LADS

3/13/20

- Communicate via email for now (not through canvas)

QR Decomp (Factorization).

$$B = \begin{array}{cc} & \\ m & \begin{bmatrix} \vec{b}_1 & \cdots & \vec{b}_n \end{bmatrix} \\ & n \end{array} \qquad \text{rank}(B) = n$$

Goal: Produce o.n. set $\{\vec{q}_1, \ldots, \vec{q}_n\}$ with $\{\vec{q}_1, \ldots, \vec{q}_k\}$ o.n. basis for $\text{span}(\vec{b}_1, \ldots, \vec{b}_k)$

The k^{th} step in Gram-Schmidt

$$\vec{v}_k = \vec{b}_k - (\vec{b}_k \cdot \vec{q}_1)\vec{q}_1 + \cdots - (\vec{b}_k \cdot \vec{q}_{k-1})\vec{q}_{k-1}$$

$$\vec{q}_k = \frac{\vec{v}_k}{\|v_k\|_2}$$

This yields Q and

$QR = A$ so $R = Q^T A$

since $Q^T Q = I$

$\boxed{eg!}$

$$B = \begin{bmatrix} -1 & -1 & 1 \\ 1 & 3 & 3 \\ -1 & -1 & 5 \\ 1 & 3 & 7 \end{bmatrix}$$

eq2

$$\vec{b}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \qquad \vec{q}_1 = \frac{\vec{b}_1}{\|\vec{b}_1\|} = \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix}$$

$$\vec{V}_2 = \vec{b}_2 - (\vec{b}_2 \cdot \vec{q}_1)\,\vec{q}_1 = \begin{bmatrix} -1 \\ 3 \\ -1 \\ 3 \end{bmatrix} - \left(\begin{bmatrix} -1 \\ 3 \\ -1 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix} \right) \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 3 \\ -1 \\ 3 \end{bmatrix} - \underset{4}{\underset{\parallel}{\left[1/2 + 3/2 + 1/2 + 3/2 \right]}} \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 3 \\ -1 \\ 3 \end{bmatrix} - \begin{bmatrix} -2 \\ 2 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\vec{q}_2 = \frac{\vec{V}_2}{\|\vec{V}_2\|} = \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

$$\vec{v_3} = \vec{b_3} - (\vec{b_3} \cdot \vec{q_1})\vec{q_1} - (\vec{b_3} \cdot \vec{q_2})\vec{q_2}$$

$$= \begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \end{bmatrix} - \left(\begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \end{bmatrix} \cdot \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix} \right) \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix} - \left(\begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \end{bmatrix} \cdot \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix} \right) \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \end{bmatrix} - \underbrace{\left(-1/2 + 3/2 - 5/2 + 7/2\right)}_{= \, 2} \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix} - \underbrace{\left(\tfrac{1}{2} + \tfrac{3}{2} + \tfrac{5}{2} + \tfrac{7}{2}\right)}_{= \, 8} \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ 2 \\ 2 \end{bmatrix}$$

$$q_3 = \frac{\vec{v_3}}{\|v_3\|} = \frac{1}{\sqrt{16}} \begin{bmatrix} -2 \\ -2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

$$\text{so } Q = \begin{bmatrix} -1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix}$$

Q has orthonormal columns.

