STK4051/9051 - Computational statistics

Trial exam spring 2019

Exercise 1
Consider first the standard Weibull distribution with density function

$$f_0(x_0) = \alpha x_0^{\alpha-1} e^{-x_0^\alpha}$$

and cummulative distribution function

$$F_0(x_0) = 1 - e^{-x^\alpha}.$$

(a) Explain how the inversion method can be used to generate samples from f_0.

Consider now the general Weibull distribution with density function

$$p(x) = \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-(x/\beta)^\alpha}$$

(b) Show that if x_0 has a standard Weibull density then $x = \beta x_0$ has a general Weibull density. Discuss how this result can be used to generate random variables from the general Weibull distribution.

Assume now we want to generate two dependent random variables that have marginal distributions that are of the Weibull form. Direct specification of dependence for the Weibull distribution can be difficult, but can be greatly simplified through transformation (this is called a copula approach in the literature).

(c) Let $\Phi()$ be the cummulative distribution function for the standard Normal distribution. Show that if $y \sim N(0,1)$, then

$$x = F_0^{-1}(\Phi(y))$$

has a standard Weibull distribution.

(d) Assume now that you are able to simulate $\boldsymbol{y} = (y_1, y_2)$ from a bivariate Normal distribution $N(\mathbf{0}, \boldsymbol{\Sigma})$ where $\Sigma_{1,1} = \Sigma_{2,2} = 1$ and $\Sigma_{1,2} = \Sigma_{2,1} = \rho$. Explain how you can use this to simulate two dependent Weibull distributed variables.

Exercise 2
Consider the following algorithm, which we will call Barker's algorithm (after Barker (1965) who suggested it):
Given the current state $\boldsymbol{x}^{(t)}$:

- Draw $\boldsymbol{y}$ from the proposal distribution $K(\boldsymbol{x}^{(t)}, \boldsymbol{y})$ (or transition kernel).
- Draw $U \sim \text{Uniform}[0,1]$ and update

$$\boldsymbol{x}^{(t+1)} = \begin{cases} \boldsymbol{y}, & \text{if } U \leq r_B(\boldsymbol{x}^{(t)}, \boldsymbol{y}) \\ \boldsymbol{x}^{(t)} & \text{otherwise} \end{cases}$$

where

$$r_B(\boldsymbol{x}, \boldsymbol{y}) = \frac{\pi(\boldsymbol{y})K(\boldsymbol{y}, \boldsymbol{x})}{\pi(\boldsymbol{x})K(\boldsymbol{x}, \boldsymbol{y}) + \pi(\boldsymbol{y})K(\boldsymbol{y}, \boldsymbol{x})}.$$

We will assume that $K(\boldsymbol{x}, \boldsymbol{y}) > 0$ for all $\boldsymbol{x}, \boldsymbol{y}$.

(a) Show that $\{\boldsymbol{x}_t\}$ is a Markov chain with invariant distribution $\pi(\boldsymbol{x})$.

(b) Explain how we can used the simulations $\{\boldsymbol{x}^{(t)}\}$ to estimate $E_\pi h = \int_{\boldsymbol{x}} h(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}$.

What kind of properties of the Markov chain will influence on the precision of such an estimate?

Assume A_1 and A_2 are two transition-kernels for Markov chains with the same stationary distribution π. Let v_1 be the variance of the estimate on $E_\pi h$ based on simulations using A_1 and v_2 the variance of the estimate of $E_\pi h$ using A_2.
Assume $A_1(\boldsymbol{x}, \boldsymbol{y}) \geq A_2(\boldsymbol{x}, \boldsymbol{y})$ for all $\boldsymbol{y} \neq \boldsymbol{x}$. One can then show that $v_1 \leq v_2$ (this you do not have to prove).

(c) Let

$$r_M(\boldsymbol{x}, \boldsymbol{y}) = \min\left\{1, \frac{\pi(\boldsymbol{y})K(\boldsymbol{y}, \boldsymbol{x})}{\pi(\boldsymbol{x})K(\boldsymbol{x}, \boldsymbol{y})}\right\}.$$

Show that $r_M(\boldsymbol{x}, \boldsymbol{y}) \geq r_B(\boldsymbol{x}, \boldsymbol{y})$ for all $\boldsymbol{x}, \boldsymbol{y}$.

