Linear Algebra

Orthogonal Matrices and QR Decomposition: Part 1

Automotive Intelligence Lab.

HANYANG UNIVERSITY

AI LAB Automotive Intelligence

Contents

Orthogonal Matrices

Introduction of Orthogonal Matrices

Important and special matrices for several decompositions

- QR decomposition
- Eigen decomposition
- Singular value decomposition

Letter Q

- Often used to indicate orthogonal matrices.

Mathematical Expression of Orthogonal Matrices

Two properties of orthogonal matrices

- Orthogonal columns
  - All columns are $\boxed{\text{pair-wise}}$ orthogonal.
- Unit-norm columns
  - The norm (geometric length) of each column is exactly $\boxed{1}$.

$$\begin{bmatrix} | & | & | & & | \\ q_1 & q_2 & q_3 & \cdots & q_n \\ | & | & | & & | \end{bmatrix}$$

$$Q = \begin{bmatrix} | & | & | \\ q_1 & q_2 & q_3 \\ | & | & | \end{bmatrix} \qquad Q^T \cdot Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Translate those two properties into a mathematical expression.

- $\langle a, b \rangle$: alternative notation for the dot product
- q_i: i^{th} column of matrix

$$\langle q_i, q_j \rangle = \begin{cases} 0, & if \;\; i \neq j \\ 1, & if \;\; i = j \end{cases}$$

Mathematical expression of orthogonal matrices

- Dot product of a column with itself is 1.
- Dot product of a column with any other column is 0.

Characteristic of Orthogonal Matrices

Definition of Matrix multiplication

- Dot products between all rows of the left matrix with all columns of the right matrix

Q^T is a matrix that multiplies Q to produce the identity matrix.

- Exact same definition as the **matrix** inverse.
- Inverse of an orthogonal matrix is its transpose..
  - Matrix inverse: tedious and prone to numerical in accuracies.
  - Matrix transpose: fast and accurate

if) Q orthogonal

$Q^{-1} = Q^T$

Identity matrix is an example of an orthogonal matrix.

$$Q^T Q = I$$

Characteristic of an orthogonal matrix

Example of Orthogonal Matrices

Practice in MATLAB with below matrices

- Does each column have unit length?
  - $\boxed{\text{Yes}}$.
- Does each column orthogonal to other columns?
  - $\boxed{\text{Yes}}$.
- Compute QQ^T.
  - Is that still the identity matrix? Try it to find out!

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \qquad \frac{1}{3}\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & -2 \\ -2 & 2 & -1 \end{bmatrix}$$

Example of an orthogonal matrices

```
% Clear workspace, command
window, and close all
figures
clc; clear; close all;

% Matrces Q1 and Q2
Q1 = [1 -1; 1 1]/sqrt(2);
Q2 = [1 2 2; 2 1 -2; -2 2 -
1]/3;

% Orthogonal matrices
Q1TQ1 = Q1' * Q1;
Q2TQ2 = Q2' * Q2;

% Display results
disp("Q1T * Q1");
disp(Q1TQ1);
disp("Q2T * Q2");
disp(Q2TQ2);
```

MATLAB code to compute QQ^T

Gram-Schmidt

Process of Gram Schmidt

$Q^T M.$

$M \to \underline{QR}$

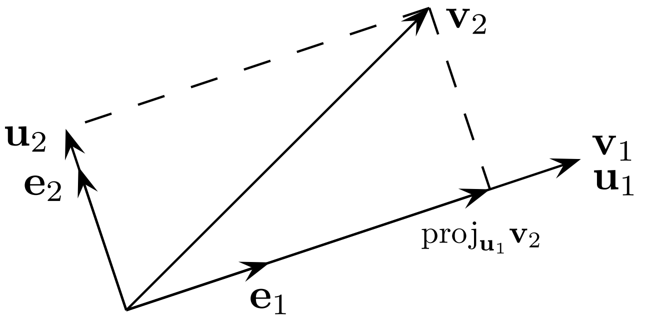
orthogonal.

- **Way of making two or more vectors perpendicular to each other**

- **Technical definition of Gram Schmidt**
  - Method of constructing an $\boxed{\text{orthogonal.}}$ basis
    - From a set of vectors in an **inner space**.
    - Most commonly Euclidean space $\boldsymbol{R^n}$ equipped with standard inner product.

- **Takes a finite, linearly independent set of vectors $\boldsymbol{S} = \{\boldsymbol{v_q}, \dots, \boldsymbol{v_k}\}$.**
  - Generate an orthogonal set $\boldsymbol{S'} = \{\boldsymbol{u_1}, \dots, \boldsymbol{u_k}\}$.
    - Spans the same $k - dimensional$ subspace of R^n as $\boldsymbol{S}$.

