Lecture 06
Gram-Schmidt Orthogonalization; Householder Reflectors

Songting Luo

Department of Mathematics
Iowa State University

MATH 562 Numerical Analysis II

Outline

Outline

1. Gram-Schmidt Orthogonalization

2. Householder Reflectors

Gram-Schmidt Orthogonalization

- A method to construct QR factorization is to orthogonalize the column vectors of $\mathbf{A}$:
- Basic idea: Gram-Schmidt orthogonalization.
  - Take jth column $\mathbf{a}_j$ of $\mathbf{A}$, subtract its orthogonal projection to $\mathbf{q}_1, \dots, \mathbf{q}_{j-1}$, and normalize to obtain $\mathbf{q}_j$:

$$\mathbf{v}_j = \mathbf{a}_j - \sum_{i=1}^{j-1} \mathbf{q}_i^* \mathbf{a}_j \mathbf{q}_i, \ \mathbf{q}_j = \mathbf{v}_j / \|\mathbf{v}_j\|.$$

- Orthogonal vectors produced by Gram-Schmidt can be written in terms of projectors

$$\mathbf{q}_j = \frac{\mathbf{P}_j \mathbf{a}_j}{\|\mathbf{P}_j \mathbf{a}_j\|}$$

  where $\mathbf{P}_j = \mathbf{I} - \hat{\mathbf{Q}}_{j-1} \hat{\mathbf{Q}}_{j-1}^*$ with $\hat{\mathbf{Q}}_{j-1} = [\mathbf{q}_1, \dots, \mathbf{q}_{j-1}]$.
- $\mathbf{P}_j$ projects orthogonally onto space orthogonal to $\langle \mathbf{q}_1, \dots, \mathbf{q}_{j-1} \rangle$ and rank of $\mathbf{P}_j$ is $m - j - 1$.

Algorithm of Gram-Schmidt Orthogonalization

Classical Gram-Schmidt method

for $j = 1$ to n

    $\mathbf{v}_j = \mathbf{a}_j$

    for $i = 1$ to $j - 1$

        $r_{ij} = \mathbf{q}_i^* \mathbf{a}_j$

        $\mathbf{v}_j = \mathbf{v}_j - r_{ij} \mathbf{q}_i$

    $r_{jj} = \|\mathbf{v}_j\|_2$

    $\mathbf{q}_j = \mathbf{v}_j / r_{jj}$

- Classical Gram-Schmidt (CGS) is **unstable**, which means that its solution is sensitive to perturbation

Alternative view to Gram-Schmidt Projection

- Orthogonal vectors produced by Gram-Schmidt can be written in terms of projectors

$$\mathbf{q}_j = \frac{\mathbf{P}_j \mathbf{a}_j}{\|\mathbf{P}_j \mathbf{a}_j\|}, \text{ where } \mathbf{P}_j = \mathbf{I} - \hat{\mathbf{Q}}_{j-1}\hat{\mathbf{Q}}_{j-1}^*, \ \hat{\mathbf{Q}}_{j-1} = [\mathbf{q}_1, \dots, \mathbf{q}_{j-1}]$$

- We may view $\mathbf{P}_j$ as product of a sequence of projections

$$\mathbf{P}_j = \mathbf{P}_{\perp \mathbf{q}_{j-1}} \mathbf{P}_{\perp \mathbf{q}_{j-2}} \cdots \mathbf{P}_{\perp \mathbf{q}_1}$$

  where $\mathbf{P}_{\perp \mathbf{q}} = \mathbf{I} - \mathbf{q}\mathbf{q}^*$.

- In stead of computing $\mathbf{v}_j = \mathbf{P}_j \mathbf{a}_j$, one could compute $\mathbf{v}_j = \mathbf{P}_{\perp \mathbf{q}_{j-1}} \mathbf{P}_{\perp \mathbf{q}_{j-2}} \cdots \mathbf{P}_{\perp \mathbf{q}_1} \mathbf{a}_j$, resulting in modified Gram-Schmidt algorithm