eg 4

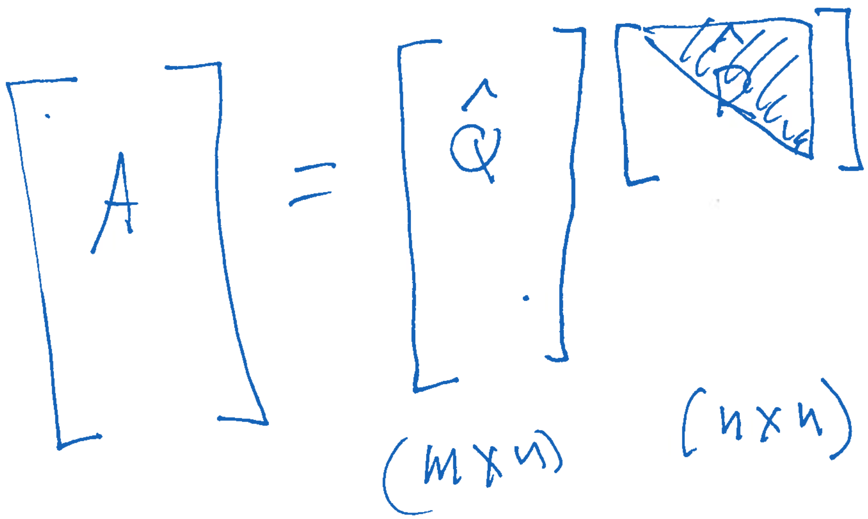
$B = QR \Rightarrow Q^T B = R$ so

$$R = \begin{bmatrix} -1/2 & 1/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 & 1/2 \\ -1/2 & -1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 \\ 1 & 3 & 3 \\ -1 & -1 & 5 \\ 1 & 3 & 7 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 4 & 2 \\ 0 & 2 & 8 \\ 0 & 0 & 4 \end{bmatrix}$$

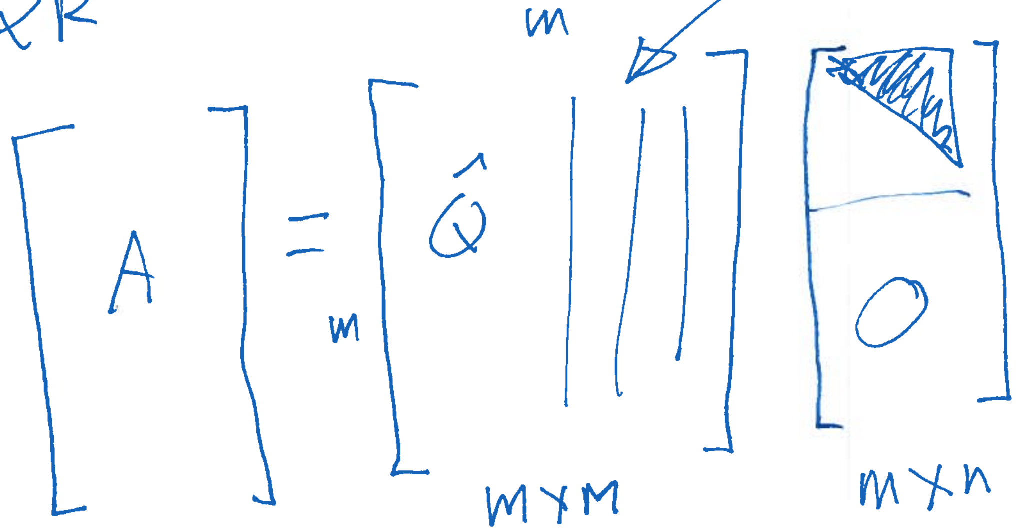
This is the thin or reduced QR.

often written $A = \hat{Q}\hat{R}$



Full QR

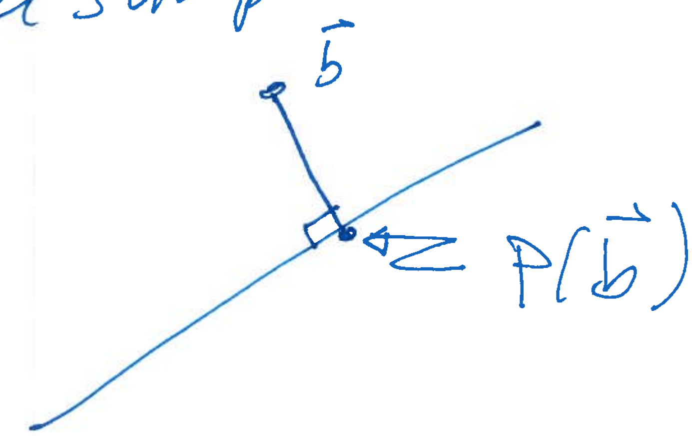
new o.n. col to complete the basis



Q is orthogonal!

