Math 2270-002 Week 13 notes
We will not necessarily finish the material from a given day's notes on that day. We may also add or subtract some material as the week progresses, but these notes represent an in-depth outline of what we plan to cover. These notes cover material in 6.4-6.6

Mon Nov 19

- 6.4   Gram Schmidt and $A = QR$ decomposition. Orthogonal matrices

Announcements:

Warm-up Exercise:

We begin on Monday with a continuation of the discussion of Gram-Schmidt orthogonalization from 6.4. Keeping track of the G.S. process carefully yields the $A = QR$ matrix product decomposition theorem, where Q is an "orthogonal matrix" consisting of an orthonormal basis for the span of the columns of A and R is an upper triangular matrix with positive entries along the diagonal. This decomposition is one way to understand why matrix determinants correspond to $\pm$ Volumes, in $\mathbb{R}^n$, and can also be useful in solving multiple linear systems of equations with the same "A" matrix more efficiently.

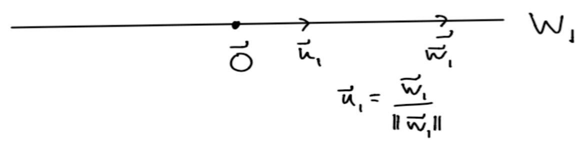
Section 6.5, *Least square solutions* is about finding approximate solutions to inconsistent matrix equations, and relies on many of the ideas we've been studying in Chapter 6 up to this point. Whenever one tries to fit experimental data to finite dimensional models it is extremely unlikely that one will get an exact fit. Least squares solutions are the "best possible", and for this reason software like Matlab automatically returns the least squares "solution" when aske to solve an inconsistent system.

Section 6.6, *Applications to linear models,* is an application of the least squares method to e.g. single or multivariate linear regression in statistics.

Recall the Gram-Schmidt process from Friday:

Start with a basis $B = \{\underline{w}_1, \underline{w}_2, \dots \underline{w}_p\}$ for a subspace W of $\mathbb{R}^n$. How can you convert it into an orthonormal basis? Here's how! The inductive process is called Gram-Schmidt orthogonalization.

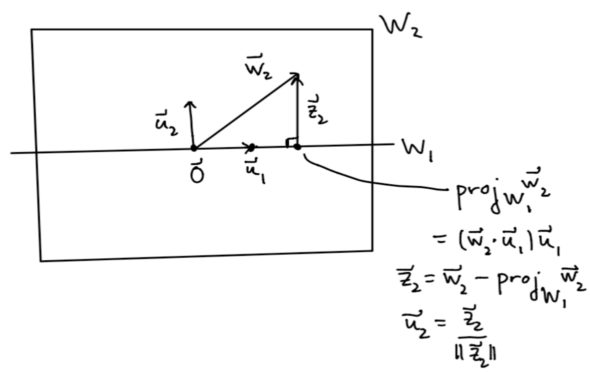
Let $W_1 = span\{\underline{w}_1\}$. Define $\underline{u}_1 = \dfrac{\underline{w}_1}{\|\underline{w}_1\|}$. Then $\{\underline{u}_1\}$ is an orthonormal basis for W_1.



Let $W_2 = span\{\underline{w}_1, \underline{w}_2\} = span\{\underline{u}_1, \underline{w}_2\}$.

Let $\underline{z}_2 = \underline{w}_2 - proj_{W_1}\underline{w}_2 = \underline{w}_2 - (\underline{w}_2 \cdot \underline{u}_1)\underline{u}_1$ so $\underline{z}_2 \perp \underline{u}_1$.

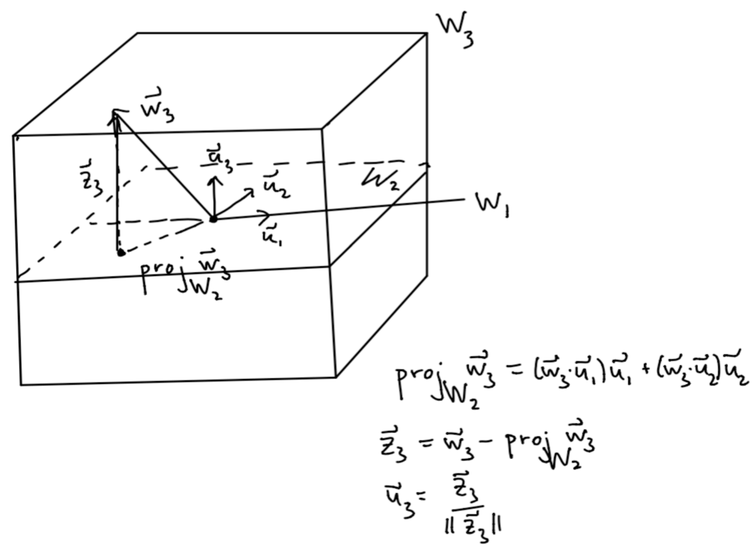
Define $\underline{u}_2 = \dfrac{\underline{z}_2}{\|\underline{z}_2\|}$. So $\{\underline{u}_1, \underline{u}_2\}$ is an orthonormal basis for W_2.



