Study Problems 3 Foundations of Computational Math 1 Fall 2024

All of these problems are based on the material in the lectures and notes in Set 5 on linear least squares and associated concepts. They are all of the basic type that you should be able to reproduce after studying the solutions.

Problem 3.1

Recall that an elementary reflector has the form $Q = I + \alpha z z^T \in \mathbb{R}^{n\times n}$ with $\|z\|_2 \neq 0$.

3.1.a. Show that Q is orthogonal if and only if

$$\alpha = \frac{-2}{z^T z} \ \text{ or } \ \alpha = 0$$

3.1.b. Given $v \in \mathbb{R}^n$, let $\gamma = \pm\|v\|$ and $z = v + \gamma e_1$. Assuming that $z \neq v$ show that

$$\frac{z^T z}{z^T v} = 2$$

3.1.c. Using the definitions and results above show that $Qv = -\gamma e_1$

Problem 3.2

Consider a Householder reflector, H, in $\mathbb{R}^2$. Show that

$$H = \begin{pmatrix} -\cos(\phi) & -\sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

where ϕ is some angle.

Problem 3.3

Suppose you are given the nonsingular tridiagonal matrix $T \in \mathbb{R}^{n\times n}$ For example, if $n = 6$ then

$$\begin{pmatrix} \alpha_1 & \beta_1 & 0 & 0 & 0 & 0 \\ \gamma_2 & \alpha_2 & \beta_2 & 0 & 0 & 0 \\ 0 & \gamma_3 & \alpha_3 & \beta_3 & 0 & 0 \\ 0 & 0 & \gamma_4 & \alpha_4 & \beta_4 & 0 \\ 0 & 0 & 0 & \gamma_5 & \alpha_5 & \beta_5 \\ 0 & 0 & 0 & 0 & \gamma_6 & \alpha_6 \end{pmatrix}$$

3.3.a Suppose you use Householder reflectors to transform T to upper triangular, i.e.,

$$H_{n-1} \dots H_1 T = R.$$

What is the zero/nonzero structure of R?

3.3.b What is the structure of each of the reflectors H_i?

3.3.c Let $T^{(i)} = H_i H_{i-1} \dots H_1 T$. What is the structure of $T^{(i)}$?

3.3.d What is the computational complexity of the factorization, i.e., what is k in $O(n^k)$? (You do not have to determine the constant in the complexity expression.)

To answer the questions on structure, in addition to characterizing it algebraically, include a $*$, 0 matrix pattern for $n = 6$ to make it clear.

Problem 3.4

3.4.a

This part of the problem concerns the computational complexity question of operation count.

For both LU factorization and Householder reflector-based orthogonal factorization, we have used elementary transformations, T_i, that can be characterized as rank-1 updates to the identity matrix, i.e.,

$$T_i = I + x_i y_i^T, \;\; x_i \in \mathbb{R}^n \text{ and } y_i \in \mathbb{R}^n$$

Gauss transforms and Householder reflectors differ in the definitions of the vectors x_i and y_i. Maintaining computational efficiency in terms of a reasonable operation count usually implies careful application of associativity and distribution when combining matrices and vectors.

Suppose we are to evaluate

$$z = T_3 T_2 T_1 v = (I + x_3 y_3^T)(I + x_2 y_2^T)(I + x_1 y_1^T)v$$

where $v \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$. Show that by using the properties of matrix-matrix multiplication and matrix-vector multiplication, the vector z can be evaluated in $O(n)$ computations (a good choice of version for an algorithm) or $O(n^2)$ computations (a bad choice of version for an algorithm) or $O(n^3)$ computations (a very bad choice of version for an algorithm).

3.4.b

This part of the problem concerns the computational complexity question of storage space.

Recall, that we discussed and programmed an **in-place** implementation of LU factorization that was very efficient in terms of storage space. An array with n^2 entries initialized with $array(I,J) = \alpha_{ij}$ could be used to store the n^2 entries needed to specify L and U, i.e., λ_{ij} for $j < i$, $2 \leq i \leq n$ and $1 \leq j \leq n-1$, and μ_{ij} for $i < j$, $1 \leq i \leq n$ and $1 \leq j \leq n$.

Let $A \in \mathbb{R}^{n \times k}$, $n \geq k$, and $rank(A) = k$. Consider the use of Householder reflectors, H_i, $1 \leq i \leq k$, to transform A to upper trapezoidal form, i.e.,

$$H_k H_{k-1} \cdots H_2 H_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

$$R \in \mathbb{R}^{k \times k} \text{ nonsingular upper triangular}$$

Suppose you are given an array with $n \times k$ entries initialized with $array(I,J) = \alpha_{ij}$ and you are to implement your algorithm using minimal storage.

(i) Are you able to store all of the information needed to specify the H_i, $1 \leq i \leq k$ and R within the array with $n \times k$ entries? Justify your answer.

(ii) If you are not able to store all of the information in the array, how much extra storage do you need and what do you store in it?

Problem 3.5

Let x and y be two vectors in $\mathbb{R}^n$.

3.5.a. Show that given x and y the value of $\|x - \alpha y\|_2$ is minimized when

$$\alpha_{min} = \frac{x^T y}{y^T y}$$

3.5.b. Show that $x = y\alpha_{min} + z$ where $y^T z = 0$, i.e., x is easily written as the sum of two orthogonal vectors with specifed minimization properties.

Problem 3.6

Let $A \in \mathbb{R}^{n \times k}$ have rank k, i.e., have k linearly independent columns. The linear least squares problem

$$\min_{x \in \mathbb{R}^k} \|b - Ax\|_2$$

has a unique solution x_{min} for any $b \in \mathbb{R}^n$. The mapping $b \mapsto x_{min}$ defines a linear transformation, $A^\dagger$, from $\mathbb{R}^n$ to $\mathbb{R}^k$ called the pseudoinverse.

The pseudoinverse for rectangular full column-rank matrices behaves much as the inverse for nonsingular matrices. To see this answer the following questions and show the following identities are true :

3.6.a. Use the normal equations to write $A^\dagger$ in terms of A.

3.6.b. If $A \in \mathbb{R}^{n\times n}$ what is $A^\dagger$?

3.6.c. $AA^\dagger A = A$

3.6.d. $A^\dagger A A^\dagger = A^\dagger$

3.6.e. $A^\dagger A = (A^\dagger A)^T$

3.6.f. $AA^\dagger = (AA^\dagger)^T$

3.6.g. If $A \in \mathbb{R}^{n\times k}$ has orthonormal columns then $A^\dagger = A^T$. Why is this important for consistency with simpler forms of least squares problems that we have discussed?