Lab6

Cui Qingxuan, Nisal Amashan

2025-03-04

Contents

1 Collaborations

Nisal Amashan: Responsible for the question 1.

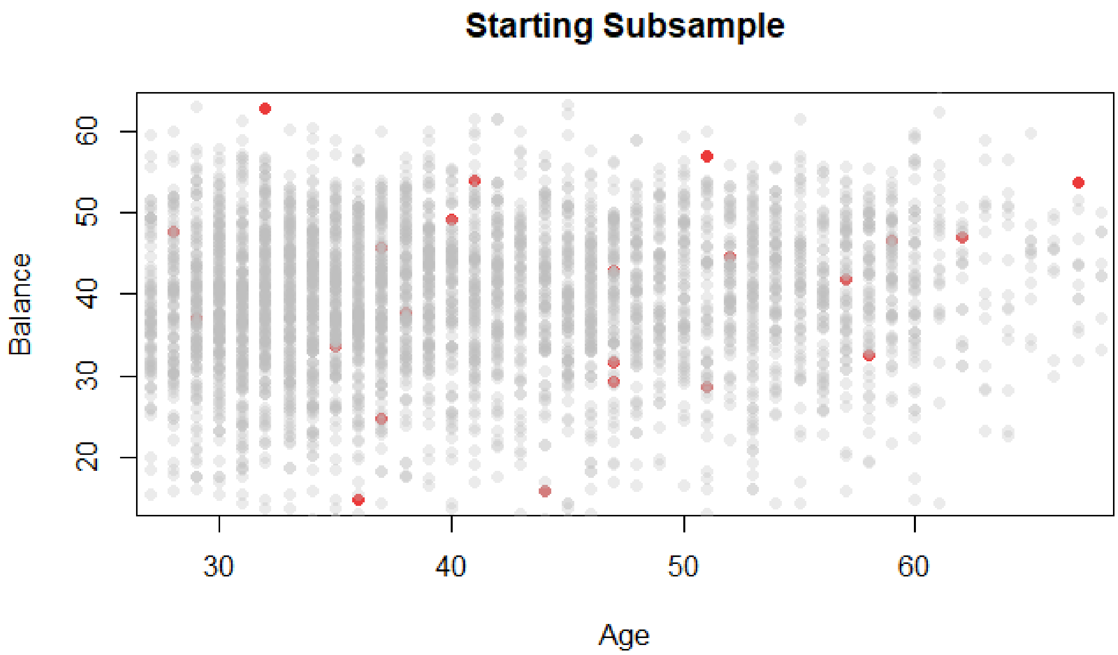
Cui Qingxuan: Responsible for the question 2.

2 Question 1

3 Question 2

3.1 Plot the Random Selected Data

A total of 22 data points were selected based on a uniform distribution. In the plot below, the selected points are highlighted in red, while the remaining data points from the bank dataset are shown in grey with transparency to provide context.



3.2 Implement Annealing Algorithm

After 50 annealing stages, the algorithm achieved a simulated minimum value of 16,625.77.

The parameters used in the simulated annealing process are as follows:

```
simulated_annealing(target_function=crit,
                    x_index=start_index,
                    temperature=80,
                    stages=50,
                    iteration_for_each_stage=10,
                    alpha=0.9,
                    beta=1.1,
                    data=bankdata,
                    annealing_method='linear')
```

