Lecture 13: Gram-Schmidt orthogonalization

Tuesday, October 6, 2015 9:30 AM

Admin: Midterm October 8
Reading: Gram-Schmidt Chapter 5.5

Recall:

Orthogonal basis → pairwise orthogonal vectors

Orthonormal basis → orthogonal, length-one vectors

MORAL: Orthonormal bases behave just like the standard basis.

eg., for $\vec{u} = \sum_j \alpha_j \vec{v}_j$, $\vec{v} = \sum_j \beta_j \vec{v}_j$,

$$\vec{u} \cdot \vec{v} = \sum_j \alpha_j^* \beta_j$$

FACT: For a subspace $\mathcal{U} \subseteq \mathbb{R}^n$ with orthonormal basis $\{\vec{u}_1, \ldots, \vec{u}_k\}$,

orthogonal projection onto $\mathcal{U}$

$$P_{\mathcal{U}} = \sum_{j=1}^{k} \vec{u}_j \vec{u}_j^T$$

Example: (coordinate expansion)

$$\vec{v} = P_{\mathbb{R}^n} \vec{v}$$

$$= \sum_{j=1}^{n} \vec{u}_j \vec{u}_j^T \vec{v}$$

($\vec{u}_j^T \vec{v} = \vec{u}_j \cdot \vec{v}$)

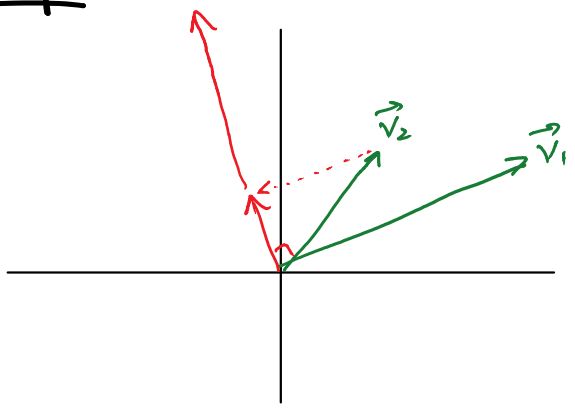
$$\boxed{\vec{v} = \sum_j (\vec{u}_j \cdot \vec{v}) \vec{u}_j}$$

Today:

HOW TO GET AN ORTHONORMAL BASIS

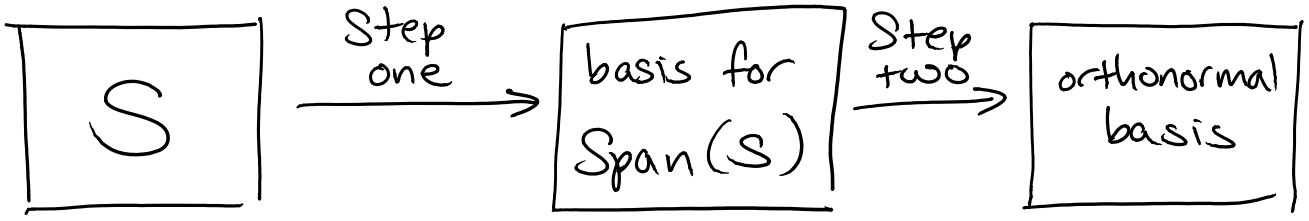
Example:

Example:



General problem: Let $S = \{\vec{v}_1, \ldots, \vec{v}_k\}$ be a set of vectors in $\mathbb{R}^n$ or $\mathbb{C}^n$.

Goal: Find an orthonormal basis for Span(S).



Step One: Find any basis for Span(S).
(because the vectors might be linearly dependent)

Let $A = \begin{pmatrix} \text{---}\ v_1^T\ \text{---} \\ \text{---}\ v_2^T\ \text{---} \\ \vdots \\ \text{---}\ v_k^T\ \text{---} \end{pmatrix}$. Span(S) = $R(A^T)$ rowspace.

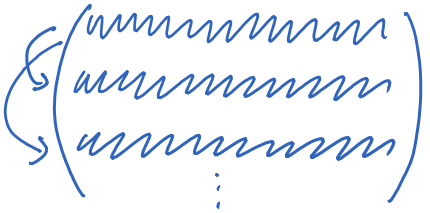
Gaussian elimination on A leaves a basis.

Step Two: Adjust the basis to be orthonormal.

Assume S is a basis (ie., lin. indep.).

$\Rightarrow \dim(\text{Span}(S)) = k$

Gaussian elimination



keep going

Same idea! $\{\vec{v} \quad \vec{v} \quad \vec{v}\} \longrightarrow \{\vec{v}_1, P_{v_1}^{\perp} v_2, \ldots, P_{v_1}^{\perp} v_k\}$ keep

Same idea! $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k\} \longrightarrow \{\vec{v}_1, P_{v_1}^{\perp} v_2, \ldots, P_{v_1}^{\perp} v_k\}$ keep going

First fix $\vec{v}_2, \ldots, \vec{v}_k$ to be $\perp$ to $\vec{v}_1$

Then fix vectors 3-k to be $\perp$ to vector 2

⋮ etc.