- **Application to column vectors of full column rank matrix**
  - Yields the $\boldsymbol{QR}$ decomposition.
    - Decomposed into orthogonal and a triangular matrix.
      - We will study QR decomposition in next section!



Basic principles of the Gram-Schmidt process

Reference: https://en.wikipedia.org/wiki/Gram%E2%80%93Schmidt_process

Vector Projection

- **Vector projection of a vector $\boldsymbol{v}$ on a nonzero vector $\boldsymbol{u}$.**
  - $< \boldsymbol{v}, \boldsymbol{u} >$: inner product of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.
  - $proj_{\boldsymbol{u}}(\boldsymbol{v})$: orthogonal projection of $\boldsymbol{v}$ onto the line spanned by $\boldsymbol{u}$.
  - If $\boldsymbol{u}$ is zero vector,
    - $proj_u \boldsymbol{v}$ is defined as zero vector.

$$\boldsymbol{proj_u(v)} = \frac{< \boldsymbol{v}, \boldsymbol{u} >}{< \boldsymbol{u}, \boldsymbol{u} >}\boldsymbol{u}$$

Vector projection

Expression of Gram Schmidt Using Vector Projection

- **Given k vectors** $v_1, \dots, v_k$.
  - Gram Schmidt process defines vectors $u_1, \dots, u_k$ as shown in below expression.
    - $\boldsymbol{u_1}, \dots, \boldsymbol{u_k}$ is required system of orthogonal vectors.
      - Known as Gram-Schmidt orthogonalization.
    - Normalized vector $\boldsymbol{e_1}, \dots \boldsymbol{e_k}$ form an orthonormal set.
      - Known as Gram-Schmidt orthogonalization.

$$\boldsymbol{u_1} = \boldsymbol{v_1} \qquad \boldsymbol{e_1} = \frac{\boldsymbol{u_1}}{\|\boldsymbol{u_1}\|}$$

$$\boldsymbol{u_2} = \boldsymbol{v_2} - \boldsymbol{proj_{u_1}}(\boldsymbol{v_2}) \qquad \boldsymbol{e_2} = \frac{\boldsymbol{u_2}}{\|\boldsymbol{u_2}\|}$$

$$\boldsymbol{u_3} = \boldsymbol{v_3} - \boldsymbol{proj_{u_1}}(\boldsymbol{v_3}) - \boldsymbol{proj_{u_2}}(\boldsymbol{v_3}) \qquad \boldsymbol{e_2} = \frac{\boldsymbol{u_2}}{\|\boldsymbol{u_2}\|}$$

$$\vdots \qquad \vdots$$

$$\boldsymbol{u_k} = \boldsymbol{v_k} - \sum_{j=1}^{k-1} \boldsymbol{proj_{u_j}}(\boldsymbol{v_k}) \qquad \boldsymbol{e_k} = \frac{\boldsymbol{u_k}}{\|\boldsymbol{u_k}\|}$$

Expression of Gram Schmidt using vector projection

Check Formula Validity

- **First, compute $< \boldsymbol{u_1}, \boldsymbol{u_2} >$**
  - Substituting previous formula for $\boldsymbol{u_2}$.
    - $\boldsymbol{u_2 = v_2 - proj_{u_1}(v_2)}$
  - Get **zero**.

- **Then, Compute $< \boldsymbol{u_1}, \boldsymbol{u_3} >$**
  - Substituting previous formula for u_3.
    - $\boldsymbol{u_3 = v_3 - proj_{u_1}(v_3) - proj_{u_2}(v_3)}$
  - Get **zero**.

$$\boldsymbol{u_1 = v_1} \qquad \boldsymbol{e_1 = \frac{u_1}{\|u_1\|}}$$

$$\boldsymbol{u_2 = v_2 - proj_{u_1}(v_2)} \qquad \boldsymbol{e_2 = \frac{u_2}{\|u_2\|}}$$

$$\boldsymbol{u_3 = v_3 - proj_{u_1}(v_3) - proj_{u_2}(v_3)} \qquad \boldsymbol{e_2 = \frac{u_2}{\|u_2\|}}$$

$$\vdots \qquad \vdots$$

$$\boldsymbol{u_k = v_k - \sum_{j=1}^{k-1} proj_{u_j}(v_k)} \qquad \boldsymbol{e_k = \frac{u_k}{\|u_k\|}}$$

Expression of Gram Schmidt using vector projection

Geometrically Check Formula Validity

$$u_1 = v_1 \qquad e_1 = \frac{u_1}{\|u_1\|}$$

$$u_2 = v_2 - proj_{u_1}(v_2) \qquad e_2 = \frac{u_2}{\|u_2\|}$$

$$u_3 = v_3 - proj_{u_1}(v_3) - proj_{u_2}(v_3) \qquad e_2 = \frac{u_2}{\|u_2\|}$$

$$\vdots \qquad \vdots$$

$$u_k = v_k - \sum_{j=1}^{k-1} proj_{u_j}(v_k) \qquad e_k = \frac{u_k}{\|u_k\|}$$

Expression of Gram Schmidt using vector projection

- **To compute $\boldsymbol{u_i}$,**
  - Projects $\boldsymbol{v_i}$ orthogonally onto subspace U.
    - U: generated by $u_1, \dots, u_{i-1}$
      - Same as subspace generated by $v_1, \dots, v_{i-1}$
    - Vector $\boldsymbol{u_i}$ defined to be the difference between $\boldsymbol{v_i}$.
  - This projection guaranteed to be **orthogonal to all of the vectors in the subspace $\boldsymbol{U}$**.

