STA410 Assignment 3

Zikun Chen
1001117882

November 15, 2017

Question 1

(a) Let

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{bmatrix}$$

Then

$$\begin{aligned} trace(AB) &= (AB)_{11} + (AB)_{22} + \cdots + (AB)_{mm} \\ &= a_{11}b_{11} + a_{12}b_{21} + \cdots + a_{1n}b_{n1} \\ &+ a_{21}b_{12} + a_{22}b_{22} + \cdots + a_{2n}b_{n2} \\ &+ \cdots \\ &+ a_{m1}b_{1m} + a_{m2}b_{2m} + \cdots + a_{mn}b_{nm} \\ &= a_{11}b_{11} + a_{21}b_{12} + \cdots + a_{m1}b_{1m} && \text{look vertically} \\ &+ a_{12}b_{21} + a_{22}b_{22} + \cdots + a_{m2}b_{2m} \\ &+ \cdots \\ &+ a_{1n}b_{n1} + a_{2n}b_{n2} + \cdots + a_{mn}b_{nm} \\ &= (BA)_{11} + (BA)_{22} + \cdots + (BA)_{nn} = trace(BA) \end{aligned}$$

(b) Let

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & s_{nn} \end{bmatrix}$$

$$\boldsymbol{V} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Then

$$\boldsymbol{V}^T S \boldsymbol{V} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & s_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} v_j v_i$$

$$\boldsymbol{V}^T \boldsymbol{V} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} v_1^2 & v_1 v_2 & \dots & v_1 v_n \\ v_2 v_1 & v_2^2 & \dots & v_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ v_n v_1 & v_n v_2 & \dots & v_n^2 \end{bmatrix}$$

Therefore,

$$\begin{aligned} tr(S\boldsymbol{V}^T\boldsymbol{V}) &= tr(\begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & s_{nn} \end{bmatrix} \begin{bmatrix} v_1^2 & v_1 v_2 & \dots & v_1 v_n \\ v_2 v_1 & v_2^2 & \dots & v_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ v_n v_1 & v_n v_2 & \dots & v_n^2 \end{bmatrix}) \\ &= tr(\begin{bmatrix} \sum_{j=1}^n s_{1j} v_j v_1 & \cdots & \cdots & \cdots \\ \vdots & \sum_{j=1}^n s_{2j} v_j v_2 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \dots & \sum_{j=1}^n s_{nj} v_j v_n \end{bmatrix}) = \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} v_j v_i \end{aligned}$$

Similarly,

$$\begin{aligned} tr(SE[\boldsymbol{V}^T\boldsymbol{V}]) &= tr(\begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & s_{nn} \end{bmatrix} \begin{bmatrix} E[v_1^2] & E[v_1 v_2] & \dots & E[v_1 v_n] \\ E[v_2 v_1] & E[v_2^2] & \dots & E[v_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ E[v_n v_1] & E[v_n v_2] & \dots & E[v_n^2] \end{bmatrix}) \\ &= tr(\begin{bmatrix} \sum_{j=1}^n s_{1j} E[v_j v_1] & \cdots & \cdots & \cdots \\ \vdots & \sum_{j=1}^n s_{2j} E[v_j v_2] & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \dots & \sum_{j=1}^n s_{nj} E[v_j v_n] \end{bmatrix}) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} E[v_j v_i] \end{aligned}$$

And since E is a linear operator and S is not random

$$\begin{aligned} E[tr(S\boldsymbol{V}^T\boldsymbol{V})] &= E[\sum_{i=1}^{n}(S\boldsymbol{V}^T\boldsymbol{V})_{ii}] \\ &= \sum_{i=1}^{n} E[(S\boldsymbol{V}^T\boldsymbol{V})_{ii}] \\ &= \sum_{i=1}^{n} E[\sum_{j=1}^{n} s_{ij}v_j v_i] \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}E[v_j v_i] = tr(SE[\boldsymbol{V}^T\boldsymbol{V}]) \end{aligned}$$

So $E[\boldsymbol{V}^T S\boldsymbol{V}] = tr(SE[\boldsymbol{V}^T\boldsymbol{V}]) = tr(S)$ since $E[\boldsymbol{V}^T\boldsymbol{V}] = I$

(c) Look at

$$E[\boldsymbol{V}^T S\boldsymbol{V}] = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} s_{ij}s_{kl}E(V_iV_jV_kV_l)$$

When all the indices are equal $i = k = j = l$, we get the term:

$$\sum_{i=1}^{n} s_{ii}^2 E(V_i^4)$$

When any of the i, j, k, l differs from any other indices, for example, WLOG, assume i is different from any indices from j, k, l:

Then $E[V_iV_jV_kV_l] = E[V_i]E[V_jV_kV_l] = 0$ since V_i's are independent and have mean 0. This is true even if the other indices are equal $j = k = l$.

Therefore, i, j, k, l must have two pairs that are equal. We split the situation into different cases below.