Gram-Schmidt procedure (inputs $\vec{v}_1, \ldots, \vec{v}_k$)

1. Project $\vec{v}_2, \ldots, \vec{v}_k$ to be orthogonal to $\vec{v}_1$, using $P_{v_1}^{\perp} = I - P_{v_1} = I - \frac{\vec{v}_1 \vec{v}_1^T}{\|v_1\|^2}$.
2. Now $P_{v_1}^{\perp} v_2, \ldots, P_{v_1}^{\perp} v_k$ span a $(k-1)$-dim subspace of Span(S), orthogonal to $\vec{v}_1$.
   Recurse to find an orthogonal basis for it.
   ie. $P_{v_1}^{\perp} v_3 \longmapsto P_{P_{v_1}^{\perp} v_2}^{\perp}\left(P_{v_1}^{\perp} v_3\right)$, etc.
3. Renormalize the vectors (divide by their lengths)

Example: Find an orthonormal basis for the span of

$(1,0,0,-1)$, $(1,2,0,-1)$, $(3,1,1,-1)$.

Answer: $\vec{v}_1$ $\vec{v}_2$ $\vec{v}_3$

$$v_1 \xrightarrow{\text{normalize}} v_1' = \frac{v_1}{\|v_1\|} = \frac{1}{\sqrt{2}}(1,0,0,-1)$$

$$v_2 \xrightarrow{\perp v_1} v_2 - (v_1' \cdot v_2) v_1' = (1,2,0,-1) - (1,0,0,-1)$$

$$= (0,2,0,0)$$

$$\xrightarrow{\text{normalize}} v_2' = (0,1,0,0)$$

$$v_3 \xrightarrow{\perp v_1} v_3 - (v_1' \cdot v_3) v_1' = (3,1,1,-1) - \frac{1}{2}(v_1 \cdot v_3) v_1$$

$$= (3,1,1,-1) - 2(1,0,0,-1)$$

$$v_3' = (1,1,1,1)$$

$$\xrightarrow{\perp v_2'} v_3' - (v_2' \cdot v_3') v_2'$$

$= (1, 0, 1, 1)$

normalize $\rightarrow \frac{1}{\sqrt{3}}(1, 0, 1, 1)$

$$\Rightarrow \left\{ \frac{1}{\sqrt{2}}(1,0,0,-1), (0,1,0,0), \frac{1}{\sqrt{3}}(1,0,1,1) \right\}$$

Sanity check: Why $v_2 \longrightarrow v_2 - \frac{(v_1 \cdot v_2)}{\|v_1\|^2} v_1$?

1. $v_1 \cdot \left(v_2 - \frac{(v_1 \cdot v_2)}{\|v_1\|^2} v_1 \right) = v_1 \cdot v_2 - \frac{(v_1 \cdot v_2)}{\|v_1\|^2}(v_1 \cdot v_1)$

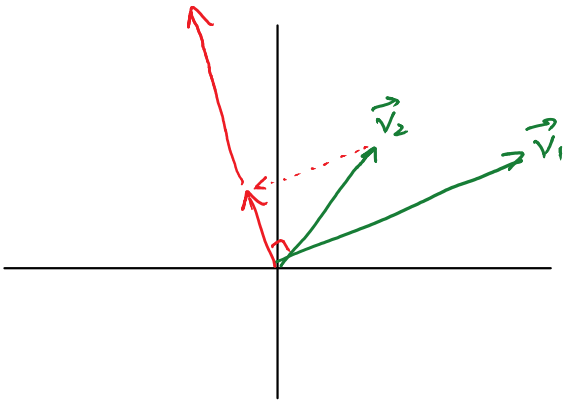
$= 0$

result is $\perp$ to v_1 ✓

2. $\text{Span}\{v_1, v_2\} = \text{Span}\{v_1, v_2 - \alpha v_1\}$ ✓

linear combination leave spanned space alone

(same as Gaussian Elim. preserves rowspace)



Observe: Two natural orders for Gram-Schmidt:

① For vector i from 1 to $k-1$:
- fix all subsequent vectors to be $\perp$ to vector i

② For vector i from 2 to k:
- project vector i to be $\perp$ to all preceding vectors

these are equivalent! (but method ① is more numerically stable)

Exercise: In Matlab/Octave/Mathematica, perform Gram-Schmidt on 5 random vectors in $\mathbb{R}^{10}$. Verify that the order doesn't matter. (Or does it?)

```
n = 10;
```

```
d = 5;
% choose d random vectors in R^n, with normally distributed coordinates
vectors = randn(n, d)
```

(2)
```
A = vectors;
for i = 1:d
  for j = 1:i-1
    A(:,i) = A(:,i) - (A(:,j)'*A(:,i)) * A(:,j);
  end
  A(:,i) = A(:,i) / sqrt(A(:,i)'*A(:,i));
end
```
} project ith col. to be ⊥ to previous columns

(1)
```
B = vectors;
for i = 1:d
  B(:,i) = B(:,i) / sqrt(B(:,i)'*B(:,i));
  for j = i+1:d
    B(:,j) = B(:,j) - (B(:,i)'*B(:,j)) * B(:,i);
  end
end
```
project subsequent columns to be ⊥ to column i

