Examination Computational Statistics

Linköpings Universitet, IDA, Statistik

Course:	732A90 Computational Statistics
Date:	2021/09/23, 8–13
Teacher:	Krzysztof Bartoszek
	During the exam please contact Hao Chi Kiang (hao.chi.kiang@liu.se)
Provided aids:	material in the zip file **exam_material_732A90.zip**
Grades:	A= $[18-20]$ points
	B= $[16-18)$ points
	C= $[14-16)$ points
	D= $[12-14)$ points
	E= $[10-12)$ points
	F= $[0-10)$ points
Instructions:	Provide a detailed report that includes plots, conclusions and interpretations.
	If you are unable to include a plot in your solution file clearly indicate the section of R code that generates it.
	Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in an appendix.
	In a number of questions you are asked to do plots. Make sure that they are informative, have correctly labelled axes, informative axes limits and are correctly described. Points may be deducted for poorly done graphs.
	Name your solution files as: **[your id]_[own file description].[format]**
	If you have problems with creating a pdf you may submit your solutions in text files with unambiguous references to graphics and code that are saved in separate files.
	There are **TWO** assignments (with sub–questions) to solve.
	Provide a separate solution file for each assignment.
	Include all R code that was used to obtain your answers in your solution files.
	Make sure it is clear which code section corresponds to which question.
	If you also need to provide some hand–written derivations please number each page according to the pattern: Question number . page in question number i.e. Q1.1, Q1.2, Q1.3,. . ., Q2.1, Q2.2, . . ., Q3.1, ...
	Scan/take photos of such derivations preferably into a single pdf file but if this is not possible multiple pdf or .bmp/.jpg/.png files are fine.
	Please do not use other formats for scanned/photographed solutions.
	Please submit all your solutions via LISAM or e–mail. If emailing, please email them to **BOTH** krzysztof.bartoszek@liu.se and KB_LiU_exam@protonmail.ch .
	During the exam you may ask the examiner questions by emailing them to KB_LiU_exam@protonmail.ch **ONLY**. Other exam procedures in LISAM.

NOTE: If you fail to do a part on which subsequent question(s) depend on describe (maybe using dummy data, partial code e.t.c.) how you would do them given you had done that part. You *might* be eligible for partial points.

Assignment 1 (10p)

Lagrange's theorem states that for a continuous function $f : [a, b] \to \mathbb{R}$ that is differentiable on (a, b) there exists a $c \in (a, b)$ such that

$$f(b) - f(a) = f'(c)(b - a). \tag{1}$$

Unfortunately the theorem does not tell us how to find this c.

Question 1.1 (4p)

Implement a golden ratio search procedure that for user provided function f and constants a and b finds c satisfying Eq. (1). The search procedure should be implemented as a function that takes f, a and b and anything else necessary as parameters. Provide code showing how your implementation works for some example f, a and b.
TIP: Remember that a derivative can be numerically approximated as

$$f'(x) \approx \frac{f(x + \Delta) - f(x)}{\Delta}$$

for small Δ.

Question 1.2 (2p)

You are given the function $f(x) = \sin^2(x) + \cos^2(x)$. For $a = -\pi$ and $b = \pi$ use your implemented optimization procedure to find all c satisfying Eq. (1). Alternatively, you may find the cs analytically, show all derivations.

Question 1.3 (4p)

For $f(x) = \sin^2(x) + \cos^2(x)$ compare your implemented optimization with R's `optim()`? What starting points of the optimization do you need to take in both cases? Is your solution unique?

Assignment 2 (10p)

We consider the following statistical model for Bernoulli (i.e. taking on values 0 and 1) random variables $Y_1, \ldots, Y_n$. We have that $P(Y_i = 1) = F(\vec{x}_i^T \vec{\beta})$, where $\vec{x}_i \in \mathbb{R}^p$ is vector of known predictors associated with Y_i and $\vec{\beta} \in \mathbb{R}^p$ is vector of unknown regression parameters. The function F is defined as

$$F(x) = \frac{e^x}{1 + e^x}.$$

Under this model the joint likelihood is

$$\prod_{i=1}^{n} P(Y_i = y_i | \vec{\beta}) = \prod_{i=1}^{n} \Big([F(\vec{x}_i^T \vec{\beta})]^{y_i} [1 - F(\vec{x}_i^T \vec{\beta})]^{(1-y_i)} \mathbf{1}_{\{0,1\}}(y_i). \Big) \tag{2}$$

The goal of the inference procedure is to find $\vec{\beta}$.

Question 2.1 (3p)

The prior distribution for $\vec{\beta}$ is $\mathcal{N}_p(\vec{b}, \mathbf{B})$ for some vector $\vec{b}$ and symmetric–positive–definite matrix $\mathbf{B}$. Choose some even $p > 2$, vector $\vec{b}$ and $\mathbf{B}$. Sample a random vector $\vec{\beta}$ from this prior distribution. You may only use `runif()` to sample.
TIP: The function `chol()` might be useful.
Choose a large n (but not too large so that the matrix operations in Q2.3 are doable) and using whatever random generator in `R` generate the $n \times p$ values of predictors, i.e. the vectors $\vec{x}_i$.

Question 2.2 (2p)

Using the sampled $\vec{\beta}$ and $\{\vec{x}_i\}$ randomly draw the binary vector of $y_1, \ldots, y_n$ values according to Eq. (2).

Question 2.3 (5p)

The aim now is to implement a Gibbs sampler (actually this particular one is called a Pólya–Gamma Gibbs sampler) that will recover $\vec{\beta}$. Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be the matrix whose i–th row is $\vec{x}_i$, denote by $\mathbf{D}(\vec{w})$ the diagonal matrix with the vector $\vec{w}$ on its diagonal. Define for $\vec{w} \in \mathbb{R}^n_+$

$$\mathbf{\Sigma}(\vec{w}) = (\mathbf{X}^T \mathbf{D}(\vec{w}) \mathbf{X} + \mathbf{B}^{-1})^{-1}$$

and

$$\vec{\mu}(\vec{w}) = \mathbf{\Sigma}(\vec{w}) \left[\mathbf{X}^T \left(\vec{y} - \frac{1}{2} \vec{1}_n \right) + \mathbf{B}^{-1} \vec{b} \right],$$

where $\vec{y} = (y_1, \ldots, y_n)^T$ for your previously simulated $Y_1, \ldots, Y_n$ random variables and $\vec{1}_n$ is a vector of n ones. The $m+1$ iteration of the Pólya–Gamma Gibbs sampler is as follows:
1. Draw $W_1, W_2, \ldots, W_n$ independently with

$$W_i \sim PG(1, |\vec{x}_i^T \vec{\beta}^{(m)}|),$$

where $PG(h, z)$ is the Pólya–Gamma distribution. Sampling from it can be done using `BayesLogit::rpg(num,h,z)`. Call the sampled vector $\vec{w} = (w_1, \ldots, w_n)^T$.
2. Draw

$$\vec{\beta}^{(m+1)} \sim \mathcal{N}_p(\vec{\mu}(\vec{w}), \mathbf{\Sigma}(\vec{w}))$$

Implement the Pólya–Gamma Gibbs sampler. For step 2. use the sampler you implemented in Q2.1. Use your implemented Pólya–Gamma Gibbs sampler to recover $\vec{\beta}$ from your $\vec{y}$ values. Provide plots of how it performs, how quickly it converges (or does not). Use in your plots that you know the true value of $\vec{\beta}$.
REMEMBER: Your sampler needs some initial guess for $\vec{\beta}$.