Case 1 - $i = j \neq k = l$, we get the term:

$$\begin{aligned} \sum_{i=1}^{n}\sum_{k=1}^{n} s_{ii}s_{kk}E(V_i^2V_k^2) &= \sum_{k=1}^{n} s_{ii}s_{kk}E(V_i^2)E(V_k^2) \\ &= \sum_{k=1}^{n} s_{ii}s_{kk}Var(V_i)Var(V_k) = \sum_{k=1}^{n} s_{ii}s_{kk} \end{aligned}$$

since V_i and V_k are independent and have zero mean.

Similarly,

Case 2 - $i = k \neq j = l$, we get the term:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}^2$$

Case 3 - $i = l \neq j = k$, we get the term:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}s_{ji}$$

In conclusion:

$$E[\boldsymbol{V}^T S \boldsymbol{V}] = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} s_{ij}s_{kl}E(V_iV_jV_kV_l)$$

$$= \sum_{i=1}^{n} s_{ii}^2 E(V_i^4) + \sum_{k=1}^{n} s_{ii}s_{kk} + \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}s_{ji}$$

$$= \sum_{i=1}^{n} s_{ii}^2 E(V_i^4) + \text{constant}$$

since matrix S is not random, so are its elements.

Furthermore, we want to minimize $E(V_i^4)$ so as to minimize $Var(tr(\hat{S}))$

$$E(V_i^4) = E((V_i^2)^2) = Var(V_i^2) + E(V_i^2)^2 = Var(V_i^2) + 1$$

Therefore, we want to minimize $Var(V_i^2) \leq 0$

Let V_i be ± 1 with probability 0.5, then $E(V_i) = 0.5 \cdot (-1) + 0.5 \cdot (+1) = 0$ and $Var(V_i) = E(V_i^2) - E(V_i)^2 = 1 - 0 = 1$ as required in the question.

More importantly, $V_i^2 = 1$ with probability 1, so $Var(V_i^2) = 0$ is minimized.

(d) **Program:**

```
num.param <- function(x, span, m) {
  n <- length(x)
  that <- NULL
  for (i in 1:m) {
    # generate rv V to minimize variance of the trace estimate
    v <- NULL
    for (j in 1:n){
      u <- runif(1, 0,1)
      if (u < 0.5) {
        v <- c(v, 1)
      }
      else {
        v <- c(v, -1)
      }
    }
    # estimate vhat generated from loess function
```

```
    r <- loess(v~x, span=span)
    vhat <- r$fitted
    that <- c(t, sum(v*vhat))
  }
  # compute the average (expectation) of v^T*S*v
  numparam = mean(that)
  # estimate the standard error
  sderror=sd(that)/sqrt(n)
  result <- list(numparam = numparam, sderror = sderror)
}

> x <- rnorm(10000, 0, 1)
> r <- num.param(x, span = 0.7, 100)
> r
$numparam
[1] 5.716757

$sderror
[1] 0.03361709
```

Question 2

(a) **Derivation:**

Method of moments equations:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{\alpha}{\lambda}$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{\alpha}{\lambda^2}$$

Therefore

$$\hat{\lambda} = \frac{\bar{x}}{\sigma^2} = \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\alpha} = \hat{\lambda}\bar{x} = \frac{\bar{x}^2}{\sigma^2} = \frac{n\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

(b) MLE:

$$
\begin{aligned}
ln(L(\alpha,\lambda)) &= ln(\prod_{i=1}^{n} f(x_i;\alpha,\lambda)) \\
&= \sum_{i=1}^{n} ln(\frac{\lambda^{\alpha} x_i^{\alpha-1} exp(-\lambda x_i)}{\Gamma(\alpha)}) \\
&= \sum_{i=1}^{n} [\alpha ln(\lambda) + (\alpha-1) ln(x_i) - \lambda x_i - ln(\Gamma(\alpha))] \\
&= n\alpha ln(\lambda) - nln(\Gamma(\alpha)) + (\alpha-1)\sum_{i=1}^{n} ln(x_i) - \lambda \sum_{i=1}^{n} x_i
\end{aligned}
$$

Score:

$$\frac{\partial ln(L)}{\partial \alpha} = nln(\lambda) - \frac{n\Gamma(\alpha)^{'}}{\Gamma(\alpha)} + \sum_{i=1}^{n} ln(x_i)$$

$$\frac{\partial ln(L)}{\partial \lambda} = \frac{n\alpha}{\lambda} - \sum_{i=1}^{n} x_i$$

$$S(\alpha,\lambda) = \begin{bmatrix} nln(\lambda) - \frac{n\Gamma(\alpha)^{'}}{\Gamma(\alpha)} + \sum_{i=1}^{n} ln(x_i) \\ \frac{n\alpha}{\lambda} - \sum_{i=1}^{n} x_i \end{bmatrix}$$

