tr(AB)=tr(BA)
tr(ABC)=tr(CAB)=tr(BCA)
tr(A+B)=tr(A)+tr(B)
tr(kA)=ktr(A)
trace is a linear operator,
linear operator means it can exchange with
other linear operator
E(tr)=tr(E)
aTAa->1x1=tr(aTAa)

Assignment #3 STA410H1F/2102H1F

due Friday November 15, 2019

Instructions: Solutions to problems 1 and 2 are to be submitted on Quercus (PDF files only). You are strongly encouraged to do problems 3–6 but these are not to be submitted for grading.

1. In lecture, we showed that we could estimate the trace of a matrix A ($\mathrm{tr}(A)$) by

$$\widehat{\mathrm{tr}}(A) = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{V}_i^T A \boldsymbol{V}_i$$

where $\boldsymbol{V}_1, \cdots, \boldsymbol{V}_m$ are independent random vectors with $E[\boldsymbol{V}_i \boldsymbol{V}_i^T] = I$.

(a) Suppose that the elements of each $\boldsymbol{V}_i$ are independent, identically distribution random variables with mean 0 and variance 1. Show that $\mathrm{Var}(\widehat{\mathrm{tr}}(A)$ is minimized by taking the elements of $\boldsymbol{V}_i$ to be ± 1 each with probability 1/2.

(Hint: This is easier than it looks – $\mathrm{Var}(\boldsymbol{V}^T A \boldsymbol{V}) = E[(\boldsymbol{V}^T A \boldsymbol{V})^2] - \mathrm{tr}(A)^2$ so it suffices to minimize

$$E[(\boldsymbol{V}^T A \boldsymbol{V})^2] = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} a_{ij}a_{k\ell}E(V_iV_jV_kV_\ell).$$

Given our conditions on the elements of $\boldsymbol{V}_i$, $V_1, \cdots, V_n$, most of $E(V_iV_jV_kV_\ell)$ are either 0 or 1. You should be able to show that

$$E[(\boldsymbol{V}^T A \boldsymbol{V})^2] = \sum_{i=1}^{n} a_{ii}^2 E(V_i^4) + \text{constant}$$

and find V_i to minimize $E(V_i^4)$ subject to $E(V_i^2) = 1$.)

(b) We also noted that if B is a symmetric $n \times n$ matrix whose eigenvalues are less than 1 in absolute value then we have the following formula for the determinate of the matrix $I - B$:

$$\det(I - B) = \exp\left(-\sum_{k=1}^{\infty}\frac{1}{k}\mathrm{tr}(B^k)\right)$$

and so we can estimate this determinant by

$$\widehat{\det}(I - B) = \exp\left(-\frac{1}{m}\sum_{i=1}^{m}\left\{\boldsymbol{V}_i^T B \boldsymbol{V}_i + \frac{1}{2}\boldsymbol{V}_i^T B^2 \boldsymbol{V}_i + \cdots + \frac{1}{r}\boldsymbol{V}_i^T B^r \boldsymbol{V}_i\right\}\right)$$

where r is "large enough".

In linear regression, one measure of the leverage of a subset of ℓ observations $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_\ell, y_\ell)$ is $1 - \det(I - H_{11})$ where H_{11} is an $\ell \times \ell$ matrix defined in terms of the linear regression "hat" matrix as follows:

$$H = X(X^T X)^{-1} X^T = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}.$$

From above, we can estimate $\det(I - H_{11})$ by

$$\widehat{\det}(I - H_{11}) = \exp\left(-\frac{1}{m}\sum_{i=1}^{m}\left\{\boldsymbol{V}_i^T H_{11}\boldsymbol{V}_i + \frac{1}{2}\boldsymbol{V}_i^T H_{11}^2 \boldsymbol{V}_i + \cdots + \frac{1}{r}\boldsymbol{V}_i^T H_{11}^r \boldsymbol{V}_i\right\}\right)$$

for some r.

Show that we can compute $H_{11}\boldsymbol{V}$ and $H_{11}^k \boldsymbol{V}$ for $k \geq 2$ using the hat matrix H as follows:

$$H\begin{pmatrix} \boldsymbol{V} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} H_{11}\boldsymbol{V} \\ H_{21}\boldsymbol{V} \end{pmatrix}$$

and

$$H\begin{pmatrix} H_{11}^{k-1}\boldsymbol{V} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} H_{11}^{k}\boldsymbol{V} \\ H_{21}H_{11}^{k-1}\boldsymbol{V} \end{pmatrix}.$$

(c) On Quercus, there is a function `leverage` (in the file `leverage.txt`) that computes the leverage for a given subset of observations for a design matrix X. (This function uses the QR decomposition of X to compute $H\boldsymbol{V}_i$; the functions are `qr`, which computes the QR decomposition of X and `qr.fitted`. which computes $H\boldsymbol{y} = QQ^T\boldsymbol{y}$.)