Let $W_3 = span\{\underline{\boldsymbol{w}}_1, \underline{\boldsymbol{w}}_2, \underline{\boldsymbol{w}}_3\}$.

Let $\underline{\boldsymbol{z}}_3 = \underline{\boldsymbol{w}}_3 - proj_{W_2}\underline{\boldsymbol{w}}_3$, so $\underline{\boldsymbol{z}}_3 \perp W_2$.

Define $\underline{\boldsymbol{u}}_3 = \dfrac{\underline{\boldsymbol{z}}_3}{\|\underline{\boldsymbol{z}}_3\|}$. Then $\{\underline{\boldsymbol{u}}_1, \underline{\boldsymbol{u}}_2, \underline{\boldsymbol{u}}_3\}$ is an orthonormal basis for W_3.



Inductively,

Let $W_j = span\{\underline{\boldsymbol{w}}_1, \underline{\boldsymbol{w}}_2, \ldots \underline{\boldsymbol{w}}_j\} = span\{\underline{\boldsymbol{u}}_1, \underline{\boldsymbol{u}}_2, \ldots \underline{\boldsymbol{u}}_{j-1}, \underline{\boldsymbol{w}}_j\}$.

Let $\underline{\boldsymbol{z}}_j = \underline{\boldsymbol{w}}_j - proj_{W_{j-1}}\underline{\boldsymbol{w}}_j = \underline{\boldsymbol{w}}_j - (\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_1)\underline{\boldsymbol{u}}_1 - (\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_2)\underline{\boldsymbol{u}}_2 - \ldots (\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_{j-1})\underline{\boldsymbol{u}}_{j-1}$.

Define $\underline{\boldsymbol{u}}_j = \dfrac{\underline{\boldsymbol{z}}_j}{\|\underline{\boldsymbol{z}}_j\|}$. Then $\{\underline{\boldsymbol{u}}_1, \underline{\boldsymbol{u}}_2, \ldots \underline{\boldsymbol{u}}_j\}$ is an orthonormal basis for W_j .

Continue up to $j = p$.

Exercise 1 Perform Gram-Schmidt on the $\mathbb{R}^3$ basis

$$B = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} \right\}.$$

This will proceed as the Friday exercise until the third step, i.e.

$$\underline{\boldsymbol{u}}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \underline{\boldsymbol{u}}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

.

$A = Q\,R$ decomposition:

We're denoting the original basis for W by $B = \{\underline{\boldsymbol{w}}_1, \underline{\boldsymbol{w}}_2, \dots \underline{\boldsymbol{w}}_p\}$. Denote the orthonormal basis we've constructed with Gram-Schmidt by $O = \{\underline{\boldsymbol{u}}_1, \underline{\boldsymbol{u}}_2, \dots \underline{\boldsymbol{u}}_p\}$. Because O is orthonormal it's easy to express these two bases in terms of each other. Notice

$$W_j = span\ \{\underline{\boldsymbol{w}}_1, \underline{\boldsymbol{w}}_2, \dots \underline{\boldsymbol{w}}_j\} = span\ \{\underline{\boldsymbol{u}}_1, \underline{\boldsymbol{u}}_2, \dots \underline{\boldsymbol{u}}_j\} \qquad \text{for each } 1 \le j \le p.$$