Hessian:

$$\frac{\partial^2 ln(L)}{\partial \alpha^2} = -n\frac{\Gamma(\alpha)^{''}\Gamma(\alpha) - (\Gamma(\alpha)^{'})^2}{(\Gamma(\alpha))^2}$$

$$\frac{\partial^2 ln(L)}{\partial \alpha \partial \lambda} = \frac{\partial^2 ln(L)}{\partial \lambda \partial \alpha} = \frac{n}{\lambda}$$

$$\frac{\partial^2 ln(L)}{\partial \lambda^2} = -\frac{n\alpha}{\lambda^2}$$

$$H(\alpha,\lambda) = \begin{bmatrix} n\frac{\Gamma(\alpha)^{''}\Gamma(\alpha) - (\Gamma(\alpha)^{'})^2}{(\Gamma(\alpha))^2} & -\frac{n}{\lambda} \\ -\frac{n}{\lambda} & \frac{n\alpha}{\lambda^2} \end{bmatrix}$$

Note we took negative for the second derivatives in the matrix for the N-R algorithm.

Newton-Raphson Iteration:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + [H(\hat{\theta}_k)]^{-1} S(\hat{\theta}_k)$$

where $\hat{\theta}_k = [\hat{\alpha}_k, \hat{\lambda}_k]^T$

To estimate the variance-covariance matrix, we can look at $[H(\hat{\theta}_k)]^{-1}$

In addition, we take smaller Newton-Raphson update in case when the usual Newton-Raphson update takes one or both of the parameters below 0.

Program:

```
gamma.nr <- function(x,alpha,lambda,eps=1.e-8,max.iter=100) {
  n <- length(x)
  m1 <- mean(x)
  var <- sum((x-m1)^2)/n

  # use MoM to estimate alpha and lambda if missing
  if (missing(alpha)) {
    lambda <- m1/var
    alpha <- lambda*m1
  }
  theta <- c(alpha,lambda)

  # compute the scores based on the initial estimates
  sum_x <- m1*n
  sum_lnx <- sum(log(x))
  score1 <- n*log(lambda) - n*digamma(alpha)/gamma(alpha) + sum_lnx
  score2 <- n*alpha/lambda - sum_x
  score <- c(score1,score2)
  iter <- 1
  while (max(abs(score))>eps && iter<=max.iter) {
    rate <- 1
    # compute observed Fisher information
    info.11 <- n*(trigamma(alpha)*gamma(alpha) - digamma(alpha)^2)/gamma(alpha)^2
    info.12 <- -n/lambda
    info.22 <- n*alpha/lambda^2
    info <- matrix(c(info.11,info.12,info.12,info.22),ncol=2)
    # Newton-Raphson iteration
    theta <- theta + rate*solve(info,score)

    # Record the alpha and lambda from previous iteration for recalculation if needed
    old_alpha <- alpha
    old_lambda <- lambda

    # use a smaller update rate until both parameters are positive
    while (theta[1]<=0 || theta[2]<=0) {
      rate <- 0.7*rate
      info.11 <- n*(trigamma(old_alpha)*gamma(old_alpha)
                  - digamma(old_alpha)^2)/gamma(old_alpha)^2
      info.12 <- -n/old_lambda
```

```
      info.22 <- n*old_alpha/old_lambda^2
      info <- matrix(c(info.11,info.12,info.12,info.22),ncol=2)
      theta <- theta + rate*solve(info,score)
    }
    alpha <- theta[1]
    lambda <- theta[2]
    iter <- iter + 1

    # update score
    score1 <- n*log(lambda) - n*digamma(alpha)/gamma(alpha) + sum_lnx
    score2 <- n*alpha/lambda - sum_x
    score <- c(score1,score2)
  }
  if (max(abs(score))>eps) print("No convergence")
  else {
    print(paste("Number of iterations =",iter-1))
    loglik <- n*alpha*log(lambda) - n*log(gamma(alpha))
        + (alpha-1)*sum_lnx - lambda*sum_x
    info.11 <- n*(trigamma(alpha)*gamma(alpha) - digamma(alpha)^2)/gamma(alpha)^2
    info.12 <- -n/lambda
    info.22 <- n*alpha/lambda^2
    info <- matrix(c(info.11,info.12,info.12,info.22),ncol=2)
    r <- list(alpha=alpha,lambda=lambda,loglik=loglik,info=info)
    r
  }
}

> # e.g. estimaton for Exp(alpha=1, lambda=3) using NR
> alpha <- 1
> lambda <- 3
> x <- rgamma(10000, shape = alpha, rate = lambda)
> r <- gamma.nr(x)
[1] "Number of iterations = 4"
> r
$alpha
[1] 1.002258

$lambda
[1] 2.972346

$loglik
[1] 10931.1

$info
          [,1]      [,2]
[1,] 13118.868 -3364.346
```

```
[2,] -3364.346  1134.438

> # estimated variance-covariance matrix
> solve(r$info)
             [,1]         [,2]
[1,] 0.0003183301 0.0009440559
[2,] 0.0009440559 0.0036812338
```