Use this to argue that the Metropolis-Hastings algorithm is more efficient than Barker's algorithm.

Based on the differences between these two algorithms, do you think this is a reasonable result?

Exercise 3
The Gibbs sampler applies to vectors of random variables. We shall in this exercise consider random pairs (X, Y). The algorithm is as follows:

Algorithm

Select X (initialization)
Repeat
Sample Y from its conditional distribution given X.
Sample X from its conditional distribution given Y.

It can under general conditions be proved that a simulation of (X, Y) appears in the limit as the loop is continued on and on. We shall below actually prove this result in the simple example considered.

Let (X, Y) be bivariate normal, with means $E(X) = E(Y) = 0$, variances $\text{var}(X) = \text{var}(Y) = 1$ and correlation $\text{corr}(X, Y) = \rho$. The conditional distribution of Y given $X = x$ is then normal with mean ρx and variance $1 - \rho^2$, and the conditional distribution of X given $Y = y$ is defined by symmetry. Let $\{Z_n\}$ and $\{V_n\}$ be sequences of independent normal variables $(0, 1)$. Also assume independence between sequences.

(a) Show that the Gibbs sampler sets up the double recursion

$$Y_n = \rho X_n + \sqrt{1-\rho^2} Z_n, \qquad X_n = \rho Y_{n-1} + \sqrt{1-\rho^2} V_n.$$

It will be proved that as $n \to \infty$, (X_n, Y_n) converges to a sample of (X, Y) for any starting point $X_0 = \mu_0$. We shall also study the rate of convergence. Consider $\{X_n\}$ first.

(b) Show that $X_n = \rho^2 X_{n-1} + \varepsilon_n$, where $\varepsilon_n = \sqrt{1-\rho^2}(\rho Z_{n-1} + V_n)$.

Note that ε_n is normal with mean 0 and variance $\sigma_\varepsilon^2 = 1 - \rho^4$. Stochastic processes of the form $X_n = aX_{n-1} + \varepsilon_n$ is known as autoregressive of order one (or AR(1) for short). They are known to converge in distribution to a limit if $|a| < 1$. Take this result for granted. Explain why it applies here.

(c) Why is $E(X_n) = \rho^2 E(X_{n-1})$? Use this to establish that $E(X_n) = \rho^{2n}\mu_0$.

(d) Show that $\text{var}(X_n) = \rho^4 \text{var}(X_{n-1}) + \sigma_\varepsilon^2$. Since $\text{var} X_0 = 0$, this yields

$$\text{var}(X_n) = \frac{\sigma_\varepsilon^2}{1-\rho^4}(1 - \rho^{4n}).$$

Prove it.

(e) What is the limit for $E(X_n)$ and $\text{var}(X_n)$ when $n \to \infty$? Insert for σ_ε^2.

(f) Explain by reason of symmetry that the same results applies to $\{Y_n\}$.

(g) Show that $E(X_n Y_n) = \rho E(X_n^2)$ and use this to show that $E(X_n Y_n)$ converges to the right value. (Note that in this case $E(X_n Y_n) = \text{corr}(X_n, Y_n)$.)

(h) Summarize your findings. What is the limit distribution of (X_n, Y_n)? Discuss the convergence speed. What is its dependence on ρ?

Exercise 4
Consider the following state space model:

$x_t = \phi x_{t-1} + \varepsilon_t$	state equation
$y_t \sim \text{Poisson}(\exp\{1 + x_t\})$	observation equation

where x_0 and $\varepsilon_1, \varepsilon_2, ...$ are independent and standard normal distributed. We want to estimate ϕ based on observations $y_1, ..., y_T$. We will do this in a Bayesian way and assume we have a prior distribution $N(0, \sigma_\phi^2)$ on ϕ.

A possible way to estimate ϕ in such situations is to extend the state model to the following model:

$$\begin{aligned}
\phi_t =& \phi_{t-1} && \text{state equation 1}\\
x_t =& \phi_{t-1}x_{t-1} + \varepsilon_t && \text{state equation 2}\\
y_t \sim& \text{Poisson}(\exp\{1 + x_t\}) && \text{observation equation}
\end{aligned}$$

where $\phi_0 \sim N(0, \sigma_\phi^2)$. Non-linear filters try to compute the posterior distribution for (ϕ_t, x_t) based on $y_1, ..., y_t$. Since $\phi = \phi_T$, the posterior distribution for (ϕ_T, x_T) based on $y_1, ..., y_T$ gives us the posterior distribution for ϕ given $y_1, ..., y_T$.
Simulation methods for non-linear filters can therefore be used on the bivariate state vector (ϕ_t, x_t).