So,

$$\underline{\boldsymbol{w}}_1 = (\underline{\boldsymbol{w}}_1 \cdot \underline{\boldsymbol{u}}_1)\underline{\boldsymbol{u}}_1$$

$$\underline{\boldsymbol{w}}_2 = (\underline{\boldsymbol{w}}_2 \cdot \underline{\boldsymbol{u}}_1)\underline{\boldsymbol{u}}_1 + (\underline{\boldsymbol{w}}_2 \cdot \underline{\boldsymbol{u}}_2)\underline{\boldsymbol{u}}_2$$

$$\vdots$$

$$\underline{\boldsymbol{w}}_j = (\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_1)\underline{\boldsymbol{u}}_1 + (\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_2)\underline{\boldsymbol{u}}_2 + \dots \ (\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_j)\underline{\boldsymbol{u}}_j$$

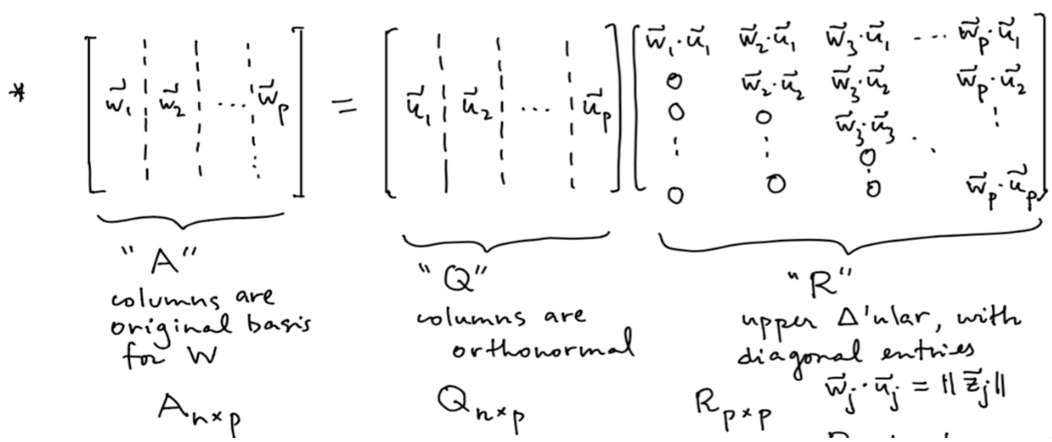
$$\vdots$$

$$\underline{\boldsymbol{w}}_p = \sum_{l=1}^{p} (\underline{\boldsymbol{w}}_l \cdot \underline{\boldsymbol{u}}_l)\underline{\boldsymbol{u}}_l \ .$$

Notice that the coefficients of the last terms in the sums above, namely $(\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_j)$ can be computed as

$$(\underline{\boldsymbol{w}}_j \cdot \underline{\boldsymbol{u}}_j) = \underline{\boldsymbol{z}}_j \cdot \frac{\underline{\boldsymbol{z}}_j}{\|\underline{\boldsymbol{z}}_j\|} = \|\underline{\boldsymbol{z}}_j\|.$$

In matrix form (column by column) we have



Thus any matrix with linearly independent columns may be written in factored form as above, ($W = Col\,A$),

$$A_{n \times p} = Q_{n \times p}\, R_{p \times p}.$$

This factorization contains geometric information and can simplify the computational work needed to solve matrix equations $A\,\underline{\boldsymbol{x}} = \underline{\boldsymbol{b}}$.

From previous page...

$$* \qquad A_{n \times p} = Q_{n \times p} R_{p \times p}$$

<u>shortcut</u> (or what to do if you forgot the formulas for the entries of R) If you just know Q you can recover R by multiplying both sides of the $*$ equation on the previous page by the transpose Q^T of the Q matrix:

$$\begin{bmatrix} \text{---} \ \vec{u}_1^{\,T} \ \text{---} \\ \text{---} \ \vec{u}_2^{\,T} \ \text{---} \\ \vdots \\ \text{---} \ \vec{u}_p^{\,T} \ \text{---} \end{bmatrix} \begin{bmatrix} \vec{w}_1 & \vec{w}_2 & \cdots & \vec{w}_p \end{bmatrix} = \begin{bmatrix} \text{---} \ \vec{u}_1^{\,T} \ \text{---} \\ \text{---} \ \vec{u}_2^{\,T} \ \text{---} \\ \vdots \\ \vec{u}_p^{\,T} \end{bmatrix} \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_p \end{bmatrix} R = IR = R!$$

$$A = Q\,R$$
$$Q^T A = Q^T Q\,R = I\,R = R.$$

<u>Example)</u> From last Friday,

$$B = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right\}, \quad O = \left\{ \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \right\}.$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{w}}_1 \cdot \underline{\boldsymbol{u}}_1 & \underline{\boldsymbol{w}}_2 \cdot \underline{\boldsymbol{u}}_1 \\ 0 & \underline{\boldsymbol{w}}_2 \cdot \underline{\boldsymbol{u}}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 2\sqrt{2} \\ 0 & 2\sqrt{2} \end{bmatrix} = Q\,R.$$

