COL 726 Homework 2

27 January–10 February, 2020

Updated text is highlighted in blue.

1. (a) Find the singular value decomposition of the 2×2 matrix $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

   (b) Given a matrix $\mathbf{A} \in \mathbb{C}^{m\times m}$ with singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, find a singular value decomposition of $\begin{bmatrix} \mathbf{A} & \mathbf{A} \\ \mathbf{A} & \mathbf{A} \end{bmatrix} \in \mathbb{C}^{2m\times 2m}$. Does your answer agree with with that of part (a) when applied to $\mathbf{A} = [1]$?

2. (a) Consider two full-rank matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{m\times n}$ with $m \geq n$. Prove that $\mathbf{AB}^*$ and $\mathbf{BA}^*$ are projectors if and only if $\mathbf{A}^*\mathbf{B} = \mathbf{B}^*\mathbf{A} = \mathbf{I}$. You may use the fact that $\mathbf{A}^*\mathbf{A}$ and $\mathbf{B}^*\mathbf{B}$ are invertible.

   (b) Under what additional conditions is $\mathbf{AB}^*$ an orthogonal projector, other than the trivial case $\mathbf{A} = \mathbf{B}$? Give an answer in terms of the singular value decompositions of $\mathbf{A}$ and $\mathbf{B}$.

3. Show how to use the QR decomposition to find bases for the following subspaces of $\mathbb{C}^m$. You may assume all matrices are full rank.

   (a) An orthonormal basis for $\operatorname{null}(\mathbf{C})$, where $\mathbf{C} \in \mathbb{C}^{p\times m}$ with $p < m$.
   Hint: Consider the full QR decomposition of $\mathbf{C}^*$.

   (b) An orthonormal basis for $\operatorname{null}(\mathbf{C}) \cap \operatorname{null}(\mathbf{D})$, where $\mathbf{C} \in \mathbb{C}^{p_1\times m}, \mathbf{D} \in \mathbb{C}^{p_2\times m}$ with $p_1, p_2 < p_1 + p_2 < m$.

   (c) A basis for $\operatorname{range}(\mathbf{A}) \cap \operatorname{null}(\mathbf{C})$, where $\mathbf{A} \in \mathbb{C}^{m\times n}, \mathbf{C} \in \mathbb{C}^{p\times m}$ with $p < n < m$.

   (d) A basis for $\operatorname{range}(\mathbf{A}) \cap \operatorname{range}(\mathbf{B})$, where $\mathbf{A} \in \mathbb{C}^{m\times n_1}, \mathbf{B} \in \mathbb{C}^{m\times n_2}$ with $n_1, n_2 < m < n_1 + n_2$.

4. Just as we have many different norms on $\mathbb{C}^m$, we can have different inner products other than the standard inner product $\mathbf{u}^*\mathbf{v}$. In particular, for any Hermitian positive definite matrix $\mathbf{W} \in \mathbb{C}^{m\times m}$, we can define a weighted inner product $(\mathbf{u}, \mathbf{v})_\mathbf{W} := \mathbf{u}^*\mathbf{W}\mathbf{v}$ and weighted norm $\|\mathbf{u}\|_\mathbf{W} = \sqrt{(\mathbf{u}, \mathbf{u})_\mathbf{W}}$.

   (a) A weighted least-squares problem is the problem of minimizing $\|\mathbf{Ax} - \mathbf{b}\|_\mathbf{W}$ for given $\mathbf{A} \in \mathbb{C}^{m\times n}$ and $\mathbf{b} \in \mathbb{C}^m$. Assuming $\mathbf{A}$ is full rank with $m > n$, show that if $\mathbf{x}$ satisfies $(\mathbf{Ax} - \mathbf{b}, \mathbf{a})_\mathbf{W} = 0$ for all $\mathbf{a} \in \operatorname{range}(\mathbf{A})$, then it is the unique minimizer. Conclude that $\mathbf{A}^*\mathbf{W}\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{W}\mathbf{b}$.

(b) The analogue of the QR decomposition is a factorization $\mathbf{A} = \mathbf{ZR}$ (or $\hat{\mathbf{Z}}\hat{\mathbf{R}}$) where the columns of $\mathbf{Z}$ satisfy $(\mathbf{z}_i, \mathbf{z}_i)_\mathbf{W} = 1$, $(\mathbf{z}_i, \mathbf{z}_j)_\mathbf{W} = 0$ for $i \neq j$. Design and implement either a modified Gram-Schmidt or Householder algorithm for computing such a factorization. Describe your algorithm in the report, and submit your code implementing a function `[Z, R] = weighted_qr(A, W)`.