Suppose that $y_i = g(x_i) + \varepsilon_i$ for $i = 1, \cdots, n$ for some smooth function g and consider the following two parametric models for g:

$$g_1(x) = \beta_0 + \sum_{k=1}^{5}\{\beta_{2k-1}\cos(2k\pi x) + \beta_{2k}\sin(2k\pi x)\}$$

and

$$g_2(x) = \beta_0 + \beta_1\phi_1(x) + \cdots + \beta_{10}\phi_{10}(x)$$

where $\phi_1(x), \cdots, \phi_{10}(x)$ are B-spline functions. Suppose that $x_1, \cdots, x_{1000}$ are equally spaced points on $[0, 1]$ with $x_i = i/1000$. The B-spline functions and the respective design matrices can be constructed using the following R code:

```
> x <- c(1:1000)/1000
> X1 <- 1
> for (k in 1:5) X1 <- cbind(X1,cos(2*k*pi*x),sin(2*k*pi*x))
> library(splines) # loads the library of functions to compute B-splines
> X2 <- cbind(1,bs(x,df=10))
```

Note that both `X1` and `X2` are 1000×11 matrices. You can see the B-spline functions $\phi_1(x), \cdots, \phi_{10}(x)$ as follows:

```
> plot(x,X2[,2])
> for (i in 3:11) points(x,X2[,i])
```

Estimate the leverage of the points $\{x_i : (k-1)/20 < x_i \leq k/20\}$ for $k = 1, \cdots, 20$ for both designs; for each design, you will obtain 20 leverages. Comment on the differences between the leverages estimated for the two designs. To estimate the leverages for the two designs, you may want to modify the function `leverage` to estimate the leverages for both designs using the same values of $\boldsymbol{V}_1, \cdots, \boldsymbol{V}_m$ (why?).

Final 年年考

2. Suppose that $X_1, \cdots, X_n$ are independent Gamma random variables with common density

$$f(x; \alpha, \lambda) = \frac{\lambda^\alpha x^{\alpha-1} \exp(-\lambda x)}{\Gamma(\alpha)} \quad \text{for } x > 0$$

where $\alpha > 0$ and $\lambda > 0$ are unknown parameters.

(a) The mean and variance of the Gamma distribution are α/λ and α/λ^2, respectively. Use these to define method of moments estimates of α and λ based on the sample mean and variance of the data $x_1, \cdots, x_n$

(b) Derive the likelihood equations for the MLEs of α and λ and derive a Newton-Raphson algorithm for computing the MLEs based on $x_1, \cdots, x_n$. Implement this algorithm in R and test on data generated from a Gamma distribution (using the R function `rgamma`). Your function should also output an estimate of the variance-covariance matrix of the MLEs – this can be obtained from the Hessian of the log-likelihood function.

Important note: To implement the Newton-Raphson algorithm, you will need to compute the first and second derivatives of $\ln \Gamma(\alpha)$. These two derivatives are called (respectively) the digamma and trigamma functions, and these functions are available in R as `digamma` and `trigamma`; for example,

```
> gamma(2) # gamma function evaluated at 2
[1] 1
> digamma(2) # digamma function evaluated at 2
[1] 0.4227843
> trigamma(2) # trigamma function evaluated at 2
[1] 0.6449341
```

Supplemental problems:

3. Consider LASSO estimation in linear regression where we define $\widehat{\boldsymbol{\beta}}_\lambda$ to minimize

$$\sum_{i=1}^{n}(y_i - \bar{y} - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

for some $\lambda > 0$. (We assume that the predictors are centred and scaled to have mean 0 and variance 1, in which case $\bar{y}$ is the estimate of the intercept.) Suppose that the least squares estimate (i.e. for $\lambda = 0$) is non-unique — this may occur, for example, if there is some exact linear dependence in the predictors or if $p > n$. Define

$$\tau = \min_{\beta}\sum_{i=1}^{n}(y_i - \bar{y} - \boldsymbol{x}_i^T\boldsymbol{\beta})^2$$

and the set

$$\mathcal{C} = \left\{\boldsymbol{\beta} : \sum_{i=1}^{n}(y_i - \bar{y} - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 = \tau\right\}.$$

We want to look at what happens to the LASSO estimate $\widehat{\boldsymbol{\beta}}_\lambda$ as $\lambda \downarrow 0$.

(a) Show that $\widehat{\boldsymbol{\beta}}_\lambda$ minimizes

$$\frac{1}{\lambda}\left\{\sum_{i=1}^{n}(y_i - \bar{y} - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 - \tau\right\} + \sum_{j=1}^{p}|\beta_j|.$$

(b) Find the limit of

$$\frac{1}{\lambda}\left\{\sum_{i=1}^{n}(y_i - \bar{y} - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 - \tau\right\}$$

as $\lambda \downarrow 0$ as a function of $\boldsymbol{\beta}$. (What happens when $\boldsymbol{\beta} \notin \mathcal{C}$?) Use this to deduce that as $\lambda \downarrow 0$, $\widehat{\boldsymbol{\beta}}_\lambda \to \widehat{\boldsymbol{\beta}}_0$ where $\widehat{\boldsymbol{\beta}}_0$ minimizes $\sum_{j=1}^{p}|\beta_j|$ on the set $\mathcal{C}$.