Check the answer:

```
A'*A
```

```
sum(sum(abs(A-B)))
```
→ 0 ✓

```
ans =

   1.0000e+00   4.8572e-17   3.1225e-17  -9.7145e-17   8.3267e-17
   4.8572e-17   1.0000e+00   4.8572e-17  -7.6328e-17  -5.5511e-17
   3.1225e-17   4.8572e-17   1.0000e+00   7.8063e-17  -1.3878e-17
  -9.7145e-17  -7.6328e-17   7.8063e-17   1.0000e+00   3.4694e-17
   8.3267e-17  -5.5511e-17  -1.3878e-17   3.4694e-17   1.0000e+00
```

Observe: On m vectors in $\mathbb{R}^n$, running time of G-S. is $\boxed{O(m^2 n)}$ $(= O(n^3)$ if $m = n)$

Question: What happens if we don't start with a linearly independent set of vectors?

Answer: It still works! Just some vectors will be zeroed out.

$\Rightarrow$ No need to run Gaussian elimination first.

Example: Gram-Schmidt on

$$\{(1,0), (1,-2), (0,1)\}$$

project to $(1,0)$:

$(1,-2) \longrightarrow (1,-2) - ((1,0)\cdot(1,-2))(1,0)$

$= (0,-2) \longrightarrow (0,-1)$ renormalizing

$(0,1) \longrightarrow (0,1) - ((1,0)\cdot(0,1))(1,0)$

$= (0,1)$

project to $(0,-1)$:

$(0,1) \longrightarrow (0,1) - ((0,-1)\cdot(0,1))(0,-1)$

project Y to $(0,-1)$: $(0,1) \longrightarrow (0,1) - \underbrace{((0,-1)\cdot(0,1))}_{-1}(0,-1)$

$= (0,0)$

$\Rightarrow$ $\boxed{\{(1,0),(0,-1)\}}$ an orthonormal basis

(just be careful about dividing by 0!!)

Recall: LU-decomposition

$$A = \begin{pmatrix}\text{permutation}\\ \text{of}\\ \text{rows}\end{pmatrix} \cdot \begin{pmatrix}\text{lower triangular}\end{pmatrix} \cdot \begin{pmatrix}\text{upper triangular}\end{pmatrix}$$

(inverse) history of G. elim.

result of Gaussian elimination

Gaussian elimination $O(n^3)$ steps to solve n equations in n unknowns

- But if you store the LU decomposition, then further equations can each be solved in $O(n^2)$ steps by back-substitution (for L and for U).

Main idea: Solving a lower or upper triangular system of equations is much faster than solving a general system: $O(n^2)$ versus $O(n^3)$.

Similarly...

It is easy to solve $Ax = b$

when the columns of A are orthonormal!

$$\begin{pmatrix} | & | & & | \\ \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \vec{x} \end{pmatrix} = \begin{pmatrix} \vec{b} \end{pmatrix}$$

$\Leftrightarrow x_1\vec{u}_1 + x_2\vec{u}_2 + \cdots + x_n\vec{u}_n = \vec{b}$

$\Rightarrow x_1 = \vec{u}_1 \cdot \vec{b}$, $x_2 = \vec{u}_2 \cdot \vec{b}$, ... $O(n^2)$ steps

QR decomposition: Any $m \times n$ matrix A with linearly independent columns can be factored as

$$A_{m \times n} = \underset{m}{\underset{n}{Q}} \cdot \underset{n}{\underset{n}{R}}$$

Q: orthonormal columns spanning $R(A)$

R: upper bar history of Gram-Schmidt process

This gives another way of amortizing the cost of solving

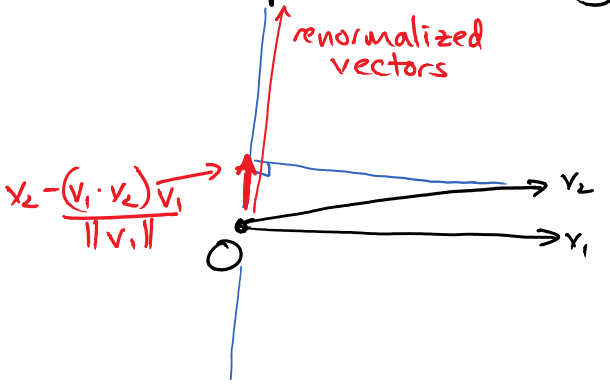
$Ax = b_1$, $Ax = b_2$, $Ax = b_3$, $Ax = b_4$, ...

Run G.-S. once $O(n^3)$, then $O(n^2)$ for each equation.

DON'T DO THIS!

Gram-Schmidt gets ugly fast.

Example: Instability



If a vector v_j is close to $\text{Span}\{v_1, \ldots, v_{j-1}\}$, then renormalization will blow up the length a lot, amplifying errors!

(QR decomposition can still be useful (see example 5.5.3 for using it to find x minimizing $\|Ax - b\|$). Numerically, the "Householder method" is slightly better than naive Gram-Schmidt; that's what Matlab's qr(·) function uses.)