<u>Exercise 2</u>) Verify that R could have been recovered via the formula

$$Q^T A = R$$

From previous page ...

$$\begin{bmatrix} 1 & 0 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 2\sqrt{2} \\ 0 & 2\sqrt{2} \end{bmatrix}.$$

<u>Exercise 3)</u> Verify that the $A = Q\,R$ factorization in this example may be further factored as

$$\begin{bmatrix} 1 & 0 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 2\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}.$$

- So, the transformation $T(\underline{\boldsymbol{x}}) = A\,\underline{\boldsymbol{x}}$ is a composition of (1) an area-preserving shear, followed by (2) a diagonal scaling that increases area by a factor of $\sqrt{2} \cdot 2\sqrt{2} = 4$, followed by a rotation of $\frac{\pi}{4}$, which does not effect area. Since determinants of products matrices are the products of determinants (we checked this back when we studied determinants), and area expansion factors of compositions are also the products of the area expansion factors, the generalization of this example gives another explanation of why the determinant of A (or its absolute value in general) coincides with the area expansion factor, in the 2×2 case. You show in your homework that the only possible Q matrices in the 2×2 case are rotations as above, or reflections across lines through the origin. In the latter case, the determinant of Q is -1, and the determinant of A is negative.

Example from Exercise 1:.

$$B=\left\{\begin{bmatrix}1\\1\\0\end{bmatrix},\begin{bmatrix}0\\4\\0\end{bmatrix},\begin{bmatrix}1\\-2\\3\end{bmatrix}\right\} \qquad O=\left\{\begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\\0\end{bmatrix},\begin{bmatrix}-\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix}\right\}.$$

Exercise 4a Find the $A = Q\,R$ factorization based on the data above, for

$$A=\begin{bmatrix}1&0&1\\1&4&-2\\0&0&3\end{bmatrix}$$

$$\text{solution } A=\begin{bmatrix}\frac{1}{\sqrt{2}}&-\frac{1}{\sqrt{2}}&0\\\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&0\\0&0&1\end{bmatrix}\begin{bmatrix}\sqrt{2}&2\sqrt{2}&-\frac{1}{\sqrt{2}}\\0&2\sqrt{2}&-\frac{3}{\sqrt{2}}\\0&0&3\end{bmatrix}$$

Exercise 4b Further factor R into a diagonal matrix times a volume-preserving shear and interpret the transformation $T(\underline{\mathbf{x}}) = A\,\underline{\mathbf{x}}$ as a composition of (1) a volume preserving shear, followed by (2) a coordinate scaling that increases volume by a factor of 12, followed by a rotation about the x_3 axis in $\mathbb{R}^3$, which preserves volume. The generalization of this example gives another explanation of why the determinant of A (or its absolute value in general) is the volume expansion factor for the transformation $T(\underline{\mathbf{x}}) = A\,\underline{\mathbf{x}}$.

Definition A square $n \times n$ matrix Q is called *orthogonal* if its columns are ortho-normal. (You can read more about orthogonal matrices at e.g. Wikipedia.)

Theorem. Let Q be an orthogonal matrix. Then

a) $Q^{-1} = Q^T$.

b) The rows of Q are also ortho-normal.

c) the transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ given by

$$T(\underline{\boldsymbol{x}}) = Q\,\underline{\boldsymbol{x}}$$

preserves dot products and magnitudes, (so also volumes, since cubes generated by perpendicular vectors will be transformed into equal-volume cubes). In other words, for all $\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}} \in \mathbb{R}^n$,

$$T(\underline{\boldsymbol{x}}) \cdot T(\underline{\boldsymbol{y}}) = \underline{\boldsymbol{x}} \cdot \underline{\boldsymbol{y}}$$

$$\|T(\underline{\boldsymbol{x}})\| = \|\underline{\boldsymbol{x}}\|.$$

d) The only matrix transformations $T : \mathbb{R}^n \to \mathbb{R}^n$ that preserve dot products are orthogonal transformations. (These transformations are often referred to as *isometries*.)

