UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in:	STK4051 — Computational statistics
Day of examination:	Tuesday May 28 2019.
Examination hours:	14.30 – 18.30.
This problem set consists of 5 pages.	
Appendices:	None
Permitted aids:	None

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

Some distributions that will be used:

$$\phi(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \qquad \text{Gaussian distribution}$$

$$\Phi(x;\mu,\sigma) = \int_{-\infty}^{x} \phi(z;\mu,\sigma)dz \qquad \text{Cummulative Gaussian}$$

$$\text{Gamma}(x,\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \qquad \text{Gamma distribution}$$

Problem 1

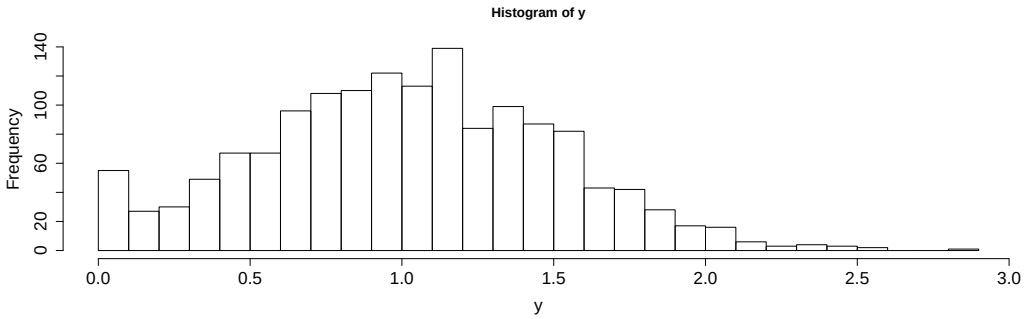
Consider a model

$$x_i \sim N(\mu,\sigma^2), \quad i = 1,...,n$$

$$y_i = \begin{cases} x_i & \text{if } x_i > 0; \\ 0 & \text{otherwise.} \end{cases}$$

We only observe $y_i, i = 1,...,n$ and we want to estimate μ and σ^2.
The histogram below shows $n = 1500$ observations simulated from such a model.

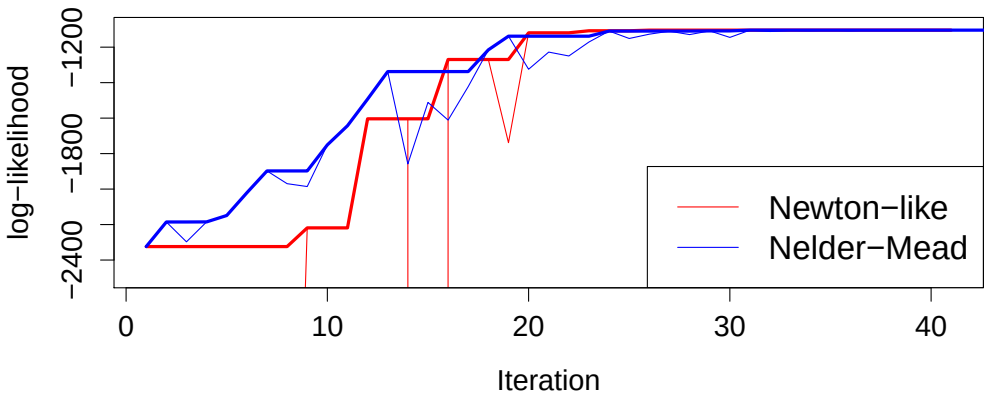
(Continued on page 2.)



(a) Assume that the observations are sorted so that the first n_1 observations are positive while the next $n_2 = n - n_1$ observations are zero. Write down the log-likelihood for $\boldsymbol{\theta} = (\mu, \sigma^2)$ in this case.

Why is is reasonable to reparametrize to $\rho = \log(\sigma^2)$ if one wants to optimize this function?

(b) The plot below shows results obtained by applying a Newton-like ascent method and a Nelder-Mead algorithm to optimize the log-likelihood. The thick lines shows the best values obtained so far during the iterations. The (almost) vertical red lines correspond to iterations where the log-likelihood value was very low and outside the limits of the plot. Iterations here are described as each call to the (log)-likelihood function.



Describe shortly these algorithms with a focus on their main differences, advantages and disadvantages.

(c) Consider now the use of the EM algorithm in this case where the x_i's can be considered as the complete data.

Write down an expression for the log-likelihood for the complete data.

Assume that you have functions available that can calculate

$$m_p(\mu, \sigma^2) = E[x^p | x < 0], \quad p = 1, 2$$

(Continued on page 3.)

