SI231b: Matrix Computations

Lecture 10: Orthogonal Projection Computations

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Oct. 12, 2022

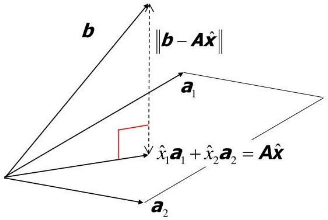
Recap

Overdetermined System: $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m > n)$, the least square (LS) solution $\mathbf{x}_{LS}$,

$$\mathbf{x}_{LS} = \arg\min \|\mathbf{b} - \mathbf{Ax}\|_2^2,$$

where $\|\cdot\|_2$ represents the vector 2-norm and $\mathbf{A}$ is full rank.

1. find $\tilde{\mathbf{b}} \in \mathcal{R}(\mathbf{A})$ such that $\|\mathbf{b} - \tilde{\mathbf{b}}\|_2$ is minimized
2. solve $\mathbf{Ax}_{LS} = \tilde{\mathbf{b}}$ to obtain $\mathbf{x}_{LS}$



Key: orthogonal projection on $\mathcal{R}(\mathbf{A})$

Previous analysis show that $\mathbf{P} \in \mathbb{R}^{m \times m}$ seperates $\mathbb{R}^m$ into two subspaces

- $\mathcal{R}(\mathbf{P})$
- $\mathcal{N}(\mathbf{P})$

and

$$\mathbb{R}^m = \mathcal{R}(\mathbf{P}) \oplus \mathcal{N}(\mathbf{P}) \quad \text{can you prove this?}$$

$\mathbf{P}$ projects $\mathbb{R}^m$ onto $\mathcal{R}(\mathbf{P})$ along $\mathcal{N}(\mathbf{P})$.

Theorem

A projector $\mathbf{P}$ is orthogonal if and only if $\mathbf{P} = \mathbf{P}^T$.

Projection with Orthonormal Basis

When $\{q_1, \ q_2, \ \cdots, \ q_n\}$ form an orthonormal basis of $\mathcal{R}(\mathbf{P})$, then the orthogonal projector is given by

$$\mathbf{P} = \mathbf{Q}\mathbf{Q}^T,$$

where $\mathbf{Q} = [q_1, \ q_2, \ \cdots, \ q_n]$

Can you explain why?

Projection with Arbitrary Basis

When $\{a_1, \ a_2, \ \cdots, \ a_n\}$ form a basis of $\mathcal{R}(\mathbf{P})$, then the orthogonal projector is given by

$$\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T,$$

where $\mathbf{A} = [a_1, \ a_2, \ \cdots, \ a_n]$

How to obtain?

Given a basis $\{a_1,\ a_2,\ \cdots,\ a_n\}$ of a subspace $\mathcal{S}$, how to compute its orthogonal/orthonormal basis $\{q_1,\ q_2,\ \cdots,\ q_n\}$?

Key: through iterative process and using the fact that

- $\text{span}\{a_1, a_2, \cdots, a_k\} = \text{span}\{q_1, q_2, \cdots, q_k\},\ k = 1, 2, \cdots, n$
- $\text{span}\{a_1, a_2, \cdots, a_k\} \subset \text{span}\{a_1, a_2, \cdots, a_k, a_{k+1}\}$

Gram-Schmidt orthogonalization.

Key: orthogonal projection of vector **a** onto vector **b**

$$\text{proj}_{\mathbf{b}}(\mathbf{a}) = \frac{<\mathbf{a}, \mathbf{b}>}{<\mathbf{b}, \mathbf{b}>}\mathbf{b},$$

where $<>$ represents the inner product of two vectors.

Gram-Schmidt Orthogonalization

How to compute the orthonormal basis?

