UiO : Matematisk institutt
Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2024
Ex5

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

3.8. Thirteen chemical measurements were carried out on each of 178 wines from three regions of Italy [53]. These data are available from the website for this book. Using one or more heuristic search methods from this chapter, partition the wines into three groups for which the total of the within-group sum of squares is minimal. Comment on your work and the results. This is a search problem of size 3^p where $p = 178$. If you have access to standard cluster analysis routines, check your results using a standard method like that of Hartigan and Wong [317].

```
#Calculating objective function
rss = function(x,lab)
{
 rss = 0
 for(j in 1:3)
 {
   z = x[lab==j,]
   mu = colMeans(z)
   res = t(z)-mu
   rss = rss + sum(colSums(res^2))
 }
 rss
}
```

```
#Steepest descent
more = TRUE;r=rep(NA,3)
while(more)
{
  more = FALSE
  for(i in 1:n)
  {
    lab.cur = lab[i]
    for(j in 1:3)
    {
      lab[i] = j
      r[j] = rss(wine[,-1],lab)
    }
    jopt = which.min(r)
    lab[i] = jopt
    if(jopt!=lab.cur)
      more = TRUE
  }
  show(min(r))
}
show(table(wine[,1],lab))
```

```
#Simulated annealing
lab = sample(1:3,n,replace=T)
rss.cur = rss(wine[,-1],lab)
for(it in c(1:100))
{
  temp = 10/(1+it)
  for(i in 1:n)
  {
    lab.cur = lab[i]
    lab[i] = sample(1:3,1)
    rss.prop = rss(wine[,-1],lab)
    if(runif(1)<exp(-(rss.prop-rss.cur)/temp))
      rss.cur = rss.prop
    else
      lab[i] = lab.cur
  }
  show(rss(wine[,-1],lab))
}
show(table(wine[,1],lab))
```

4.3. distribution. Some data points have missing values in one or more coordinates. Only 27 of the 50 observations are complete.

a. Derive the EM algorithm for joint maximum likelihood estimation of μ and Σ. It is easiest to recall that the multivariate normal density is in the exponential family.

b. Determine the MLEs from a suitable starting point. Investigate the performance of the algorithm, and comment on your results.

Solution to exercise (4.3)

(a). Define $\boldsymbol{y}_i$ to be the complete data and $\boldsymbol{x}_i$ the observed data. Note in general that we have

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \Sigma_{yy} \end{pmatrix} \right)$$

then

$$\begin{aligned} E[\boldsymbol{y}|\boldsymbol{x}] &= \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x) \\ V[\boldsymbol{y}|\boldsymbol{x}] &= \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} \end{aligned}$$

$$\begin{aligned} \mathrm{tr}\{\boldsymbol{ABC}\} &= \mathrm{tr}\{\boldsymbol{BCA}\} = \mathrm{tr}\{\boldsymbol{CAB}\} \\ \frac{\partial \log(|\boldsymbol{\Sigma}|)}{\partial \boldsymbol{\Sigma}} &= \boldsymbol{\Sigma}^{-1} \\ \frac{\partial \mathrm{tr}\{\boldsymbol{A\Sigma}^{-1}\boldsymbol{B}\}}{\partial \boldsymbol{\Sigma}} &= -(\boldsymbol{\Sigma}^{-1}\boldsymbol{BA\Sigma}^{-1})^T \end{aligned}$$

Normal distribution is in the exponential family

- The Exponential family:

$$f_y(\mathbf{y}|\boldsymbol{\theta}) = c_1(\mathbf{y})c_2(\boldsymbol{\theta})\exp\{\boldsymbol{\theta}^T\mathbf{s}(\mathbf{y})\}$$

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2$$

$$\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{1/2}}\exp -\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)$$

$$\frac{|Q|^{\frac{1}{2}}}{(2\pi)^{\frac{p}{2}}}\exp -\frac{1}{2}(y-\mu)^T Q(y-\mu)$$

$$= \frac{|Q|^{\frac{1}{2}}}{(2\pi)^{\frac{p}{2}}}\exp -\frac{1}{2}\mu^T Q\mu \cdot \exp -\frac{1}{2}(y^T Qy - 2\mu^T Qy)$$

Alternative:

- The Exponential family:

$$f_y(\mathbf{y}|\boldsymbol{\theta}) = c_1(\mathbf{y})c_2(\boldsymbol{\theta})\exp\{\boldsymbol{\theta}^T\mathbf{s}(\mathbf{y})\}$$

$$L^{compl}(\boldsymbol{\theta}) = \prod^{n} \frac{1}{(2\pi)^{3/2}|\boldsymbol{\Sigma}|^{1/2}} - \exp\left(-0.5(\boldsymbol{y}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu})\right)$$

- Algorithm

E-step $\boldsymbol{s}^{(t)} = E[\boldsymbol{s}(\boldsymbol{Y})|\boldsymbol{x};\boldsymbol{\theta}^{(t)}]$
M-step $\boldsymbol{\theta}^{(t+1)}$ solves $E[\boldsymbol{s}(\boldsymbol{Y})|\boldsymbol{\theta}] = \boldsymbol{s}^{(t)}$

Sufficient statistics for normal distribution:

$$\overline{\boldsymbol{y}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i \qquad \overline{\boldsymbol{y^2}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\boldsymbol{y}_i^T$$

$$E\left(\overline{\boldsymbol{y}}|\boldsymbol{\theta}^{(t)},\boldsymbol{x}\right) = \frac{1}{n}\sum_{i=1}^{n}E(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^{(t)})$$

$$E(\overline{\boldsymbol{y}}|\boldsymbol{\theta}) = \boldsymbol{\mu}$$

$$E[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)$$
$$V[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$$

$$\overline{\boldsymbol{y^2}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^T$$

M-step $$E\left(\overline{\boldsymbol{y^2}} \middle| \boldsymbol{\theta}\right) = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\boldsymbol{T}}$$

E-step $$E\left(\overline{\boldsymbol{y^2}} \middle| \boldsymbol{\theta}^{(t)}, \boldsymbol{x}\right) = \frac{1}{n}\sum_{i=1}^{n} E\left(\boldsymbol{y}_i \boldsymbol{y}_i^T \middle| \boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n} V\left(\boldsymbol{y}_i \middle| \boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right) + E\left(\boldsymbol{y}_i \middle| \boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right) E\left(\boldsymbol{y}_i \middle| \boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right)^T$$

$$E[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)$$
$$V[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$$

From scratch…

$$L^{compl}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{3/2}|\boldsymbol{\Sigma}|^{1/2}} - \exp\left(-0.5(\boldsymbol{y}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu})\right)$$

$$l^{compl}(\boldsymbol{\theta}) = -\frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Sigma}|) - 0.5\sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu})$$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) = -\frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Sigma}|) - 0.5\sum_{i=1}^{n} E[(\boldsymbol{y}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu})|\boldsymbol{x}_i, \boldsymbol{\theta}^t]$$

$$= -\frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Sigma}|) - 0.5\sum_{i=1}^{n} E[\mathrm{tr}\{(\boldsymbol{y}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu})\}|\boldsymbol{x}_i, \boldsymbol{\theta}^t]$$

$$= -\frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Sigma}|) - 0.5\sum_{i=1}^{n} E[\mathrm{tr}\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu})(\boldsymbol{y}_i - \boldsymbol{\mu})^T\}|\boldsymbol{x}_i, \boldsymbol{\theta}^t]$$

$$= -\frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Sigma}|) - 0.5\sum_{i=1}^{n} \mathrm{tr}\{\boldsymbol{\Sigma}^{-1}E[(\boldsymbol{y}_i - \boldsymbol{\mu})(\boldsymbol{y}_i - \boldsymbol{\mu})^T|\boldsymbol{x}_i, \boldsymbol{\theta}^t]\}$$

$$= -\frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Sigma}|) - 0.5\mathrm{tr}\{\boldsymbol{\Sigma}^{-1}\sum_{i=1}^{n} E[(\boldsymbol{y}_i - \boldsymbol{\mu})(\boldsymbol{y}_i - \boldsymbol{\mu})^T|\boldsymbol{x}_i, \boldsymbol{\theta}^t]\}$$