Tues Nov 20

- 6.5 Least squares solutions, and projection revisited.

Announcements:

Warm-up Exercise:

Least squares solutions, section 6.5

In trying to fit experimental data to a linear model you must often find a "solution" to

$$A\,\underline{\mathbf{x}} = \underline{\boldsymbol{b}}$$

where no exact solution actually exists. Mathematically speaking, the issue is that $\underline{\boldsymbol{b}}$ is not in the range of the transformation

$$T(\underline{\mathbf{x}}) = A\,\underline{\mathbf{x}},$$

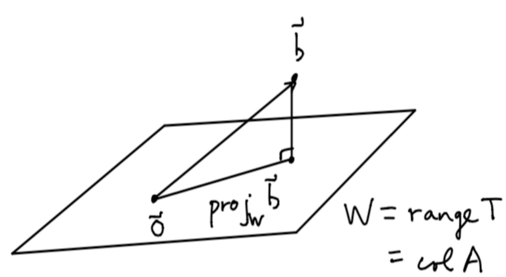
i.e.

$$\underline{\mathbf{x}} \notin Range\ T = Col\ A.$$

In such a case, the *least squares solution(s)* $\hat{\mathbf{x}}$ solve(s)

$$A\,\hat{\mathbf{x}} = proj_{Col\ A}\,\underline{\boldsymbol{b}}\,.$$

Thus, for the least squares solution(s), $A\,\hat{\mathbf{x}}$ is as close to $\underline{\mathbf{b}}$ as possible. Note that there will be a unique least squares solution $\hat{\boldsymbol{x}}$ if and only if $Nul\ A = \{\underline{\mathbf{0}}\}$, i.e. if and only if the columns of A are linearly independent, i.e. all columns are pivot columns.



Exercise 1 Find the least squares solution to

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}.$$

Note that an implicit equation of the plane spanned by the two columns of A is

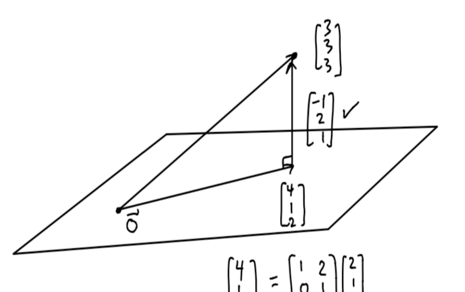
$$-y_1 + 2\,y_2 + y_3 = 0.$$

Since $[3\ 3\ 3]^T$ does not satisfy the implicit equation, there is no exact solution to this problem.

You may use the Gram-Schmidt ortho-normal basis for *Col A*, namely

$$\mathrm{O} = \left\{ \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \right\}.$$

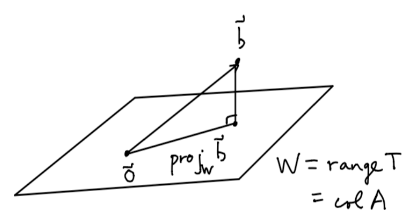
Solution:



There's actually a smart way to find the least squares solutions that doesn't require an orthonormal basis for *Col A*. To understand it fully depends on concepts we talked about last week (and was one reason we spent a long time talking about orthogonal complements to subspaces). As a further result, it will turn out that one can compute projections onto a subspace with elementary matrix operations and without first constructing an orthonormal basis for the subspace !!! Consider the following chain of equivalent conditions on $\underline{\mathbf{x}}$:

$$A\,\underline{\mathbf{x}} = proj_{Col\,A}\,\underline{\mathbf{b}}$$

$$\underline{\mathbf{z}} = \underline{\mathbf{b}} - A\,\underline{\mathbf{x}} \in (Col\,A)^{\perp} = Nul\,A^T$$



$$A^T(\underline{\mathbf{b}} - A\,\underline{\mathbf{x}}) = \underline{\mathbf{0}}$$

$$A^T\underline{\mathbf{b}} - A^TA\,\underline{\mathbf{x}} = \underline{\mathbf{0}}$$

$$A^TA\,\underline{\mathbf{x}} = A^T\underline{\mathbf{b}}\,.$$

This last equation will always be consistent because projections exist. And if the columns of *A* are linearly independent the solutions to the top equation, and hence the final equation, will be unique. So the matrix A^TA will be invertible in that case. The final matrix equation is called the *normal equation* for least squares solutions.