(c) Show that $\widehat{\boldsymbol{\beta}}_0$ is the solution of a linear programming problem. (Hint: Note that $\mathcal{C}$ can be expressed in terms of $\boldsymbol{\beta}$ satisfying p linear equations.)

4. Consider minimizing the function

$$g(x) = x^2 - 2\alpha x + \lambda|x|^\gamma$$

where $\lambda > 0$ and $0 < \gamma < 1$. (This problem arises, in a somewhat more complicated form, in shrinkage estimation in regression.) The function $|x|^\gamma$ has a "cusp" at 0, which mean that if λ is sufficient large then g is minimized at $x = 0$.

(a) g is minimized at $x = 0$ if, and only if,

$$\lambda \geq \frac{2}{2-\gamma}\left[\frac{2-2\gamma}{2-\gamma}\right]^{1-\gamma}|\alpha|^{2-\gamma}. \tag{1}$$

Otherwise, g is minimized at x^* satisfying $g'(x^*) = 0$. Using R, compare the following two iterative algorithms for computing x^* (when condition (1) does not hold):

(i) Set $x_0 = \alpha$ and define

$$x_k = \alpha - \frac{\lambda\gamma}{2}\frac{|x_{k-1}|^\gamma}{x_{k-1}} \quad k = 1, 2, 3, \cdots$$

(ii) The Newton-Raphson algorithm with $x_0 = \alpha$.

Use different values of α, γ, and λ to test these algorithms. Which algorithm is faster?

(b) Functions like g arise in so-called bridge estimation in linear regression (which are generalizations of the LASSO) – such estimation combines the features of ridge regression (which shrinks least squares estimates towards 0) and model selection methods (which produce exact 0 estimates for some or all parameters). Bridge estimates $\widehat{\boldsymbol{\beta}}$ minimize (for some $\gamma > 0$ and $\lambda > 0$),

$$\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|^\gamma. \tag{2}$$

See the paper by Huang, Horowitz and Ma (2008) ("Asymptotic properties of bridge estimators in sparse high-dimensional regression models" *Annals of Statistics.* **36,** 587–613) for details. Describe how the algorithms in part (a) could be used to define a coordinate descent algorithm to find $\widehat{\boldsymbol{\beta}}$ minimizing (2) iteratively one parameter at a time.

(c) Prove that g is minimized at 0 if, and only if, condition (1) in part (a) holds.

5. Suppose that A is a symmetric non-negative definite matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. Consider the following algorithm for computing the maximum eigenvalue λ_1:

$$\text{Given } \boldsymbol{x}_0, \text{ define for } k = 0, 1, 2, \cdots, \boldsymbol{x}_{k+1} = \frac{A\boldsymbol{x}_k}{\|A\boldsymbol{x}_k\|_2} \text{ and } \mu_{k+1} = \frac{\boldsymbol{x}_{k+1}^T A\boldsymbol{x}_{k+1}}{\boldsymbol{x}_{k+1}^T\boldsymbol{x}_{k+1}}.$$

Under certain conditions, $\mu_k \to \lambda_1$, the maximum eigenvalue of A; this algorithm is known as the **power method** and is particularly useful when A is sparse.

(a) Suppose that $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$ are the eigenvectors of A corresponding to the eigenvalues $\lambda_1, \cdots, \lambda_n$. Show that $\mu_k \to \lambda_1$ if $\boldsymbol{x}_0^T\boldsymbol{v}_1 \neq 0$ and $\lambda_1 > \lambda_2$.

(b) What happens to the algorithm if if the maximum eigenvalue is not unique, that is, $\lambda_1 = \lambda_2 = \cdots = \lambda_k$?

6. (a) Suppose that A is an invertible matrix that can be written as $I - B$ where B has its eigenvalues in the interval $(-1, 1)$. Show that

$$\mathrm{tr}(A^{-1}) = \sum_{k=0}^{\infty} \mathrm{tr}(B^k)$$

(where $B^0 = I$).

(b) We can use Hutchinson's method to estimate $\mathrm{tr}(A^{-1})$ by exploiting the formula in part (a), truncating the infinite series at some finite point r (where $B^k = 0$ for $k > r$). The key lies in writing $A = \alpha(I - B)$ for some constant α and matrix B whose eigenvalues lie in $(-1, 1)$; then

$$\mathrm{tr}(A^{-1}) = \alpha^{-1} \sum_{k=0}^{\infty} \mathrm{tr}(B^k).$$

Suppose that $\lambda_1, \cdots, \lambda_n > 0$ are the eigenvalues of A and define μ so that

$$\mu \geq \max_{1 \leq i \leq n} \lambda_i.$$

In terms of μ, how would you define α and B?

(c) Suppose that A is symmetric positive definite with elements $\{a_{ij} : 1 \leq i, j \leq n\}$. Show that we can take μ in part (b) to be

$$\mu = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|.$$

(Hint: Show that $\mu = \|A\|_\infty$.)