Statistical Machine Learning 2018

Assignment 2
Deadline: 28th of October 2018

Instructions:

- You can work **alone or in pairs** (= max 2 people). **Write the full name and S/U-number of all team members on the first page of the report.**
- Write a **self-contained report** with the answers to each question, **including** comments, derivations, explanations, graphs, etc. This means that the elements and/or intermediate steps required to derive the answer have to be in the report. (Answers like 'No' or 'x=27.2' by themselves are not sufficient, even when they are the result of running your code.)
- If an exercise specifically asks for code, put **essential code snippets** in your answer to the question in the report, and explain briefly what the code does. In addition, hand in **complete (working and documented) source code** (MATLAB recommended, other languages are allowed but not "supported").
- In order to avoid extremely verbose or poorly formatted reports, we impose a **maximum page limit** of 20 pages, including plots and code, with the following formatting: fixed **font size** of 11pt on an **A4 paper**; **margins** fixed to 2cm on all sides. All figures should have axis labels and a caption or title that states to which exercise (and part) they belong.
- Upload reports to **Brightspace** as a **single pdf** file: '`SML_A2_<Namestudent(s)>.pdf`' and one zip-file with the executable source/data files (e.g. matlab m-files). For those working in pairs, only one team member should upload the solutions.
- Assignment 2 consists of 3 exercises, weighted as follows: 3 points, 2 points, and 5 points. The **grading** will be based solely on the report pdf file. The source files are considered supplementary material (e.g. to verify that you indeed did the coding).
- For any problems or questions, send us an email, or just ask.
  Email addresses: `tomc@cs.ru.nl` and `b.kappen@science.ru.nl`

Exercise 1 – weight 3

The financial services department of an insurance company receives numerous phone calls each day from people who want to make a claim against their policy. Most claims are genuine, however about 1 out of every 6 are thought to be fraudulent. To tackle this problem the company has installed a trial version of a software voice-analysis system that monitors each conversation and gives a numerical score z between 0 and 1, depending on allegedly suspicious vocal intonations of the customer. Unfortunately, nobody seems to know anymore how to interpret the score in this particular version of the system ...

Tests revealed that the conditional probability density of z, given that a claim was valid ($c = 1$) or false ($c = 0$) are

$$\begin{aligned} p(z|c = 0) &= \alpha_0(1 - z^2) \, , \\ p(z|c = 1) &= \alpha_1 z(z + 1) \, . \end{aligned}$$

1. Compute the normalization constants α_0 and α_1. How does the z score relate to the validity of the claim? What values for z would you expect when the claim is valid / false?
2. Use the sum and product rule to show that the probability distribution function $p(z)$ can be written as
$$p(z) = \frac{(3z + 1)(z + 1)}{4}. \tag{1}$$
3. Use Bayes' rule to compute the posterior probability distribution function $p(c|z)$. Plot these distributions in MATLAB as a function of z. How can these posterior probabilities help in making a decision regarding the validity of the claim?
4. Compute the optimal decision boundary (based on our numerical score z) that minimizes the misclassification rate. For which z should we classify $c = 0$ (false) and for which z should we classify $c = 1$ (valid)? Explain your decision.
5. Compute the misclassification rate, given the optimal decision boundary determined previously. Interpret the result you have obtained. Is the z score useful in determining the validity of the claim? Compare this with your prior guess from 1.

Exercise 2 – weight 2

The government of the United Kingdom has decided to call a referendum regarding the country's European Union membership. The citizens of the UK will be asked the following question at the referendum: "Should the United Kingdom remain a member of the European Union or leave the European Union?". The European Commission (EC) is interested in the potential outcome of this referendum and has contracted a polling agency to study this issue.

Suppose that a person's vote follows a Bernoulli distribution with parameter θ and suppose that the EC's prior distribution for θ, the proportion of British citizens that would be in favor of leaving the EU, is beta distributed with $\alpha = 90$ and $\beta = 110$.

1. Determine the mean and variance of the prior distribution. Plot the prior density function.
2. A random sample of 1000 British citizens is taken, and 60% of the people polled support leaving the European Union. What are the posterior mean and variance for θ? Plot the posterior density function together with the prior density. Explain how the data from the sample changed the prior belief.