Modified Gram-Schmidt Algorithm

Modified Gram-Schmidt method

for $j = 1$ to n

$\quad \mathbf{v}_j = \mathbf{a}_j$

for $i = 1$ to n

$\quad r_{ii} = \|\mathbf{v}_i\|_2$

$\quad \mathbf{q}_i = \mathbf{v}_i / r_{ii}$

$\quad$ for $j = i + 1$ to n

$\qquad r_{ij} = \mathbf{q}_i^* \mathbf{v}_j$

$\qquad \mathbf{v}_j = \mathbf{v}_j - r_{ij} \mathbf{q}_i.$

CGS v.s. MGS

- Key difference between CGS and MGS is how r_{ij} is computed.
- CGS above is column-oriented (in the sense that **R** is computed column by column) and MGS above is row-oriented, but this is NOT the main difference between CGS and MGS. There are also column-oriented MGS and row-oriented CGS.
- MGS is numerically more stable than CGS (less sensitive to round-off errors)

Example: CGS v.s. MGS

- Consider matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}$$

  where ϵ is small such that $1 + \epsilon^2 = 1$ with round-off error

- For both CGS and MGS

  $\mathbf{v}_1 \leftarrow (1, \epsilon, 0, 0)^T,\ r_{11} = \sqrt{1+\epsilon^2} \approx 1,\ \mathbf{q}_1 = \mathbf{v}_1 / r_{11} = (1, \epsilon, 0, 0)^T$
  $\mathbf{v}_2 \leftarrow (1, 0, \epsilon, 0)^T,\ r_{12} = \mathbf{q}_1^T \mathbf{a}_2 = 1(\text{ or } = \mathbf{q}_1^T \mathbf{v}_2 = 1)$
  $\mathbf{v}_2 \leftarrow \mathbf{v}_2 - r_{12}\mathbf{q}_1 = (0, -\epsilon, \epsilon, 0)^T,$
  $r_{22} = \sqrt{2}\epsilon,\ \mathbf{q}_2 = (0, -1, 1, 0)/\sqrt{2},$
  $\mathbf{v}_3 \leftarrow (1, 0, 0, \epsilon)^T,\ r_{13} = \mathbf{q}_1^T \mathbf{a}_3 = 1(\text{ or } = \mathbf{q}_1^T \mathbf{v}_3 = 1)$
  $\mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{13}\mathbf{q}_1 = (0, -\epsilon, 0, \epsilon)^T$

Example: CGS v.s. MGS

- For CGS

$$r_{23} = \mathbf{q}_2^T \mathbf{a}_3 = 0,\ \mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{23}\mathbf{q}_2 = (0, -\epsilon, 0, \epsilon)^T$$

$$r_{33} = \sqrt{2}\epsilon, \mathbf{q}_3 = \mathbf{v}_3/r_{33} = (0, -1, 0, 1)^T/\sqrt{2}$$

  - Note that $\mathbf{q}_2^T \mathbf{q}_3 = (0, -1, 1, 0)(0, -1, 0, 1)^T/2 = 1/2$.

- For MGS

$$r_{23} = \mathbf{q}_2^T \mathbf{v}_3 = \epsilon/\sqrt{2}, \mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{23}\mathbf{q}_2 = (0, -\epsilon/2, -\epsilon/2, \epsilon)^T$$

$$r_{33} = \sqrt{6}\epsilon/2, \mathbf{q}_3 = \mathbf{v}_3/r_{33} = (0, -1, -1, 2)^T/\sqrt{6}$$

  - Note that $\mathbf{q}_2^T \mathbf{q}_3 = (0, -1, 1, 0)(0, -1, -1, 2)^T/\sqrt{12} = 0$.

Operation Count

- It is important to assess the efficiency of algorithms. But how?
  - We could implement different algorithms and do head-to-head comparison, but implementation details might affect true performance
  - We could estimate cost of all operations, but it is very tedious
  - Relatively simple and effective approach is to estimate amount of floating-point operations, or "flop", and focus on asymptotic analysis as sizes of matrices approach infinity
- Count each operation $+, -, *, /$, and $\sqrt{}$ as one flop, and make no distinction of real and complex numbers

Theorem

CGS and MGS require $\sim 2mn^2$ *flops to compute a QR factorization of an* $m \times n$ *matrix.*

Outline

1. Gram-Schmidt Orthogonalization

2. Householder Reflectors

Gram-Schmidt as Triangular Orthogonalization

- Every step of Gram-Schmidt can be viewed as multiplication with triangular matrix. For example, at first step:

$$[\mathbf{v}_1|\mathbf{v}_2|\cdots|\mathbf{v}_n]\underbrace{\begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \frac{-r_{13}}{r_{11}} & \cdots \\ & 1 & & \\ & & 1 & \\ & & & \ddots \end{bmatrix}}_{\mathbf{R}_1} = [\mathbf{q}_1|\mathbf{v}_2^{(2)}|\cdots|\mathbf{v}_n^{(2)}]$$