<u>Exercise 2</u> Re-do Exercise 1 using the normal equation, i.e find the least squares solution $\hat{\mathbf{x}}$ to

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}.$$

And then note that $A\,\hat{\mathbf{x}}$ is $proj_{Col\,A}\underline{\mathbf{b}}$, i.e. you found the projection of $[3\ 3\ 3]^T$ without ever finding and using an ortho-normal basis!!!

Matlab assumes you want the least squares solution when you hand it an inconsistent system. This is because, as we'll discuss tomorrow, whenever an applied mathematician, engineer or scientist is using a finite-dimensional linear model for an actual experimental process, there is almost no chance that the actual data will fit the model exactly.

script:

```
% matlab assumes you want least squares solutions to inconsistent systems
A=[1 2; 0 1; 1 0]
b=[3;3;3]
aug=[A,b]
rref(aug)  %system is inconsistent (last column is pivot column)
x=linsolve(A,b)  %least squares solution
```

executes to produce:

```
A =

     1     2
     0     1
     1     0


b =

     3
     3
     3


aug =

     1     2     3
     0     1     3
     1     0     3


ans =

     1     0     0
     0     1     0
     0     0     1


x =

    2.0000
    1.0000
```

$\underline{\text{Exercise 3}}$ In the case that $A^T A$ is invertible we may take the normal equation for finding the least squares solution to $A\,\underline{\mathbf{x}} = \underline{\mathbf{b}}$ and find $A\,\hat{\underline{\mathbf{x}}} = proj_{Col\,A}\underline{\mathbf{b}}$ directly:

$$A^T A\,\underline{\mathbf{x}} = A^T\underline{\mathbf{b}}$$

$$\underline{\mathbf{x}} = \left(A^T A\right)^{-1} A^T\underline{\mathbf{b}}$$

$$proj_{Col\,A}\underline{\mathbf{b}} = A\,\underline{\mathbf{x}} = A\left(A^T A\right)^{-1} A^T\underline{\mathbf{b}}.$$

Verify for the third time that for $W = span\left\{\begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right\}$, $proj_W\begin{bmatrix}3\\3\\3\end{bmatrix} = \begin{bmatrix}4\\1\\2\end{bmatrix}$ by "plug and chug".

Wed Nov 21

- 6.6 Fitting data to "linear" models.

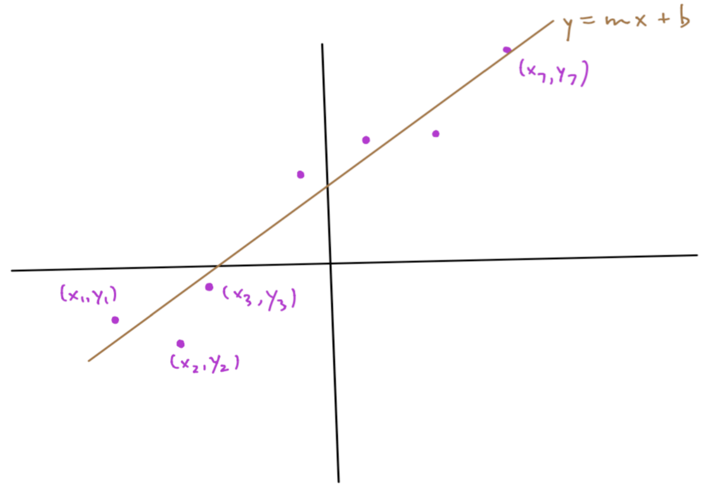
Announcements:

Warm-up Exercise:

Applications of least-squares to data fitting.

- Find the best line formula $y = m\,x + b$ to fit n data points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. We seek $\begin{bmatrix} m \\ b \end{bmatrix}$ so that

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = m \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + b \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$



In matrix form, find $\begin{bmatrix} m \\ b \end{bmatrix}$ so that

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}. \qquad A \begin{bmatrix} m \\ b \end{bmatrix} = \mathbf{y}\,.$$

There is no exact solution unless all the data points are actually on a single line!

