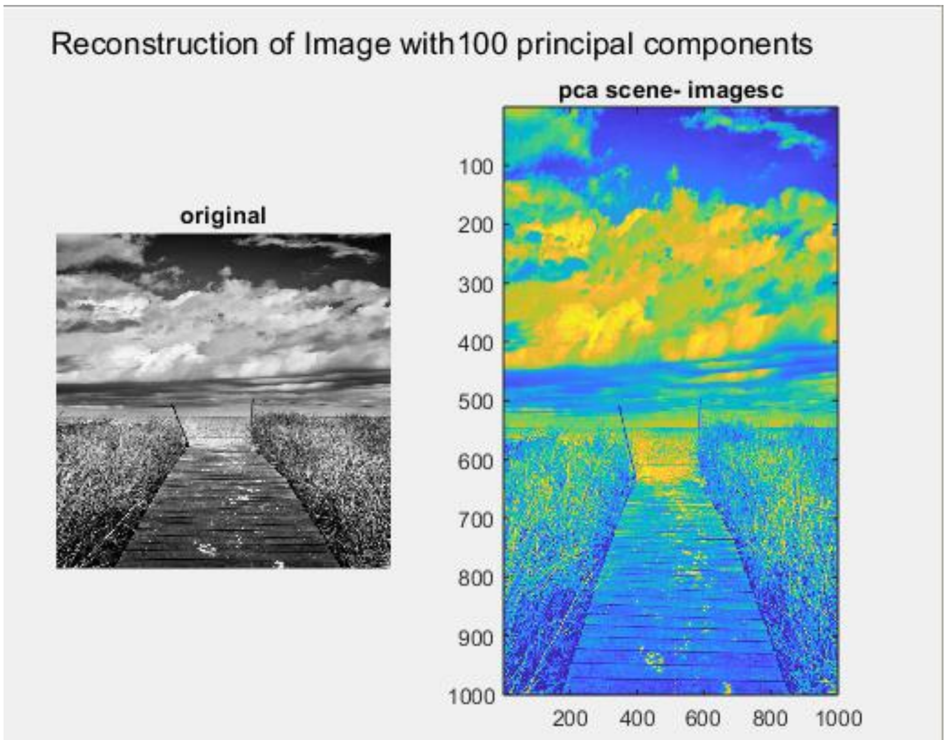


Figure 1.2. I and W*X' reconstructed with 100 principal components ,the difference is negligible

Exercise 2 – How many eigenvalues does it take to explain 95% of the total variance?

We know that the coefficients and eigenvalues stored in the latent variable of PCA are in descending order of variance. From Figure1.3 We can see that the most variance (> 80%) is stored in the very first component.

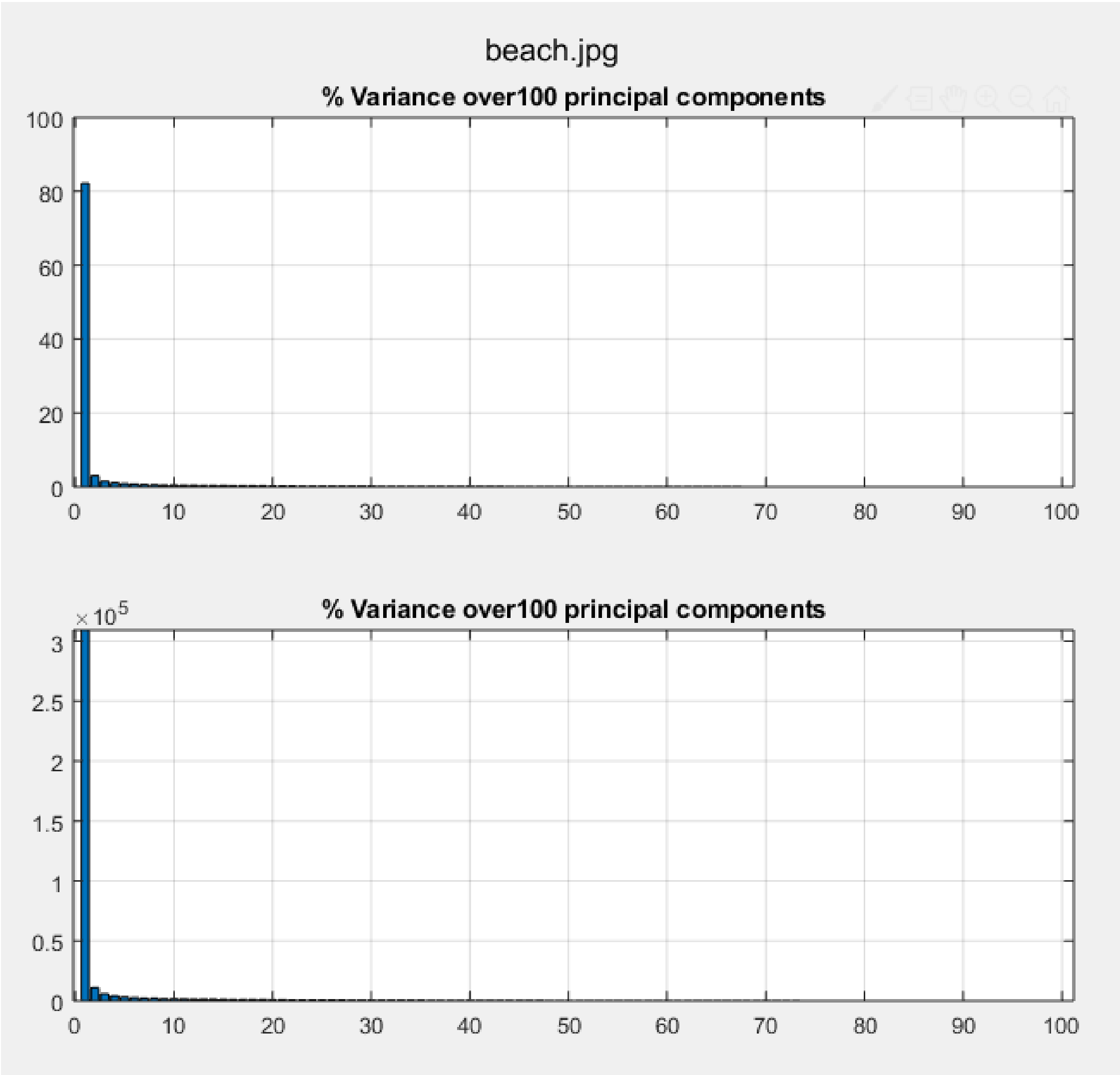


Figure 1.3. Variance % and Variance over 100 principal components

So we can sum eigenvalues one after another until we get to the 95% of the total sum of the latents. It results that the 95% of the variance is explained by the first “i” eigenvalues. Same thing would be looking at the variable “explained” from the full pca output:

```
[X,W,latent,tsquared, explained, mu] = pca(S);
```

And we can see the first ‘i’ values explain the 95% of the variance.