Further,

$$
\begin{aligned}
&E[(\boldsymbol{y}_i-\boldsymbol{\mu})(\boldsymbol{y}_i-\boldsymbol{\mu})^T|\boldsymbol{x}_i,\boldsymbol{\theta}^t] \\
&\quad =E[(\boldsymbol{y}_i-E[\boldsymbol{y}_i|\boldsymbol{x}_i]+E[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t]-\boldsymbol{\mu})(\boldsymbol{y}_i-E[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t]+E[\boldsymbol{y}_i|\boldsymbol{x}_i]-\boldsymbol{\mu})^T|\boldsymbol{x}_i,\boldsymbol{\theta}^t] \\
&\quad =E[(\boldsymbol{y}_i-E[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t])(\boldsymbol{y}_i-E[\boldsymbol{y}_i|\boldsymbol{x}_i])^T|\boldsymbol{x}_i,\boldsymbol{\theta}^t]+ \\
&\qquad E[(E[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t]-\boldsymbol{\mu})(E[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t]-\boldsymbol{\mu})^T|\boldsymbol{x}_i,\boldsymbol{\theta}^t] \\
&\quad =V(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t)+(E[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t]-\boldsymbol{\mu})(E[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^t]-\boldsymbol{\mu})^T \\
&\quad =\boldsymbol{V}_i^t+(\hat{\boldsymbol{y}}_i^t-\boldsymbol{\mu})(\hat{\boldsymbol{y}}_i^t-\boldsymbol{\mu})^T
\end{aligned}
$$

Inserting this, we get

$$
Q(\boldsymbol{\theta},\boldsymbol{\theta}^t)=-\frac{3n}{2}\log(2\pi)-\frac{n}{2}\log(|\boldsymbol{\Sigma}|)-0.5\mathrm{tr}\{\boldsymbol{\Sigma}^{-1}\sum_{i=1}^{n}\boldsymbol{V}_i^t+(\hat{\boldsymbol{y}}_i^t-\boldsymbol{\mu})(\hat{\boldsymbol{y}}_i^t-\boldsymbol{\mu})^T\}
$$

$$
E\left[\left(y_i-E(y_i|x,\theta^t)\right)\cdot\left(E(y_i|x,\theta^t)-\mu\right)|x_i,\theta^t\right]=
$$

$$
E\left[\left(y_i-E(y_i|x,\theta^t)\right)|x_i,\theta^t\right]\cdot\left(E(y_i|x,\theta^t)-\mu\right)=
$$

$$
\left(E[y_i|x_i,\theta^t]-E(y_i|x,\theta^t)\right)\cdot\left(E(y_i|x,\theta^t)-\mu\right)\ =0
$$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) = -\frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Sigma}|) - 0.5\text{tr}\{\boldsymbol{\Sigma}^{-1}\sum_{i=1}^{n}\boldsymbol{V}_i^t + (\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu})(\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu})^T\}$$

Differentiating wrt $\boldsymbol{\mu}$, we get

$$\boldsymbol{\Sigma}^{-1}\sum_{i=1}^{n}(\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu}^{t+1}) = \mathbf{0}$$

$$\sum_{i=1}^{n}(\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu}^{t+1}) = \mathbf{0}$$

$$\boldsymbol{\mu}^{t+1} = \frac{1}{n}\sum_{i=1}^{n}\hat{\boldsymbol{y}}_i^t$$

$$\text{tr}\{\boldsymbol{ABC}\} = \text{tr}\{\boldsymbol{BCA}\} = \text{tr}\{\boldsymbol{CAB}\}$$

$$\frac{\partial \log(|\boldsymbol{\Sigma}|)}{\partial \boldsymbol{\Sigma}} = \boldsymbol{\Sigma}^{-1}$$

$$\frac{\partial \text{tr}\{\boldsymbol{A}\boldsymbol{\Sigma}^{-1}\boldsymbol{B}\}}{\partial \boldsymbol{\Sigma}} = -(\boldsymbol{\Sigma}^{-1}\boldsymbol{B}\boldsymbol{A}\boldsymbol{\Sigma}^{-1})^T$$