<u>Least squares solution</u>:

$$A^T A \begin{bmatrix} m \\ b \end{bmatrix} = A^T \mathbf{y}\,.$$

$$A^T A \begin{bmatrix} m \\ b \end{bmatrix} = A^T \underline{\mathbf{y}}$$

As long as the columns of A are linearly independent (i.e.at least two different values for x_j) there is a unique solution $[m, b]^T$. Furthermore, you are actually solving

$$A \begin{bmatrix} m \\ b \end{bmatrix} = proj_W \underline{\mathbf{y}}$$

where

$$W = span\left\{ \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ : \\ 1 \end{bmatrix} \right\},$$

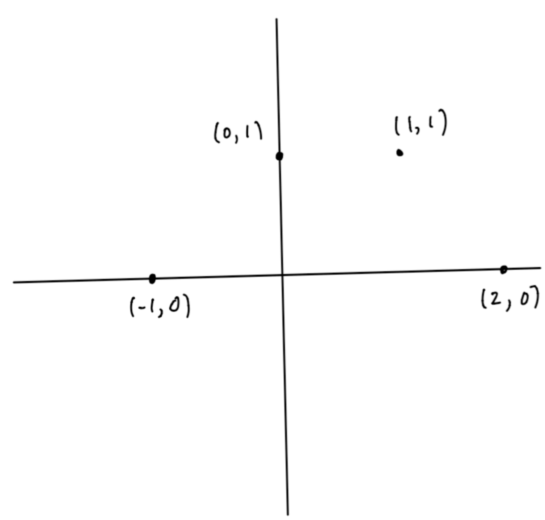
so

$$\left\| \begin{bmatrix} y_1 \\ y_2 \\ : \\ y_n \end{bmatrix} - m \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix} - b \begin{bmatrix} 1 \\ 1 \\ : \\ 1 \end{bmatrix} \right\|^2$$

is as small as possible. In other words, you've minimized the sum of the *squared vertical deviations* from points on the line to the data points,

$$\sum_{i=1}^{n} (y_i - mx_i - b_i)^2.$$

<u>Exercise 1</u> Find the least squares line fit for the 4 data points $\{(-1, 0), (0, 1), (1, 1), (2, 0)\}$. Sketch.



Example 2 Find the best quadratic fit to the same four data points. This is still a "linear" model!! In other words, we're looking for the best quadratic function

$$p(x) = c_0 + c_1 x + c_2 x^2$$

to fit to the four data points

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

We want to solve

$$c_0 \begin{bmatrix} 1 \\ 1 \\ : \\ 1 \end{bmatrix} + c_1 \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix} + c_2 \begin{bmatrix} x_1^2 \\ x_2^2 \\ : \\ x_n^2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ : \\ y_n \end{bmatrix}.$$

For our example this is the system

$$c_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + c_1 \begin{bmatrix} -1 \\ 0 \\ 1 \\ 2 \end{bmatrix} + c_2 \begin{bmatrix} 1 \\ 0 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

with Matlab and the least squares normal equation (which matlab will apply automatically as well), we can let technology solve

$$A^T A \, \underline{\mathbf{c}} = A^T \underline{\boldsymbol{b}}$$

although this problem is small enough that one could also work it by hand.

This Matlab script

```
%in the following example the least square solution is
%actually an exact solution.
C=[1,-1,1; 1,0,0; 1,1,1; 1,2,4]
b2=[0;1;1;0]
c=linsolve(C,b2)   %least squares solution
c2=(transpose(C)*C)^(-1)*transpose(C)*b2  %also least squares solution
rref([C,b2])  %system was consistent
```

yields

```
C =

     1    -1     1
     1     0     0
     1     1     1
     1     2     4


b2 =

     0
     1
     1
     0
```

```
c =

    1.0000
    0.5000
   -0.5000


c2 =

    1.0000
    0.5000
   -0.5000


ans =

    1.0000         0         0    1.0000
         0    1.0000         0    0.5000
         0         0    1.0000   -0.5000
         0         0         0         0
```

For a plot, this script:

```
%plots...
t=linspace(-1.5,2.5,100) %left endpt, right endpt, numpoints
% "t" above is a vector 100 equally spaced numbers between -1.5 and 2.5
% the definition below is for an equally sized vector containing the
% parabolic approximation.  we use "t." to extract a scalar value from the
% vector
y=c(1)+c(2)*t+c(3)*t.*t
lucky1=plot(t,y,'black')
title('best parabolic fit')
xlabel('t')  %horizontal variable label
ylabel('y')   %vertical variable label
hold on   % the "hold" command lets us combine plots into one display
scatter([-1,0,1,2],[0,1,1,0],'red')
hold off
```

produces this display: