06-12-2023

Hengde Ouyang

2023-06-12

1. Learning Rate Eta

1. Normally learning rate will range from 0 to 1. Setting a learning rate larger than 1 might not be appropriate.
   So next, I set some learning rates larger than 1 just for experiments, it might not be used in practice.

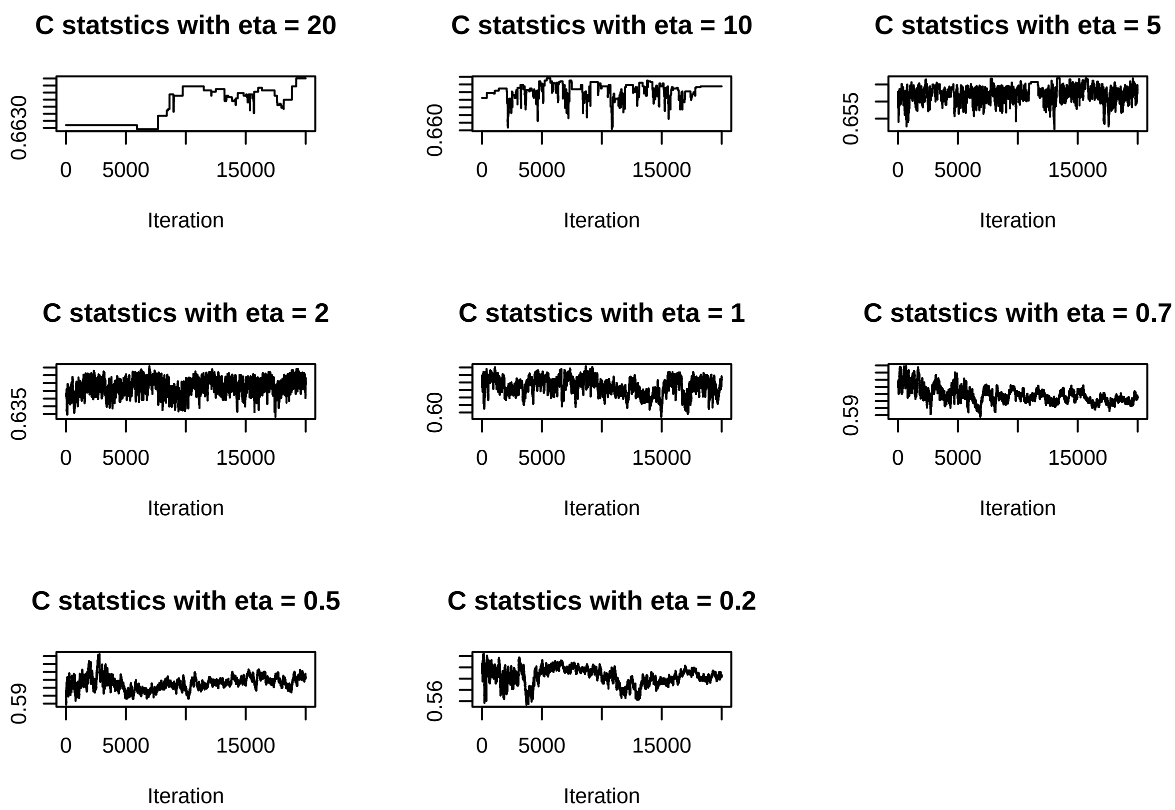
2. In our simulation, we try different eta

```
system.time({
  eta = 20*length(Y)
  result_lung1 = MH_Sampling(Y,delta,tau,
                       A,beta0,sigma0,
                       var.prop,m,
                       B,eta,Wmat_option)
  eta = 10*length(Y)
  result_lung2 = MH_Sampling(Y,delta,tau,
                       A,beta0,sigma0,
                       var.prop,m,
                       B,eta,Wmat_option)
  eta = 5*length(Y)
  result_lung3 = MH_Sampling(Y,delta,tau,
                       A,beta0,sigma0,
                       var.prop,m,
                       B,eta,Wmat_option)
  eta = 2*length(Y)
  result_lung4 = MH_Sampling(Y,delta,tau,
                       A,beta0,sigma0,
                       var.prop,m,
                       B,eta,Wmat_option)
  eta = 1*length(Y)
  result_lung5 = MH_Sampling(Y,delta,tau,
                       A,beta0,sigma0,
                       var.prop,m,
                       B,eta,Wmat_option)
  eta = 0.7*length(Y)
  result_lung6 = MH_Sampling(Y,delta,tau,
                       A,beta0,sigma0,
                       var.prop,m,
                       B,eta,Wmat_option)
  eta = 0.5*length(Y)
```

```
  result_lung7 = MH_Sampling(Y,delta,tau,
                             A,beta0,sigma0,
                             var.prop,m,
                             B,eta,Wmat_option)
  eta = 0.2*length(Y)
  result_lung8 = MH_Sampling(Y,delta,tau,
                             A,beta0,sigma0,
                             var.prop,m,
                             B,eta,Wmat_option)

})
```

```
##    user  system elapsed
##  477.23   38.57  870.00
```




```
Wmat = HarrellC_Wmat(Y,delta,tau)

# Randomly choose one beta
index = sample(1:8,size = 1)

result_all = data.frame(Eta=c(20,10,5,2,1,0.7,0.5,0.2),
                        AcceptanceRate=c(result_lung1$accept_rate,
                                         result_lung2$accept_rate,
                                         result_lung3$accept_rate,
                                         result_lung4$accept_rate,
```

```
                                        result_lung5$accept_rate,
                                        result_lung6$accept_rate,
                                        result_lung7$accept_rate,
                                        result_lung8$accept_rate),
                     HarrellC = c(HarrellC(colMeans(result_lung1$THETA),Wmat),
                                  HarrellC(colMeans(result_lung2$THETA),Wmat),
                                  HarrellC(colMeans(result_lung3$THETA),Wmat),
                                  HarrellC(colMeans(result_lung4$THETA),Wmat),
                                  HarrellC(colMeans(result_lung5$THETA),Wmat),
                                  HarrellC(colMeans(result_lung6$THETA),Wmat),
                                  HarrellC(colMeans(result_lung7$THETA),Wmat),
                                  HarrellC(colMeans(result_lung8$THETA),Wmat)),
                     beta_mean = c(mean(result_lung1$BETA[,index]),
                                   mean(result_lung2$BETA[,index]),
                                   mean(result_lung3$BETA[,index]),
                                   mean(result_lung4$BETA[,index]),
                                   mean(result_lung5$BETA[,index]),
                                   mean(result_lung6$BETA[,index]),
                                   mean(result_lung7$BETA[,index]),
                                   mean(result_lung8$BETA[,index])),
                     beta_var = c(var(result_lung1$BETA[,index]),
                                   var(result_lung2$BETA[,index]),
                                   var(result_lung3$BETA[,index]),
                                   var(result_lung4$BETA[,index]),
                                   var(result_lung5$BETA[,index]),
                                   var(result_lung6$BETA[,index]),
                                   var(result_lung7$BETA[,index]),
                                   var(result_lung8$BETA[,index])))
result_all
```

```
##    Eta AcceptanceRate  HarrellC   beta_mean     beta_var
## 1 20.0    0.003772727 0.6619625 -0.03554927 1.744377e-05
## 2 10.0    0.026000000 0.6642357 -0.02944730 1.285406e-04
## 3  5.0    0.197954545 0.6628149 -0.03703997 4.337835e-04
## 4  2.0    0.614181818 0.6597841 -0.08346922 1.909975e-03
## 5  1.0    0.794181818 0.6507861 -0.15675711 6.859054e-03
## 6  0.7    0.877909091 0.6353476 -0.40040980 7.291627e-02
## 7  0.5    0.902500000 0.6262550 -0.48049603 3.874735e-02
## 8  0.2    0.911000000 0.6317484 -0.20680509 4.397003e-02
```

