MA 589 — Computational Statistics

Project 1

(Due: Friday, 09/28/18)

1. (WaRming up) Write (`R`) functions that return:

   (a) The inverse or the transpose inverse of an upper triangular matrix. Call this function `inv.upper.tri` and provide a `transpose` argument to specify if the transpose is requested. Hint: use `backsolve`.

   (b) The L_2 norm of vector $\mathbf{v}$ with n entries, defined as $\texttt{norm2}(\mathbf{v}) = \|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2} = \sqrt{v^T v}$. Quick check: if `u <- 1e200 * rep(1, 100)`, what is `norm2(u)`?[1]

   (c) The column-normalization U of matrix A, $U_{ij} = A_{ij}/\|A_{\cdot,j}\|$ (call this function `normalize.cols`, and feel free to use `norm2` above).

   (d) The projection of vector $\mathbf{a}$ into $\mathbf{u}$ (called as `proj(a, u)`) ,

   $$\mathsf{proj}_{\mathbf{u}}(\mathbf{a}) = \frac{\mathbf{u}^\top \mathbf{a}}{\|\mathbf{u}\|^2}\mathbf{u}.$$

   Quick check: what is `proj(1:100, u)`, `u` as in (b) above?

   (e) The Vandermonde matrix of vector $\mathbf{a} = [a_i]_{i=1,\ldots,n}$ and degree d,

   $$\mathbb{V}(\mathbf{a}, d) = \begin{bmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^d \\ 1 & a_2 & a_2^2 & \cdots & a_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_n & a_n^2 & \cdots & a_n^d \end{bmatrix} = [a_i{}^{j-1}]_{i=1,\ldots,n,\ j=1,\ldots,d+1}$$

   (called as `vandermonde(a, d)`).

2. The *machine epsilon*, ϵ, can be defined as the smallest floating point (with base 2) such that $1 + \epsilon > 1$, that is, $1 + \epsilon/2 == 1$ in machine precision.

   (a) Write a function that returns this value by starting at `eps = 1` and iteratively dividing by 2 until the definition is satisfied.

   (b) Write a function that computes $f(x) = \log(1 + \exp(x))$ and then evaluate: $f(0)$, $f(-80)$, $f(80)$, and $f(800)$.

   (c) How would you specify your function to avoid computations if $x \ll 0$ ($x < 0$ and $|x|$ is large)? (Hint: ϵ.)

   (d) How would you implement your function to not overflow if $x \gg 0$?

[1] `1e201`, of course.

3. (QR via Gram-Schmidt) If $A = [\mathbf{a}_1\, \mathbf{a}_2\, \cdots \mathbf{a}_p]$ is a n-by-p matrix with $n > p$ then we can obtain a "thin" QR decomposition of A using Gram-Schmidt orthogonalization[2]. To get Q, we start with $\mathbf{u}_1 = \mathbf{a}_1$, then compute iteratively, for $i = 2, \ldots, p$,

$$\mathbf{u}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} \mathsf{proj}_{\mathbf{u}_j}(\mathbf{a}_i),$$

and finally, with $\mathbf{q}_i = \mathbf{u}_i / \|\mathbf{u}_i\|$, set $Q = [\mathbf{q}_1\, \mathbf{q}_2\, \cdots \mathbf{q}_p]$ as a column-normalized version of $U = [\mathbf{u}_i]_{i=1,\ldots,p}$.

   (a) Show that $C = Q^\top A$ is upper triangular and that C is the Cholesky factor of $A^\top A$.

   (b) Write a R function that computes the Q orthogonal factor of a Vandermonde matrix with base vector $\mathbf{x}$ and degree d *without computing the Vandermonde matrix explicitly*, that is, as your function iterates to compute $\mathbf{u}_i$, compute and use the columns of the Vandermonde matrix on the fly.