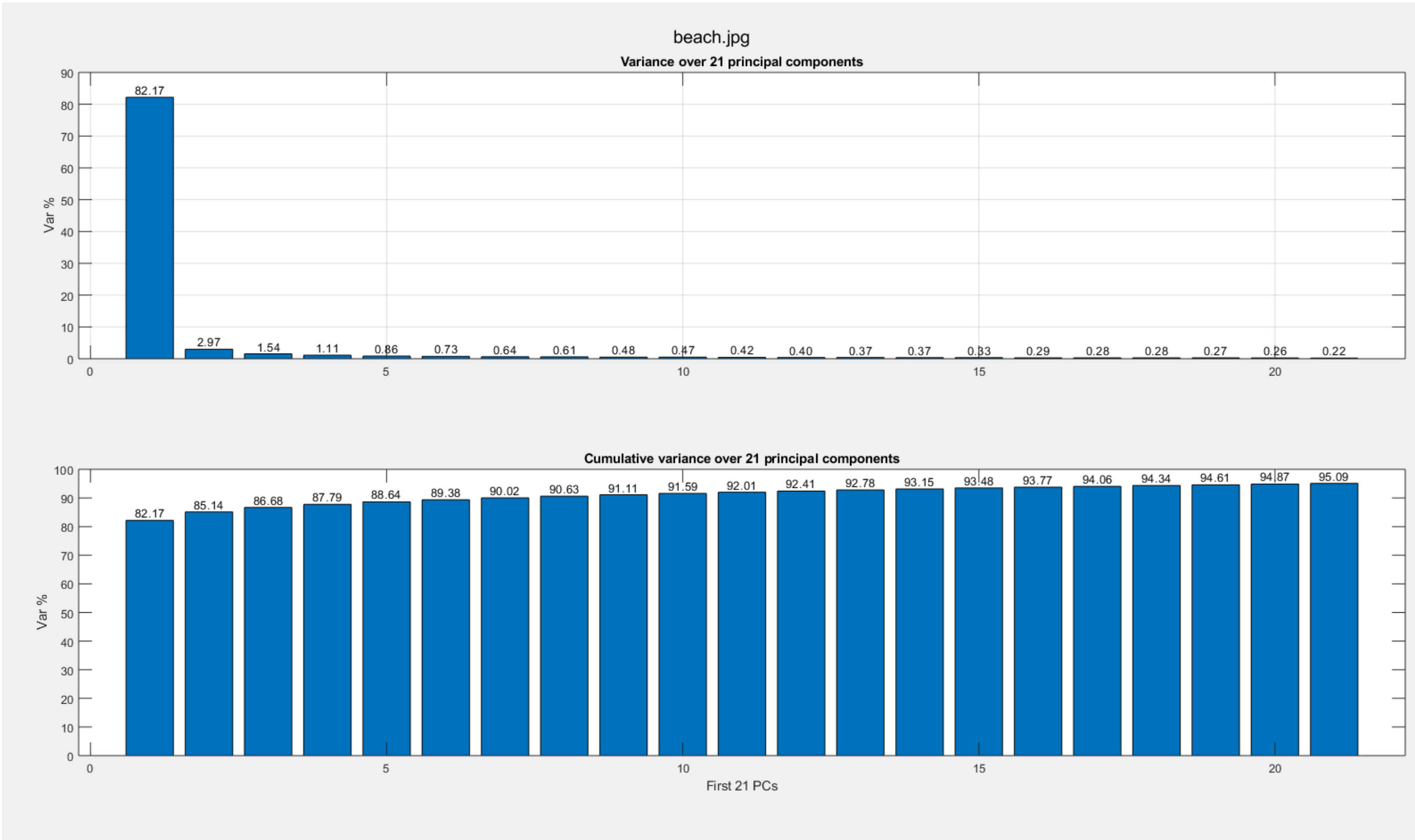


Figure 1.4. Variance and Cum-Variance over 21 principal components

If we reconstruct the image keeping the first i = 21 principal components we get still a very good approximation.

We can see in Figure1.5 how the images appears when is reconstructed with i = 21 components that capture the 95% of the variance of the data.

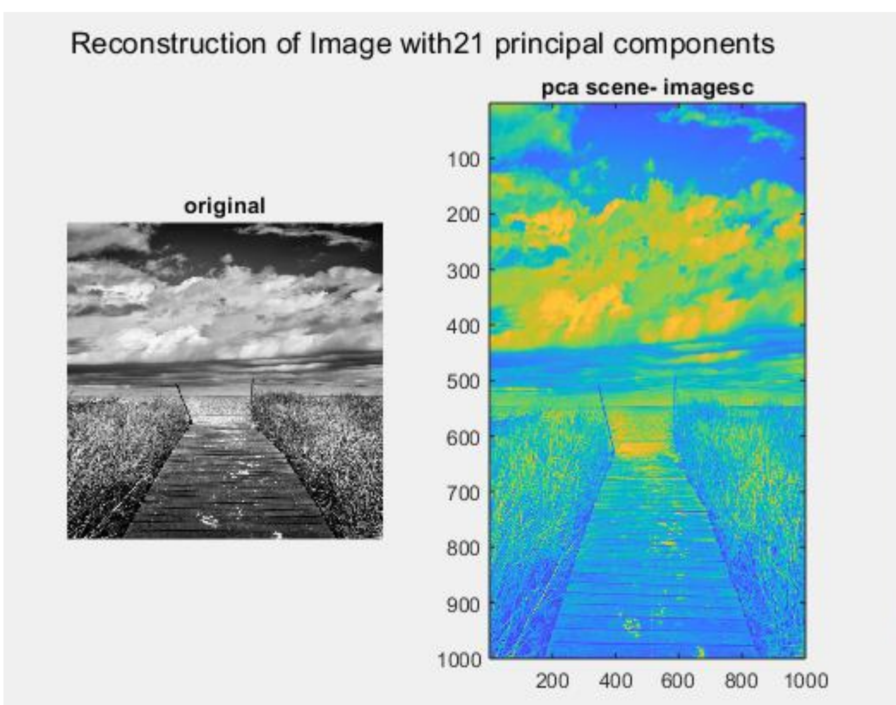


Figure 1.5. Reconstruction of image with 21 principal components.

Exercise 3 – Reconstruct S using only the first 6 PCs and compare it to the original

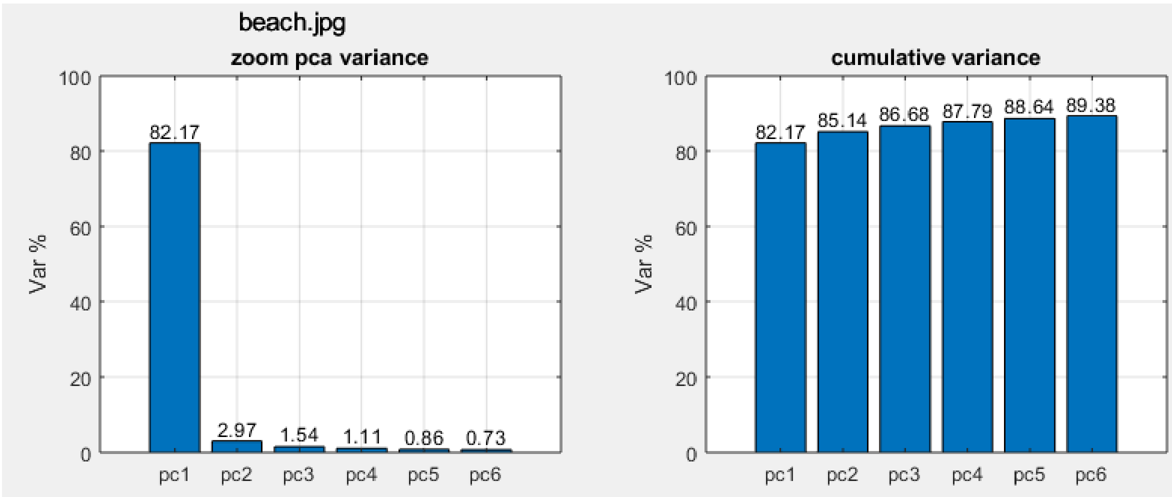


Figure 1.6. Histogram of variance and cumulative variance of the first 6 principal components.