Orthogonal projection of vector $\mathbf{a}$ onto vector $\mathbf{b}$

$$\mathrm{proj}_{\mathbf{b}}(\mathbf{a}) = \frac{< \mathbf{a}, \mathbf{b} >}{< \mathbf{b}, \mathbf{b} >}\mathbf{b},$$

where $<>$ represents the inner product of two vectors.

$$\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}$$

$$\tilde{\mathbf{q}}_2 = \mathbf{a}_2 - (\mathbf{q}_1^T\mathbf{a}_2)\mathbf{q}_1$$

$$\mathbf{q}_2 = \frac{\tilde{\mathbf{q}}_2}{\|\tilde{\mathbf{q}}_2\|}$$

$$\vdots$$

$$\tilde{\mathbf{q}}_k = \mathbf{a}_k - (\mathbf{q}_1^T\mathbf{a}_k)\mathbf{q}_1 - (\mathbf{q}_2^T\mathbf{a}_k)\mathbf{q}_2 - \cdots - (\mathbf{q}_{k-1}^T\mathbf{a}_k)\mathbf{q}_{k-1}$$

$$\mathbf{q}_k = \frac{\tilde{\mathbf{q}}_k}{\|\tilde{\mathbf{q}}_k\|}$$

Can you also explain in the context of projection onto subspaces?

Algorithm: Gram-Schmidt Orthogonalization (numerically unstable)

input: a collection of linearly independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$

$\tilde{\mathbf{q}}_1 = \mathbf{a}_1$, $\mathbf{q}_1 = \tilde{\mathbf{q}}_1 / \|\tilde{\mathbf{q}}_1\|_2$

for $i = 2, \ldots, n$

$\quad \tilde{\mathbf{q}}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} (\mathbf{q}_j^T \mathbf{a}_i) \mathbf{q}_j$

$\quad \mathbf{q}_i = \tilde{\mathbf{q}}_i / \|\tilde{\mathbf{q}}_i\|_2$

end

output: $\mathbf{q}_1, \ldots, \mathbf{q}_n$

The (classic) Gram-Schmidt (CGS)

- gives orthogonal $\tilde{\mathbf{q}}_i$ in exact arithmetic
- is numerical unstable due to round off error accumulation on modern computers (returns non-orthogonal $\tilde{\mathbf{q}}_i$)

Modified Gram-Schmidt (MGS)

Instead of computing $\tilde{\mathbf{q}}_k = \mathbf{a}_k - (\mathbf{q}_1^T \mathbf{a}_k)\mathbf{q}_1 - (\mathbf{q}_2^T \mathbf{a}_k)\mathbf{q}_2 - \cdots - (\mathbf{q}_{k-1}^T \mathbf{a}_k)\mathbf{q}_{k-1}$, but

$$
\begin{aligned}
\tilde{\mathbf{q}}_k^{(1)} &= \mathbf{a}_k - (\mathbf{q}_1^T \mathbf{a}_k)\mathbf{q}_1 \\
\tilde{\mathbf{q}}_k^{(2)} &= \tilde{\mathbf{q}}_k^{(1)} - (\mathbf{q}_2^T \tilde{\mathbf{q}}_k^{(1)})\mathbf{q}_2 \\
&\vdots \\
\tilde{\mathbf{q}}_k^{(j)} &= \tilde{\mathbf{q}}_k^{(j-1)} - (\mathbf{q}_j^T \tilde{\mathbf{q}}_k^{(j-1)})\mathbf{q}_j \\
&\vdots
\end{aligned}
$$

Both CGS and MGS take $\mathcal{O}(2mn^2)$ flops

Given $\mathbf{a}_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$, $\mathbf{a}_2 = \begin{bmatrix} 1 & 0 & \epsilon & 0 \end{bmatrix}^T$, $\mathbf{a}_3 = \begin{bmatrix} 1 & 0 & 0 & \epsilon \end{bmatrix}^T$, compare classical and modified Gram-Schmidt for

$$\mathcal{V} = \text{span}\left\{\mathbf{a}_1,\ \mathbf{a}_2,\ \mathbf{a}_3\right\}$$

where the approximation $1 + \epsilon^2 = 1$ can be made.