3. Based on the "Martin-Syring" paper, the primary role played by the learning rate is to control the spread of the Gibbs posterior.

4. When n is large, the prior influence will be rather limited.

   "we see that eta can only be affecting the spread of the Gibbs posterior, with small eta making the posterior wider, more diffuse, and large eta making the posterior narrower, more concentrated at which the empirical risk is minimized"

$$R_n(\theta) = -log\{\tilde{C}_{\tau,v}(\theta, G; D)\}$$

However, 3 and 4 have an important assumption, that is, n is large.
For the lung data set, n=167, is it large enough?

5. Learning rate selection has something to do with credible region. (The paper uses the highest posterior density (HPD) as an example)

6. The paper gives us an algorithm to do learning rate selection.

Data:

$$T_i = (X_i, Y_i),\ X_i \in \mathbf{R}^p, i = 1, ..., n$$

Gibbs posterior distribution:

$$\pi_n^{(\eta)}(\theta) \propto e^{-\eta n R_n(\theta)} \pi(\theta)$$

Highest posterior density (HPD) region:

$$C_\alpha^{(\eta)}(T^n) = \{\theta : \pi_n^{(\eta)}(\theta) > k(\alpha, \eta)\}$$

where $\pi_n^{(\eta)}$ is the Gibbs posterior density and $k(\alpha; \eta)$ is the cutoff chosen to ensure the the region has probability $1 - \alpha$

Converge probability function:

$$c_\alpha(\eta) = c_\alpha(\eta\,; P) = P\{\theta \in C_\alpha^{(\eta)}(T^n)\}$$

For the algorithm, we first use a bootstrap approximation for the converge probability function:

$$\hat{c}_\alpha^{boot}(\eta) = \frac{1}{B} \sum_{b=1}^{B} I\{\hat{\theta}_n \in C_\alpha^{(\eta)}(\tilde{T}_b^n)\}$$

where $\hat{\theta}_n$ is the empirical risk minimizer.

Then:

$$\eta_s = \eta_{s-1} + \kappa_s\{\hat{c}_\alpha^{boot}(\eta_{s-1}) - (1 - \alpha)\}, s \geq 1$$

where $(\kappa_s) \subset (0, 1)$ is a deterministic sequence of step-sizes. e.g. $\kappa_s \propto (1 + s)^{(-r)},\ r \in (0.5, 1]$