3. Examine the effect of changing the prior hyperparameters (α, β) on the posterior by looking at several other hyperparameter configurations. Which values for α and β correspond to a non-informative prior? What is the interpretation of α and β for the beta prior? What does the choice of α and β in Question 1 tell you about the strength of the prior belief?

4. Imagine you are now a reporter for the polling agency and you have been sent on field duty to gather more data. Your mission is to go out on the streets and randomly survey people on their thoughts regarding the upcoming referendum. Given all the available information you have acquired, what is the probability that the first person you talk to will vote 'Leave'?

   Hint: Derive the predictive distribution for the next vote using the posterior distribution for θ computed in Question 2. For a reminder on predictive distributions, see subsection 1.2.6 in Bishop, in particular Equation (1.68).

Exercise 3 – Sequential learning (weight 5)

Part 1 – Obtaining the prior

Consider a four dimensional variable $[x_1, x_2, x_3, x_4]^T$, distributed according to a multivariate Gaussian with mean $\tilde{\boldsymbol{\mu}} = [1, 0, 1, 2]^T$ and covariance matrix $\tilde{\boldsymbol{\Sigma}}$ given as

$$\tilde{\boldsymbol{\Sigma}} = \left(\begin{array}{cc|cc} 0.14 & \text{-}0.3 & 0.0 & 0.2 \\ \text{-}0.3 & 1.16 & 0.2 & \text{-}0.8 \\ \hline 0.0 & 0.2 & 1.0 & 1.0 \\ 0.2 & \text{-}0.8 & 1.0 & 2.0 \end{array} \right) \tag{2}$$

We are interested in the conditional distribution over $[x_1, x_2]^T$, given that $x_3 = x_4 = 0$. We know this conditional distribution will also take the form of a Gaussian:

$$p\big([x_1, x_2]^T \,|\, x_3 = x_4 = 0\big) = \mathcal{N}([x_1, x_2]^T | \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p) \tag{3}$$

for which the mean and covariance matrix are most easily expressed in terms of the (partitioned) precision matrix (see Bishop,§2.3.1).

1. Use the partitioned precision matrix $\tilde{\boldsymbol{\Lambda}} = \tilde{\boldsymbol{\Sigma}}^{-1}$ to give an explicit expression for the mean $\boldsymbol{\mu}_p$ and covariance matrix $\boldsymbol{\Sigma}_p$ of this distribution and calculate their values. (This distribution will be taken as the *prior* information for the rest of this exercise, hence the subscript p). You may use the MATLAB command `inv` to calculate matrix inverses.

2. [MATLAB] - Create a function that can generate random number pairs, distributed according to the distribution in (3). Initialize your random generator and then draw a *single* pair

   $$\boldsymbol{\mu}_t = [\mu_{t1}, \mu_{t2}]^T \tag{4}$$

   from this distribution. (These will be the 'true' means, hence the subscript t).

   Hint: you can use the MATLAB function `mvnrnd` (which resides in the `Statistics` toolbox[1]).

3. [MATLAB] - Make a plot of the probability density of the distribution (3).

   Hint: use the MATLAB function `mvnpdf` (which resides in the `Statistics` toolbox[1]) to calculate the probability density of a multivariate Gaussian random variable. The MATLAB functions `meshgrid` and `surf` may also prove useful.

[1] In OCTAVE, you have to install the `Statistics` package in order to use this function.

Part 2 – Generating the data

Here we assume we are dealing with a 2d-Gaussian data generating process

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{5}$$

For the mean $\boldsymbol{\mu}$, we will use the value $\boldsymbol{\mu}_t$ drawn in (4) in order to *generate* the data. Subsequently, we will pretend that we do not know this "true" value $\boldsymbol{\mu}_t$ of $\boldsymbol{\mu}$, and estimate $\boldsymbol{\mu}$ from the data. For the covariance matrix $\boldsymbol{\Sigma}$ we will use the "true" value

$$\boldsymbol{\Sigma}_t = \begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 4.0 \end{pmatrix} \tag{6}$$

to generate the data.

1. [MATLAB] - Generate at least 1000 data pairs $\{x_i, y_i\}$, distributed according to (5) with $\boldsymbol{\mu} = \boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_t$ and save them to file in a plain-text format.

   Hint: Use MATLAB functions `save` and `load` with the `-ascii` flag.

