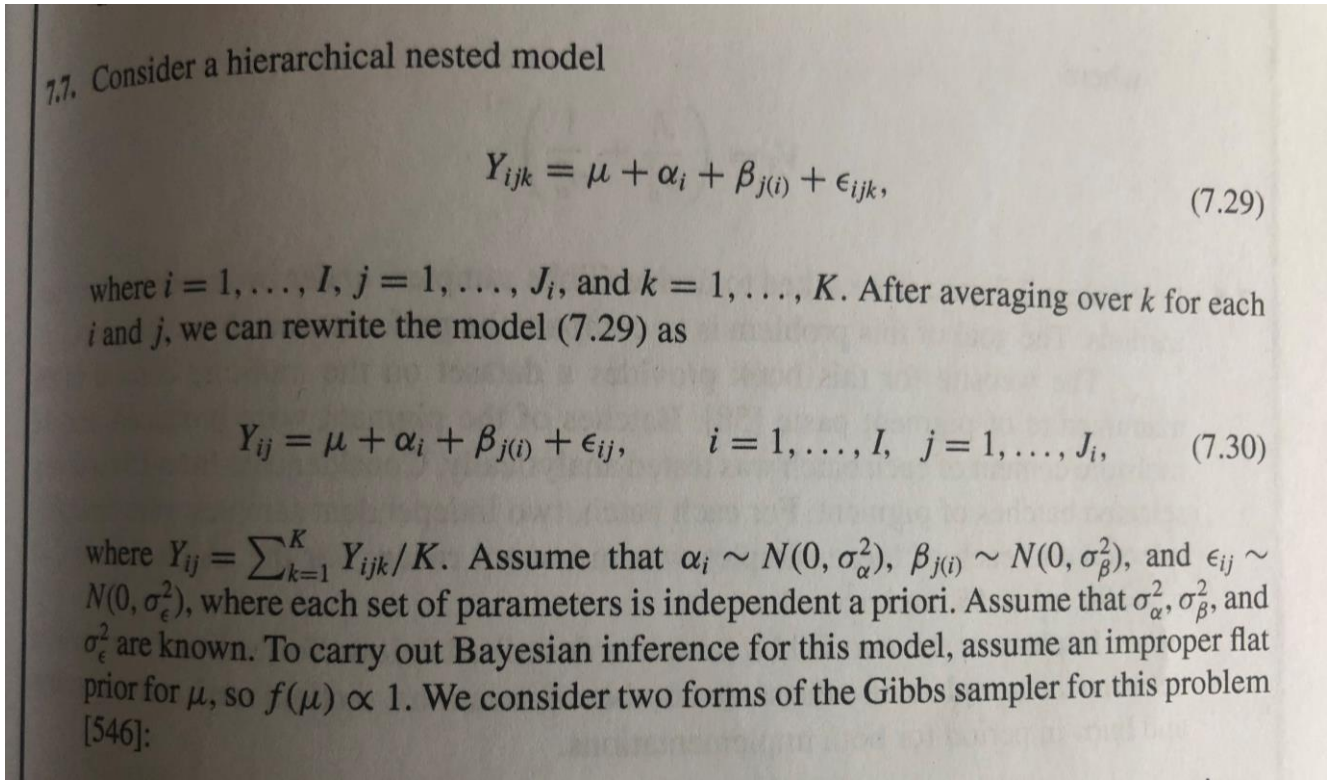
UiO : Matematisk institutt
Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2024

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

Do 7.8 =7.7 model in Stan



7.7. Consider a hierarchical nested model

$$Y_{ijk} = \mu + \alpha_i + \beta_{j(i)} + \epsilon_{ijk}, \tag{7.29}$$

where $i = 1, \ldots, I$, $j = 1, \ldots, J_i$, and $k = 1, \ldots, K$. After averaging over k for each i and j, we can rewrite the model (7.29) as

$$Y_{ij} = \mu + \alpha_i + \beta_{j(i)} + \epsilon_{ij}, \qquad i = 1, \ldots, I, \quad j = 1, \ldots, J_i, \tag{7.30}$$

where $Y_{ij} = \sum_{k=1}^{K} Y_{ijk}/K$. Assume that $\alpha_i \sim N(0, \sigma_\alpha^2)$, $\beta_{j(i)} \sim N(0, \sigma_\beta^2)$, and $\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$, where each set of parameters is independent a priori. Assume that σ_α^2, σ_β^2, and σ_ϵ^2 are known. To carry out Bayesian inference for this model, assume an improper flat prior for μ, so $f(\mu) \propto 1$. We consider two forms of the Gibbs sampler for this problem [546]:

Rcode

```
d = read.table("V:/PROMAX/PROMAX-Odd/Lecture/STK4051-9051/Lectures_Odd/Ex15/pigment.dat",header=T)
stanFileName="V:/PROMAX/PROMAX-Odd/Lecture/STK4051-9051/Lectures_Odd/Ex15/Exercise7_8_Rstan.stan"


dStan = list(I = 15 ,
             J = 2,
             moist = d$Moisture,
             batch = d$Batch,
             samp = d$Sample)

N=10000
nSamp=100
##MCMC
fit1 = stan(file = stanFileName, data = dStan,
            iter=1.1*N,
            warmup=0.1*N,
            thin=N/nSamp,
            chains=4,
            seed=231171,
            refresh=10000,
            control=list(adapt_delta=0.80))
```

Stan code part 1

```
data {
  int<lower=0> I;          // number of batches
  int<lower=0> J;          // numer of samples
  real moist[I*J];           // Moisture vector
  int<lower=1,upper=I>  batch[I*J];
  int<lower=1,upper=J>  samp[I*J];
}
transformed data{
  real<lower=0> sigmaA=sqrt(86);        // batch effect std
  real<lower=0> sigmaB=sqrt(58);        // sample effect std
  real<lower=0> sigmaE=sqrt(1);         // Random effect std
  matrix[I,J] y;
  for (i in 1:I*J){
      y[batch[i],samp[i]]=moist[i] ;
  }
}
```

Stan code part 2

```
parameters {
  real mu;                // population treatment effect
  vector[I] alpha;        // batch effect
  matrix[I,J] beta ;       // sample effect per batch
}

transformed parameters {
  matrix[I,J] theta;
  for (i in 1:I)
    for(j in 1:J)
      theta[i,j] = mu + alpha[i]+beta[i,j];        // joint treatment effects
}
model {
  target += normal_lpdf(alpha| 0, sigmaA);   // prior log-density alpha
  for (j in 1:J){
    target += normal_lpdf(beta[,j] | 0, sigmaB);    // prior log-density beta
    target += normal_lpdf(y[,j] | theta[,j], sigmaE);  // log-likelihood
  }
}
```

Exercise 8
Consider a mixture model where

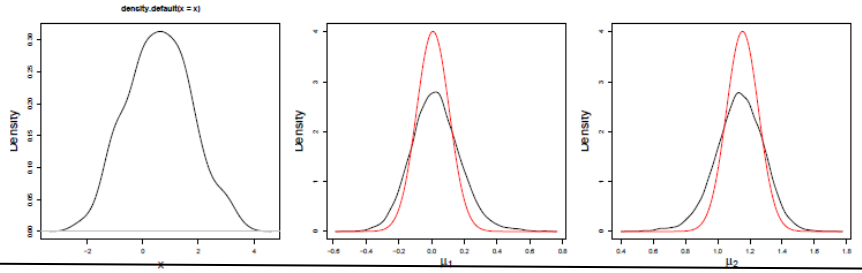
$$\begin{aligned} \Pr(C_i = k) &= \pi_k, & k = 1, 2 \\ p(x_i | C_i = k) &= N(\mu_k, \sigma_k^2) \\ p(\mu_k) &= N(0, \sigma_\beta^2), & k = 1, 2 \end{aligned}$$

Our focus is now Bayesian inference where we are interested in the posterior distribution $p(\boldsymbol{\mu}|\boldsymbol{y})$ with $\boldsymbol{\mu} = (\mu_1, \mu_2)$ and $\boldsymbol{x} = (x_1, ..., x_n)$.

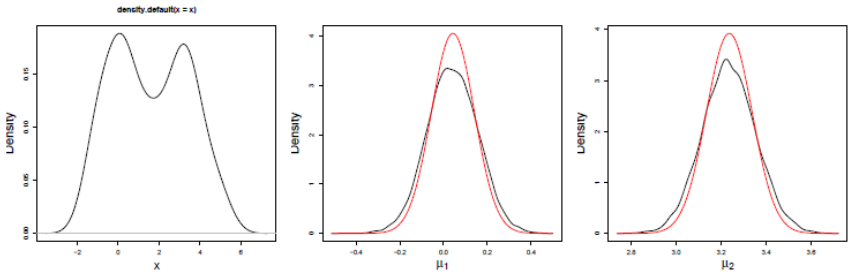
(a) The variational approximation is based on a mean-field assumption with Gaussian distributions assumed for each variable of interest.