Can similarly obtain

$$\boldsymbol{\Sigma}^{t+1} = \frac{1}{n}\sum_{i=1}^{n}[\boldsymbol{V}_i^t + (\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu}^{t+1})(\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu}^{t+1})^T]$$

```r
EM.trinormal = function(x)
{
  n = nrow(x)
  #Make table of missing values
  num.mis = apply(is.na(x),1,sum)
  #Initial values based on complete data
  x.compl = x[num.mis==0,]
  mu = colMeans(x.compl)
  Sigma = var(x.compl)

  #Make a matrix with estimates of all observations (equal to observed when given)
  y.hat = x
  #Array for conditional covariance matrix for all observations
  V = array(0,c(n,3,3))
  more = TRUE;eps=0.001
  while(more)
  {
    mu.new = rep(0,3)
    Sigma.new = matrix(0,nrow=3,ncol=3)
    for(i in 1:n)
     {
       #First find y.hat and V
       if(num.mis[i]>0)
        {
          ind = c(1:3)[!is.na(x[i,])]
          y.hat[i,] = mu[1:3]+Sigma[1:3,ind]%*%solve(Sigma[ind,ind])%*%matrix(x[i,ind]-mu[ind],ncol=1)
          V[i,,] = Sigma-Sigma[1:3,ind]%*%solve(Sigma[ind,ind])%*%Sigma[ind,1:3]
        }
     }
    #Then estimate
    mu.new = colMeans(y.hat)
    Sigma.new = matrix(0,3,3)
    for(i in 1:n)
      Sigma.new = Sigma.new + V[i,,]+(y.hat[i,]-mu)%o%(y.hat[i,]-mu)
    Sigma.new = Sigma.new/n
    more = (max(abs(mu-mu.new))>eps) | (max(abs(Sigma-Sigma.new))>eps)
    mu = mu.new
    Sigma = Sigma.new
  }
  list(mu=mu,Sigma=Sigma)
}
```

Initialize mu and sigma

Set data for all

Only update data which have missing

$$E[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)$$
$$V[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$$

4.3. distribution. Some data points have missing values in one or more coordinates. Only 27 of the 50 observations are complete.

a. Derive the EM algorithm for joint maximum likelihood estimation of μ and Σ. It is easiest to recall that the multivariate normal density is in the exponential family.

b. Determine the MLEs from a suitable starting point. Investigate the performance of the algorithm, and comment on your results.

$$\boldsymbol{\mu}^{t+1} = \frac{1}{n}\sum_{i=1}^{n} \hat{\boldsymbol{y}}_i^t$$

$$E[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)$$
$$V[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$$

Can similarly obtain

$$\boldsymbol{\Sigma}^{t+1} = \sum_{i=1}^{n}[\boldsymbol{V}_i^t + (\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu}^{t+1})(\hat{\boldsymbol{y}}_i^t - \boldsymbol{\mu}^{t+1})^T]$$

(a). Using the R-script for Exercise 4.3, write a function that has as input the data and some initial estimate of the parameters and returns the ML estimates.

(*b*). Write a script that generates $B = 1000$ bootstrap samples by sampling with replacement from the $n = 48$ observations. Run the script and obtain

(*d*). Consider now a parametric bootstrap approach, where first you generate $\boldsymbol{y}_i \sim N(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ for $i = 1, ..., n$ and then put those observations to missing that are missing in the original data set. Run this procedure and obtain

Bootstrap

```r
#Non-parametric Bootstrapping
B = 1000
n = nrow(x)
mu.boot = matrix(nrow=B,ncol=3)
for(b in 1:B)
{
  ind = sample(1:n,n,replace=T)
  x.new = x[ind,]
  res.new = EM.trinormal(x.new)
  mu.boot[b,] = res.new$mu
}
```

```r
#Parametric Bootstrapping
B = 1000
n = nrow(x)
mu.boot = matrix(nrow=B,ncol=3)
for(b in 1:B)
{
  x.new = rmvnorm(n,mu.hat,Sigma.hat)
  #Insert missing values
  x.new[is.na(x)] <- NA
  res.new = EM.trinormal(x.new)
  mu.boot[b,] = res.new$mu
}
```

```r
library(mvtnorm)
```

```
          mu[1] mu[2] mu[3]
Estimate  0.879 2.850 9.026
Bias     -0.001 0.000 0.006
Variance  0.033 0.018 0.059
SE        0.182 0.135 0.243
2.5%      0.520 2.578 8.571
97.5%     1.237 3.105 9.504
```

```r
bias.mu = colMeans(mu.boot)-mu.hat
var.mu = apply(mu.boot,2,var)
ci.mu = apply(mu.boot,2,quantile,probs=c(0.025,0.975))
tab = rbind(mu.hat,bias.mu,var.mu,sqrt(var.mu),ci.mu)
colnames(tab) = c("mu[1]","mu[2]","mu[3]")
rownames(tab)[1:4] = c("Estimate","Bias","Variance","SE")
show("Non-parametric bootstrapping")
show(round(tab,3))
```

```
          mu[1]  mu[2] mu[3]
Estimate  0.879  2.850 9.026
Bias     -0.003 -0.004 0.004
Variance  0.031  0.017 0.056
SE        0.176  0.130 0.237
2.5%      0.546  2.610 8.562
97.5%     1.251  3.130 9.487
```

(c). For the bootstrap procedure above, do it take into account

- wrong model specifications?
- the actual amount of missing data?