Classical Gram-Schmidt

- $\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|_2} = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$
- $\mathbf{q}_2 = \mathbf{a}_2 - \mathbf{q}_1^T \mathbf{a}_2 \mathbf{q}_1 = \begin{bmatrix} 0 & -\epsilon & \epsilon & 0 \end{bmatrix}^T$

  $\mathbf{q}_2 = \frac{\mathbf{q}_2}{\|\mathbf{q}_2\|_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}^T$
- $\mathbf{q}_3 = \mathbf{a}_3 - \mathbf{q}_1^T \mathbf{a}_3 \mathbf{q}_1 - \mathbf{q}_2^T \mathbf{a}_3 \mathbf{q}_2 = \begin{bmatrix} 0 & -\epsilon & 0 & \epsilon \end{bmatrix}^T$

  $\mathbf{q}_3 = \frac{\mathbf{q}_3}{\|\mathbf{q}_3\|_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}^T$

Orthogonality is lost

Modified Gram-Schmidt

- $\tilde{\mathbf{q}}_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$

  $\mathbf{q}_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$

- $\tilde{\mathbf{q}}_2 = \mathbf{a}_2 - \tilde{\mathbf{q}}_1^T \mathbf{a}_2 \tilde{\mathbf{q}}_1 = \begin{bmatrix} 0 & -\epsilon & \epsilon & 0 \end{bmatrix}^T$

  $\mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}^T$

- $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1^T \mathbf{a}_3 \mathbf{q}_1 = \begin{bmatrix} 0 & -\epsilon & 0 & \epsilon \end{bmatrix}^T$

  $\tilde{\mathbf{q}}_3 = \tilde{\mathbf{q}}_3 - \mathbf{q}_2^T \tilde{\mathbf{q}}_3 \mathbf{q}_2 = \begin{bmatrix} 0 & -\frac{\epsilon}{2} & -\frac{\epsilon}{2} & \epsilon \end{bmatrix}^T$

  $\mathbf{q}_3 = \frac{1}{\sqrt{6}} \begin{bmatrix} 0 & -1 & -1 & 2 \end{bmatrix}^T$

Orthogonality is preserved

Reduced QR Factorization

For a full rank matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m > n)$, the Gram-Schmidt procedure gives

$$\underbrace{\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \vdots \\ & & \ddots & \\ & & & r_{nn} \end{bmatrix}}_{\mathbf{R}}$$

with $r_{kk} \neq 0$. This is called the *reduced QR factorization* of $\mathbf{A}$.

Full QR Factorization

Extending the reduced QR factorization by adding $m - n$ columns to $\mathbf{Q}$ so that

$$\tilde{\mathbf{Q}} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n & q_{n+1} & \cdots & q_m \end{bmatrix}$$

is an orthogonal matrix ($\tilde{\mathbf{Q}} \in \mathbb{R}^{m \times m}$)

- orthogonal matrix: a square matrix with orthonormal columns, i.e., $\tilde{\mathbf{Q}}^T \tilde{\mathbf{Q}} = \mathbf{I}_m$

Then $\mathbf{A} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$ with $\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix}$

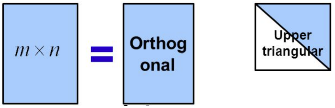


Figure 1: Reduced QR Factorization

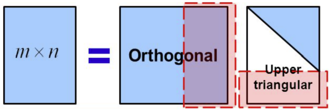


Figure 2: Full QR Factorization

One of the Top 10 Algorithms in the 20th Century[1]

Given a rectangular matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{A}$ can be factorized into the form

$$\mathbf{A} = \mathbf{QR}$$

where

- $\mathbf{Q} \in \mathbb{R}^{m\times m}$ is an orthogonal matrix
- $\mathbf{R} \in \mathbb{R}^{m\times n}$ is upper-triangular

Reduced QR Factorization

For $m > n$, the reduced QR factorization given by

- $\mathbf{Q} \in \mathbb{R}^{m\times n}$ has orthonormal columns
- $\mathbf{R} \in \mathbb{R}^{n\times n}$ is upper-triangular
- also called 'economic' QR factorization in some cases

[1] https://doi.ieeecomputersociety.org/10.1109/MCISE.2000.814652

Readings

You are supposed to read

- Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*, SIAM, 1997.

  Lecture 6, 8, 11