From the Histogram we can see how the first PC is responsible for the 82,17% of the variance and that the first 6 PCs retain the 89,38% of the original information.

We reconstruct the Image, as previously done:

n_pca = 6,

I_pca = W(:,1:n_pca)*X(:,1:n_pca)',

We can see in Figure 1.7 that the reconstructed image appears smoother than the original image. This makes sense, since part of the background information (mostly noise) has been lost.

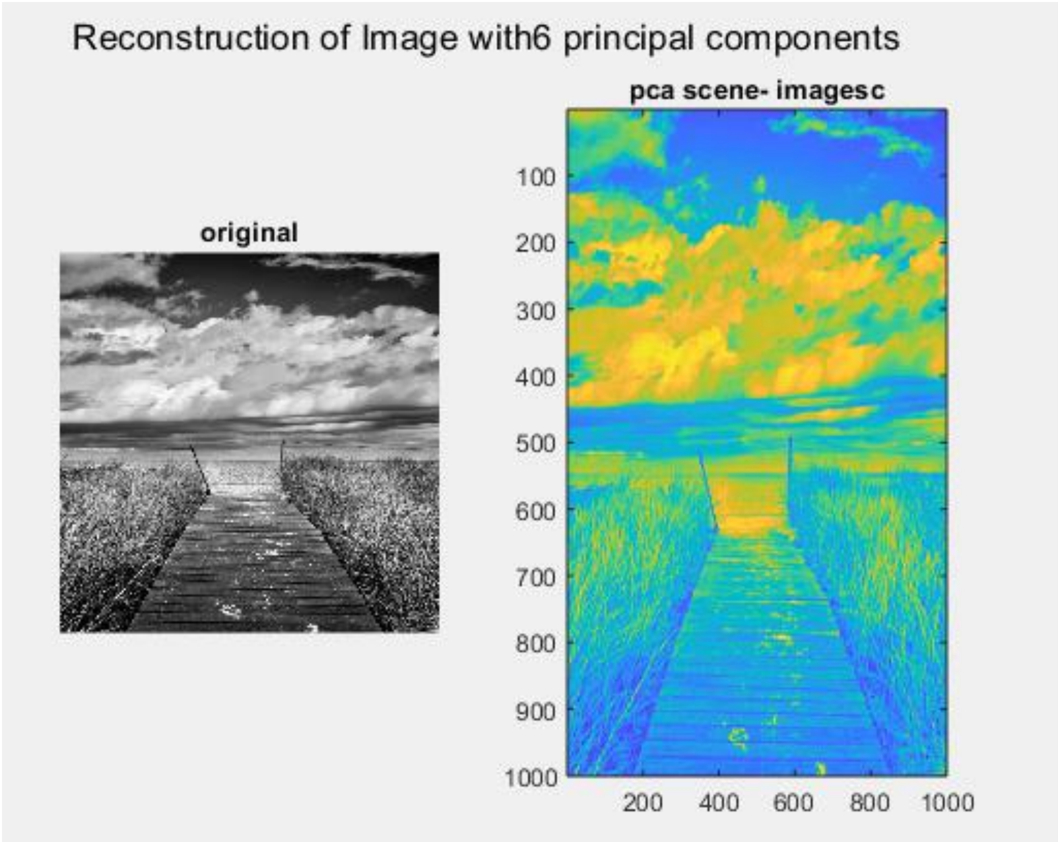


Figure 1.7. Reconstruction of image with 6 principal components.

Homework 2 – Independent components analysis ICA

Exercise 1 – Create and linearly combine two Laplacian distributions

Given the Laplace distribution,

$$Lp(k) = \mu - \sigma \bullet sign(c) \bullet \log(1 - 2 \bullet | c - 0.5|),$$

where c is a random value, μ and σ are the mean and standard deviation of the distribution respectively, we define M=2 Laplace distributions of N=10000 samples each (Fig.1.**A**),

$$\mathbf{d}_1 = [d_1(1), \ldots, d_1(k), \ldots, d_1(N)] \quad \text{where } d_1(k) = Lp(k),$$

$$\mathbf{d}_2 = [d_2(1), \ldots, d_2(k), \ldots, d_2(N)] \quad \text{where } d_2(k) = Lp(k),$$

$$D_{2xN} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

We then create M linear combinations,

$$\mathbf{S}_{NxM} = \mathbf{A}_{MxM} \bullet \mathbf{D}_{MxN} \,,$$

Where **A** is a MxM matrix of steady-state scalar variables,

$$\mathbf{A}_{\text{MxM}} = \begin{bmatrix} a_{11} & \dots & a_{1M} \\ & \vdots & \\ a_{M1} & \dots & a_{MM} \end{bmatrix},$$

Which in the considered case,

$$\mathbf{A}_{2x2} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_2 \end{bmatrix} = \begin{bmatrix} 10 & 20 \\ -2 & 10 \end{bmatrix}$$

And **S** is the MxN matrix containing the resulting M linear combinations (Fig.2.1.**B**),

$$\mathbf{S}_{\text{MxN}} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix},$$

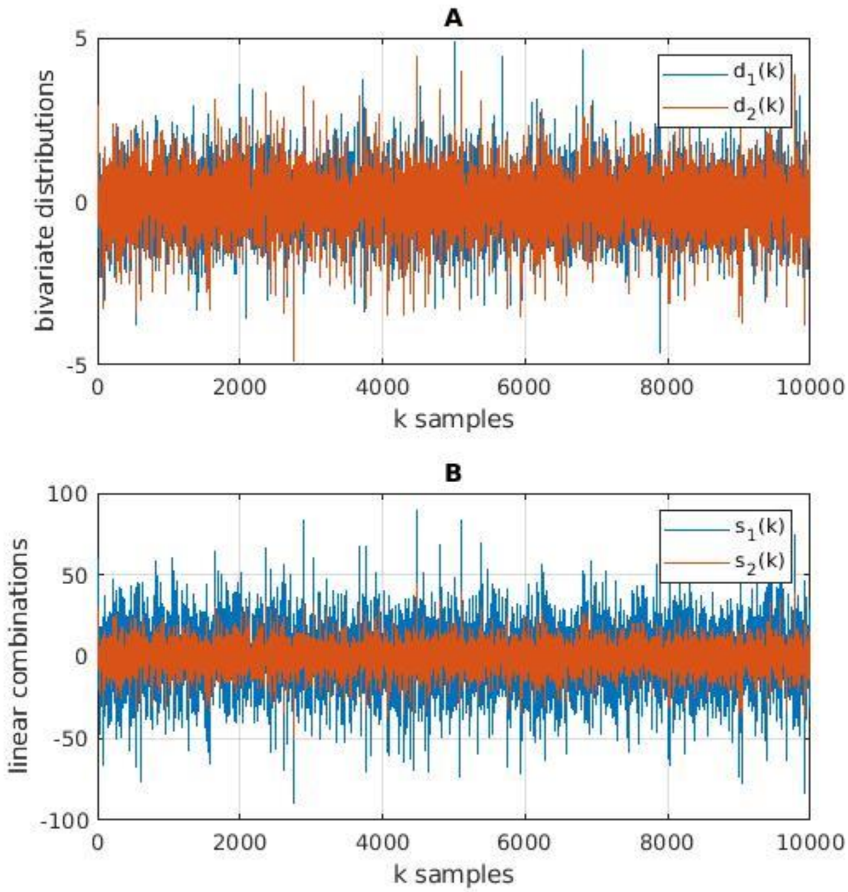


Figure 2.1. A the two laplacian distributions and **B** their linear combinations.