(e). For the parametric bootstrap procedure, do it take into account

- wrong model specifications?
- the actual amount of missing data?

Comment on possible differences from the non-parametric version.

Exercise 11 (Stochastic gradient and neural network) (a). Take the *Stoc_grad_NN.R* script from the course web page. Modify the script to $n = 10\,000$. Run the script and inspect how the algorithm perform. Repeat the algorithm several times and look at the variability of the results.

Hint: You probably have to modify the script a bit, perhaps divide it into two. For the first part, the simulation of data, you want to keep that fixed (either by storing the data or by be sure that you use the same seed every time. However, for the run of the SG algorithm you should vary the seed!

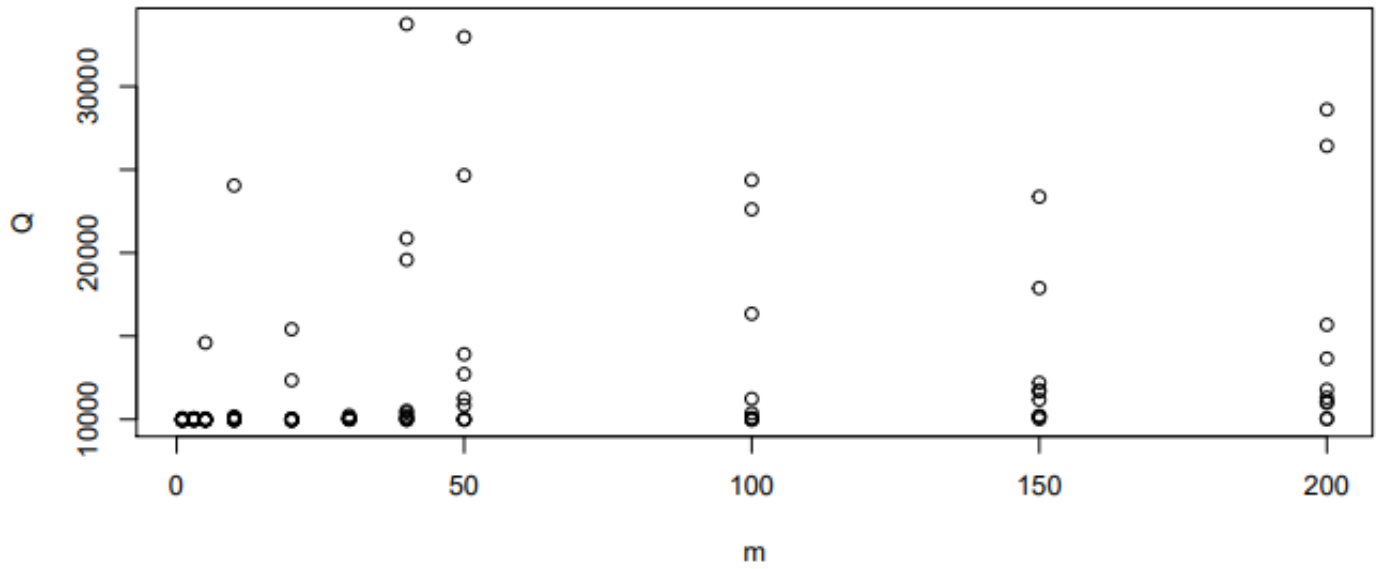
(a). 10 repetitions with random initial points on the α's and linear regression estimates on the β's given the α's of the algorithm with $n = 10\,000$ gave $Q - values$ ranging from 10025.54 to 10368.43:

10217.33 10144.09 10105.26 10368.43 10097.87 10188.39
10120.00 10025.54 10079.28 10200.39

This indicates that the algorithms is quite sensitive to starting values.

(b). Try to modify the size of the subsample and look at the performance.

