Homework 3, CS 6210, Fall 2020
Instructor: Austin R. Benson
Due Friday, October 16, 2020 at 10:19am ET on CMS (before lecture)

Policies

Collaboration. You are encouraged to discuss and collaborate on the homework, but you have to write up your own solutions and write your own code.
Programming language. You can use any programming language for the coding parts of the assignment. Code snippets that we provide and demos in class will use Julia.
Typesetting. Your write-up should be typeset with LaTeX. Handwritten homeworks are not accepted.
Submission. Submit your write-up as a single PDF on CMS: `https://cmsx.cs.cornell.edu`.

Problems

1. *Underdetermined least squares.*

   Recall our linear least squares problem:

$$\min_x \|Ax - b\|_2^2. \tag{1}$$

   For this question, let $A \in \mathbb{R}^{m \times n}$ with $m < n$ and suppose that $\text{rank}(A) = m$ (i.e., A has full row rank). In class, we considered the case of $m > n$, which is called *overdetermined* since there are more equations than unknowns. The $m < n$ case is called *underdetermined*.

   (a) Show that Eq. (1) does not have a unique minimizer.

   (b) One reasonable way to select a minimizer is to pick the one of minimal norm. Let $S \subset \mathbb{R}^n$ be the set of minimizers for Eq. (1). The *least 2-norm solution* is

$$x_{\text{ln}} = \arg\min_{x \in S} \|x\|_2^2. \tag{2}$$

   Show that $x_{\text{ln}} = \left((A^T)^+\right)^T b$, where $(A^T)^+$ is the Moore–Penrose pseudoinverse for A^T.

   (c) Let $\lambda > 0$ be a constant. Show that the solution to

$$\min_x \|Ax - b\|_2^2 + \lambda^2 \|x\|_2^2 \tag{3}$$

   is unique.

   (d) Let x_λ be the solution to Eq. (3). Show that $\lim_{\lambda \to 0} x_\lambda = x_{\text{ln}}$.

2. *Updating a QR factorization.*

   Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, with $m > n$ and A full rank. Suppose that we have already used a QR factorization of A to solve the linear least squares problem

$$\min \|Ax - b\|_2^2.$$

   Now let $a_1 \in \mathbb{R}^n$ and $b_1 \in \mathbb{R}$. Design an $O(n^2)$ algorithm to solve

$$\arg\min_{x \in \mathbb{R}^n} \left\| \begin{bmatrix} a_1^T \\ A \end{bmatrix} x - \begin{bmatrix} b_1 \\ b \end{bmatrix} \right\|$$

   This is useful when, e.g., we collect new data and want to update our solution.

3. *Tall-and-skinny QR.*

   Let $A \in \mathbb{R}^{m \times n}$ with $m \geq 4n$ and consider A in the following blocked form

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix},$$

   where $A_i \in \mathbb{R}^{m_i \times n}$ with $m_i \geq n$. Suppose that we compute the following:

   (i) A QR factorizations of each block: $A_i = Q_i R_i$ for $i = 1, \ldots, 4$.

   (ii) Two more QR factorizations $\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = Q_5 R_5$ and $\begin{bmatrix} R_3 \\ R_4 \end{bmatrix} = Q_6 R_6$.

   (iii) A final QR factorization $\begin{bmatrix} R_5 \\ R_6 \end{bmatrix} = Q_7 R_7$.

   (a) Show how to write a QR factorization of A in terms of $Q_1, \ldots, Q_7$ and $R_1, \ldots, R_7$.

   (b) When could this type of algorithm be useful?

4. *Conditioning and iterative refinement.*

   (a) Implement algorithms to compute a QR factorization using (i) modified Gram-Schmit, (ii) Cholesky QR, and (iii) Householder transformations.

   (b) Create a sequence of increasingly ill-conditioned matrices $A \in \mathbb{R}^{m \times n}$ and compute QR factorizations $A = \hat{Q}\hat{R}$ using each of your three implementations. Plot $\|\hat{Q}^T\hat{Q} - I\|_2$ as a function of the condition number of A for each algorithm.

   (c) We can "refine" a computed factorization $A = \hat{Q}\hat{R}$ when $\hat{Q}$ is not orthogonal due to roundoff error. To do this, we compute a QR factorization of $\hat{Q}$: $\hat{Q} = \bar{Q}\bar{R}$. Then $A = \bar{Q}(\bar{R}\hat{R})$ is a QR factorization of A. This procedure is called iterative refinement. Perform one step of iterative refinement with each of your three algorithms. Plot $\|\bar{Q}^T\bar{Q} - I\|_2$ as a function of the condition number of A.