LECTURE 16

CGS, MGS, Householder triangularization

end class early for alphabetizing situation

Q. Write down definitions for the following

① Orthonormal basis $\{b_1, \dots, b_n\}$

② Orthogonal matrix Q

③ Orthogonal projector P

eg. " P is an orthogonal projector iff ________ "

④ Normalization

Recall we described Gauss Elim $\longleftrightarrow$ LU decomposition.

Goal: QR decomp as result of same algorithm.

Historical / intellectual steps:

algs for computing QR.

① Classical Gram-Schmidt (CGS)

② Modified Gram-Schmidt (MGS) - better (why?)

③ Householder triangularization (next time)

- Understand how these algorithms work
- the drawbacks / issues!
- How to implement most efficiently

CGS via orthogonal projectors

$P^2 = P$, $\text{range}(P) \perp \text{null}(P)$

Setting $A \in \mathbb{R}^{m \times n}$, $n \le m$, full rank

$\Rightarrow \{a_{\cdot 1}, \dots, a_{\cdot n}\} = \{\vec{a}_1, \dots, \vec{a}_n\}$ are linearly indep

want: ON basis for $\mathbb{R}^n$ from A.

P_j = orth proj to orth. comp. $\langle q_1, \dots, q_{j-1} \rangle^\perp$

Algorithm

for $j = 1:n$
- $v_j = a_j$
- for $i = 1 : j-1$
  - $r_{ij} = q_i^T a_j$
  - $v_j = v_j - r_{ij} q_i$
- $r_{jj} = \| v_j \|$
- $q_j = v_j / r_{jj}$

$j=1$

$v_1 = a_1$

$r_{11} = \|a_1\|$

$q_1 = \dfrac{a_1}{\|a_1\|}$

$j=2$

$v_2 = a_2$

for $i \in \{1\}$:

$r_{12} = q_1^T a_2$

$v_2 = v_2 - r_{12} \cdot q_1$

$\Big\}$ $P_2 a_2 = v_2$. store

$r_{22} = \|v_2\|$

$q_2 = v_2 / r_2 = \dfrac{v_2}{\|v_2\|}$

- $P_1 = I$
- P_j = orth. proj onto orth complement $\langle q_1, \dots, q_{j-1} \rangle^\perp$

each $q_j \perp \underbrace{\langle q_1, \dots, q_{j-1} \rangle}_{\hat{Q}_{j-1}}$

$P_j = I - \hat{Q}_{j-1} \hat{Q}_{j-1}^T$ projects onto orth complement $\langle q_1, \dots, q_{j-1} \rangle^\perp$

Rmk. Note the r_{ij} are computed columnwise.

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}$$
$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ & r_{22} & r_{23} & r_{24} \\ & & r_{33} & r_{34} \\ & & & r_{44} \end{bmatrix} = R.$$

Issues
- lots of calculations in inner for loop; numerical errors compound...
- rank $P_j = m - (j-1)$. Projecting onto small dimension is dangerous - matters a lot if you have error in that direction.

* Unstable for high dimensions!

(HW06 – you'll do an example)

MGS

Idea: Instead of P_j = proj onto orth complement at each step, we use a composition of all the rank $m-1$ projectors

$$P_j = (P_{\perp q_{j-1}})(P_{\perp q_{j-2}})\cdots(P_{\perp q_1})$$

$$P_{\perp q_i} = I - q_i q_i^T$$

(and $P_1 = I$ still)

Observe that if there were no error, mathematically we're doing exactly the same process; orth. proj onto the same subspace.

But the seqn of arith operations differs:

Algorithm

for $i = 1:n$

$\quad v_i = a_i$

for $i = 1:n$

$\quad r_{ii} = \|v_i\|$

$\quad q_i = v_i / r_{ii}$ normalize first.

$\quad$ for $j = i+1:n$ $\{q_1, \dots, q_i\}$ already done.

$\quad\quad r_{ij} = q_i^T v_j$

$\quad\quad v_j = v_j - r_{ij} q_i$

remove components in the $q_1, \dots, q_i$ directions.

for all remaining vectors (earlier than for CGS).

Rmk. Order of computation of the r_{ij}:

1. $r_{11} \quad r_{12} \quad r_{13} \quad r_{14} \longrightarrow$
2. $r_{22} \quad r_{23} \quad r_{24} \longrightarrow$
3. $r_{33} \quad r_{34} \longrightarrow$
4. $r_{44} \longrightarrow$