Updated QR Factorization

10 points

Suppose that we have computed a QR factorization of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with $A = \boldsymbol{QR}$ and wish to efficiently (i.e. with less than $O(n^3)$ cost) compute a QR factorization of $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T$. In this problem, we will derive how to update the previously computed QR factorization of $\boldsymbol{A}$.

1. In order to obtain an updated QR factorization for $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T$, we will begin by obtaining the form $\boldsymbol{Q}(\boldsymbol{R} + \boldsymbol{w}\boldsymbol{v}^T)$. Provide a formula for $\boldsymbol{w}$.
2. Show that $\boldsymbol{R} + \boldsymbol{w}\boldsymbol{v}^T = \boldsymbol{Q}_1(\boldsymbol{Q}_1^T\boldsymbol{R} + c\boldsymbol{e}_1\boldsymbol{v}^T)$, where c is a scalar and $\boldsymbol{Q}_1$ is a series of Givens rotations. Specify which Givens rotations must be done and show that they reduce the matrix $\boldsymbol{w}\boldsymbol{v}^T$ to upper triangular form.

   Use the notation $\boldsymbol{G}_{ij}$ to denote a Givens rotation that introduces a zero in the jth entry using the ith entry (in the case of reducing a vector).
3. Now, show that $\boldsymbol{Q}_1(\boldsymbol{Q}_1^T\boldsymbol{R} + c\boldsymbol{e}_1\boldsymbol{v}^T) = \boldsymbol{Q}_1(\boldsymbol{H})$, where $\boldsymbol{H}$ is an *upper Hessenberg matrix*.

   Upper Hessenberg matrices (https://en.wikipedia.org/wiki/Hessenberg_matrix) are *almost triangular*. In addition to on and above the diagonal, they may also have nonzeros on the first off-diagonal below the diagonal.
4. Finally, derive how we can reduce $\boldsymbol{Q}_1(\boldsymbol{H})$ using Givens rotations into a QR factorization.
5. Show that the cost of the process outlined above is $O(n^2)$. Compare this cost to explicitly computing the QR factorization of $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T$.

Be sure to show all your work and provide justifications for every step to receive full credit.

Please submit your response to this written problem as a PDF file below. You may do either of the following:

- write your response out by hand, scan it, and upload it as a PDF.

  We will not accept unprocessed pictures taken with your phone.

  If you decide to use your phone for scanning, make sure to use an app such as CamScanner (https://www.camscanner.com/) to get a readable PDF. Alternatively, there's a fast and convenient scanner in the Engineering IT office in 2302 Siebel that can just email you a PDF. (It's the Fax-machine-looking thing--not the scanner that's attached to one of the computers.)
- create the PDF using software.

  If you're looking for an easy-ish way to type math, check out TeXmacs (http://texmacs.org/) or LyX (http://www.lyx.org/). Both are installed in the virtual machine. (Under "Applications / Accessories / GNU TeXmacs editor" and "Applications / Office / LyX document processor" respectively.)

Submit your response to each problems in this homework as a separate PDF. If you have multiple PDFs that you need to merge into one, try PDF Split and Merge (http://www.pdfsam.org/download/).

NOTE: Please make sure your solutions are legible and easy to follow. If they are not, we may deduct up to five points *per problem*.

Review uploaded file (blob:https://relate.cs.illinois.edu/b78a605b-be7f-4db3-855c-52368bccbf23) · Embed viewer

Uploaded file*

选择文件 未选择任何文件

Your answer is mostly correct. (85.0 %)

The following feedback was provided:

Part4: You shouldn't use same sets of Givens Rotation. See solution. -.5
Part5: Missing comparison with explicit QR. -1

1) A rank-one update in $\boldsymbol{A}$ is represented by $\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^T$. We are also already given the QR factorization of $\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R}$.

$$\begin{aligned}
&\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^T \\
=&\boldsymbol{Q}\boldsymbol{R}+\boldsymbol{u}\boldsymbol{v}^T \\
=&\boldsymbol{Q}(\boldsymbol{R}+\boldsymbol{Q}^T\boldsymbol{u}\boldsymbol{v}^T) \\
=&\boldsymbol{Q}(\boldsymbol{R}+\boldsymbol{w}\boldsymbol{v}^T)
\end{aligned}$$

So, $\boldsymbol{w}=Q^T\boldsymbol{u}$.

2) Consider $\boldsymbol{R}+\boldsymbol{w}\boldsymbol{v}^T$. We notice that since $\boldsymbol{R}$ is upper triangular, we need to first work on reducing $\boldsymbol{w}\boldsymbol{v}^T$ to be upper triangular. We can also note that if we want to reduce $\boldsymbol{w}\boldsymbol{v}^T$ to be upper triangular, we can accomplish this by reducing $\boldsymbol{w}$.