2. From now on, we will assume (pretend) the 'true' mean $\boldsymbol{\mu}_t$ is unknown and estimate $\boldsymbol{\mu}$ from the data. Calculate the maximum likelihood estimate of $\boldsymbol{\mu}_{\text{ML}}$ and $\boldsymbol{\Sigma}_{\text{ML}}$ for the data, and also an unbiased estimate of $\boldsymbol{\Sigma}$ (see Bishop, §2.3.4). Compare with the true values $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$.

Part 3 – Sequential learning algorithms

We will now estimate the mean $\boldsymbol{\mu}$ from the generated data and the known variance $\boldsymbol{\Sigma}_t$ *sequentially*, i.e., by considering the data points one-by-one.

1. [MATLAB] - Write a procedure that processes the data points $\{\mathbf{x}_n\}$ in the generated file one-by-one, and after each step computes an updated estimate of $\boldsymbol{\mu}_{\text{ML}}$, the maximum likelihood of the mean (using Bishop, eq.2.126).

Now we also use the prior information $p(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$. From the prior, the generated data and the known variance $\boldsymbol{\Sigma}_t$, we will estimate the mean $\boldsymbol{\mu}$.

2. Work out the details of sequential Bayesian inference (see eq.2.144) for the mean $\boldsymbol{\mu}$. Apply Bayes' theorem in eq. 2.113-2.117 at each step $n = 1, \ldots, N$ to compute the new posterior mean $\boldsymbol{\mu}^{(n)}$ and covariance $\boldsymbol{\Sigma}^{(n)}$ after a new point $(\mathbf{x}_n)$ has arrived from the old posterior mean $\boldsymbol{\mu}^{(n-1)}$ and covariance $\boldsymbol{\Sigma}^{(n-1)}$. Use this updated posterior as the prior in the next step. The first step starts from the original prior (3).
   Note: do not confuse the posterior $\boldsymbol{\Sigma}^{(n)}$ with the known $\boldsymbol{\Sigma}_t$ of the data generating process. For some more hints, see appendix.

3. [MATLAB] - Write a procedure that processes the data points $\{\mathbf{x}_n\}$ in the generated file one-by-one, and after each step computes an updated estimate of $\boldsymbol{\mu}_{\text{MAP}}$ - the maximum of the posterior distribution, using the results of the previous exercise.

4. [MATLAB] - Plot both estimates (ML and MAP) in a single graph (1d or 2d) as a function of the number of data points observed. Indicate the true values $\{\mu_{t1}, \mu_{t2}\}$ as well. Evaluate your result.

4 Hints

Below are some hints for **Exercise 3 - Part 3 - Question 2**.

Bayes rule is also valid if earlier acquired information is taken into account. For example, if this is earlier seen data $D_{n-1} = \{x_1, \dots, x_{n-1}\}$. Bayes rule conditioned on this earlier data is

$$P(\mu|x_n, D_{n-1}) \propto P(\mu|D_{n-1}) P(x_n|\mu, D_{n-1}).$$

Since$D_n = \{x_1, \dots, x_n\}$ this is written more conveniently as

$$P(\mu|D_n) \propto P(\mu|D_{n-1}) P(x_n|\mu, D_{n-1}).$$

If, given the model parameters μ, the probability distribution of x_n is independent of earlier data D_{n-1}, we can further reduce this to

$$P(\mu|D_n) \propto P(\mu|D_{n-1}) P(x_n|\mu)$$

You should be able to see the relation with (2.144) and see in particular that the factor between square brackets in (2.144) is to be identified with $P(\mu|D_{n-1})$.

Another important insight is that if $P(\mu|D_{n-1})$ and $P(x_n|\mu)$ are of the form (2.113) and (2.114), i.e., if $P(\mu|D_{n-1})$ is a Gaussian distribution over μ with a certain mean and covariance (you are free to give these any name, e.g. $\mu^{(n-1)}$, $\Sigma^{(n-1)}$) and if $P(x_n|\mu)$ is also Gaussian with a mean that is linear μ, then you can use (2.116) and (2.117) to compute the posterior $P(\mu|D_n)$, which therefore is also Gaussian.

So it is your task to show this. To do this you have to figure out the mapping of the variables and parameters in the current exercise, i.e., what is the correspondence between $\mu, x_n, \Sigma_t, \mu^{(n-1)}, \Sigma^{(n-1)}$ etc. with $x, \mu, \Lambda, y, A, b, L$. Don't forget that some quantities can also be zero or and other may be identity matrices.