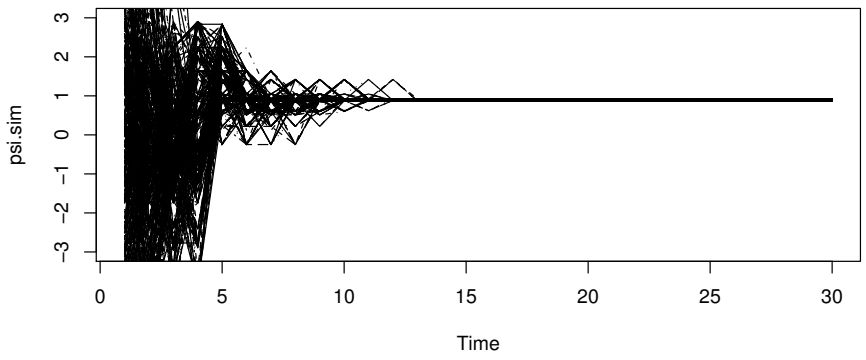
(a) Explain the general principles behind sequential importance sampling (SIS).

Discuss why resampling in general is important in connection to SIS algorithms.

(b) Simulations from the posterior distribution for (ϕ_t, x_t) based on $y_1, ..., y_t$ was performed through the following SIS algorithm:

- Draw $\tilde{x}_t^j$ from $N(\phi_{t-1}^j x_{t-1}^j, 1)$ for $j = 1, ..., M$
- Put $\widetilde{\phi}_t^j = \phi_{t-1}^j$ for $j = 1, ..., M$.
- Calculate the weights $w_t^j = p(y_t | x_t = \tilde{x}_t^j)$ for $j = 1, ..., M$ and the normalized weights $q_t^j = w_t^j / \sum_{j'} w_t^{j'}$.
- Draw $(x_t^1, \phi_t^1), ..., (x_t^M, \phi_t^M)$ from $\{(\tilde{x}_t^1, \tilde{\phi}_t^1), ..., (\tilde{x}_t^M, \widetilde{\phi}_t^M)\}$ with replacement and with probabilities $q_t^1, ..., q_t^M$.

The figure below shows simulations of ϕ_t for $t = 1, ..., T$ based on a SIS algorithm with resampling. Each curve corresponds to a sequence of simulated ϕ's, $\phi_1^j, ..., \phi_T^j$. The different simulations $\phi_t^j, j = 1, ..., M$ for a fixed t are (approximately) from the posterior distribution for ϕ_t given $y_1, ..., y_t$. Here $T = 30$ and $M = 50$.



Why do the number of different values of the simulated ϕ's decrease with t? What kind of problems do this make in the estimation of ϕ?

(c) A more efficient algorithm can be obtained by integrating out the unknown ϕ when simulating the x-process.

One can show (you do not have to do this) that

$$p(\phi|x_1, ..., x_t, y_1, ..., y_t) = N(\hat{\phi}_t, \sigma_t^2)$$

where

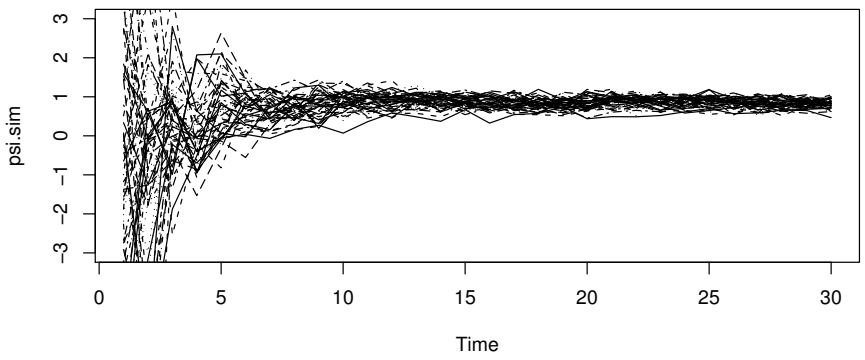
$$\hat{\phi}_t = \frac{\sigma_\phi^2 \sum_{i=2}^t x_i x_{i-1}}{1 + \sigma_\phi^2 \sum_{i=2}^t x_{i-1}^2}, \quad \sigma_t^2 = \frac{\sigma_\phi^2}{1 + \sigma_\phi^2 \sum_{i=2}^t x_{i-1}^2}$$

Use this to explain how you can simulate from the distribution $p(x_{t+1}|x_1, ..., x_t, y_1, ..., y_t)$.