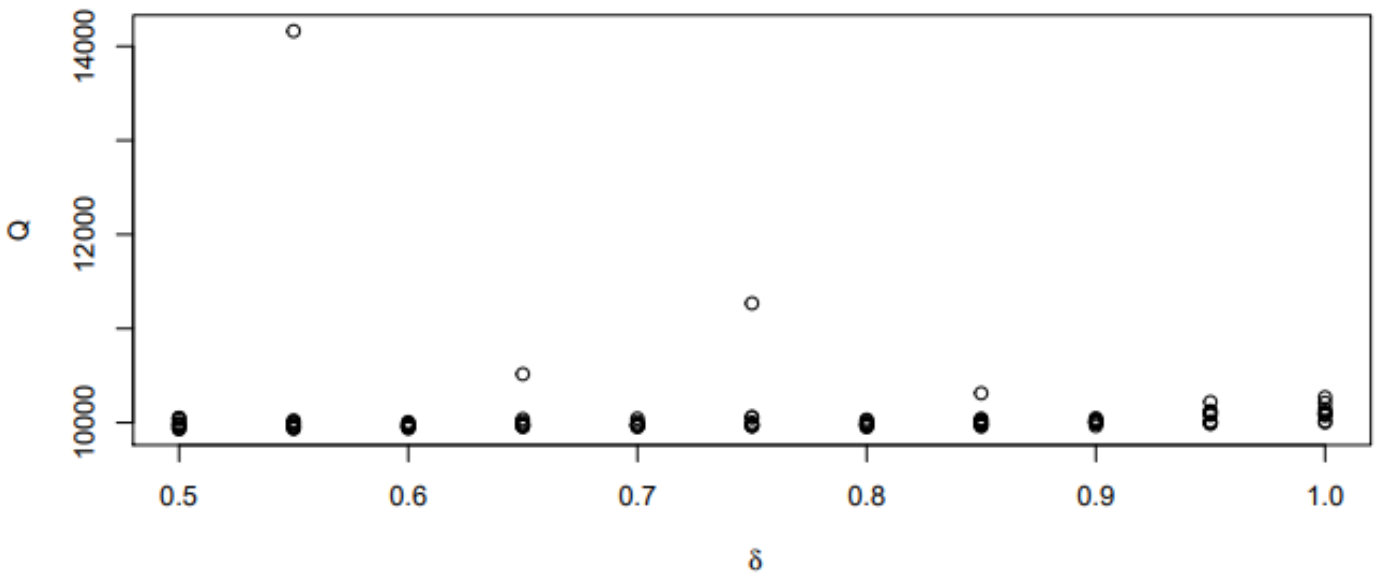
(b). Using $m = 1, 3, 5, 10, 20, 30, 40, 50, 100, 150, 200$, repeating the procedure 10 times for each value of m gave the following results:



The minimum value 9925.46 was obtained for $m = 20$ while for $m = 1$ we obtained 9925.61. It seems like the procedure is working best for small m.

(c). Try to modify the learning rate $\gamma_t = c/(1+t)^\delta$ with $\delta \in [0.5, 1]$. How do the algorithm perform for different choices?

(c). Using $\delta = 0.500.550.600.650.700.750.800.850.900.951.00$ and $m = 1$, repeating the procedure 10 times for each value of δ gave the following results:



The smallest value obtained was 9931.51 for $\delta = 0.5$. It is not very clear what is preferable here, but top large values should be avoided.

(*d*). Try to change the initial values of the β's to be completely random as well. How do the algorithm perform then?

(*d*). Running now with $m = 1$ and $\delta = 0.5$ 22 times gave the following values of Q:

10041.637	9955.682	10016.338	9950.949	9948.319	9960.379
9941.765	9938.026	9956.522	9986.015	9938.413	9972.349
10025.531	10078.068	9926.557	10004.114	9947.645	9941.453
9964.286	9967.233				

with the smallest value equal to 9926.557. Note that this is smaller than the values obtained by using LS estimates on β!

Exercise 12 (SG and Geostatistics) (a). Consider the ***Geostat_SG.R*** script from the course web page. Try out different values of m and evaluate the performance of the algorithm.

(b). Include a call to the ***geoR*** package (last commands in the script) which should give the optimal solution. Does the results from the SG algorithm look reasonable?

(c). Try to increase n by doubling it and run the routine from the ***geoR*** package. How many times are you able to double it before the routines take too much time to produce results?

(d). Now try to increase n to 1000 and 10 000. Try out different values of m and learningrates to obtain as good performance as possible.