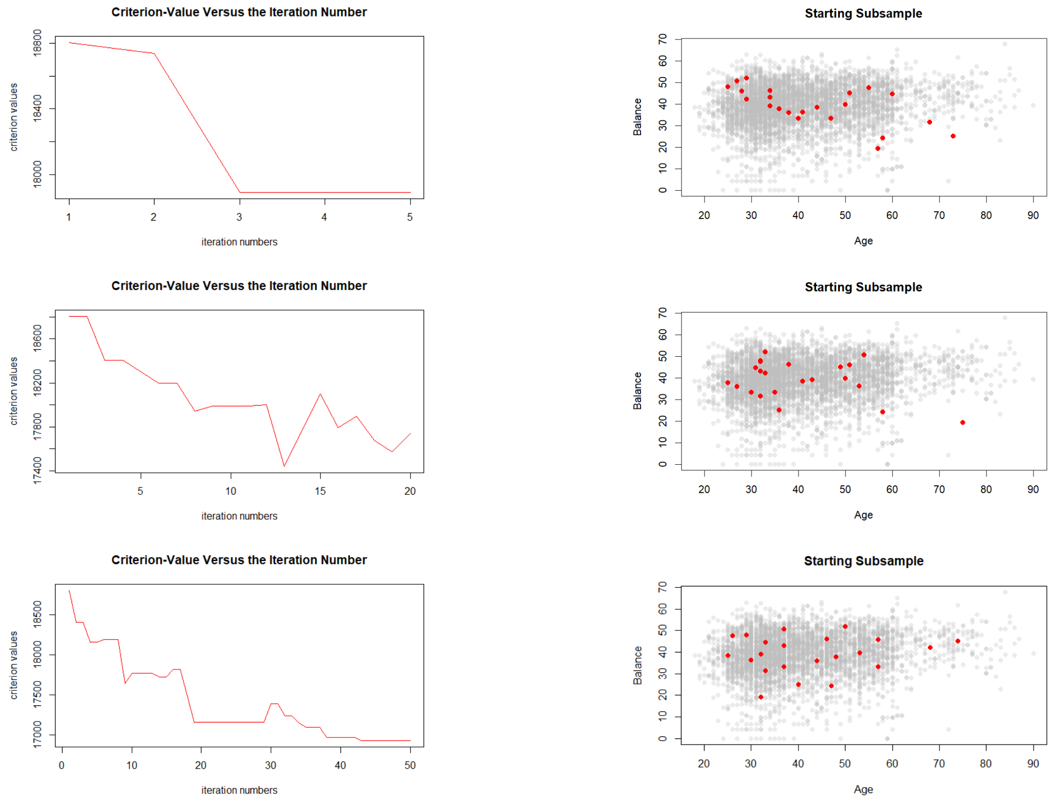
3.3 Compare Different Combinations

To evaluate the impact of different parameter settings, we tested three different configurations, as summarized in the table below. Among these, result 3 yielded the best optimization performance.

	result1	result2	result3
iteration	5	20	50
starting_temperature	20	50	80
anneaing_method	exp	log	linear
optimal_criterion	17864.32	17536.20	16625.77

Additionally, we visualize the results using the plots below. The left column presents the optimization curves, which show the criterion value as a function of iteration number, while the right column displays the corresponding selected 22 points for each configuration.