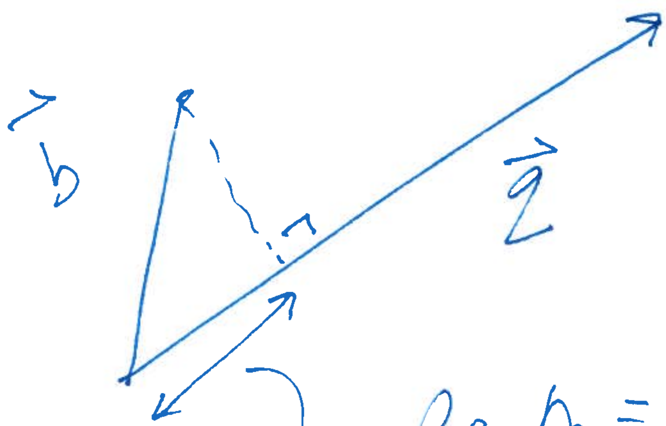
$\boxed{\text{QR and orthogonal projection}}$

Imp. in optimization since <u>shortest</u> (min). distance from a pt. to a subspace is orthogonal projection



What is the formula for $P =$ orthogonal projection?

One dim!



$\vec{q}$ is unit vector

length $= \vec{b}\cdot\vec{q}$

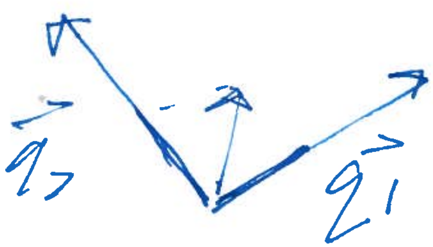
vector is $(\vec{b}\cdot\vec{q})\vec{q}$

$$(\vec{b}\cdot\vec{q})\vec{q} = \vec{q}(\vec{q}\cdot\vec{b}) = \vec{q}(\vec{q}^T\vec{b})$$

$$= \underbrace{(\vec{q}\vec{q}^T)}_{\text{outer product} \,=\, \text{rank 1 matrix}}\vec{b}$$

So $P(\vec{b}) = (\vec{q}\vec{q}^T)\vec{b}$ so $P = \vec{q}\vec{q}^T$

Now projection onto a plane



$$\vec{p}(\vec{b}) = (\vec{b} \cdot \vec{q}_1)\vec{q}_1 + (\vec{b} \cdot \vec{q}_2) q_2 .$$

$$= (q_1 q_1^T)\vec{b} + (\vec{q}_2 \cdot q_2^T)\vec{b}$$

$$= \begin{bmatrix} \vec{q}_1 & \vec{q}_2 \end{bmatrix} \begin{bmatrix} q_1^T \\ q_2^T \end{bmatrix} \vec{b}$$

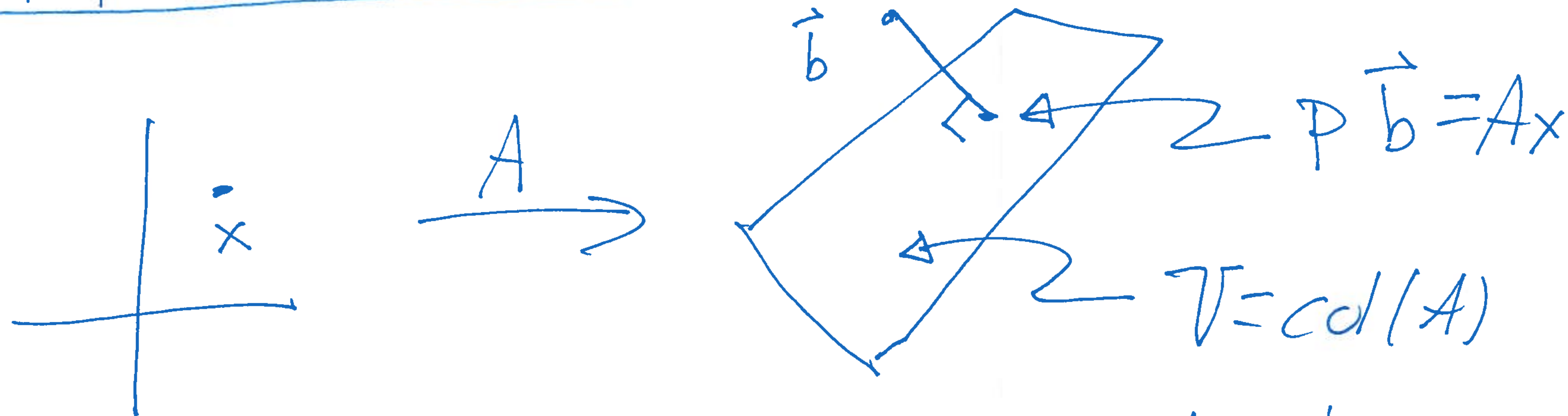
So $P = QQ^T$ with $Q = \begin{bmatrix} \vec{q}_1 & \vec{q}_2 \end{bmatrix}$

