3.7. References and Other Topics for Chapter 3

The best recent reference on least squares problems is [33], which also discusses variations on the basic problem discussed here (such as constrained, weighted, and updating least squares), different ways to regularize rank-deficient problems, and software for sparse least squares problems. See also chapter 5 of [121] and [168]. Perturbation theory and error bounds for the least squares solution are discussed in detail in [149]. Rank-revealing QR decompositions are discussed in [28, 30, 48, 50, 126, 150, 196, 206, 236]. In particular, these papers examine the tradeoff between cost and accuracy in rank determination, and in [206] there is a comprehensive performance comparison of the available methods for rank-deficient least squares problems.

3.8. Questions for Chapter 3

QUESTION 3.1. *(Easy)* Show that the two variations of Algorithm 3.1, CGS and MGS, are mathematically equivalent by showing that the two formulas for r_{ji} yield the same results in exact arithmetic.

QUESTION 3.2. *(Easy)* This question will illustrate the difference in numerical stability among three algorithms for computing the QR factorization of a matrix: Householder QR (Algorithm 3.2), CGS (Algorithm 3.1), and MGS (Algorithm 3.1). Obtain the Matlab program QRStability.m from HOMEPAGE/Matlab/QRStability.m. This program generates random matrices with user-specified dimensions `m` and `n` and condition number `cnd`, computes their QR decomposition using the three algorithms, and measures the accuracy of the results. It does this with the *residual* $\|A - Q \cdot R\| / \|A\|$, which should be around machine epsilon ε for a stable algorithm, and the *orthogonality of* Q $\|Q^T \cdot Q - I\|$, which should also be around ε. Run this program for small matrix dimensions (such as `m`$= 6$ and `n`$= 4$), modest numbers of random matrices (`samples`$= 20$), and condition numbers ranging from `cnd`$= 1$ up to `cnd`$= 10^{15}$. Describe what you see. Which algorithms are more stable than others? See if you can describe how large $\|Q^T \cdot Q - I\|$ can be as a function of choice of algorithm, `cnd` and ε.

QUESTION 3.3. *(Medium; Hard)* Let A be m-by-n, $m \geq n$, and have full rank.

1. *(Medium)* Show that $\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \cdot \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$ has a solution where x minimizes $\|Ax - b\|_2$. One reason for this formulation is that we can apply iterative refinement to this linear system if we want a more accurate answer (see section 2.5).

2. *(Medium)* What is the condition number of the coefficient matrix in terms of the singular values of A? Hint: Use the SVD of A.

3. *(Medium)* Give an explicit expression for the inverse of the coefficient matrix, as a block 2-by-2 matrix. Hint: Use 2-by-2 block Gaussian elimination. Where have we previously seen the (2,1) block entry?

4. *(Hard)* Show how to use the QR decomposition of A to implement an iterative refinement algorithm to improve the accuracy of x.

QUESTION 3.4. *(Medium) Weighted least squares:* If some components of $Ax-b$ are more important than others, we can weight them with a scale factor d_i and solve the weighted least squares problem $\min \|D(Ax-b)\|_2$ instead, where D has diagonal entries d_i. More generally, recall that if C is symmetric positive definite, then $\|x\|_C \equiv (x^TCx)^{1/2}$ is a norm, and we can consider minimizing $\|Ax-b\|_C$. Derive the normal equations for this problem, as well as the formulation corresponding to the previous question.

QUESTION 3.5. *(Medium; Z. Bai)* Let $A \in \mathbb{R}^{n\times n}$ be positive definite. Two vectors u_1 and u_2 are called A-orthogonal if $u_1^TAu_2 = 0$. If $U \in \mathbb{R}^{n\times r}$ and $U^TAU = I$, then the columns of U are said to be A-orthonormal. Show that every subspace has an A-orthonormal basis.

QUESTION 3.6. *(Easy; Z. Bai)* Let A have the form

$$A = \begin{bmatrix} R \\ S \end{bmatrix},$$

where R is n-by-n and upper triangular, and S is m-by-n and dense. Describe an algorithm using Householder transformations for reducing A to upper triangular form. Your algorithm should not "fill in" the zeros in R and thus require fewer operations than would Algorithm 3.2 applied to A.