SGD for dependent data

- Consider spatial data:

$$\boldsymbol{Y} = \begin{pmatrix} Y(s_1) \\ Y(s_2) \\ \vdots \\ Y(s_n) \end{pmatrix} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\mu} = \mu \boldsymbol{I}$ and $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{R} + \boldsymbol{\tau}^2 \boldsymbol{I}$, that is

$$\text{cov}[Y(s_i), Y(s_j)] = \begin{cases} \sigma^2 r(||s_i - s_j||; \boldsymbol{\phi}) & s_i \neq s_j \\ \sigma^2 + \tau^2 & s_i = s_j \end{cases}$$

- Realisation of a process defined continuously in a space $\mathcal{S}$
- Log-likelihood with $\theta = (\boldsymbol{\mu}, \sigma^2, \tau^2, \boldsymbol{\phi})$

$$l(\theta) = -\frac{n}{2} \log 2\pi - \frac{1}{2} \log(|\boldsymbol{\Sigma}|) - \frac{1}{2} (\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu})$$

- In general, computational burden is $O(n^3)$, problematic for large n

KL and Geostatistics

- Liang et al. (2013): Approximate KL by

$$\widehat{KL}(f_\theta, g) = C - \frac{1}{\binom{n}{m}} \sum_{k=1}^{\binom{n}{m}} \log(f_\theta(\boldsymbol{y}_k|\boldsymbol{s}_k))$$

  where $(\boldsymbol{y}_k, \boldsymbol{s}_k)$ is a subset of $(\boldsymbol{y}, \boldsymbol{s})$ of size m.
- Find θ as the solution of

$$\frac{\partial}{\partial\boldsymbol{\theta}} \widehat{KL}(f_\theta, g) = -\frac{1}{\binom{n}{m}} \sum_{k=1}^{\binom{n}{m}} H(\theta, \boldsymbol{y}_k, \boldsymbol{s}_k)$$

$$H(\theta, \boldsymbol{y}_k, \boldsymbol{s}_k) = \frac{\partial}{\partial\boldsymbol{\theta}} \log(f_\theta(\boldsymbol{y}_k|\boldsymbol{s}_k))$$

  by the stochastic gradient algorithm!

Example

$$\log f(\boldsymbol{y}_k|\boldsymbol{s}_k) = -\frac{m}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(y_k - \mu\mathbf{1_m})^T\Sigma_k^{-1}(y_k - \mu\mathbf{1_m})$$

$$(\boldsymbol{\Sigma}_k)_{i,j} = \text{cov}(Y(\boldsymbol{s}_{k,i}) - Y(\boldsymbol{s}_{k,j}) = \tau^2 I(j=j) + \sigma^2 \exp(-(||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi)$$
$$(\boldsymbol{R}_k)_{i,j} = \exp(-(||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi)$$

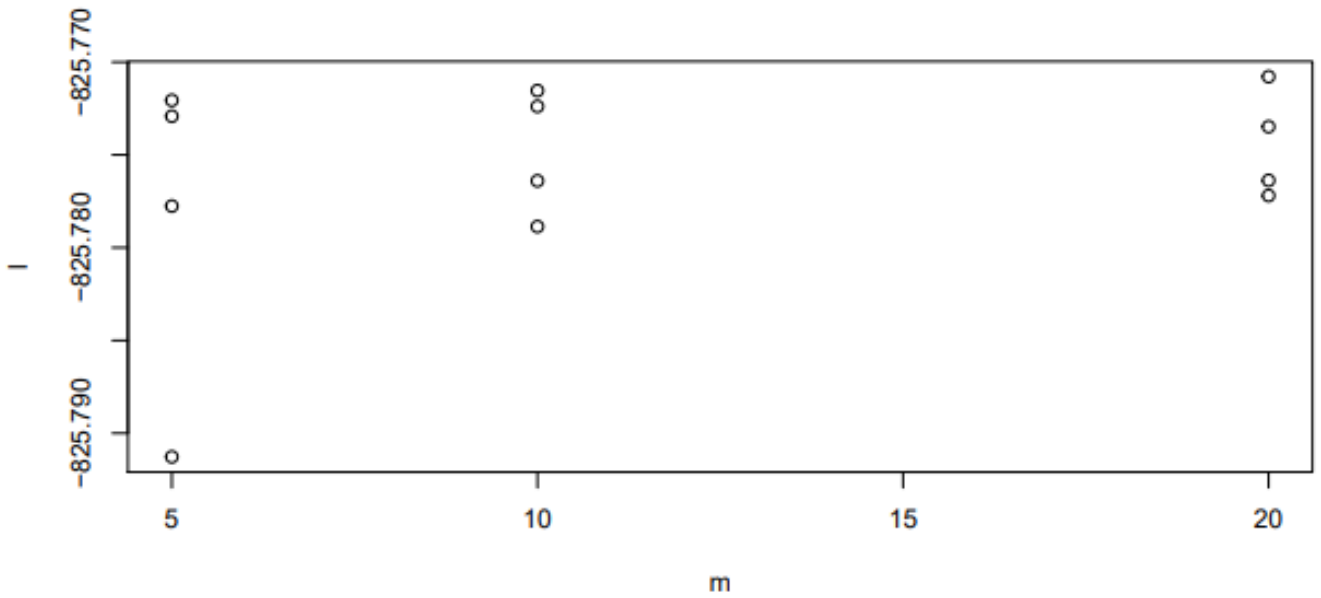
$$H_\mu(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = \mathbf{1}_m^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$
$$H_{\sigma^2}(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = -\frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\boldsymbol{R}_k) + \frac{1}{2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{R}_k\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$
$$H_{\tau^2}(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = -\frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}) + \frac{1}{2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)^T\boldsymbol{\Sigma}_k^{-2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$
$$H_\phi(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = -\frac{\sigma^2}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\frac{d\boldsymbol{R}_k}{d\phi}) + \frac{\sigma^2}{2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)^T\boldsymbol{\Sigma}_k^{-1}\frac{d\boldsymbol{R}_k}{d\phi}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$