Optimization Curves & Distribution of Results



4 Appendix

```
##############################################################
### Computational statistics, Linköping University, VT2025 ###
### Criterion function Lab 6, Q2 (space filling design)    ###
### Frank Miller                                           ###
###                                                        ###
### We are using in this lab partial data from the         ###
### original bankdata available at                         ###
### https://archive.ics.uci.edu/ml/datasets/Bank+Marketing ###
### See also the publication:                              ###
### Sérgio Moro, P. Cortez, P. Rita (2014). A data-driven  ###
### approach to predict the success of bank telemarketing. ###
### Decision Support Systems.                              ###
##############################################################

# you need to save the following dataset at the right place and/or add/set the path where it is located
load("bankdata.Rdata")

nclients <- dim(bankdata)[1]  # number of individuals in the dataset, here 4364
```

```r
# criterion function: sum of minimal distance to an element in the subset
# dat is the full dataset (here: bankdata), subs is the set of ids for the subset selected
# subs should be a vector of elements in 1, ..., 4364; for this question, it should be of length 22
# example call: crit(bankdata, 1:22), selecting the first 22 individuals
# result of this function is the criterion to be minimized
crit <- function(dat, subs){
  s <- length(subs)
  dist <- matrix(rep(0, nclients*s), ncol=s)
  for (i in 1:s){
    dist[, i] <- sqrt((dat[,1]-dat[subs[i],1])^2+(dat[,2]-dat[subs[i],2])^2)
  }
  sum(apply(dist, 1, min))
}

# it is good to identify the individuals in the full set by their id (1, ..., 4364),
# then we can sample from this set for the starting subset:
fullset <- 1:nclients

# Plot the selected 22 points
set.seed(12345)
start_index = floor(runif(22, min=1, max=length(bankdata[,1])))
start_sub = bankdata[start_index, ]
plot(start_sub[, 1], start_sub[, 2], type="p", col="red", xlab="Age", ylab="Balance", main="Starting Su
points(bankdata[, 1], bankdata[, 2], col=adjustcolor("grey", alpha.f=0.3), pch=16)


# Simulated Annealing Algorithm
annealing = function(target, x_index, tem, iter, m, alpha, beta, data, annealing='exp'){
  # iter: j, stage numbers
  # t: starting temperature
  # target: target function, here is crit
  # proposal: proposal distribution, to generate the x_star
  # x_index: index of x0
  # m: number of iterations per stages
  # bankdata
  # alpha: the factor to update temperature
  # beta: the factore to update m
  # data: bankdata
  x_cri = c()
  for(j in c(1:iter)){
    x_cri = append(x_cri, crit(bankdata, x_index))
    for(t in c(1:m)){
      # sample a candidate xt from proposal distribution
      # we may need to generate 22 data points
      # shuffle
      candidate_index = sample(x_index)
      # exchange the first 5 index
      candidate_index[1:10] = sample(1:nrow(data), 10)

      # compute criteria function
      h = exp((crit(bankdata, x_index) - crit(bankdata, candidate_index)) / tem)

      # update xt+1
      p = min(h, 1)
      # x_index = ifelse(runif(1) < p, candidate_index, x_index)
      if(runif(1) < p) x_index = candidate_index
      # set t<-t+1 and next round to t==m
    }
  # update temperature t and m
    if(annealing == 'exp'){
      tem = alpha * tem
    }
    else if(annealing == 'log'){
      tem = tem / (1+log(alpha))
    }
    else if(annealing=='linear'){
      tem = tem - alpha
    }
    m = beta * m
  }
  return(list(x_cri, x_index))
}


result1=annealing(target=crit,
                 x_index=start_index,
                 tem=20,
                 iter=5,
                 m=10,
                 alpha=0.9,
                 beta=1.1,
                 data=bankdata,
                 annealing='exp')

result2=annealing(target=crit,
                 x_index=start_index,
                 tem=50,
                 iter=20,
                 m=10,
                 alpha=0.9,
                 beta=1.1,
                 data=bankdata,
                 annealing='log')

result3=annealing(target=crit,
                 x_index=start_index,
                 tem=80,
                 iter=50,
                 m=10,
                 alpha=0.9,
                 beta=1.1,
                 data=bankdata,
                 annealing='linear')


plot(c(1:5), result1[[1]], type='l', col='red', xlab='iteration numbers', ylab='criterion values', main=
plot(bankdata[, 1], bankdata[, 2], type="p", col=adjustcolor("grey", alpha.f=0.3), xlab="Age", ylab="Ba
points(bankdata[result2[[2]], 1], bankdata[result3[[2]], 2], col="red", pch=16)

plot(bankdata[, 1], bankdata[, 2], type="p", col=adjustcolor("grey", alpha.f=0.3), xlab="Age", ylab="Ba
points(bankdata[result1[[2]], 1], bankdata[result3[[2]], 2], col="red", pch=16)
plot(c(1:20), result2[[1]], type='l', col='red', xlab='iteration numbers', ylab='criterion values', mai

plot(bankdata[, 1], bankdata[, 2], type="p", col=adjustcolor("grey", alpha.f=0.3), xlab="Age", ylab="Ba
points(bankdata[result3[[2]], 1], bankdata[result3[[2]], 2], col="red", pch=16)
plot(c(1:50), result3[[1]], type='l', col='red', xlab='iteration numbers', ylab='criterion values', mai
```