(d) Consider now the SIS algorithm (with resampling) which at time t goes through the following steps:

- Draw $\tilde{x}_t^j$ from $p(x_t|x_1, ..., x_{t-1}, y_1, ..., y_{t-1})$ for $j = 1, ..., M$.
- Calculate the weights $w_t^j = p(y_t|x_t = \tilde{x}_t^j)$ for $j = 1, ..., M$ and the normalized weights $q_t^j = w_t^j / \sum_{j'} w_t^{j'}$.
- Draw $x_t^1, ..., x_t^M$ from $\{\tilde{x}_t^1, ..., \tilde{x}_t^M\}$ with replacement and the probabilities $q_t^1, ..., q_t^M$.
- Draw $\phi_t^j \sim p(\phi|x_1^j, ..., x_t^j, y_1, ., y_t)$

The figure below shows simulations of ϕ_t based on this algorithm.



Which advantages does this algorithm have compared to the one given in (b)?

In order to estimate the posterior expectation of ϕ, is it necessary to simulate the ϕ's at all? If not, explain how inferense on ϕ then can be performed. What is this technique called?

Exercise 5 (Weight loss programme)
Venables and Ripley [1999] contain a dataset (originally from Dr. T Davies) describing weights (y_i) of obese patients after different number of days (x_i)

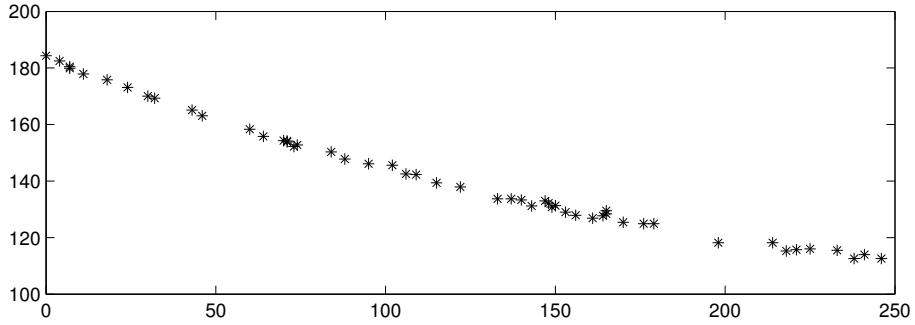


Figure 1: Weight loss from an obese patient

since start of a weight reduction programme. The data is plotted in Figure 1. Venables and Ripley [1999] suggests the following model for this dataset:

$$y_i = \beta_0 + \beta_1 e^{-\beta_2 x_i} + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

for $i = 1, ..., n$. Here $\boldsymbol{\theta} = (\beta_0, \beta_1, \beta_2, \sigma^2)$ is a set of parameters that needs to be estimated. Estimation will be based on maximum likelihood, i.e. maximizing

$$l(\boldsymbol{\theta}) = -\frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 e^{-\beta_2 x_i})^2$$

(a) Show that maximisation with respect to $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)$ is equivalent to minimizing

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 e^{-\beta_2 x_i})^2.$$

What is the optimal value for σ^2 given $\boldsymbol{\beta}$?

(b) Describe Newton's method and perform the calculations needed to implement this algorithm.
For reference, the following results where obtained using Newton's method in this case. Note however that for small perturbations of these starting values, numerical problems occured.