$$\frac{d(\boldsymbol{R}_k)_{i,j}}{d\phi} = ||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi^2 \cdot \exp(-(||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi)$$

```r
#Geostatistical model
rm(list=ls())
library(mvtnorm)
set.seed(345)
mu=1
sigma = 1
tau = 1
phi=0.25
n = 500*2
s = cbind(runif(n),runif(n))  # sampling spatial locations
d = as.matrix(dist(s,diag=TRUE,upper=TRUE))
Sigma = sigma^2*exp(-d/phi)+tau^2*diag(1,n)

Sigma.chol = chol(Sigma)
y = mu + rnorm(n)%*%Sigma.chol # sampling y"values"



logl = function(y,d,mu,tau2,sig2,phi)
{
  n = length(y)
  Sigma = sig2*exp(-d/phi)+tau2*diag(1,n)
#  res = matrix(y-mu,ncol=1)
#  show(c(mu,sig2,tau2,phi))
#  show(class(Sigma))
#  l = -0.5*n*log(2*pi)-0.5*determinant(Sigma,log=TRUE)$mod-0.5*sum(res*solve(Sigma,res))
  d = dmvnorm(y,rep(mu,n),Sigma,log=TRUE)
#  show(c(d,l))
  d
}
tr = function(M)
{
  sum(diag(M))
}
```

```
for(it in 1:100)
{
   # sample & set up
  ind = sample(1:n,m,replace=FALSE)
  y.k = y[ind]
  s.k = s[ind,]
  d.k = as.matrix(dist(s.k,diag=TRUE,upper=TRUE))
  R.k = exp(-d.k/phi.hat)
  Sigma.k = sig2.hat*R.k+tau2.hat*diag(1,m)
  dR.k = d.k*R.k/phi.hat^2
  res = y.k-mu.hat
  res2 = solve(Sigma.k,res)
  res3 = R.k%*%res2
  res4 = dR.k%*%res2
  H.mu = sum(res2)
  H.sig2 = -0.5*tr(solve(Sigma.k,R.k))+0.5*sum(res2*res3)
  H.tau2 = -0.5*tr(solve(Sigma.k))+0.5*sum(res2*res2)
  H.phi = -0.5*tr(solve(Sigma.k,sig2.hat*dR.k))+0.5*sig2.hat*sum(res2*res4)

  ## stepsize
  alpha = 10/(m*(1+it)^delta)

  ## update
  mu.hat = mu.hat + alpha*H.mu
  a = log(sig2.hat) + alpha*H.sig2/sig2.hat
  sig2.hat = exp(a)
  a = log(tau2.hat) + alpha*H.tau2/tau2.hat
  tau2.hat = exp(a)
  a = log(phi.hat) + alpha*H.phi/phi.hat
  phi.hat = exp(a)
  if(it%%100==0)
  {
   #show(c(H.mu,H.sig2,H.tau2,H.phi))
   l = logl(y,d,mu.hat,tau2.hat,sig2.hat,phi.hat)
   lvec = c(lvec,l)
   itvec = c(itvec,it)
   show(c(it,l,mu.hat,tau2.hat,sig2.hat,phi.hat))
  }
```

(a). $m = 5, 10, 20)$, 10 000 iterations, 4 repetitions gave



(*b*). `geoR` gave log-likelihood of -791.7.