Gram-Schmidt and Linearly Independent Sequence

- **Gram-Schmidt process applies to linearly independent countably infinite sequence $\{\boldsymbol{v}_i\}_i$.**

- **Result of application**
  - An orthogonal(or orthonormal) sequence $\{\boldsymbol{u}_i\}_i$.
    - For natural number n: algebraic span of $v_1, \dots, v_n$ is same as that of $u_1, \dots, u_n$.

- **How about applied to a linearly dependent sequence?**
  - Outputs $\mathbf{0}$ vector in the i^{th} step.
    - Assuming that v_i is a linear combination of $v_1, \dots, v_{i-1}$.
  - To produce orthonormal basis.
    - Algorithm should test for zero vectors in the output.
    - Algorithm should discard zero vectors.
      - Because no multiple of a zero vector can have a length of $\boxed{1}$.
    - Number of vectors output by algorithm will be dimension of the space spanned by the original inputs.

Euclidean Space

- **Consider following set of vectors in R^2 as Eq 1..**
  - With conventional inner product
- **Then, perform Gram-Schmidt as Eq 2..**
  - To obtain orthogonal set of vectors!

$$\boldsymbol{S} = \left\{ \boldsymbol{v}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \boldsymbol{v}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right\}$$

Eq 1. Set of vectors

$$\boldsymbol{u}_1 = \boldsymbol{v}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$\boldsymbol{u}_2 = \boldsymbol{v}_2 - proj_{\boldsymbol{u}_1}(\boldsymbol{v}_2) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - proj_{\begin{bmatrix} 3 \\ 1 \end{bmatrix}} \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \frac{8}{10} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -2/5 \\ 6/5 \end{bmatrix}$$

Eq 2. Gram-Schmidt

Check Whether Orthogonal or Not

- **Check that vectors u_1 and u_2 are indeed orthogonal as Eq 1..**
  - If dot product of two vectors is $\mathbf{0}$, then they are $\boxed{\text{orthogonal.}}$.
- **In case of non-zero vectors,**
  - We can normalize vectors by dividing out their sizes as Eq 2..

$$\langle \boldsymbol{u}_1, \boldsymbol{u}_2 \rangle = \left\langle \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} -2/5 \\ 6/5 \end{bmatrix} \right\rangle = -\frac{6}{5} + \frac{6}{5} = 0$$

Eq 1. Dot product of two vectors

$$\boldsymbol{e}_1 = \frac{1}{\sqrt{10}} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$\boldsymbol{e}_2 = \frac{1}{\sqrt{\frac{40}{25}}} \begin{bmatrix} -2/5 \\ 6/5 \end{bmatrix} = \frac{1}{\sqrt{10}} \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

Eq 2. Normalizing vectors

Code Exercise of Gram-Schmidt algorithm using MATLAB

Code Exercise (09_01)

- Follow the order of Gram-Schmidt algorithm in previous slide.

```
%% Gram-Schmidt Algorithm

% Clear workspace, command window, and close all figures
clc; clear; close all;

% Initialize the matrices
A = [8 1 6; 3 5 7; 4 9 2];
Q = zeros(3);

% Perform the Gram-Schmidt process
for i = 1:size(A, 2)
    % Start with the original vector
    v = A(:, i);

    % Subtract the projections onto all previously obtained orthogonal vectors
    for j = 1:i-1
        v = v - (Q(:, j)' * A(:, i)) / (Q(:, j)' * Q(:, j)) * Q(:, j);
    end

    % Normalize the vector to make it orthogonal
    Q(:, i) = v / norm(v);
end

% Display the original and orthogonalized matrices
disp('Original Matrix A:');
disp(A);
disp('Orthogonalized Matrix Q:');
disp(Q);

% Verify orthogonality by computing dot proudct
disp('Dot products between different vectors of Q (should be close to zero):');
for i = 1:size(Q, 2)
    for j = i+1:size(Q, 2)
        fprintf('Dot product between Q(:, %d) and Q(:, %d): %f\n', i, j, dot(Q(:, i), Q(:, j)));
    end
end
```

MATLAB code

THANK YOU
FOR YOUR ATTENTION