- Gram-Schmidt therefore multiplies triangular matrices to orthogonalize column vectors, and in turns can be viewed as triangular orthogonalization $\mathbf{A}\underbrace{\mathbf{R}_1\cdots\mathbf{R}_n}_{\hat{\mathbf{R}}^{-1}} = \hat{\mathbf{Q}}$ where $\mathbf{R}_i$ is triangular matrix
- A "dual" approach would be orthogonal triangularization, i.e., multiply $\mathbf{A}$ by unitary matrices to make it triangular matrix

Householder Triangularization

- Method introduced by Alston Scott Householder in 1958
- It multiplies unitary matrices to make column triangular, e.g.,

$$\underset{\mathbf{A}}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}} \xrightarrow{\mathbf{Q}_1} \underset{\mathbf{Q}_1\mathbf{A}}{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}} \xrightarrow{\mathbf{Q}_2} \underset{\mathbf{Q}_2\mathbf{Q}_1\mathbf{A}}{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix}} \xrightarrow{\mathbf{Q}_3} \underset{\mathbf{Q}_3\mathbf{Q}_2\mathbf{Q}_1\mathbf{A}}{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}$$

- After n steps, we get a product of unitary matrices,

$$\underbrace{\mathbf{Q}_n \cdots \mathbf{Q}_2\mathbf{Q}_1}_{\mathbf{Q}^*}\mathbf{A} = \mathbf{R}$$

  and in turn we get full QR factorization $\mathbf{A} = \mathbf{QR}$.
- $\mathbf{Q}_k$ introduces zeros below diagonal of kth column while preserving zeros below diagonal in preceding columns
- The key question is how to find $\mathbf{Q}_k$?

Householder Reflectors

- First consider $\mathbf{Q}_1$: $\mathbf{Q}_1\mathbf{a}_1 = \|\mathbf{a}_1\|\mathbf{e}_1$, where $\mathbf{e}_1 = (1, 0, \cdots, 0)^T$. Why the length is $\|\mathbf{a}_1\|$?
- $\mathbf{Q}_1$ reflects $\mathbf{a}_1$ across hyperplane H orthogonal to $\mathbf{v} = \|\mathbf{a}_1\|\mathbf{e}_1 - \mathbf{a}_1$, and there fore

$$\mathbf{Q}_1 = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^*}{\mathbf{v}^*\mathbf{v}}$$

- More generally,

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix}$$

  where $\mathbf{I}$ is $(k-1) \times (k-1)$ and $\mathbf{F}$ is $(m-k+1) \times (m-k+1)$ such that $\mathbf{F}\mathbf{x} = \|\mathbf{x}\|_2\mathbf{e}_1$, where $\mathbf{x}$ is $(a_{k,k}, a_{k+1,k}, \cdots, a_{m,k})^T$.
- What is $\mathbf{F}$? It has similar form as $\mathbf{Q}_1$ with $\mathbf{v} = \|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$.

Choice of Reflectors

- We could choose $\mathbf{F}$ such that $\mathbf{Fx} = -\|\mathbf{x}\|\mathbf{e}_1$ instead of $\mathbf{Fx} = \|\mathbf{x}\|\mathbf{e}_1$ or more generally, $\mathbf{Fx} = z\|\mathbf{x}\|\mathbf{e}_1$ with $|z| = 1$ for $z \in \mathbb{C}$.
- This leads to an infinite number of possible QR factorizations of $\mathbf{A}$
- If we require $z \in \mathbb{R}$, we still have two choices
- Numerically, it is undesirable for $\mathbf{v}^*\mathbf{v}$ to be close to zero for $\mathbf{v} = z\|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$, and $\|\mathbf{v}\|$ is larger if $z = -sign(x_1)$
- Therefore, $\mathbf{v} = -sign(x_1)\|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$. Since $\mathbf{I} - 2\frac{\mathbf{vv}^*}{\mathbf{v}^*\mathbf{v}}$ is independent of sign, clear out the factor -1 and obtain $\mathbf{v} = sign(x_1)\|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$.
- For completeness, if $x_1 = 0$, set z to 1 (instead of 0).