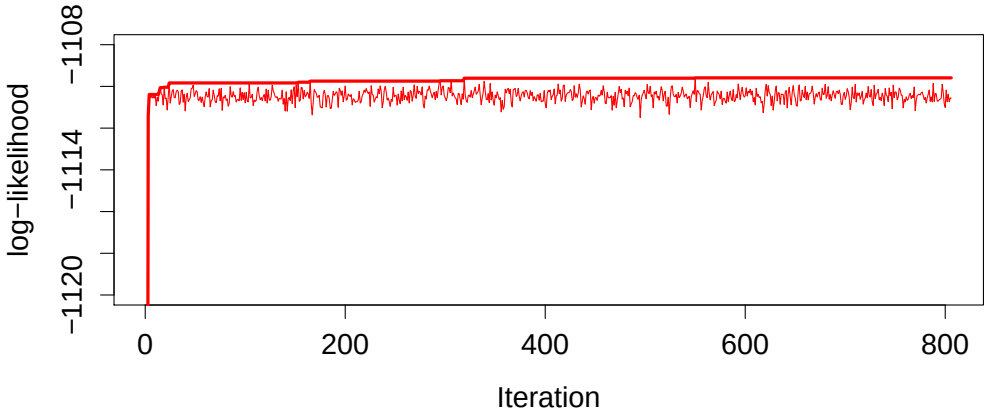
for $x \sim N(\mu, \sigma^2)$ and for any values of μ and σ^2.

Derive updating equations for μ and σ^2 expressed by the m_p terms.

(d) Calculating the m_p values exactly can be a bit problematic, but can be approximated by Monte Carlo simulation.

Describe how such estimates can be achieved.

Below is a plot showing the values of the log-likelihood at different iterations of the EM algorithm with Monte Carlo estimates of m_p inserted. Try to explain the behaviour of this modified EM algorithm (also here the think line corresponds to the best values so far).



Problem 2

Assume that we have two transition densities $P_1(\boldsymbol{z}|\boldsymbol{x})$ and $P_2(\boldsymbol{z}|\boldsymbol{x})$, each defining a Markov chain and transitions from $\boldsymbol{x}$ to $\boldsymbol{z}$. Assume that each of the transition densities make a target density $\pi(\boldsymbol{x})$ invariant, that is

$$\int_{\boldsymbol{x}} \pi(\boldsymbol{x}) P_j(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{x} = \pi(\boldsymbol{z}), \quad j = 1, 2.$$

Define now a new transition density $P(\boldsymbol{z}|\boldsymbol{x})$ where the transition densities are obtained by first drawing $\boldsymbol{y} \sim P_1(\boldsymbol{y}|\boldsymbol{x})$ and then drawing $\boldsymbol{z} \sim P_2(\boldsymbol{z}|\boldsymbol{y})$.

(a) Show that also this new transition density keeps $\pi(\boldsymbol{x})$ invariant.

(b) Assume now that $\boldsymbol{x} = (x_1, x_2)$ and that P_1 changes x_1 by drawing x_1 from $\pi(x_1|x_2)$ while P_2 changes x_2 by drawing x_2 from $\pi(x_2|x_1)$.

Show that both P_1 and P_2 keep $\pi(x_1, x_2)$ invariant.

What does this imply for the systematic scan Gibbs sampler?

(Continued on page 4.)

(c) Assume now that we want to use a Metropolis-Hasting algorithm, but use a proposal distribution where we draw $x_1^* \sim \pi(x_1|x_2)$ and put $x_2^* = x_2$. What is the Metropolis-Hastings ratio in this case?

What does this say about the relationship between the Gibbs sampler and the Metropolis-Hastings algorithm?

(d) Assume now we have independent data $x_i \sim N(\mu, \sigma^2)$ for $i = 1, ..., n$. Assume further that we want to perform Bayesian inference on $\boldsymbol{\theta} = (\mu, \tau)$ where $\tau = 1/\sigma^2$. We assume a prior for $\boldsymbol{\theta}$ as

$$p(\boldsymbol{\theta}) \propto \tau^{\alpha-1} e^{-\beta\tau}.$$

Derive the Gibbs sampler algorithm in this case, that is show how μ and τ are simulated at each step.

Problem 3

Consider a nonlinear regression model

$$y_i = f(\boldsymbol{x}_i; \boldsymbol{\beta}) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

where $f(\boldsymbol{x}; \boldsymbol{\beta})$ is some non-linear function parametrized by $\boldsymbol{\beta}$. We will focus on the estimation of $\boldsymbol{\beta}$. The objective function that we want to minimize is

$$g(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i - f(\boldsymbol{x}_i; \boldsymbol{\beta})]^2 + J(\boldsymbol{\beta})$$

where $J(\boldsymbol{\beta})$ is some penalty function, e.g. $\lambda \sum_j \beta_j^2$. A gradient based method for minimizing $g(\boldsymbol{\beta})$ is

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \alpha \nabla g(\boldsymbol{\beta})$$

where $\nabla g(\boldsymbol{\beta})$ is the vector of partial derivatives with respect to the β_j's. We will focus on a setting where the set of observations $\{(y_i, \boldsymbol{x}_i), i = 1, ..., n\}$ is very large such that computation of $\nabla g(\boldsymbol{\beta})$ can be very costly.