We reduce $\boldsymbol{w}$ to $c\boldsymbol{e}_1$ using $n-1$ Givens rotations, where c is a scalar and $\boldsymbol{e}_1$ is a vector with a 1 in its first entry and zero everywhere else. We denote a Givens rotation with $\boldsymbol{G}_{ij}$, where the Givens rotation introduces a zero into jth entry using the ith entry (in the case of reducing a vector). We first reduce the last entry in $\boldsymbol{w}$ with $\boldsymbol{G}^T_{(n-1),n}$Then, we reduce the second to last entry with $\boldsymbol{G}^T_{(n-2),(n-1)}$We repeat this process until we have fully reduced $\boldsymbol{w}$. The sequence of Givens rotations required for this reduction will be $\boldsymbol{G}^T_{1,2}\boldsymbol{G}^T_{2,3}\cdots\boldsymbol{G}^T_{(n-2),(n-1)}\boldsymbol{G}^T_{(n-1),n}$We will denote this sequence of rotations as $\boldsymbol{Q}_1$.

$$\begin{aligned}
&\boldsymbol{R}+\boldsymbol{w}\boldsymbol{v}^T \\
=&\boldsymbol{R}+\boldsymbol{Q}_1\boldsymbol{Q}_1^T\boldsymbol{w}\boldsymbol{v}^T \\
=&\boldsymbol{R}+\boldsymbol{Q}_1c\boldsymbol{e}_1\boldsymbol{v}^T \\
=&\boldsymbol{Q}_1(\boldsymbol{Q}_1^T\boldsymbol{R}+c\boldsymbol{e}_1\boldsymbol{v}^T)
\end{aligned}$$

3) From part 2, we have that ce_1v^T is upper triangular. Therefore, we must show that Q_1^TR is upper Hessenberg in order to show that $Q_1^TR+ce_1v^T$ is upper Hessenberg. Since Q_1^T represents the Givens rotations needed, let us first consider the first Givens rotation $\boldsymbol{G}^T_{(n-1),n}$This Givens rotation acts on the

last two rows and will ignore the rest.

$$\begin{aligned}&\boldsymbol{G}^T_{(n-1),n}\boldsymbol{R}\\&=\begin{bmatrix}c & s\\-s & c\end{bmatrix}\begin{bmatrix}0 & \cdots & 0 & R_{(n-1),(n-1)} & R_{(n-1),n}\\0 & \cdots & 0 & 0 & R_{n,n}\end{bmatrix}\\&=\begin{bmatrix}0 & \cdots & 0 & \hat{R}_{(n-1),(n-1)} & \hat{R}_{(n-1),n}\\0 & \cdots & 0 & \hat{R}_{n,(n-1)} & \hat{R}_{n,n}\end{bmatrix}\end{aligned}$$

We can show that $\hat{R}_{n,(n-1)}$ will be nonzero.

$$\begin{aligned}c&=\frac{w_{n-1}}{\sqrt{w_{n-1}^2+w_n^2}}\\s&=\frac{w_n}{\sqrt{w_{n-1}^2+w_n^2}}\end{aligned}$$

Thus, $\hat{R}_{n,(n-1)}=-sR_{(n-1),(n-1)}\neq 0$ since both s and $R_{(n-1),(n-1)}$ are nonzero. Similarly, if we apply the next Givens rotation, the next subdiagonal entry will be

$$\begin{aligned}R_{(n-1),(n-2)}&=-sR_{(n-2),(n-2)}\\&=\frac{-\sqrt{w_{n-1}^2+w_n^2}}{\sqrt{w_{n-2}^2+w_{n-1}^2+w_n^2}}R_{(n-2),(n-2)}\\&\neq 0.\end{aligned}$$

Therefore, $R_{i+1,i}$ which is the subdiagonal entry below the i-th diagonal entry of R, will be

$$R_{i+1,i}=\frac{\|w_{i+1:}\|_2}{\|w_{i:}\|_2}\boldsymbol{R}_{i,i}\neq 0$$

This will introduce a nonzero subdiagonal into R after applying all the Givens rotations. Q_1^TR will be upper Hessenberg and hence $\boldsymbol{H}=\boldsymbol{Q}_1^T\boldsymbol{R}+c\boldsymbol{e}_1\boldsymbol{v}^T$ will be upper Hessenberg.

4) Since $\boldsymbol{H}$ is upper Hessenberg, we can reduce this to upper triangular using $n-1$ Givens rotations. The process for reducing an upper Hessenberg to upper triangular is similar to reducing a vector. This will result in a sequence of Givens rotations $\hat{\boldsymbol{G}}^T_{1,2}\hat{\boldsymbol{G}}^T_{2,3}\cdots\hat{\boldsymbol{G}}^T_{(n-2),(n-1)}\hat{\boldsymbol{G}}^T_{(n-1),n}$ which we will denote as $\hat{\boldsymbol{Q}}^T$.

$$=\boldsymbol{Q}_1\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^T\boldsymbol{H}=\bar{\boldsymbol{Q}}\bar{\boldsymbol{R}}$$

where $\bar{\boldsymbol{Q}}=\boldsymbol{Q}_1\hat{\boldsymbol{Q}}$ and $\bar{\boldsymbol{R}}=\hat{\boldsymbol{Q}}^T\boldsymbol{H}$ is upper triangular.

5) Updating the QR factorization will require performing $O(n)$ Givens rotations. Therefore, the cost will be $O(n^2)$. The explicit computation of the QR factorization would require $O(n^3)$ FLOPS. We decrease the cost by $O(n)$.