In Figure 2.2, the scatterplot shows the relationship existing between the linear combinations $s_1(k)$ and $s_2(k)$. We can see a strong linear positive correlation between the two.

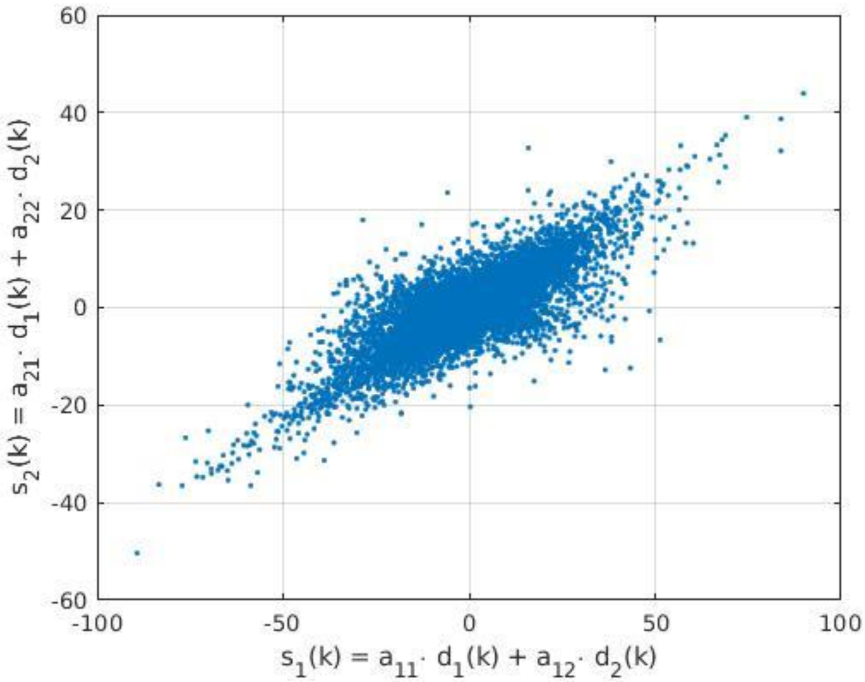


Figure 2.2. The two linear combinations plot one against the other.

Exercise 2 – PCA of the bivariate distribution

The PCA analysis on the **S** matrix returns a set of new variables, the $\mathbf{X}_{MxM}$ PC principal components,

$$\mathbf{X}_{2x2} = [\mathbf{x_1}_{2x1}, \mathbf{x_2}_{2x1}] = \begin{bmatrix} 0.93 & -0.36 \\ 0.36 & 0.93 \end{bmatrix},$$

and the transformed data, or scores, in a matrix **W** with the same dimension of $\mathbf{S}^T$. The PC scores are the representations of **S** in the principal component space. Rows of **W** correspond to the **N** observations, and columns correspond to the **M** components.

The total variance of original and reconstructed data remains the same. Although, it is distributed among the PC in the most unbalanced way. In Fig.2.3, it is represented the amount of variance each PC can possibly explain, this value is computed by finding the eigenvalues of the covariance matrix of $\mathbf{S}^T$. The first PC contains the 94,39% of the entire variance, or rather it is the PC that encodes most of the information, while the second PC represent the 5.60% of the total variance. In order to efficiently reconstruct the original data we need to select a number of PC with final total variance of at least 95% ot of the total, I.e., both PC are needed in this case.

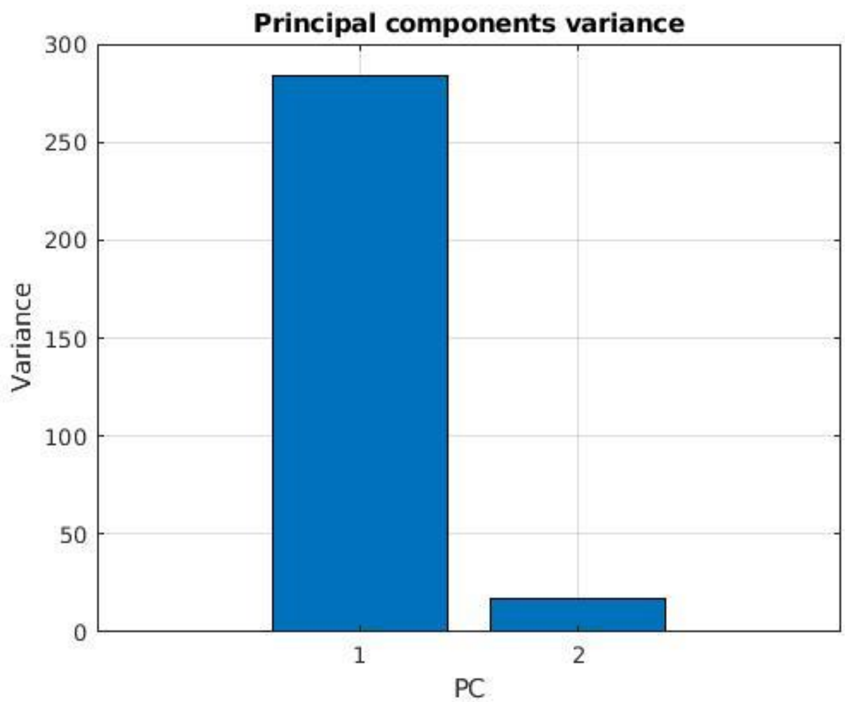


Figure 2.3. Principal components variance.

From the analysis of the score (Fig.2.4.**B**), the two distribution results still dependent, the scores radiate from the origin towards two main directions that not completely follow the two cartesian axis. In Fig.2.4.**C**, the reconstructed data

$$\mathbf{S'} = \mathbf{XW}^{T.}$$

are overlapped by the two PC, it is evident how the new variables are not following the two main direction of the data. One of the main drawbacks of using PCA with bivariate distribution, such as S, is the orthogonality assumption. As matter of fact, PCA works fine with Gaussian distribution, but it is not optimal for minimizing the dependency of non-orthogonal components. In general, PCA works better for normal distributions then for skewed or asymmetric distributions, where it can fail because it retains and maximize the variance only of the projected dimensions. PC method rotates the axis to decorrelate the data, in order to eliminate second order dependencies. However, if the distribution is not Gaussian there are higher order dependencies that can not be removed (as seen in Fig.2.4.**B**).

