Advanced Computational Statistics – Spring 2023
Assignment for Lecture 3

Frank Miller, frank.miller@liu.se,
Department of Computer and Information Science, Linköpings University

April 4, 2023

Perform the solutions individually and send your report **until April 17** by email to me. Try to keep this deadline. However, if you have problems with it, there will be a final deadline on September 30 for all assignments. Please include your name in the filename(s) of your solution file(s). Please send me one **pdf-file** with your report (alternatively, Word is ok, too), and additionally, please send me your code in one separate **plain-text file** (an R-markdown, .rmd, is possible but not required).

From the following four problems, you can choose three. It is optional, if you want to do the fourth as well.

Problem 3.1

Let $g(\mathbf{x}), \mathbf{x} \in I\!R^3$ the multivariate normal mixture with

$$g(\mathbf{x}) = \sum_{i=1}^{4} w_i f_i(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

where $f_i(\cdot; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is the density of the multivariate normal distribution with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. We use here mixing weights $w_i = 1/4, i = 1, \ldots, 4$, mean vectors

$$\boldsymbol{\mu}_1 = (0,0,0)^\top, \quad \boldsymbol{\mu}_2 = (2,2,0)^\top, \quad \boldsymbol{\mu}_3 = (2,0,2)^\top, \quad \boldsymbol{\mu}_4 = (0,2,2)^\top,$$

and the covariance matrices $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \mathbf{I}$ are the identity matrix and $\boldsymbol{\Sigma}_4 = 0.98 \cdot \mathbf{I}$ (it is not difficult to program the density of the multivariate normal, especially here when covariance matrices are simple; alternatively a package like `mvtnorm` can be used).

a. Identify first the four local maxima by using different starting values for an algorithm of your choice. Which of the local maxima is the global maximum?

b. Use now the Particle Swarm Optimization in the `R`-function `psoptim` in package `pso` with the aim to identify the global maximum. Choose different values for the swarm size and for the average percentage of informants for each particle. (If you like, you can also investigate other parameters in the same way; but this is optional). You might consider another maximum number of iterations to keep the running times reasonable. Do not forget to define an appropriate search space.

   Run the PSO repeatedly (e.g. 100 times) for each chosen configuration of swarm size and average percentage of informants. Check if the result was close to the true global maximum. Based on this, identify values for these parameters which work well for maximisation of this function g.

Problem 3.2

An experiment was conducted investigating how the growth of garden cress depends on a (potentially) toxic fertilizer. The investigated range of fertilizer was between 0 and 1.2% concentration in the water. Growth of garden cress was measured in yield in mg after 5.5 days of growth. $n = 81$ experiments were conducted and the data is on the homepage in the file `cressdata.txt` (columns: observation number, fertilizer concentration, yield).

A fifth-degree polynomial is supposed to be fit to the data using least squares with L^1 regularisation (Lasso). The objective function to be minimised is

$$g(\boldsymbol{\beta}) = \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\tilde{\boldsymbol{\beta}}\|_1, \tag{1}$$

where $\mathbf{X}$ is the design matrix with columns 1, fertilizer, ..., fertilizer^5, $\tilde{\boldsymbol{\beta}} = (\beta_1, \ldots, \beta_5)^\top$ is the parameter vector without intercept, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_5)^\top$ is the complete parameter vector and $\mathbf{y}$ is the yield-data. (We do not regularise the intercept, see e.g. Lange (2010), page 310.) $\lambda \geq 0$ is a fixed regularisation constant (we do not determine it data-dependently here).

Note that regularisation in the situation of high degree polynomial models is illustrated in Chapter 5.2 of Goodfellow, Bengio and Courville (2016). Note further that $\lambda = 0$ corresponds to the least squares estimation, where the solution of the optimisation problem is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$.

a. Program yourself the Lasso objective function (1). Choose several algorithms to solve the optimisation problem; you can use e.g. some methods in `optim` or `psoptim`. Motivate choice of parameters and starting values used in these algorithms. Determine the Lasso estimator for $\lambda = 0, 10$ and two or three other λ-values with the chosen algorithms. Report also the objective value $g(\hat{\boldsymbol{\beta}})$ in order to compare the algorithms. Which algorithm is best, which are good?

b. Choose the estimated $\boldsymbol{\beta}$ of a good algorithm and plot the predicted regression functions for the chosen λ-values together with the data. Comment on the shape of the regression functions having the illustration in Chapter 5.2 in Goodfellow et al. (2016) in mind.

Problem 3.3

Consider a unidimensional minimisation problem where the minimum is attained at 0. Consider further a particle in the PSO algorithm with $x^{(1)} = 0$ and $x^{(2)} = -2$ ($v^{(2)} = -2$). For this particle, we have $p_{\text{best}}^{(t)} = g_{\text{best}}^{(t)} = 0$ for all t (the starting value happened to identify already the minimum; the stagnation assumption is fulfilled here). We consider a standard PSO with parameters w, $c_1 = c_2 =: c$ which generates the sequence $x^{(t)}, t = 1, 2, \ldots$ for this particle.

a. Compute the sequence $E[x^{(t)}]$ for iteration number $t = 1, 2, \ldots, 40$ and plot $E[x^{(t)}]$ versus t for different combinations of w and c. Use the pairs $(0.721, 1.193)$ (default in R-package `pso`), $(0.9, 1.193), (0.721, 2.2), (0.2, 3)$ for (w, c) and at least one further pair of your choice.

b. Simulate the sequence $x^{(t)}, t = 1, 2, \ldots, 40$, around 1000 times. Compute the Monte Carlo estimate for $\text{Var}(x^{(t)})$ for each $t = 3, \ldots, 40$ (which is simply the variance of the say 1000 simulated values for $x^{(t)}$). Plot the estimated variance versus iteration number. Do this for the (w, c)-pairs which you have used in a.

c. Based on your results from a. and the empirical results from b.: Can you confirm the theoretical results about order-1 and order-2 stability?

Problem 3.4

Consider planning of an experiment when a cubic regression model

$$y_i = \beta_0 + \beta_1 w_i + \beta_2 w_i^2 + \beta_3 w_i^3 + \varepsilon_i, \quad i = 1, \ldots, n,$$

is assumed. The predictor variable w_i might be chosen in the interval $[-1, 1]$. However, due to practical circumstances, a distance of 0.05 between design points is required and at most one observation can be made at each point. Therefore, we require that observations can only be made using $w \in \{-1, -0.95, -0.9, \ldots, 1\}$. Further, the sample size n might be chosen by the experimenter, but each observation has a cost. To balance between the higher information from more observations and their higher cost, a penalized D-optimality criterion should be optimised here:

$$\text{Minimise } n/5 - \log\{\det(X^T X)\}$$

where X is the design matrix having rows $(1, w_i, w_i^2, w_i^3)$ and log is the natural logarithm. An example function is provided in the file `crit_HA3.r` on the course homepage which computes this criterion.

a. Write your own simulated annealing algorithm for the problem described here. You need to choose a reasonable proposal distribution (possibly on some neighbourhood of a design) and a cooling schedule; you might need to test different options.

b. What is the optimal design in this case based on your optimisation?