CSE 383C: Numerical Linear Algebra

Fall 2016

1 Algorithm Complexity

Let $A \in \mathbb{C}^{m \times n}$, $x, y, z \in \mathbb{C}^n$, $H \in \mathbb{C}^{m \times m}$, $u, v \in \mathbb{C}^m$.

i. **Inner Product** $x^t y$ - $O(2n)$ time, $O(1)$ space.

ii. **Outer Product** xy^t - $O(n^2)$ time, $O(n^2)$ space.

iii. **Outer Product Vector** $(xy^t)z = x(y^t z)$ - $O(3n)$ time, $O(n)$ space.

iv. **Dense Matrix Vector** Ax - $O(2mn)$ time, $O(m)$ space.

v. **Spar Matrix Vector** Ax - $O(m + n)$ time, $O(m + n)$ space.

vi. **Gram Schmidt** - $O(\frac{3}{2}mn^2)$ time.

vii. **Householder Vector** $Hv = (I - 2uu^*)v = v - 2uu^*v$ - $O(4m)$ time.

viii. **Householder** - $O(2mn^2 - \frac{2}{3}n^3)$ time.

ix. **Givens** - $O(mn^2)$ time.

2 QR Factorization

Given a matrix $A \in \mathbb{R}^{m \times n}$, we can write $A = QR$, where Q is orthonormal and R is upper triangular. This matrix factorization exists **for all** matrices.

2.1 Reduced QR

If $A \in \mathbb{R}^{m \times n}$, then $A = QR$ produces $Q \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{n \times n}$.

The typical GS orthogonalization produces this.

2.2 Full QR

If $A \in \mathbb{R}^{m \times n}$, then $A = QR$ produces $Q \in \mathbb{R}^{m \times m}$ and $R \in \mathbb{R}^{m \times n}$.

A typical GS factorization loops over the columns of A and orthogonalizes that column with respect to the previous columns of A. But if $m > n$, there are only n columns of A and so there are $m - n$ more orthogonal vectors that we need to form a basis for $\mathbb{R}^{m \times m}$.

This means we need $m - n$ more linearly independent vectors. Well we can just pick random vectors to orthogonalize because the probability of picking a vector that aligns exactly with a previous one (linearly dependent vector) is 0.

Another option is just to use Householder QR, or Givens QR.

2.3 Gram Schmidt

Given a matrix $A \in \mathbb{R}^{m \times n}$, A full rank (why? explained later), we want to form an orthogonal basis for the range of A.

Pick the first column of A, called a_1. We want an orthogonal basis for span of a_1, well we can just pick $v_1 = a_1$.

Now we are on the second column of A, called a_2. We want to now find an orthogonal vector to v_1. Well we can just find the projection onto v_1, defined as $v_1 v_1^T a_2$, and then subtract this bit off off v_1. So $v_2 = (I - v_1 v_1^T) a_2$.

We are now on the third column of A, called a_3. We want to find an orthogonal vector to $v_1,\ v_2$. Well we can find this by finding the component of a_3 that lives in $span\{v_1,\ v_2\}$, then subtracting that component from a_3. So $v_3 = (I - v_1 v_1^T - v_2 v_2^T) a_3$.

We continue until we have gone through every column, now we have an orthogonal basis for $Range(A)$, but this is not orthonormal. We can simply normalize each column $q_i = \frac{v_i}{|v_i|}$.

Now we have formed our matrix Q, and the R follows. A column of R, say r_j, tells us the linear combination of Q that we need to form the corresponding column a_j. By construction, R is upper triangular.

Why does A have to be full rank? If the columns of A are not linearly independent, then when we try to find a orthogonal vector, we will get a $v_i = 0$, and get NaNs in our answer.

2.4 Modified Gram Schmidt

In CGS, we use $v_i = a_i - \sum_{j=1}^{i-1} q_j q_j^* a_i$, but if the columns of A are almost linearly dependent, the inner product and subtraction operations will cause large numerical instabilities, and cause $q_i \cdot q_j \neq 0$.

Instead, we will initialize $v_i = a_i$, but then for every iteration, we do $v_i = v_i - q_j q_j^* v_i \ \forall j < i$. This makes it so that even though we have some instabilities in R, we focus on the orthogonality of Q, and we can bound $|Q^*Q - I| = O(\kappa(A)\epsilon_m)$.