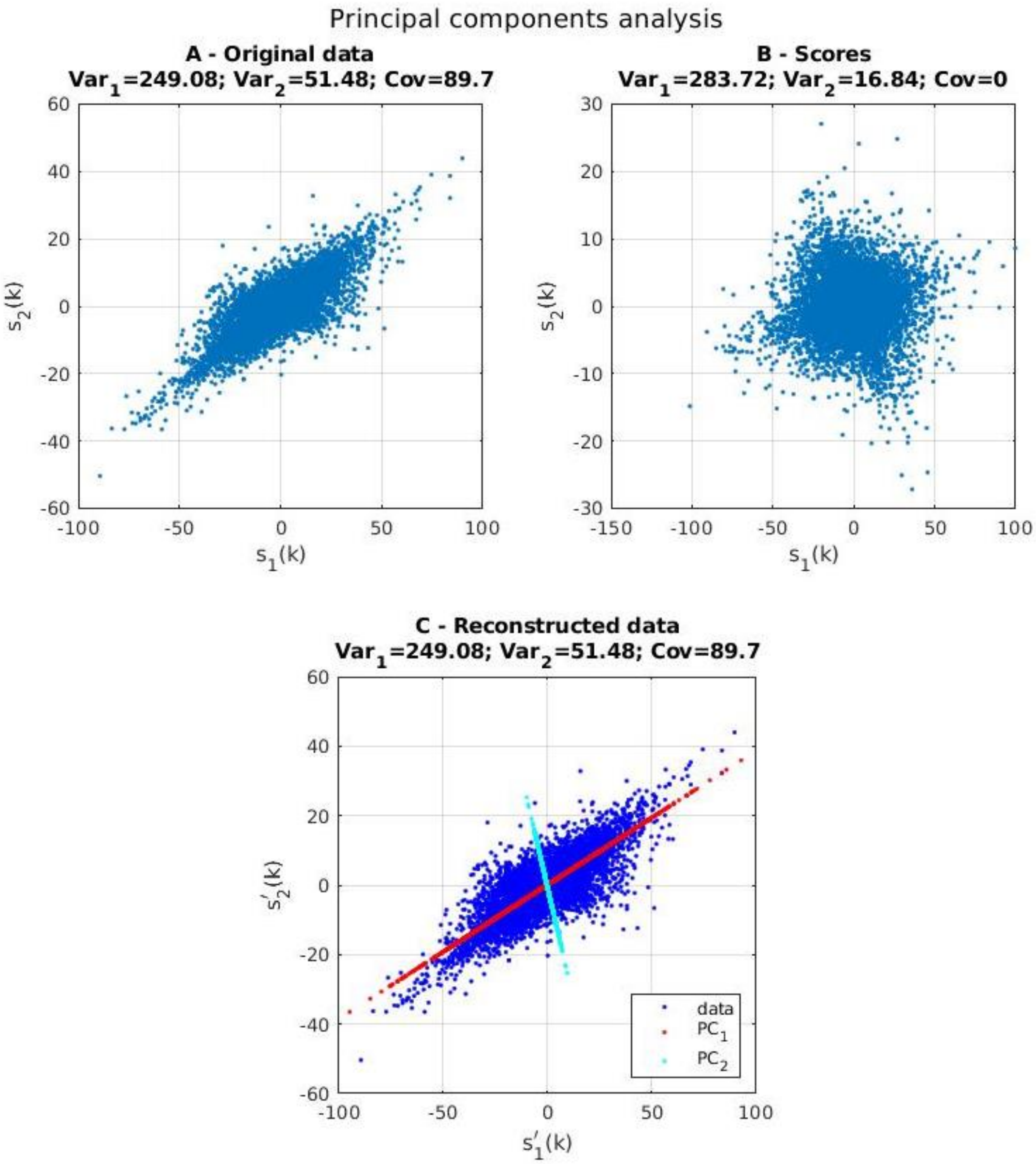


Figure 2.4. Principal components analysis: (**A**) the original data **S**; (**B**) the scores, i.e., the reconstructed data **S** in PC space; (**C**) the reconstructed data **S'**.

Exercise 3 – ICA of the bivariate distribution

The Independent Component Analysis (ICA) is a more efficient method than PCA when analyzing non-gaussian distribution such as **S**. We employed this method in order to separate the data into two independent non-gaussian distribution and the results demonstrate the efficacy of this method. In Fig.2.5.**B**-**C**, we compare the PCA and the ICA scores respectively, the latest one is independent and radiate from the origin in directions of the axis. The ICA score has equal unit variance, zero covariance, and different scale with respect to PCA score and the original data.