<u>Theorem</u>: If V has o.n. basis $\{\vec{q}_1, \dots, \vec{q}_k\}$ let $\hat{Q} = \begin{bmatrix} \vec{q}_1 & \cdots & \vec{q}_k \end{bmatrix}$

Then orthogonal projection onto V is given by $P = \hat{Q}\hat{Q}^T$

or $P\vec{b} = (\hat{Q}\hat{Q}^T)\vec{b}$

Application to least squares



The soln to least sq ~~sol~~ $Ax=b$

is $Ax = P\vec{b}$.

Now say $A = \hat{Q}\hat{R}$ factorization

so $\hat{Q}$ has columns that are an o.n.

basis for col(A).

So projection onto col(A) is given by $P = \hat{Q}\hat{Q}^T$

$A x = P\vec{b}$ becomes

$\hat{Q}\hat{R}x = \hat{Q}\hat{Q}^T\vec{b}$

$\boxed{\hat{R}x = \hat{Q}^T\vec{b}}$ since $\hat{Q}^T\hat{Q} = I$

easy to solve by back substitution.

MACHINE Learning – some examples

$$x_i \longrightarrow \boxed{\text{Machine } M} \longrightarrow M(x_i)$$

data with features → x_i

categories. → $M(x_i)$

bit maps of hand written numerals 16x16 matrix ⟶ 10 categories 0, 1, .., 9

X_i Input

has correct output y_i

So WANT $y_i = M(x_i)$ correct answer

How build M?

Rule based expert system - deterministic, fixed

New idea

M depends on parameters $\mathcal{m}$ and can learn by adjusting $\mathcal{m}$

Given $\vec{x}_1, \ldots, \vec{x}_n$ training data, $\vec{y}_1, \ldots, \vec{y}_n$ correct output

~~E~~ Pick $\mathcal{m}_1$, initial parameters

for $i = 1$ to n

$x_i \rightarrow M_{\mathcal{m}_i}(x_i)$

adjust $\mathcal{m}_i$ to shrink $|M_{\mathcal{m}_i}(x_i) - y_i| = \text{error}_i$

end

- $M_{\mathcal{m}_n}$ then used on new data and see how it does.
- NOTE: OFTEN MULTIPLE x_i are processed before adjusting $\mathcal{m}$.

What form should the machine take

- Popular and successful structure is neural net — many variants
  - First version
  - explain later what motivates this choice.

$\cancel{F}$

K-level net

$$M_{m}(x) = F_K \circ F_{K-1} \circ \dots \circ F_1(x)$$

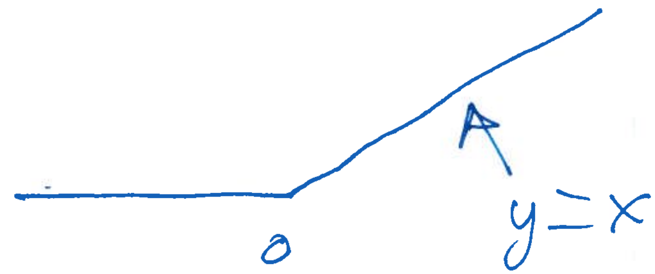
each $F_i(x, A_i, \vec{b})$ A_i is matrix

weights

bias

$$= \sigma(A_i \vec{x} + \vec{b})$$

activation function

$\sigma(y) =$ 

$\sigma(y_1, y_2, \ldots, y_n) = (\sigma(y_1), \ldots, \sigma(y_n))$

vectorized version

Deep [redacted] feed forward neural net

many layers or functions F_i