Describe what kind of assumptions the mean-field approximation is based on. Also specify which parameters that needs to be fitted in the variational approximation approach.

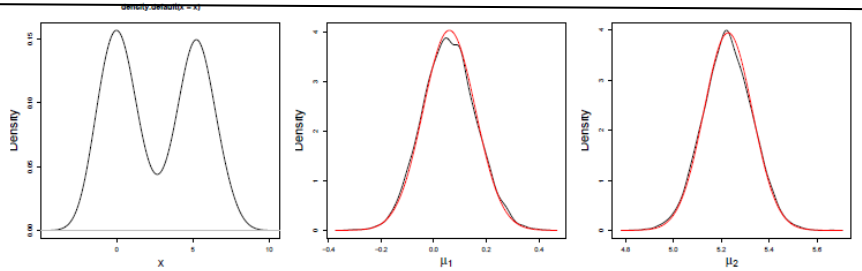
The plots below compare a variational inference approximation with output from a simple Metropolis-Hastings algorithm (where in each case the first half is discarded) for different simulated datasets (where the μ_k values differ). In each row, the first plot shows the estimated density of the observations $\boldsymbol{x}$ while the two next plots shows the estimates of $p(\mu_1|\boldsymbol{x})$ and $p(\mu_2|\boldsymbol{x})$ with the black lines corresponding to output from Metropolis-Hastings while the red lines correspond to the variational inference approximation.



Case 1



Case 2



Case 3

(b)Discuss the results seen in the Figure and relate this to the assumptions made for the variational approximation.

For the last row, the two classes are quite separated, making it relatively easy to identify which x_i's that belong to the two classes.
In that case, the mean Field approximation will become quite good. As there becomes more ucnertainty to which classes that the x_i's belong to, the mean FIeld approximation becomes worse due to that there will be more dependence between μ_1 and μ_2.

Note that if the class-membership was known we have $p(\mu_1, \mu_2|\boldsymbol{x}, \boldsymbol{c}) = p(\mu_1|\boldsymbol{x}, \boldsymbol{c})p(\mu_2|\boldsymbol{x}, \boldsymbol{c})$, that is they are independent.

Problem 1. (Monte Carlo integration / Variational inference) When encountering problems in variational inference, we measure distance between two distributions, $q(x), p(x)$ with the Kullback–Leibler divergence, which is defined as:

$$KL(q||p) = \int \log q(\boldsymbol{x})\, q(x)\, d\boldsymbol{x} - \int \log p(\boldsymbol{x})\, q(\boldsymbol{x})\, d\boldsymbol{x} \tag{1}$$

$$= E_q\{\log(q(\boldsymbol{X})) - \log p(\boldsymbol{X})\}$$

For complex distributions, this integral is not easily accessible. We will now consider a target distribution being the bivariate normal distribution with unit variance and mean zero. This distribution is defined as:

$$p(\boldsymbol{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left\{-\frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\,\boldsymbol{x}\right\} \tag{2}$$

where

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}. \tag{3}$$

In a standard approach to variational inference we consider the mean field approximation where we look for a solution of the form:

$$q(x) = q_1(x_1) \cdot q_2(x_2). \tag{4}$$

Which means that we are looking for a distribution with independent components, specifically we are looking for solutions where $q_j(x_j) = \phi(x_j; \mu_j, \sigma_j^2)$, with $\phi(x; a, b^2)$ being the normal density; mean a and variance b^2. Hint: You can use a library to evaluate the multivariate density, e.g. `dmvnorm` in the `mvtnorm` package.

a) Let $\rho = 0.9$, $\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$.

Make a function which evaluates the Kullback–Leibler divergence $KL(q||p)$ using Monte Carlo integration. How many samples do you need to have a result with Monte Carlo variability less than 0.01? (Here variability is measured in terms of standard deviation.)