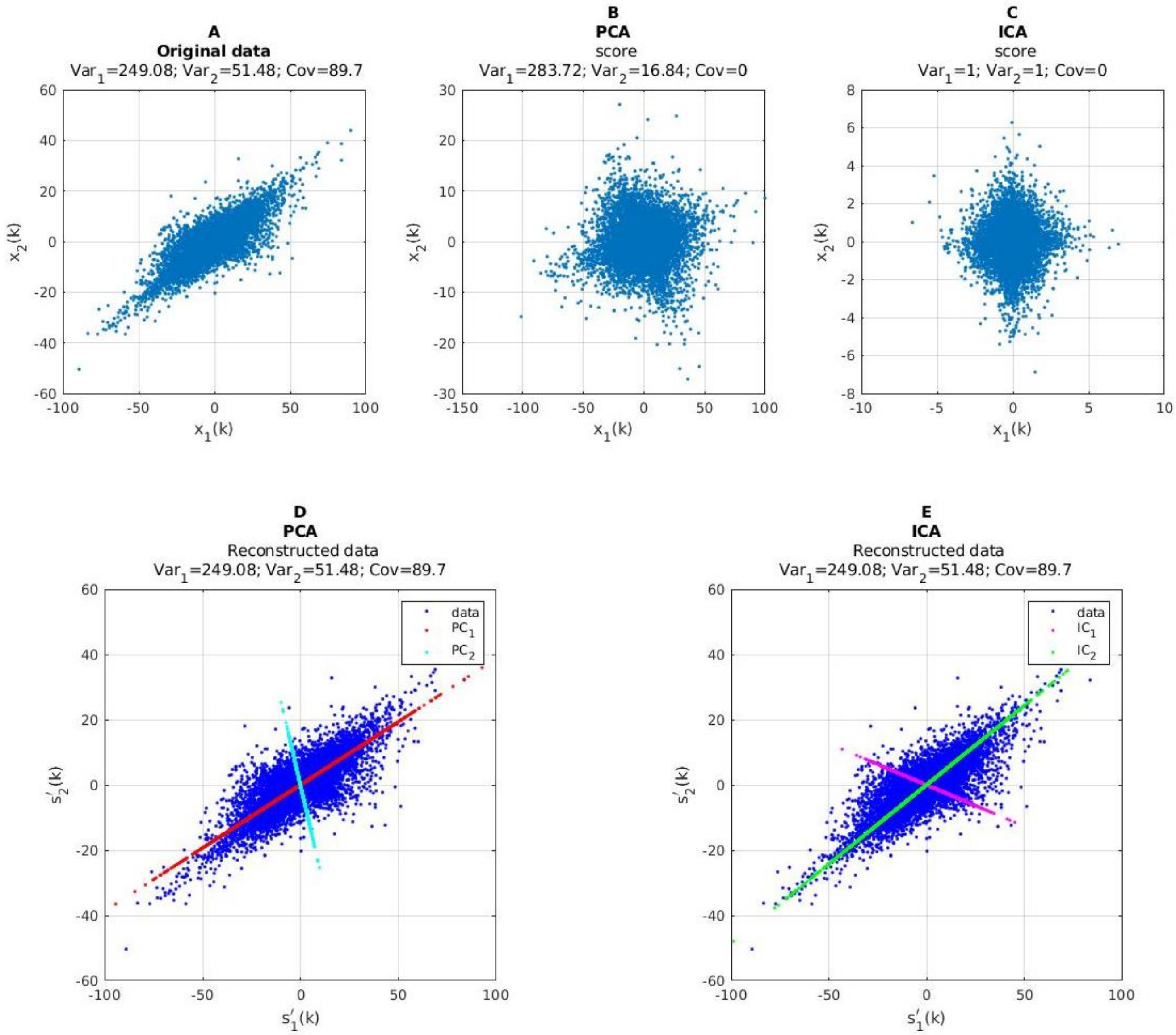


Figure 2.5. Principal vs Independent components analysis: **(A)** the original data S; **(B)** the PCA scores, i.e., the reconstructed data **S'** in PC space; **(C)** the ICA scores, i.e., the reconstructed data **S'** in IC space; (**D**) the data **S** reconstructed with PCs (in red and cyan); **(E)** the data S reconstructed with ICs (in magenta and green)**.**

Exercise 4 – Compare ICA results with the original samples

The ICA score has different scale with respect to the original data and the variance that each independent component IC can explain is equal, this could lead to a substantial loss of information or possible overfitting. As matter of fact, assigning equal variance to a variable that explain less information can lead to over fitting a specific feature, especially when working with high-dimensional space this approach could lead to a wrong reconstruction of the data.

$$\mathbf{W}_{ica} = A_{ica} \bullet \mathbf{S},$$

$$S'_{ica} = \mathbf{X}_{ica} \bullet \mathbf{W}_{ica}^{T} = \mathbf{X}_{ica} \bullet (A_{ica} \bullet \mathbf{S})^{T}.$$

Exercise 5 – ICA of linearly combined time series

Given two guitar soundtracks, we store N=10.000 samples of the 1-channel tracks with a sampling frequency,

$$F_s = 44{,}100 \text{ Hz},$$

into a matrix MxN,

$$\mathbf{D}_{MxN} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad M=2,$$

where,

$$\mathbf{d}_i = [d_i(1), \ldots, d_i(k), \ldots, d_i(N)],$$

We then create M linear combinations,

$$\mathbf{S}_{NxM} = \mathbf{A}_{MxM} \bullet \mathbf{D}_{MxN},$$

where **A** is a MxM matrix of steady-state scalar variables,

$$\mathbf{A}_{MxM} = \begin{bmatrix} 18 & 20 \\ -30 & 10 \end{bmatrix},$$

And **S** is the MxN matrix containing the resulting M linear combinations (Fig.2.6.**C**),

$$\mathbf{S}_{MxN} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}.$$

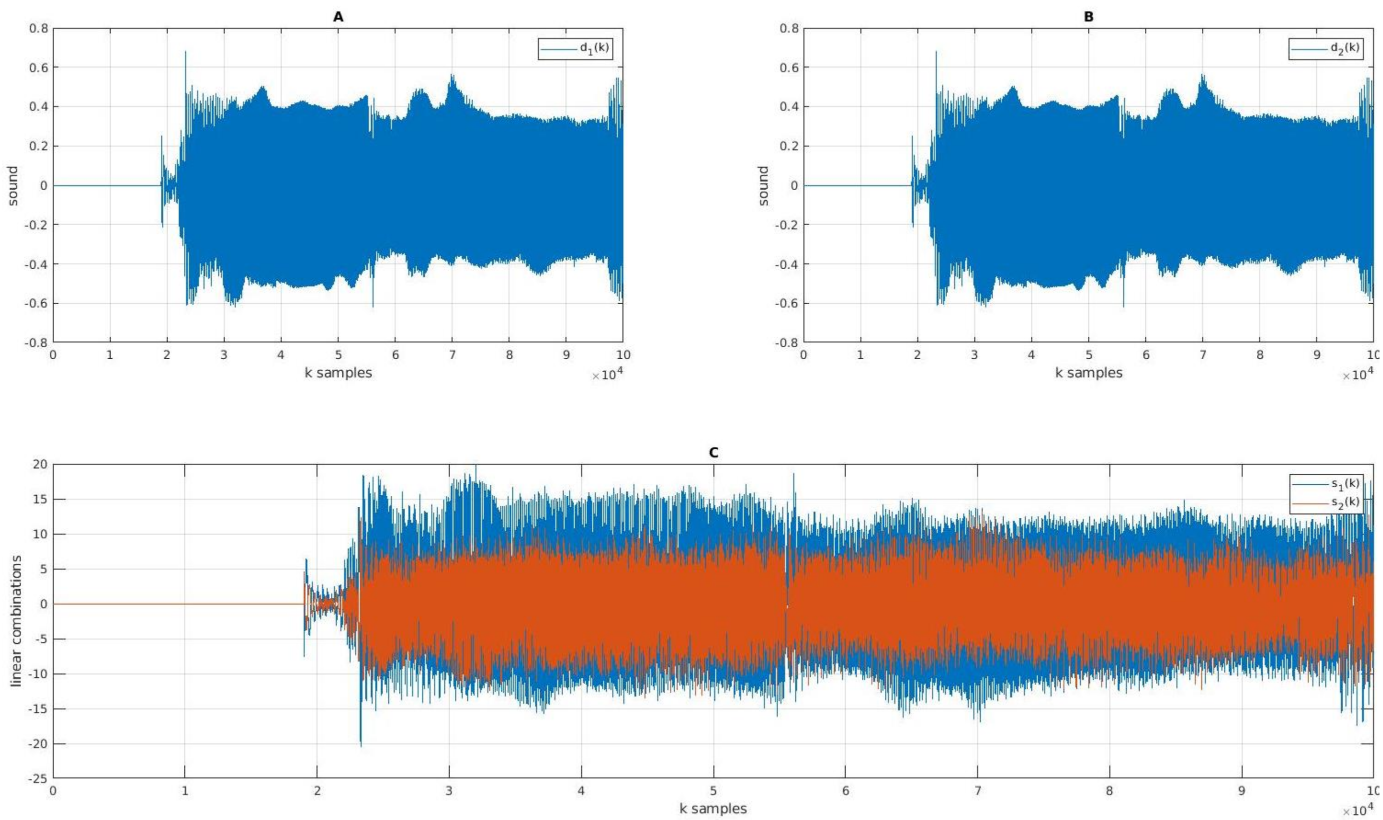


Figure 2.6. A-B the two original time series and **C** their linear combinations.