Iteration s	$\beta_0^{(s)}$	$\beta_1^{(s)}$	$\beta_2^{(s)}$	$(\sigma^2)^{(s)}$	$l^{(s)}$)
0	90.000	95.000	0.0050000	209.386	-143.259
1	84.339	100.551	0.0051991	0.72866	-69.670
2	76.350	107.131	0.0044158	1.47380	-78.827
3	76.801	106.841	0.0045417	0.65652	-68.314
4	81.664	102.393	0.0048807	0.60634	-67.281
5	81.399	102.662	0.0048866	0.56958	-66.468
6	81.374	102.684	0.0048844	0.56958	-66.468
7	81.374	102.684	0.0048844	0.56958	-66.468

(c) A run with Fisher's scoring algorithm with the same starting values as above gave the following results:

Iteration s	$\beta_0^{(s)}$	$\beta_1^{(s)}$	$\beta_2^{(s)}$	$(\sigma^2)^{(s)}$	$l(\boldsymbol{\beta}^{(s)}, (\sigma^2)^{(s)})$
0	90.000	95.00	0.0050000	209.39	-143.259
1	81.400	102.66	0.0048760	0.5758	-66.609
2	81.374	102.68	0.0048844	0.5696	-66.468
3	81.374	102.68	0.0048844	0.5696	-66.468
4	81.374	102.68	0.0048844	0.5696	-66.468

Give a general description of this algorithm and discuss its benefits compared to Newton's method

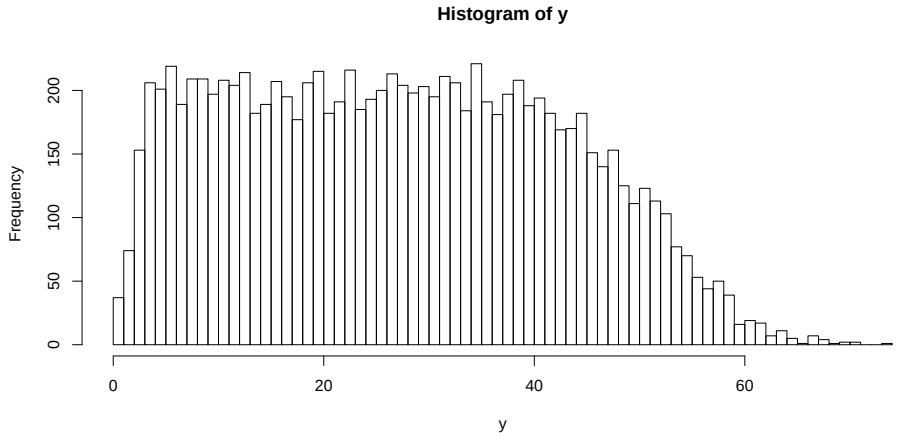
(d) Show that given β_2, the maximum values for all the other parameters can be found analytically.

What benefits do this result have with respect to optimisation?

Exercise 6
Consider a mixture model where

$$\Pr(C_i = k) = \pi_k$$
$$\Pr(y_i = y | C_i = k) = \frac{\lambda_k^{y_i} e^{-\lambda_k}}{y_i!}$$

Our aim is to obtain maximum likelihood estimates of $\boldsymbol{\theta} = \{(\pi_k, \lambda_k), k = 1, ..., K\}$ based on observations $\boldsymbol{y} = (y_1, ..., y_n)$.
The histogram below shows a simulated dataset with $K = 10$ classes and $n = 10\,000$.



(a) Write down the likelihood function based on the observations $\boldsymbol{y}$.

A call to a general optimiser using the Nelder-Mead algorithm gave the following estimates:

k	1	2	3	4	5	6	7	8	9	10
$\hat{\pi}_k$	0.107	0.000	0.146	0.164	0.049	0.126	0.004	0.202	0.183	0.020
$\hat{\lambda}_k$	5.18	8.26	11.32	19.26	27.75	28.01	28.27	37.17	47.85	48.83

with a log-likelihood value equal to 40573.98, obtained after 502 function calls.

Describe short the main features of the Nelder-Mead algorithm.

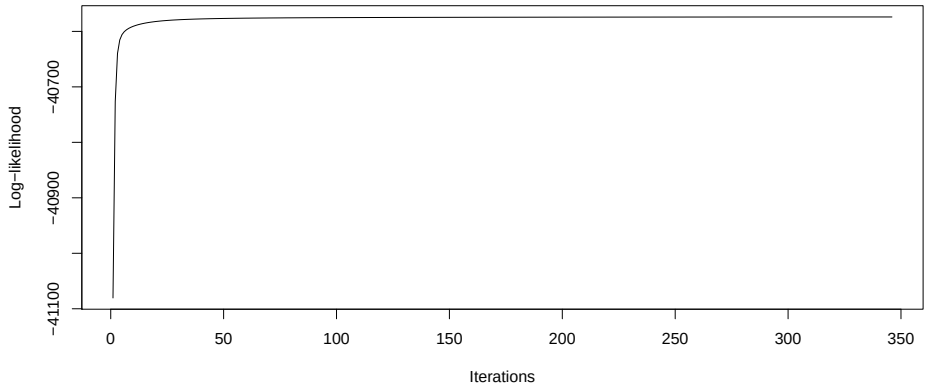
(b) An alternative to a direct optimiser is to use the EM algorithm where we treat $\{c_i\}$ as missing variables. Derive the updating equations for the parameters involved in this case.

