SI231b: Matrix Computations

Lecture 10: Orthogonal Projection Computations

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

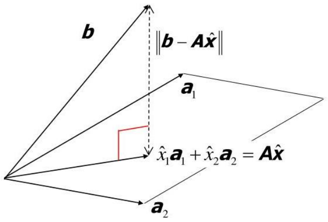
Oct. 19, 2021

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

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

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

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



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

Projection onto subspaces

Suppose $\mathcal{V} = \mathcal{U} \oplus \mathcal{W}$, then there is a projector P such that $\mathcal{R}(\mathrm{P}) = \mathcal{U}$ and $\mathcal{N}(\mathrm{P}) = \mathcal{W}$, we say that P is a projector onto $\mathcal{U}$ along $\mathcal{W}$.

Orthogonal projector

An orthogonal projector P is the one that projects onto a subspace $\mathcal{U}$ along a subspace $\mathcal{W}$ when $\mathcal{U}$ and $\mathcal{W}$ are orthogonal.

Warning: orthogonal projectors are not orthogonal matrices.

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

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

and

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

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

Theorem

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

Projection with Orthonormal Basis

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

$$P = QQ^T,$$

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

Can you explain why?

Projection with Arbitrary Basis

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

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

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

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

$$\mathrm{proj_b(a)} = \frac{<\mathrm{a}, \mathrm{b}>}{<\mathrm{b}, \mathrm{b}>}\mathrm{b},$$

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

How to compute the orthonormal basis?

Orthogonal projection of vector a onto vector b

$$\mathrm{proj_b(a)} = \frac{<\mathrm{a}, \mathrm{b}>}{<\mathrm{b}, \mathrm{b}>}\mathrm{b},$$

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

$$\begin{aligned}
\mathrm{q}_1 &= \frac{\mathrm{a}_1}{\|\mathrm{a}_1\|} \\
\tilde{\mathrm{q}}_2 &= \mathrm{a}_2 - (\mathrm{q}_1^T \mathrm{a}_2)\mathrm{q}_1 \\
\mathrm{q}_2 &= \frac{\tilde{\mathrm{q}}_2}{\|\tilde{\mathrm{q}}_2\|} \\
&\vdots \\
\tilde{\mathrm{q}}_k &= \mathrm{a}_k - (\mathrm{q}_1^T \mathrm{a}_k)\mathrm{q}_1 - (\mathrm{q}_2^T \mathrm{a}_k)\mathrm{q}_2 - \cdots - (\mathrm{q}_{k-1}^T \mathrm{a}_k)\mathrm{q}_{k-1} \\
\mathrm{q}_k &= \frac{\tilde{\mathrm{q}}_k}{\|\tilde{\mathrm{q}}_k\|}
\end{aligned}$$

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

Gram-Schmidt Orthogonalization

Algorithm: Gram-Schmidt Orthogonalization (numerically unstable)

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

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

for $i = 2, \ldots, n$

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

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

end

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

The (classic) Gram-Schmidt (CGS)

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

Modified Gram-Schmidt (MGS)

Instead of computing $\tilde{q}_k = a_k - (q_1^T a_k)q_1 - (q_2^T a_k)q_2 - \cdots - (q_{k-1}^T a_k)q_{k-1}$, but

$$\begin{aligned}
\tilde{q}_k^{(1)} &= a_k - (q_1^T a_k)q_1 \\
\tilde{q}_k^{(2)} &= \tilde{q}_k^{(1)} - (q_2^T \tilde{q}_k^{(1)})q_2 \\
&\vdots \\
\tilde{q}_k^{(j)} &= \tilde{q}_k^{(j-1)} - (q_j^T \tilde{q}_k^{(j-1)})q_j \\
&\vdots
\end{aligned}$$

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

Given $a_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$, $a_2 = \begin{bmatrix} 1 & 0 & \epsilon & 0 \end{bmatrix}^T$, $a_3 = \begin{bmatrix} 1 & 0 & 0 & \epsilon \end{bmatrix}^T$, compare classical and modified Gram-Schmidt for

$$\mathcal{V} = \text{span}\left\{a_1,\ a_2,\ a_3\right\}$$

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

Classical Gram-Schmidt

- $q_1 = \frac{a_1}{\|a_1\|_2} = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$
- $q_2 = a_2 - q_1^T a_2 q_1 = \begin{bmatrix} 0 & -\epsilon & \epsilon & 0 \end{bmatrix}^T$

  $q_2 = \frac{q_2}{\|q_2\|_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}^T$
- $q_3 = a_3 - q_1^T a_3 q_1 - q_2^T a_3 q_2 = \begin{bmatrix} 0 & -\epsilon & 0 & \epsilon \end{bmatrix}^T$

  $q_3 = \frac{q_3}{\|q_3\|_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}^T$

Orthogonality is lost

Modified Gram-Schmidt

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

  $q_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$

- $\tilde{q}_2 = a_2 - \tilde{q}_1^T a_2 \tilde{q}_1 = \begin{bmatrix} 0 & -\epsilon & \epsilon & 0 \end{bmatrix}^T$

  $q_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}^T$

- $\tilde{q}_3 = a_3 - q_1^T a_3 q_1 = \begin{bmatrix} 0 & -\epsilon & 0 & \epsilon \end{bmatrix}^T$

  $\tilde{q}_3 = \tilde{q}_3 - q_2^T \tilde{q}_3 q_2 = \begin{bmatrix} 0 & -\frac{\epsilon}{2} & -\frac{\epsilon}{2} & \epsilon \end{bmatrix}^T$

  $q_3 = \frac{1}{\sqrt{6}} \begin{bmatrix} 0 & -1 & -1 & 2 \end{bmatrix}^T$

Orthogonality is preserved

Reduced QR Factorization

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

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

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

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

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

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

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

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

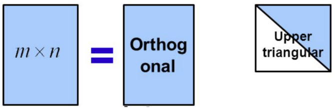


Figure 1: Reduced QR Factorization

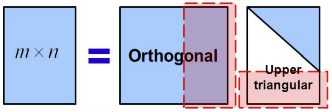


Figure 2: Full QR Factorization

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

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

$$A = QR$$

where

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

Reduced QR Factorization

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

- $Q \in \mathbb{R}^{m\times n}$ has orthonormal columns
- $R \in \mathbb{R}^{n\times n}$ 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