We employed the ICA method to separate the two linear combinations. The resulting 2 independent components are,

$$\mathbf{X}_{2x2} = [\mathbf{x}_{1\,2x1}, \mathbf{x}_{2\,2x1}] = \begin{bmatrix} 1.0 & 4.05 \\ 1.27 & -6.6 \end{bmatrix}$$

Exercise 5.a - Reconstruction error

Figure 2.7 compares the **S** original data to the **S'** reconstructed and **W** IC-space (or score) data. From Figure 2.7.**E** the data seems correctly reconstructed and the IC follows the two main directions. However, if we calculate the maximum root mean square error between **S** and **S'** we will notice a large deviation,

$$\mathrm{Max(RMSE)} = \max\ \left(\Sigma(N^{-1} \bullet (\mathbf{S}(k)-\mathbf{S}'(k))^2)\right)^{0.5} = 6.71.$$

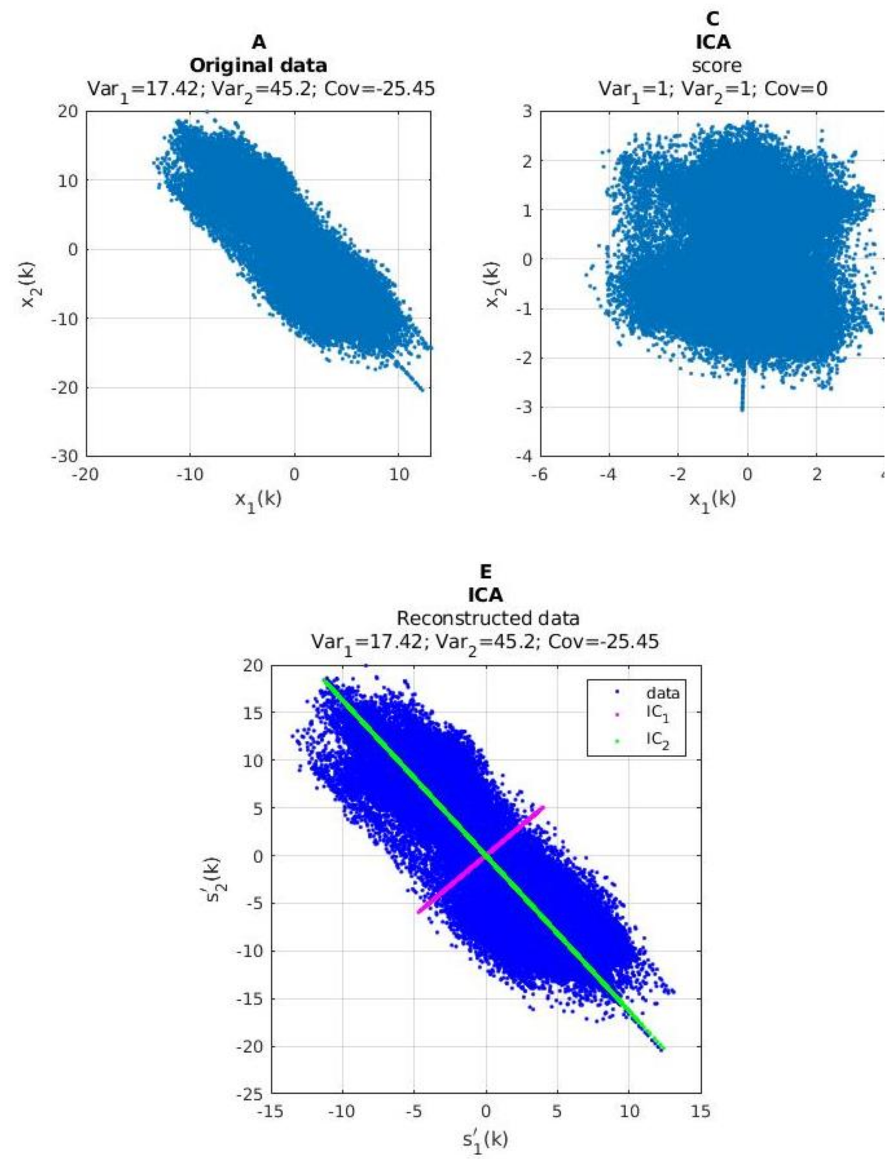


Figure 2.7. Independent components analysis: **(A)** the original data S; (**B)** the scores, i.e., the reconstructed data **S** in IC space; (**C)** the reconstructed data **S'**.

Exercise 5.b - Subjective perception evaluation

The two audio samples sound similar. Although, the reconstruction is noisy it is still possible to recognize the original sound.

Exercise 5.c - Compare the evaluation methods

Although the RMSE gives as a general evaluation about the deviation from the original data, the listener can still recognize the sound. In our opinion, the efficiency of the evaluation method depends on the application. For instance, such large error could be a problem if present in assistive device for people with hearing impairment, and the presence of noise could lead to severe discomfort.

Homework 3 – Bayesian multisensory integration

In this exercise, we have data from three experiments, one with only visual tasks, one with auditory tasks and one with a mixture of both. Each one collects 50 trials and their answers.

Our task is to obtain information about the signals of both stimuli using the data collected by estimating the mean values and the variance of each normal distribution. If only we had the results of the visual and auditory experiments, the task would be trivial, it would be enough to look for the optimal values of the mean and the variance that minimize each function neg log likelihood of both tasks.

What happens is that in addition to these data, we have the data from the joint experiment that also gives us information about the signals separately, and therefore should be considered to obtain more accurate results.

Knowing that the log likelihood functions are independent of each other, we construct a joint neg log likelihood function of the three corresponding to the signals of the three experiments, we substitute the mean and variance of the function of the audiovisual stimuli for their equivalence with the means and variances of the stimuli separately, in order to have only four variables to estimate, and we look for the optimal values that minimize this function.

The function neg log likelihood to minimize is the following:

$$-LogLikelihood = -\left(LogLikelihood_{Visual} + LogLikelihood_{Audio} + LogLikelihood_{Audio\,Visual}\right)$$

Where,

$$\text{LogLikelihood}(\mu, \sigma^2, x_1, \ldots, x_n) = (2\pi\sigma 2)^{-n/2} \exp(-(2\sigma 2)^{-1} \Sigma(x_j-\mu)^2)$$

The estimates are as follows:

Variables	Real	Calculated
'm_a'	-15.62523364	-15.3917948865441
's_a'	2.00197196121263	1.76029883124763
'm_v'	22.86545656	22.5307075121455
's_v'	2.39030759784214	2.53526702658564
'm_av'	-4.0469457068	-3.05652414282423
's_av'	1.82852413036846	2.09073560273964