$$KL(q||p) = \int \log q(\boldsymbol{x})\, q(x)\, d\boldsymbol{x} - \int \log p(\boldsymbol{x})\, q(\boldsymbol{x})\, d\boldsymbol{x}$$

$$= E_q\{\log(q(\boldsymbol{X})) - \log p(\boldsymbol{X})\}$$

$$q(x) = q_1(x_1) \cdot q_2(x_2). \qquad p(\boldsymbol{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left\{-\frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\,\boldsymbol{x}\right\}$$

```
KL1a=function(rho,s0,N)
{
  set.seed(231171)
  R =cbind(rnorm(N,mean=0,sd=s0),rnorm(N,mean=0,sd=s0))
  Sigma = matrix(c(1, rho,rho, 1),ncol=2 )

  lp=dmvnorm(R, c(0,0), Sigma, log = TRUE)
  lq= dnorm(R[,1],0,s0,log=TRUE) +dnorm(R[,2],0,s0,log=TRUE)
  show(var(lq-lp))
  #show(mean(lq-lp)/sqrt(var(lq-lp)))
  mean(lq-lp)
}


KL1a(0.9,1,1000)
#Variance about 40
N = 40/0.01^2
```

b) It is possible to show that the Kullback–Leibler divergence has its minimum for $\mu_1 = \mu_2 = 0$, and $\sigma_1 = \sigma_2 = \sqrt{1-\rho^2}$. In light of this result, discuss strengths and weaknesses of an analysis based on mean field variational inference.

The rest of problem 1 is for STK 9051 only. An algorithm for deriving the mean field approximation in the general case, is to use coordinate ascent variational inference, i.e. the CAVI algorithm. A version of the CAVI algorithm is detailed below.

Algorithm 1 (CAVI)

1) Initialize $q_2^{(0)}(x) = \phi(x; a, b^2)$,
2) While not converged iterate:
   a. Set $q_1^{(i)}(x_1) \propto \exp\{E_{q_2^{(i-1)}}(\log p(x_1|x_2))\}$
   b. Set $q_2^{(i)}(x_2) \propto \exp\{E_{q_1^{(i)}}(\log p(x_2|x_1))\}$

c) (STK 9051 only). We will now analyze use of this algorithm for the distribution $p(\boldsymbol{x})$ in (2). Start off with $q_2^{(0)}(x) = \phi(x; 1, 1^2)$. Compute $q_1^{(1)}(x)$, and derive $q_j^{(n)}(x)$. Comment on the result. You can use without proof (if needed) that:

$$\log p(x_i|x_j) = const \; -\frac{1}{2}\frac{(x_i - \rho x_j)^2}{1-\rho^2} \tag{5}$$

1c) $E_{q_2}(\log p(x_1|x_2)) = E_2\left(const\ -\frac{1}{2}\frac{(x_1-\rho x_2)^2}{1-\rho^2}\right) = const - \frac{1}{2}E_2\left(\frac{x_1^2-2\rho x_1 x_2+\rho^2 x_2^2}{1-\rho^2}\right) =$

$$const - \frac{1}{2}\frac{x_1^2 - 2\rho x_1}{1-\rho^2} = const - \frac{1}{2}\frac{(x_1-\rho)^2}{1-\rho^2}$$

After one iteration we have: $q_1^{(1)}(x_1) = \phi(x_1; \rho, 1-\rho^2)$

$$E_{q_1}(\log p(x_2|x_1)) = const - \frac{1}{2}E_1\left(\frac{x_2^2 - 2\rho x_2 x_1 + \rho^2 x_1^2}{1-\rho^2}\right)$$

$$const - \frac{1}{2}\frac{x_2^2 - 2x_2\rho^2}{1-\rho^2} = const - \frac{1}{2}\frac{(x_2-\rho^2)^2}{1-\rho^2}$$

$$q_2^{(1)}(x_1) = \phi(x_2; \rho^2, 1-\rho^2)$$

Thus after n iterations we have

$$q_1^{(n)}(x_1) = \phi(x_1; \rho^{2n-1}, 1-\rho^2)$$

$$q_2^{(n)}(x_2) = \phi(x_2; \rho^{2n}, 1-\rho^2)$$

Problem 5 (STAN) The library rstan uses Hamiltonian Monte Carlo as a generic tool for implementing Bayesian inference. The stan program below defines a statistical model with data x, and parameters nu, mu1, and mu2. The stan program is also available in the file `Oppg5.stan`.

```
data{
int<lower=1> N;
real X[N];
}

parameters{
real<lower=0,upper=1> nu;
real <lower=0> mu1;
real <upper=0> mu2;
}

model{
for(i in 1:N){
  target+=log(nu*exp(normal_lpdf(X[i]|mu1,1))+(1-nu)*exp(normal_lpdf(X[i]|mu2,1)));

}
}
```

a) The program implies a prior for the parameters and a likelihood for the data. State these statistical models.

b) Run the model using N=625, and X from `EM_mixture.dat`. Show a scatterplot of mu1 and mu2.