QUESTION 3.7. *(Medium; Z. Bai)* If $A = R + uv^T$, where R is an upper triangular matrix, and u and v are column vectors, describe an efficient algorithm to compute the QR decomposition of A. Hint: Using Givens rotations, your algorithm should take $O(n^2)$ operations. In contrast, Algorithm 3.2 would take $O(n^3)$ operations.

QUESTION 3.8. *(Medium; Z. Bai)* Let $x \in \mathbb{R}^n$ and let P be a Householder matrix such that $Px = \pm\|x\|_2e_1$. Let $G_{1,2}, \ldots, G_{n-1,n}$ be Givens rotations, and let $Q = G_{1,2}\cdots G_{n-1,n}$. Suppose $Qx = \pm\|x\|_2e_1$. Must P equal Q? (You need to give a proof or a counterexample.)

QUESTION 3.9. *(Easy; Z. Bai)* Let A be m-by-n, with SVD $A = U\Sigma V^T$. Compute the SVDs of the following matrices in terms of U, Σ, and V:

1. $(A^TA)^{-1}$,

2. $(A^TA)^{-1}A^T$,

3. $A(A^TA)^{-1}$,

4. $A(A^TA)^{-1}A^T$.

QUESTION 3.10. *(Medium; R. Schreiber)* Let A_k be a best rank-k approximation of the matrix A, as defined in Part 9 of Theorem 3.3. Let σ_i be the ith singular value of A. Show that A_k is unique if $\sigma_k > \sigma_{k+1}$.

QUESTION 3.11. *(Easy; Z. Bai)* Let A be m-by-n. Show that $X = A^+$ (the Moore–Penrose pseudoinverse) minimizes $\|AX - I\|_F$ over all n-by-m matrices X. What is the value of this minimum?

QUESTION 3.12. *(Medium; Z. Bai)* Let A, B, and C be matrices with dimensions such that the product A^TCB^T is well defined. Let $\mathcal{X}$ be the set of matrices X minimizing $\|AXB - C\|_F$, and let X_0 be the unique member of $\mathcal{X}$ minimizing $\|X\|_F$. Show that $X_0 = A^+CB^+$. Hint: Use the SVDs of A and B.

QUESTION 3.13. *(Medium; Z. Bai)* Show that the Moore–Penrose pseudoinverse of A satisfies the following identities:

$$\begin{aligned} AA^+A &= A, \\ A^+AA^+ &= A^+, \\ A^+A &= (A^+A)^T, \\ AA^+ &= (AA^+)^T. \end{aligned}$$

QUESTION 3.14. *(Medium)* Prove part 4 of Theorem 3.3: Let $H = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$, where A is square and $A = U\Sigma V^T$ is its SVD. Let $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, $U = [u_1, \ldots, u_n]$, and $V = [v_1, \ldots, v_n]$. Prove that the $2n$ eigenvalues of H are $\pm\sigma_i$, with corresponding unit eigenvectors $\frac{1}{\sqrt{2}}\begin{bmatrix} v_i \\ \pm u_i \end{bmatrix}$. Extend to the case of rectangular A.

QUESTION 3.15. *(Medium)* Let A be m-by-n, $m < n$, and of full rank. Then $\min \|Ax - b\|_2$ is called an *underdetermined least squares problem.* Show that the solution is an $(n-m)$-dimensional set. Show how to compute the unique minimum norm solution using appropriately modified normal equations, QR decomposition, and SVD.

QUESTION 3.16. *(Medium)* Prove Lemma 3.1.

ank-k approxima-
Let σ_i be the ith

hat $X = A^+$ (the
l n-by-m matrices

matrices with di-
et $\mathcal{X}$ be the set of
ique member of $\mathcal{X}$
e SVDs of A and

Penrose pseudoin-

em 3.3: Let
s SVD. Let $\Sigma =$
Prove that the $2n$
rs $\frac{1}{\sqrt{2}}\left[\begin{smallmatrix} v_i \\ \pm u_i \end{smallmatrix}\right]$. Ex-

f full rank. Then
oblem. Show that
mpute the unique
al equations, QR

QUESTION 3.17. *(Hard)* In section 2.6.3, we showed how to reorganize Gaussian elimination to perform Level 2 BLAS and Level 3 BLAS at each step in order to exploit the higher speed of these operations. In this problem, we will show how to apply a sequence of Householder transformations using Level 2 and Level 3 BLAS.