Figure 3.1 Results of estimates

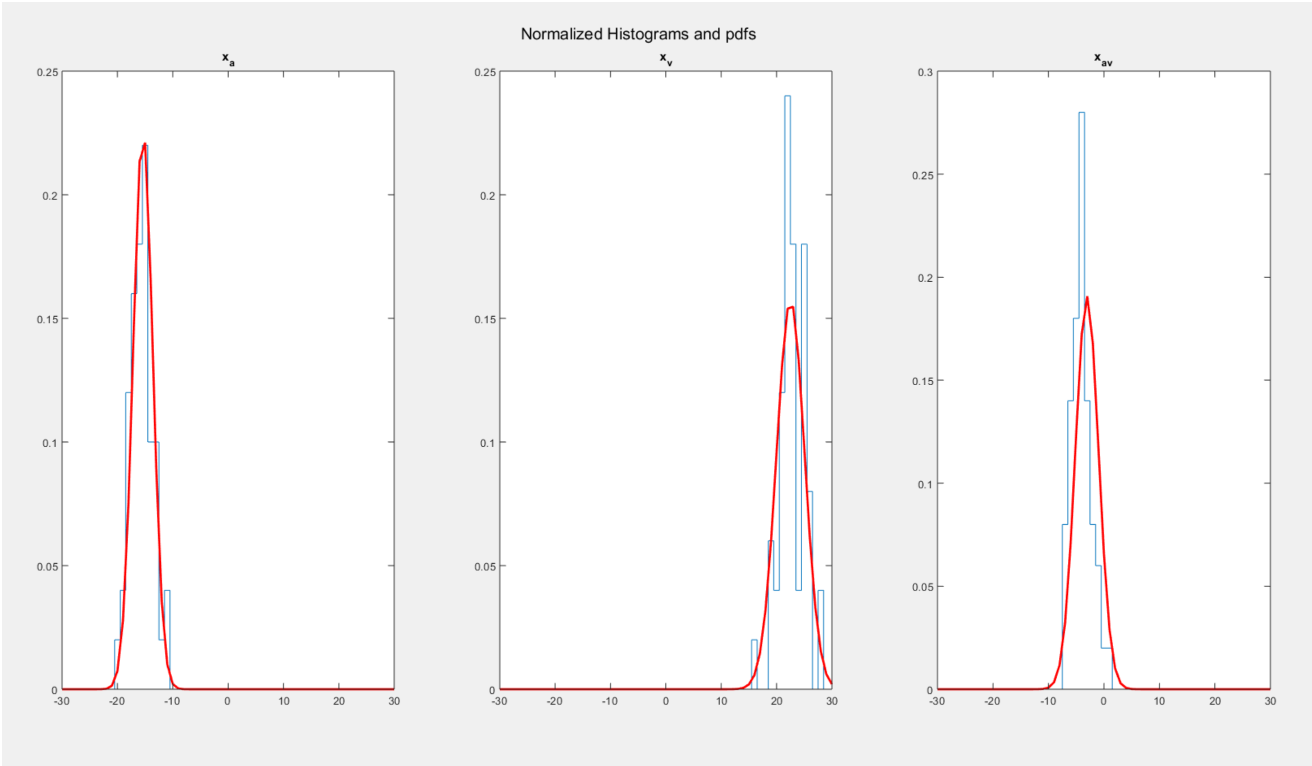


Figure 3.2 Goodness of fit, histogram of the real data, overlapped by pdf (estimated_variables)

Variables	Real	Calculated
'm_a'	-15.832550462	-16.0344807504864
's_a'	4.69142465642395	2.98992586944464
'm_v'	22.460033	22.2578545657044
's_v'	3.69968448661963	4.02608524684125
'm_av'	-3.584196146	-2.42280347103478
's_av'	3.37070735453587	5.76190008456044

Figure 3.3 Results of estimates with the wandering Observer (input lapse series)

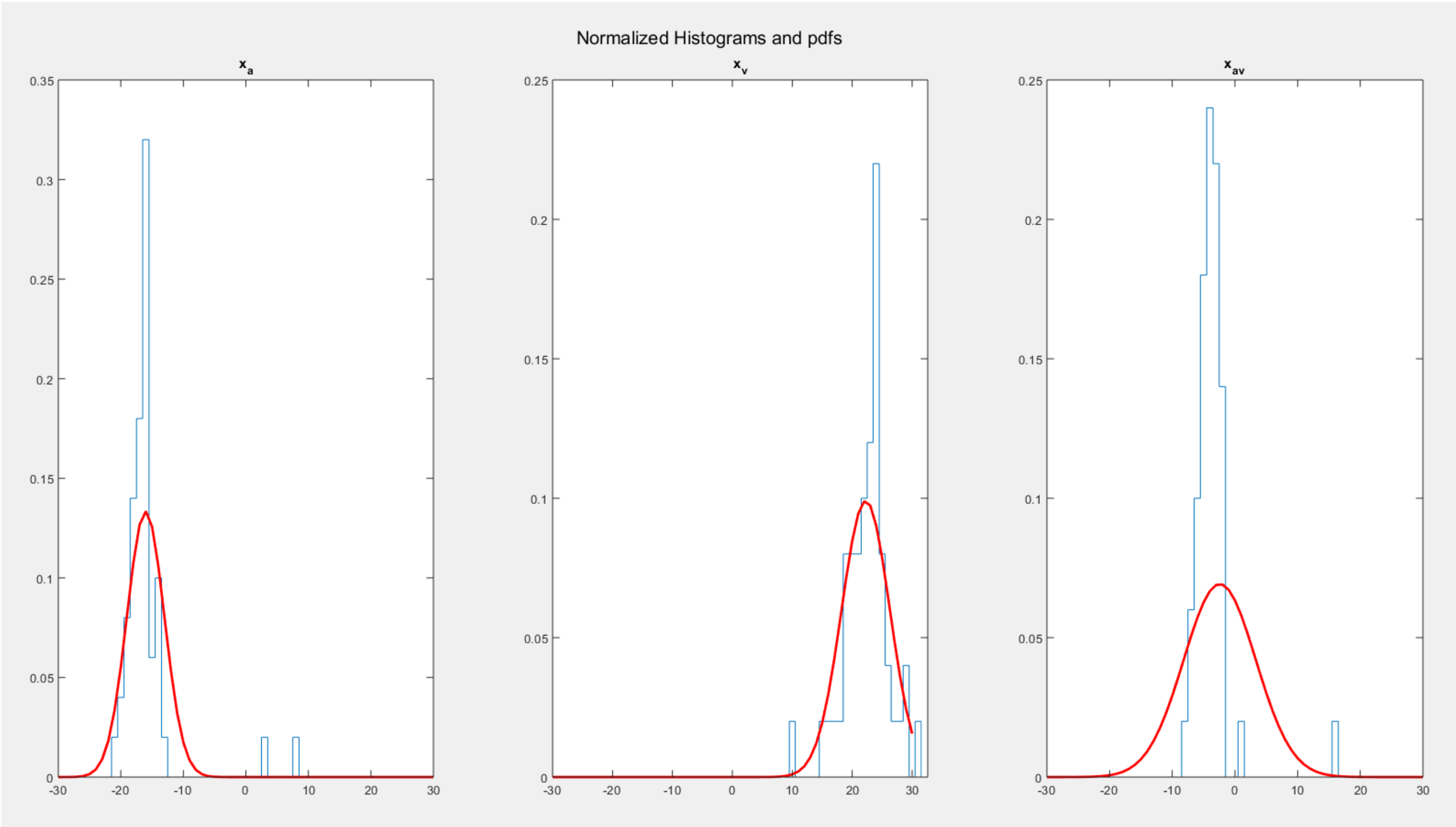


Figure 3.4 Goodness of fit, histogram of the real-modified (lapse) data, overlapped by pdf (new_estimated_variables)

As expected, the randomness of 4% impacts negatively on the estimates of the parameters and on the overall goodness of fit because there are outliers.