The plot below shows the log-likelihood values at different iterations based on the EM algorithm.

Further, the final estimates obtained in this case is given in the table below, obtained after 346 iterations with a final log-likelihood value of 40573.98.

Explain the result in the figure with respect to properties of the EM algorithm.

k	1	2	3	4	5	6	7	8	9	10
$\hat{\pi}_k$	0.110	0.136	0.097	0.107	0.103	0.025	0.149	0.074	0.119	0.081
$\hat{\lambda}_k$	5.24	11.31	17.56	22.30	27.80	31.13	35.04	40.76	46.68	49.32



(c) Comparing the results from the two algorithms, the estimates appears to be quite different. However, the log-likelihood values are quite similar. Try to give an explanation on this.

Exercise 7

Cortez et al. [2009] considers a dataset of red wine quality of the Portuguese "Vinho Verde" wine. The following variables are measured:

Input variables (based on physicochemical tests): x_1 - fixed acidity, x_2 - volatile acidity, x_3, - citric acid, x_4 - residual sugar, x_5 - chlorides, x_6 - free sulfur dioxide, x_7 - total sulfur dioxide, x_8 - density, x_9 - pH, x_{10} - sulphates, x_{11} - alcohol.

Output variable (based on sensory data): y - quality (score between 0 and 10).

A simple model for the output quality is

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i$$

with $p = 11$ and $i = 1, ..., n = 1599$. This model is including all the variables as linear terms. In practice however, we would like to perform some kind of

model selection. One way of describing possible submodels is

$$y_i = \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j x_{ij} + \varepsilon_i$$

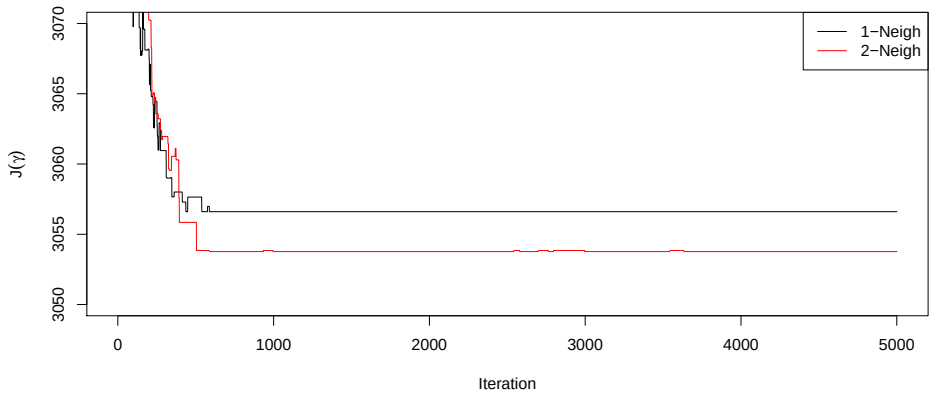
where $\gamma_j = 1$ if the covariate is to be included into the model and 0 otherwise. Define $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p)$. Our aim will be to minimise $J(\boldsymbol{\gamma})$ where

$$J(\boldsymbol{\gamma}) = -2 * \ell(\gamma) + 2 * (2 + \sum_{j=1}^{p} \gamma_j)$$

and $\ell(\gamma)$ is the log-likelihood value obtained by selecting the optimal $\boldsymbol{\beta}$ values for the given model.

(a) The figure below shows the results from two different versions of simulated annealing, the first changing one γ_j at a time, the second also allowing for two changes at a time. For the first version, one component is selected at random at each iteration. For the second version, first a random selection on whether to change one or two variables is made, thereafter the components to change are selected at random.