1. Let $u_1, \ldots, u_b$ be a sequence of vectors of dimension n, where $\|u_i\|_2 = 1$ and the first $i-1$ components of u_i are zero. Let $P = P_b \cdot P_{b-1} \cdots P_1$, where $P_i = I - 2u_iu_i^T$ is a Householder transformation. Show that there is a b-by-b lower triangular matrix T such that $P = I - UTU^T$, where $U = [u_1, \ldots, u_b]$. In particular, provide an algorithm for computing the entries of T. This identity shows that we can replace multiplication by b Householder transformations P_1 through P_b by three matrix multiplications by U, T, and U^T (plus the cost of computing T).

2. Let House(x) be a function of the vector x which returns a unit vector u such that $(I - 2uu^T)x = \|x\|_2 e_1$; we showed how to implement House(x) in section 3.4. Then Algorithm 3.2 for computing the QR decomposition of the m-by-n matrix A may be written as

   for $i = 1 : m$
   $\quad u_i = \text{House}(A(i : m, i))$
   $\quad P_i = I - 2u_iu_i^T$
   $\quad A(i : m, i : n) = P_iA(i : m, i : n)$
   endfor

   Show how to implement this in terms of the Level 2 BLAS in an efficient way (in particular, matrix-vector multiplications and rank-1 updates). What is the floating point operation count? (Just the high-order terms in n and m are enough.) It is sufficient to write a short program in the same notation as above (although trying it in Matlab and comparing with Matlab's own QR factorization are a good way to make sure that you are right!).

3. Using the results of step (1), show how to implement QR decomposition in terms of Level 3 BLAS. What is the operation count? This technique is used to accelerate the QR decomposition, just as we accelerated Gaussian elimination in section 2.6. It is used in the LAPACK routine `sgeqrf`.

QUESTION 3.18. *(Medium)* It is often of interest to solve *constrained least squares problems*, where the solution x must satisfy a linear or nonlinear constraint in addition to minimizing $\|Ax - b\|_2$. We consider one such problem here. Suppose that we want to choose x to minimize $\|Ax - b\|_2$ subject to the linear constraint $Cx = d$. Suppose also that A is m-by-n, C is p-by-n, and C has full rank. We also assume that $p \le n$ (so $Cx = d$ is guaranteed to be consistent) and $n \le m + p$ (so the system is not underdetermined). Show that there is a unique solution under the assumption that $\left[\begin{smallmatrix} A \\ C \end{smallmatrix} \right]$ has full column rank. Show how to compute x using two QR decompositions and some matrix-vector multiplications and solving some triangular systems of equations. Hint: Look at LAPACK routine `sgglse` and its description in the LAPACK manual [10] (NETLIB/lapack/lug/lapack_lug.html).

QUESTION 3.19. *(Hard; Programming)* Write a program (in Matlab or any other language) to update a geodetic database using least squares, as described in Example 3.3. Take as input a set of "landmarks," their approximate coordinates (x_i, y_i), and a set of new angle measurements θ_j and distance measurements L_{ij}. The output should be corrections $(\delta x_i, \delta y_i)$ for each landmark, an error bound for the corrections, and a picture (triangulation) of the old and new landmarks.

QUESTION 3.20. *(Hard)* Prove Theorem 3.4.

QUESTION 3.21. *(Medium)* Redo Example 3.1, using a rank-deficient least squares technique from section 3.5.1. Does this improve the accuracy of the high-degree approximating polynomials?

Nonsymm

4.1. Introdu

We discuss canoni
and algorithms fo
A (in section 4.4)
ric matrices A =
to eigenvalue pro
applications from
ential equations,
all the canonical
in a list.

One can roug
direct methods an
ods, which are in
eigenvectors. Di
$O(n^3)$ operations
atively insensitiv

The main dire
(see section 4.4.8
able service, con
observed, analyz
gence proof, even
the problem of d
(and quickly!) c
still iterate, since
zeros of polynom
method *direct* if
fixed number of i

Iterative meth
to sparse matric
only convenient