   (c) It can be shown that with $\eta_1 = 1$,

$$\eta_{i+1} = \mathbf{u}_i^\top \mathbf{u}_i, \quad \text{and} \quad \alpha_i = \frac{\mathbf{u}_i^\top \mathrm{Diag}(\mathbf{x}) \mathbf{u}_i}{\mathbf{u}_i^\top \mathbf{u}_i}, \qquad \text{for } i = 1, \ldots, d+1,$$

   where $\mathrm{Diag}(\mathbf{x})$ is a diagonal matrix with diagonal entries $\mathbf{x}$, the U matrix can be computed using the recurrence relation: $\mathbf{u}_1 = \mathbb{1}_n$, $\mathbf{u}_2 = \mathbf{x} - \alpha_1 \mathbb{1}_n$, and, for $i = 2, \ldots, d$,

$$(\mathbf{u}_{i+1})_j = (x_j - \alpha_i)(\mathbf{u}_i)_j - \frac{\eta_{i+1}}{\eta_i}(\mathbf{u}_{i-1})_j, \qquad \text{for } j = 1, \ldots, n.$$

   Write a R function that, given η and α, computes Q.

   (d) Now modify your function in (a) to also compute α and η. Quick check for the "compact" representation of Q: show that $\alpha_1 = \bar{\mathbf{x}}$, $\eta_2 = n$, and $\eta_3 = (n-1)s_{\mathbf{x}}^2$, where $\bar{\mathbf{x}}$ and $s_{\mathbf{x}}^2$ are the sample mean and variance of $\mathbf{x}$ respectively.

4. (Orthogonal staged regression) Suppose we observe $\mathbf{y} \sim N(X\beta, \sigma^2 I_n)$, X a n-by-p full rank matrix, and wish to test

$$H_0 : \beta_j = \beta_{j+1} = \cdots = \beta_p = 0$$

for some $j \geq 1$. It can be shown that the ML estimator $\hat{\beta} = (X^\top X)^{-1} X^\top \mathbf{y}$ has distribution $\hat{\beta} \sim N(\beta, \sigma^2 (X^\top X)^{-1})$.

   (a) If X has thin QR decomposition $X = QR$, show that H_0 is equivalent to testing $\gamma_j = \cdots = \gamma_p = 0$ where $\gamma = R\beta$, and so we can regress $\mathbf{y}$ on Q instead of X, that is, estimate γ from $\mathbf{y} \sim N(Q\gamma, \sigma^2 I_n)$.

[2] This is not recommended in practice because it is numerically unstable. Orthogonalizations are usually performed using Givens rotations or Householder reflections.

(b) Show that the ML estimator for γ is $\hat{\gamma} = Q^\top \mathbf{y}$ and the components of $\hat{\gamma}$ are *independent*.

(c) Using R, explain how you compute: (i) the ML estimate $\hat{\beta}$ as a function of $\hat{\gamma}$, and (ii) the correlation matrix of $\hat{\beta}$ using only `crossprod`, `normalize.cols`, and `inv.upper.tri`.

(d) As a concrete example, let us use the `cars` dataset[3] to fit a polynomial regression of degree $d = 3$. Take `dist` to be the response $\mathbf{y}$ and `speed` to be the base vector $\mathbf{x}$ to define $X = \mathbb{V}(\mathbf{x}, d)$, a Vandermonde design.

  i. Compute Q using the routine from 3.b, obtain $\hat{\gamma} = Q^\top \mathbf{y}$ and compare it to the estimate from `coef(lm(dist ~ Q - 1))`.

  ii. Compute $\hat{\beta}$ according to (c) and compare it to the estimate from `coef(lm(dist ~ vandermonde(speed, 3) - 1))`

  iii. ($*$) Compare the p-values when testing $\gamma_j = 0$ and $\beta_j = 0$ for $j = 1, \ldots, 4$ from the above regression fits. How would you explain the discrepancies based on the results from (b) and (c)? What degree would you recommend for a polynomial regression of `dist` on `speed`?

(e) ($*$) Suppose we use another routine to obtain a QR decomposition of X. Under what conditions is R the Cholesky factor of $X^\top X$? Write a function that returns a QR decomposition of X satisfying these conditions.

[3] Load it in R with `data(cars)`.