EECE 7220 / 8220: Scientific Computing

Fall 2018

Instructor: Madhu Balasubramanian, Ph.D.

Instructor's Email: `mblsbrmn@memphis.edu`

Homework: 4

Due: Nov 15, 2018

Submission Instructions:

- Upload your programs to the homework dropbox in http://elearn.memphis.edu
- Prepare a PDF report of your solution and results (solution vector, etc.) for the problems and upload the PDF report to the homework dropbox.

Learning Objective: To understand how to solve a rectangular matrix system of equations (least squares problem) using: 1) QR decomposition, 2) Moore-Penrose inverse (from *normal equation*), 3) using SVD, and 4) using an optimization procedure.

Data Generation: To evaluate your algorithms (programs), generate test data $\{(x_i, y_i)\}_{i=1}^{100}$ that has quadratic relationship as follows:

1. Let the underlying relationship between variables x_i (independent) and y_i (dependent) be quadratic. i.e. $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$.
2. Choose any non-zero values for the quadratic model parameters β_0, β_1, and β_2.
3. Generate 100 data points from the quadratic model: $\{(x_i, y_i)\}_{i=1}^{1}00$, where $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$. For example, choose $x_i = -50$ to $49; \beta_0 = 100; \beta_1 = -2;$ and $\beta_1 = 3$.
4. Now the generated data points $\{(x_i, y_i)\}_{i=1}^{100}$ lies exactly on a quadratic polynomial.
5. Add random variations to each of the data points as follows: $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$, where the random variable ϵ_i follows a normal distribution with 0 mean and σ standard deviation. i.e. $\epsilon_i \sim Normal(0, \sigma^2)$. You can choose any variance σ, e.g. $\sigma = 500$. <u>Hint</u>: In Matlab, you can use `eps = normrnd(mu, sigma, 100)` to draw 100 random ϵ_i from a normal distribution with mean `mu = 0` and standard deviation `sigma`.
6. <u>Example model parameters for data generation</u>: $\beta_0 = 100; \beta_1 = -2; \beta_2 = 3; \vec{x} = \{-50, -49, -48, \ldots, 48, 49\}\,; \sigma = 500$
7. Least squares problem to be solved: $X_{100\times3}\vec{\beta}_{3\times1} = \vec{y}_{100\times1}$

AUXILIARY ALGORITHMS

Modified Gram-Schmidt Orthogonalization Procedure:

Input : A set of r linearly independent vectors $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_r\}$

Output: A set of r orthonormal vectors $\{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_r\}$ such that $\mathtt{Span}\left(\{\vec{x}_i\}_{i=1}^{r}\right) = \mathtt{Span}\left(\{\vec{q}_i\}_{i=1}^{r}\right)$

$r_{11} \leftarrow \|\vec{x}_1\|$;
if $r_{11} = 0$ **then**
    Stop
else
    $\vec{q}_1 \leftarrow \frac{x_1}{r_{11}}$
end
for $j \leftarrow 2$ **to** r **do**
    $\hat{q} \leftarrow \vec{x}_j$;
    for $i \leftarrow 1$ **to** $j-1$ **do**
        $r_{ij} = (\hat{q}, \vec{q}_i)$;
        $\hat{q} = \hat{q} - r_{ij}\vec{q}_i$;
    end
    $r_{jj} \leftarrow \|\hat{q}\|$;
    if $r_{jj} = 0$ **then**
        Stop;
    else
        $\vec{q}_j = \frac{\hat{q}}{r_{jj}}$;
    end
end

Algorithm 1: Modified Gram-Schmidt Orthogonalization Procedure

Back-substitution Procedure *to solve* $R\vec{\beta} = \vec{y}$:

Input : 1. An upper triangular matrix $R \in \mathbb{C}^{n\times n}$; and 2. right-side vector $\vec{y} \in \mathbb{C}^{n\times 1}$

Output: $\vec{\beta} \in \mathbb{C}^{n\times 1}$

for $j \leftarrow n$ **to** 1 **do**

$$\beta_j = \frac{y_j - \sum_{k=j+1}^{n} \beta_k \, r_{jk}}{r_{jj}}$$

end

Algorithm 2: Back-substitution procedure

Homework Problem:

1. Using QR decomposition, write a program to solve for the model parameters $\vec{\beta}$ using the generated data above: $X_{100\times3}\vec{\beta}_{3\times1} = \vec{y}_{100\times1}$:

   · With $X = QR$,

   $$\begin{aligned} X\vec{\beta} &= \vec{y} \\ QR\vec{\beta} &= \vec{y} \end{aligned}$$

   · Multiplying both sides by Q^* and also using the fact that for rectangular matrices with orthogonal columns $Q^*Q = I$,

   $$\begin{aligned} Q^*QR\vec{\beta} &= Q^*\vec{y} \\ R\vec{\beta} &= Q^*\vec{y} \end{aligned}$$

   · Recognize that R is an upper-triangular matrix.

   · Solve for the unknown parameters $\vec{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$ from $R\vec{\beta} = Q^*\vec{y}$ using the *back-substitution* procedure.

2. Write a program to solve for the model parameters $\vec{\beta}$ from the *normal equation* of $X_{100\times3}\vec{\beta}_{3\times1} = \vec{y}_{100\times1}$ using the data generated above. (refer to the lectures notes for details of solution using the normal equation).

3. Using singular value decomposition (use your SVD code from homework 1), write a program to solve for the model parameters $\vec{\beta}$ from $X_{100\times3}\vec{\beta}_{3\times1} = \vec{y}_{100\times1}$ using the data generated above:

   · $X \in \mathbb{C}^{m\times n}$ where $m > n$.

   · Using SVD, $X = U\Sigma V^*$ where the columns of U and V are orthogonal.

   $$\begin{aligned} X\vec{\beta} &= \vec{y} \\ U\Sigma V^*\vec{\beta} &= \vec{y} \end{aligned}$$

   · Multiplying both sides by U^* and using the fact that $U^*U = I$ for matrices with orthogonal columns,

   $$\begin{aligned} U^*U\Sigma V^*\vec{\beta} &= U^*\vec{y} \\ \Sigma V^*\vec{\beta} &= U^*\vec{y} \\ \vec{\beta} &= (\Sigma V^*)^{-1}\, U^*\vec{y} \end{aligned}$$

- We can observe that ΣV^* is a diagonal matrix and $\vec{\beta}$ is solved much easily using SVD.

4. Using unconstrained minimization, write a program to solve for the model parameters $\vec{\beta}$ from $X_{100\times3}\vec{\beta}_{3\times1} = \vec{y}_{100\times1}$ using the data generated above. <u>Note</u>: You can use the built-in MATLAB function `fminunc()` to solve the problem.

   $\hat{\beta} = \underset{\vec{\beta}}{\operatorname{argmin}} \|\vec{y} - X\vec{\beta}\|^2$