Compulsory for STK4051 - Computational statistics

Spring 2019

Part 1 (of 2)

This is the first part of the compulsory exercise for STK4051/9051, spring semester 2019. The second part of the compulsory exercise will be made available in the end of Mars. The deadline for the complete compulsory exercise (including part 2) is **April 30**. This *must* be delivered in the Devilry system (`devilry.ifi.uio.no`).

It *is* possible to deliver the first part earlier in which case feedback also will be given earlier. For this part, you should deliver the exercise in paper to the lecturer!

Reports may be written in Norwegian or English, and should preferably be text-processed (LaTeX, Word). Give your name and "student number" on the first page. Write concisely. Relevant figures need to be included in the report. Copies of relevant parts of machine programs used (in R, or matlab, or similar) are also to be included, perhaps as an appendix to the report.

This first part contains five exercises and comprises four pages (with an extra fifth page, see below). Some R-code is available from the course web-page. You are free to use other software, but would then need to translate or write your own code for that part included in the R-script.

Exercise 1 (Simulating data). We will in this exercise use simulated data. The model we will consider is

$$\begin{aligned} X_t|C_t = k \sim & N(\mu_k, \sigma_k^2) \\ \Pr(C_t = \ell|C_{t-1} = k) = & p_{k\ell}, \qquad\qquad k, \ell = 1, ..., K \\ \Pr(C_1 = k) = & 1/K \end{aligned}$$

were we will refer to t as time in the following. We will further define

$$\boldsymbol{\theta} = \{(\pi_k, \mu_k, \sigma_k^2), k = 1, ..., K, p_{k,\ell}, k, \ell = 1, ..., K\}.$$

Data can be generated through the script `HMM_sim.R` that is available on the course webpage.

(a). Run the simulation script but change the seed to your student number!

Plot $\{x_t\}$ as a function of t. Do you see a clear pattern in the change of underlying states?

(b). Calculate the probability $\Pr(C_t = k|X_t)$, that is the class probability using only the current observation (based on the true parameter values).

Perform a "classification" for each t by using some appropriate rule based on $\Pr(C_t = k|X_t)$ and compare this with the true classes $\{C_t\}$.

Comment on the results.

Exercise 2 (Discrete optimization). We will continue to use the data simulated from Exercise 1. We will assume the model parameters $\boldsymbol{\theta}$ are known.

Our interest here will be to optimize $p(\boldsymbol{C}|\boldsymbol{x}; \boldsymbol{\theta})$ where $\boldsymbol{C} = (C_1, ..., C_n)$ and $\boldsymbol{x} = (x_1, ..., x_n)$.

(a). Show that

$$p(\boldsymbol{C}|\boldsymbol{x}; \boldsymbol{\theta}) \propto \Pr(C_1) f_{C_1}(x_1) \prod_{i=2}^{n} p_{c_{i-1}, c_i} f_{C_i}(x_i)$$

where $f_k(x_i)$ is the density for $x_i|C_i = k$. Discuss why it may be better to consider $p(\boldsymbol{C}|\boldsymbol{x}; \boldsymbol{\theta})$ on a log-scale.

(b). Implement a greedy (local search) algorithm for optimizing $p(\boldsymbol{C}|\boldsymbol{x}; \boldsymbol{\theta})$. Specify how you choose initial values. Calculate $p(\boldsymbol{C}|\boldsymbol{x}; \boldsymbol{\theta})$ (up to the proportionality constant) based on your result. Make a plot of $\boldsymbol{x}$ and $\boldsymbol{C}$ as functions of time (perhaps zoomed in on a smaller time-frame) and discuss the results.

(c). Implement simulated annealing for optmization of $p(\boldsymbol{C}|\boldsymbol{x}; \boldsymbol{\theta})$. Specify how you choose the initial temperature and the temperature schedule. Calculate $p(\boldsymbol{C}|\boldsymbol{x}; \boldsymbol{\theta})$ based on your result. Make a plot of $\boldsymbol{x}$ and $\boldsymbol{C}$ as functions of time (perhaps zoomed in on a smaller time-frame) and discuss the results.