To check that your algorithm is correct, test it on the input

$$\mathbf{A} = \mathbf{I}, \qquad \mathbf{W} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix},$$

and verify that $\mathbf{A} = \mathbf{ZR}$, that $\mathbf{R}$ is upper triangular, and that the columns of $\mathbf{Z}$ have the desired properties. You do not have to submit this part.

(c) How would you use such a decomposition to solve the weighted least-squares problem in part (a) in $O(m^2)$ time?

5. In this exercise, you will prove the backward stability of one step in the Householder algorithm. You may assume that vector addition, scalar multiplication, norms, and inner products have been shown to be backward stable. Let ϵ_m denote the machine precision.

(a) Let $\mathbf{x} = [x_1, \ldots, x_m]^T$ be a vector with $x_1 \neq 0$, and consider the corresponding Householder reflection, i.e. let $\mathbf{v}$ be the unit vector along $\mathbf{x} + \operatorname{sign}(x_1)\|\mathbf{x}\|_2\mathbf{e}_1$. Suppose the vector computed with floating-point arithmetic is $\tilde{\mathbf{v}}$ instead. Show that $\|\tilde{\mathbf{v}} - \mathbf{v}\|_2 = O(\epsilon_\mathrm{m})$.

(b) Consider a vector $\mathbf{y} \in \mathbb{C}^m$. We would like to compute $\mathbf{b} = \mathbf{y} - 2(\mathbf{v}^*\mathbf{y})\mathbf{v}$, but we use $\tilde{\mathbf{v}}$ instead of $\mathbf{v}$, then incur further floating-point error, and end up with a vector $\tilde{\mathbf{b}}$ instead. Prove that $\tilde{\mathbf{b}} = \tilde{\mathbf{y}} - 2(\mathbf{v}^*\tilde{\mathbf{y}})\mathbf{v}$ for some $\tilde{\mathbf{y}}$ such that $\|\tilde{\mathbf{y}} - \mathbf{y}\|_2 = O(\|\mathbf{y}\|_2\epsilon_\mathrm{m})$.

6. Here we will explore a least-squares strategy to fit a curve not of the form $y = f(x)$. We are given a set of points $(x_1, y_1), \ldots, (x_m, y_m)$ representing observations of a planet's positions:

x_i	10.2	9.5	8.7	7.7	6.7	5.6	4.4	3.0	1.6	0.1
y_i	3.9	3.2	2.7	2.2	1.8	1.5	1.3	1.2	1.3	1.5

The orbit is assumed to be an ellipse of the form $f(x, y) := (1 + a)x^2 + (1 - a)y^2 + 2bxy + cx + dy + e = 0$, and the best-fitting ellipse is obtained by minimizing $\sum_i f(x_i, y_i)^2$.

(a) Formulate this as a linear least-squares problem $\min_\mathbf{p} \|\mathbf{A}\boldsymbol{\theta} - \mathbf{b}\|_2^2$ where $\boldsymbol{\theta} = [a, b, c, d, e]^T$ is the vector of coefficients, and give the form of $\mathbf{A}$ and $\mathbf{b}$ in your report. Write a program to solve it using a built-in QR factorization procedure (`qr` in Matlab, `scipy.linalg.qr` in Python). Report the values of the coefficients, and make a plot showing the points on top of the fitted orbit (drawn using e.g. `fcontour`, see below). Also submit your least-squares code as a function `theta = ellipse_fit(x, y)`.

(b) This is a very ill-conditioned problem, as you can verify for yourself by perturbing the data points. Quantify this ill-conditioning by computing the condition numbers $\kappa_{\mathbf{b}\mapsto\boldsymbol{\theta}}$ and $\kappa_{\mathbf{A}\mapsto\boldsymbol{\theta}}$, using the formulas given by Trefethen and Bau.

(c) Derive an upper bound for the condition number of $\boldsymbol{\theta}$ with respect to the original data vector $\mathbf{v} = [x_1, y_1, \ldots, x_m, y_m]^T$. Use any convenient norm.

Submission: Submit a PDF of your answers for all questions to Gradescope. Submit the code for Questions 4(b) and 6(a) in a zip file to Moodle. Both submissions must be uploaded before the assignment deadline.

The zip file should contain two files, `weighted_qr`.(m|py) and `ellipse_fit`.(m|py), each implementing the relevant function with the exact inputs and outputs specified in the question. Neither function is required to produce any side-effects like printing out values or drawing plots, it just needs to return the correct output values. Any results you are asked to show should go in the PDF.

For plotting the orbit in Question 6(a), you can use `fcontour` which is a built-in function in Matlab. For Python, a simple function providing analogous functionality is provided here: `fcontour.py`.