MAST30027: Modern Applied Statistics

Week 12 Lab Sheet

1. **Metropolis-Hastings** [**We already solved problems (a), (b), and (c) in the week 10. This week, solve the problem (d), (e), and (f).**]

   Recall that $\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$, iff $\mathbf{X}$ has joint density

   $$f_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

   where $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.

   (a) Write an R function that evaluates the density of a bivariate normal distribution. The function should take as input the point $\mathbf{x}$, the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$.

   You will find the functions `solve` and `det` useful.

   (b) Write a program in R that uses the Metropolis-Hastings algorithm to generate a sample of size $n = 1000$ from the $N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}\right)$ distribution. Use the symmetric random walk proposal distribution $N\left(\mathbf{x}, \sigma^2 I\right)$ with $\sigma = 2.5$.

   Use $\mathbf{X}(0) = \begin{pmatrix} 6 \\ -6 \end{pmatrix}$ as your initial state. Report the proportion of accepted values.

   (c) Let $\mathbf{X}(n)$ be the n-th sample point. Plot $X_i(n)$ and the cumulative averages $\overline{X}_i(n) = n^{-1}\sum_{j=1}^{n} X_i(j)$, for $i = 1, 2$. The cumulative averages should give a rough idea of how quickly the $\mathbf{X}(n)$ converge in distribution.

   (d) You can use the R functions `cor` or `acf` to estimate the lag 1 autocorrelation ρ. Calculate the so called *effective sample size*: $n(1-\rho)/(1+\rho)$ for each variable.

   (e) Change the standard deviation of the proposal chain to $\sigma = 0.1$ and then $\sigma = 10$. How do the proportion of accepted values, the cumulative averages, and the effective sample size change?

   (f) Now change $\boldsymbol{\Sigma}$ to $\begin{pmatrix} 4 & 2.8 \\ 2.8 & 4 \end{pmatrix}$ and repeat the above analysis. How do things change?

2. **Variational Inference**

   To explore the inaccuracy introduced by VI, we discuss the problem of approximating a bivariate normal distribution using VI with the mean-field variational family. In practice, we already know the densify for the bivariate normal distribution. However, here we pretend that we know the density of the bivariate normal distribution 'up to constant' and approximate the exact bivariate normal distribution using VI.

   Suppose that for $\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$, we only know

   $$p(\mathbf{x}) \propto \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

   where $\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$. Let $h(\mathbf{x}) = \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$. Then, $p(\mathbf{x}) = Ch(\mathbf{x})$.

   We will apply VI with the mean-field variational family, $D = \{q(\mathbf{x}) | q(\mathbf{x}) = q_1(x_1)q_2(x_2)\}$ to find $q^*(\mathbf{x}) = q_1^*(x_1)q_2^*(x_2)$ which minimises the Kullback-Leibler (KL) divergence to the exact density

$p(\mathbf{x})$,

$$\begin{aligned} q^*(\mathbf{x}) &= \underset{q(\mathbf{x})\in D}{\arg\min} \quad KL(q(\mathbf{x})||p(\mathbf{x})) \\ &= \underset{q(\mathbf{x})\in D}{\arg\min} \quad \int_{\mathbf{x}} q(\mathbf{x})[\log q(\mathbf{x}) - \log p(\mathbf{x})]d\mathbf{x}. \end{aligned}$$

Minimising the KL divergence is equivalent to maximising the ELBO:

$$\text{ELBO}(q(\mathbf{x})) = \int_{\mathbf{x}} q(\mathbf{x})[\log h(\mathbf{x}) - \log q(\mathbf{x})]d\mathbf{x},$$

because

$$\begin{aligned} KL(q(\mathbf{x})||p(\mathbf{x})) &= \int_{\mathbf{x}} q(\mathbf{x})[\log q(\mathbf{x}) - \log p(\mathbf{x})]d\mathbf{x} \\ &= \int_{\mathbf{x}} q(\mathbf{x})[\log q(\mathbf{x}) - \log h(\mathbf{x}) - \log C]d\mathbf{x} \\ &= -\text{ELBO}(q(\mathbf{x})) + \log C. \end{aligned}$$

We will use the CAVI algorithm for optimisation. The CAVI iteratively optimises each factor as follows while holding the other factors fixed:

$$q_1^*(x_1) \propto \exp\{\mathbb{E}_{x_2}[\log h(\mathbf{x})]\}, \quad q_2^*(x_2) \propto \exp\{\mathbb{E}_{x_1}[\log h(\mathbf{x})]\},$$

where the expectations $\mathbb{E}_{x_2}$ and $\mathbb{E}_{x_1}$ are taken with respect to $q_2^*(x_2)$ and $q_1^*(x_1)$, respectively.

(a) Derive $q_1^*(x_1)$ and $q_2^*(x_2)$ and write the corresponding distribution names and their parameters.

(b) Derive the ELBO up to constant.

(c) Implement the CAVI algorithm.

(d) Obtain the approximated density $q^*(\mathbf{x}) = q_1^*(x_1)q_2^*(x_2)$ for $\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 2 & 0.1 \\ 0.1 & 2 \end{pmatrix}$. Run the CAVI algorithm at least two times using different initial values and report $q_1^*(x_1)$ and $q_2^*(x_2)$ with the highest ELBO. For each run, check that the ELBO increase at each iteration by plotting them. Compare the approximated density $q^*(\mathbf{x}) = q_1^*(x_1)q_2^*(x_2)$ to the true density $p(\mathbf{x})$ by making contour plots. You can make the contour plot for the true density $p(\mathbf{x}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$ where $\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} 2 & 0.1 \\ 0.1 & 2 \end{pmatrix}$ using the following code:

```
> library(mvtnorm)
> x.points <- seq(-3,3,length.out=100)
> y.points <- x.points
> z <- matrix(0,nrow=100,ncol=100)
> mu <- c(0,0)
> sigma <- matrix(c(2,0.1,0.1,2),nrow=2)
> for (i in 1:100) {
+   for (j in 1:100) {
+     z[i,j] <- dmvnorm(c(x.points[i],y.points[j]),
+   mean=mu,sigma=sigma)
+   }
+ }
> contour(x.points,y.points,z)
```

(e) Obtain the approximated density $q^*(\mathbf{x}) = q_1^*(x_1)q_2^*(x_2)$ for $\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 2 & 1.9 \\ 1.9 & 2 \end{pmatrix}$. Run the CAVI algorithm at least two times using different initial values and report $q_1^*(x_1)$ and $q_2^*(x_2)$ with the highest ELBO. For each run, check that the ELBO increase at each iteration by plotting them. Compare the approximated density $q^*(\mathbf{x}) = q_1^*(x_1)q_2^*(x_2)$ to the true density $p(\mathbf{x})$ by making contour plots.