(d). For this model it is actually possible to design an algorithm through dynamic programming that gives the *exact* optimum:

(i) Define $V_{1,l} = \log(\pi_l) + \log f_l(x_1)$

(ii) Define $V_{t,l} = \max_k[V_{t-1,k} + \log p_{kl} + \log f_l(x_t)]$ for $t = 2, ..., n$

(iii) Define $S_{t,l} = \text{argmax}_k[V_{t-1,k} + \log p_{kl} + \log f_l(x_t)]$.

(iv) Define $C_n^{opt} = \text{argmax}_k V_{n,k}$

(v) Define $C_t^{opt} = S_{t+1,C_{t+1}^{opt}}$ for $t = n-1, ..., 1$

This is also called the *Viterbi algorithm*.

Implement this algorithm as well and compare with previous results.

(e). Only for PhD students: Show that this algorithm actually gives the global optimum!

Hint: You can use any source you want to find this out, but you should give a proper derivation of this and include a reference to your source.

Exercise 3 (The EM-algorithm). We will continue to work on the simulated data from Exercise 1. Consider now however the problem of simultaneous estimation of $\boldsymbol{C}$ and $\boldsymbol{\theta}$. We will in this exercise apply the EM-algorithm. On the course web-page there is an R-script, `HMM.E.R` which calculates

$$\begin{aligned} q_{i|i-1}(k) &= \Pr(C_i = k|x_1, ..., x_{i-1}) && \text{Prediction} \\ q_{i|i}(k) &= \Pr(C_i = k|x_1, ..., x_i) && \text{Updating} \\ q_{i|n}(k) &= \Pr(C_i = k|x_1, ..., x_n) && \text{Smoothing} \end{aligned}$$

as well as

$$q_{i-1,i|n}(k, l = \Pr(C_i = \ell, C_{i-1} = k|x_1, ..., x_n)$$

which is based on the forward-backward algorithm discussed in the lecture.

(a). Make a function which, given the q's above, calculates estimates for the unknown parameters (the M-step in the EM-algorithm).

Include your calculations in the report!

(b). Combine your function with the one calculating the q's to implement a function which performs the EM-algorithm. Specify your convergence criterion.

(c). Include a routine which calculates the log-likelihood value for a given value of $\boldsymbol{\theta}$.

Hint: Use that

$$\begin{aligned} \ell(\boldsymbol{\theta}) &= \log f(\boldsymbol{x}; \boldsymbol{\theta}) \\ &= \log f(x_1; \boldsymbol{\theta}) + \sum_{i=2}^{n} \log f(x_i|x_1, ..., x_{i-1}; \boldsymbol{\theta}) \end{aligned}$$

and show that the terms $f(x_i|x_1, ..., x_{i-1}; \boldsymbol{\theta})$ are possible to be calculated by the output from the `HMM.E.R` script.

(d). How does the number of computational steps in this algorithm depend on n and K?

(e). Run the EM-algorithm on your siumulated data. Try out different starting values. Confirm that the log-likelihood value is non-decreasing with the number of iterations in the EM-algorithm. Discuss the results.

Exercise 4 (Direct parameter optimization). An alternative to the EM-algorithm is to implement a function which directly calculates the log-likelihood for a given set of parameters and then throw this function into a numerical optimizer. One problem that needs to be considered in this case is the constraints on the parameters involved.

(a). Specify which constraints that are involved in the parameter vector $\boldsymbol{\theta}$. How many free parameters do we then have?

(b). Suggest some reparametrization of $\boldsymbol{\theta}$ which removes the constraints on the parameters. Make sure that this reparametrization is invertible.

(c). Modify the routine you implemented in Exercise 3(c) to calculate the log-likelihood using the reparametrized parameters as input. Optimize $l(\boldsymbol{\theta})$ through some numerical optimizer.

Hint: You might expect that this optimization both is time-consuming and non-stable.

Exercise 5 (Summary). Write a (maximum) one-page summary of the previous exercises where you in particular consider the following points:

(a). Ways of obtaining uncertainty measures on the parameter estimates.

(b). The use of the EM algorithm compared with the direct optimization route for this problem.

(c). The results you obtained. In particular, is there a need to include the temporal dependence in the C_i's?