Give a short description of the simulated annealing algorithm. Discuss in particular the use of neighborhoods and relate that to the figure below.



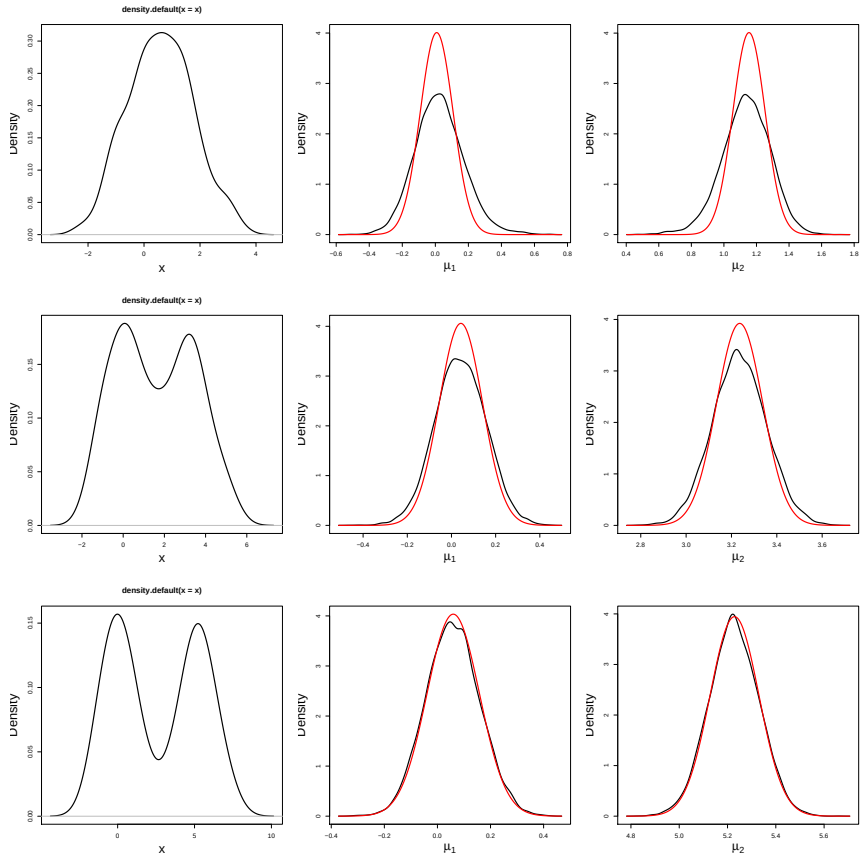
(b) Assume that the temperature is selected to be very large. What kind of algorithm does then appear?

On the other hand, if the temperature is kept fixed, what kind of algorithm do we then obtain?

Exercise 8
Consider a mixture model where

$$\begin{aligned} \Pr(C_i = k) &= \pi_k, & k = 1, 2 \\ p(x_i | C_i = k) &= N(\mu_k, \sigma_k^2) \\ p(\mu_k) &= N(0, \sigma_\beta^2), & k = 1, 2 \end{aligned}$$

Our focus is now Bayesian inference where we are interested in the posterior distribution $p(\boldsymbol{\mu}|\boldsymbol{y})$ with $\boldsymbol{\mu} = (\mu_1, \mu_2)$ and $\boldsymbol{x} = (x_1, ..., x_n)$.
The plots below compare a variational inference approximation with output from a simple Metropolis-Hastings algorithm (where in each case the first half is discarded) for different simulated datasets (where the μ_k values differ). In each row, the first plot shows the estimated density of the observations $\boldsymbol{x}$ while the two next plots shows the estimates of $p(\mu_1|\boldsymbol{x})$ and $p(\mu_2|\boldsymbol{x})$ with the black lines corresponding to output from Metropolis-Hastings while the red lines correspond to the variational inference approximation.



(a) The variational approximation is based on a mean-field assumption with Gaussian distributions assumed for each variable of interest.

Describe what kind of assumptions the mean-field approximation is based on. Also specify which parameters that needs to be fitted in the variational approximation approach.

(b) Discuss the results seen in the figure and relate this to the assumptions made for the variational approximation.

References

P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4):547–553, 2009.

W. N. Venables and B. D. Ripley. Modern Applied Statistics with S-plus. Statistics and Computing. Springer Verlag, third edition, 1999.