(a) Describe the stochastic gradient algorithm and also describe why such an algorithm can be usefull in this setting.

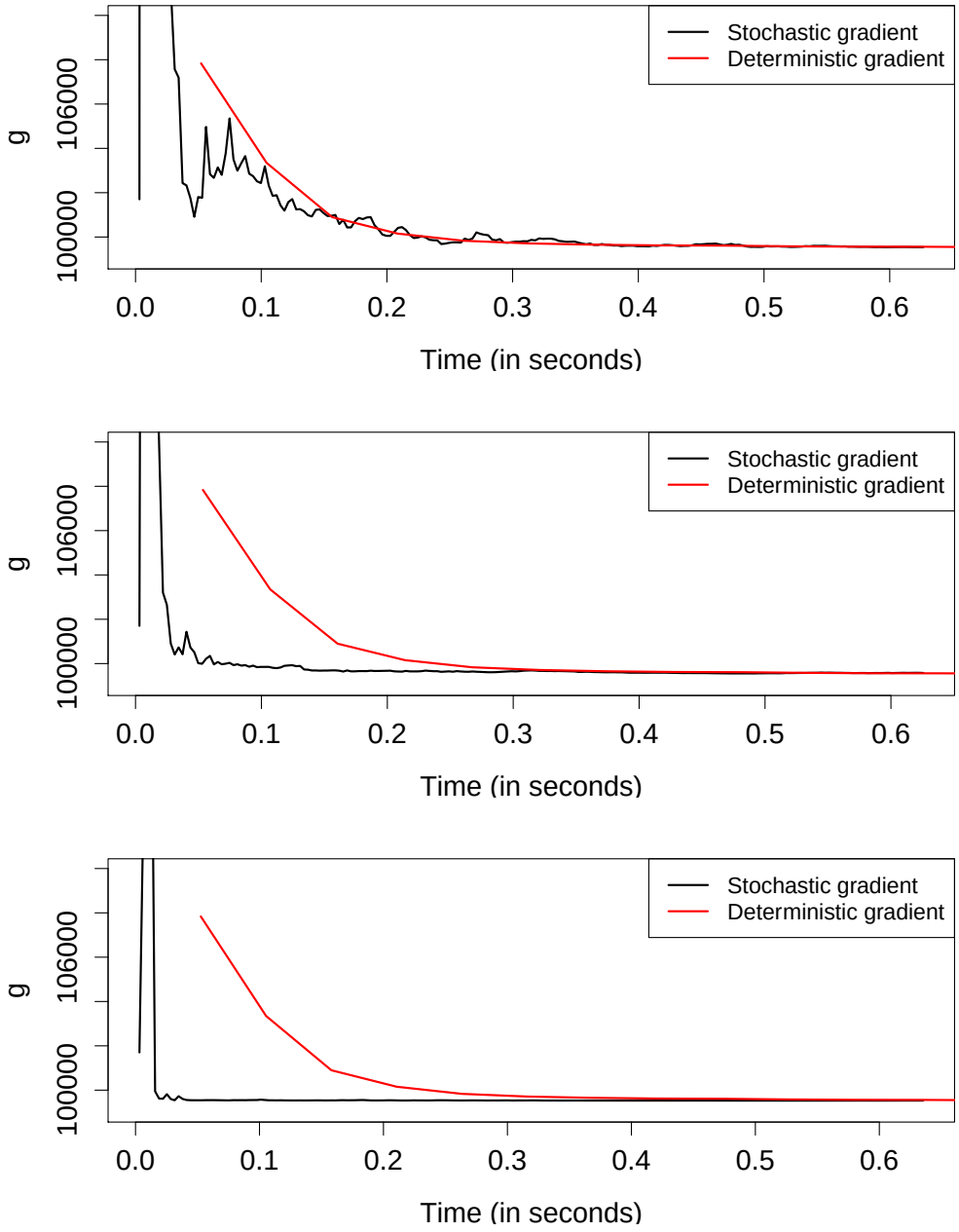
(b) Specify the conditions needed for the learning rate and discuss why these conditions are reasonable for the algorithm to converge.

(c) A popular non-linear regression model is defined by

$$f(x; \boldsymbol{\beta}) = \frac{\beta_1}{1 + \exp(\beta_2 + \beta_3 x)}.$$

(Continued on page 5.)

$n = 100\,000$ observations were simulated from this model. The plots below compares a stochastic gradient algorithm with a deterministic gradient based method. The x-axis shows the cpu time used while the y axis shows the corresponding values of the g function. Note that for the deterministic gradient method, at each iteration α is divided by 2 until a smaller g value is obtained. The different plots are obtained by using different minibatch sizes.



Explain what we mean by minibatch sizes.

Specify which plot that corresponds to the largest minibatch size and which belongs to the smallest minibatch size.