CS/ECE/ME 532 - Fall 2017 - Rebecca Willett

Homework 2: Norms and Least Squares

Due Wednesday, September 27th, 2017

Roumen Guha

1. Define the mapping $\Phi : \mathbb{R}^n \to \mathbb{R}$ as

$$\Phi(x) = \max_{1 \leq i \leq n} i|x_i|$$

Is $\Phi(x)$ a norm?

Solution: For some function $p(x)$ to be a valid norm, it must satisfy the following conditions:

(i) $p(av) = |a|p(v), a \in \mathbb{R}$: is absolutely scalable

(ii) $p(u + v) \leq p(u) + p(v)$: satisfies the triangle inequality

(iii) $p(v) \geq 0$: satisfies non-negativity

(iv) if $p(v) = 0$, then $v = 0$ is the zero vector: is point-separating

It is easy to see that $\Phi(x)$ is absolutely scalable, and always is always positive except when $x_i = 0$, which satisfies conditions (i), (iii) and (iv). Therefore, the only condition left to prove is that $\Phi(x)$ satisfies the triangle inequality:

If we try $\Phi(u + v)$, we know this will be less than or equal to $\Phi(u) + \Phi(v)$ because of the order of operations between the magnitude operation and addition operation, for example when elements of u or v are negative. Therefore, it is necessarily true that $\Phi(u + v) \leq \Phi(u) + \Phi(v)$. (I did a few example calculations before being able to answer this with some certainty)

2. Equivalence of norms. For each case below find positive constants a and b (possibly different in each case) so that for every $x \in \mathbb{R}^n$

(i) $a\|x|_1 \leq \|x\|_2 \leq b\|x|_1$ (HINT: Use Cauchy-Schwarz inequality, which states that for any vectors u, v, we have: $u^T v \leq \|u\|_2\|v\|_2$ and a vector of all ones.)

Solution: $a = 0, b = 1$ would be a pair of values that would suffice.

(ii) $a\|x|_1 \leq \|x\|_\infty \leq b\|x|_1$

Solution: $a = 0, b = 1$ would be the same pair of values that would suffice.

3. Let

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

a) What is the rank of A?

In [25]:
```
A = [1 1 1
     1 1 0
     1 0 0]

rank(A)
```

Out[25]: 3

b) Suppose that $y = Ax$. Derive an explicit formula for x in terms of y.

Solution: x = A\y

4. Answer the following questions. Make sure to explain your reasoning.

a) Are the columns of the following matrix linearly independent?

$$A = \begin{bmatrix} +0.92 & +0.92 \\ -0.92 & +0.92 \\ +0.92 & -0.92 \\ -0.92 & -0.92 \end{bmatrix}$$

In [29]:
```
A = [0.92 0.92
     -0.92 0.92
     0.92 -0.92
     -0.92 -0.92]

rank(A)
# Yes, the columns are linearly independent.
```

Out[29]: 2

b) Are the columns of the following matrix linearly independent?

$$A = \begin{bmatrix} +1 & +1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & -1 \end{bmatrix}$$

In [30]:
```
A = [ 1  1  1
     -1  1 -1
      1 -1 -1]

rank(A)
# Yes, the columns are linearly independent.
```

Out[30]: 3

c) Are the columns of the following matrix linearly independent?

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 3 & 4 & 5 \\ 5 & 6 & 8 \end{bmatrix}$$

In [31]:
```
A = [1 2 2
     3 4 5
     5 6 8]

rank(A)
# No, I was able to obtain x_1 from the equation x_3 - 0.5*x_2
```

Out[31]: 2

d) What is the rank of the following matrix?

$$A = \begin{bmatrix} +5 & 2 \\ -5 & +2 \\ +5 & -2 \end{bmatrix}$$

In [33]:
```
A = [5 2
     -5 2
     5 -2]

rank(A)
```

Out[33]: 2

e) Suppose the matrix in part **(c)** is used in the least squares optimization $\min_x \|b - Ax\|_2$. Does a unique solution exist?

Solution: No unique solution exists, because you only need two of the three column vectors to form a basis, as the three column vectors are not linearly independent.

5. Consider the following matrix and vector:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

a) Find the solution $\hat{x}$ to $\min_x \|b - Ax\|_2$.

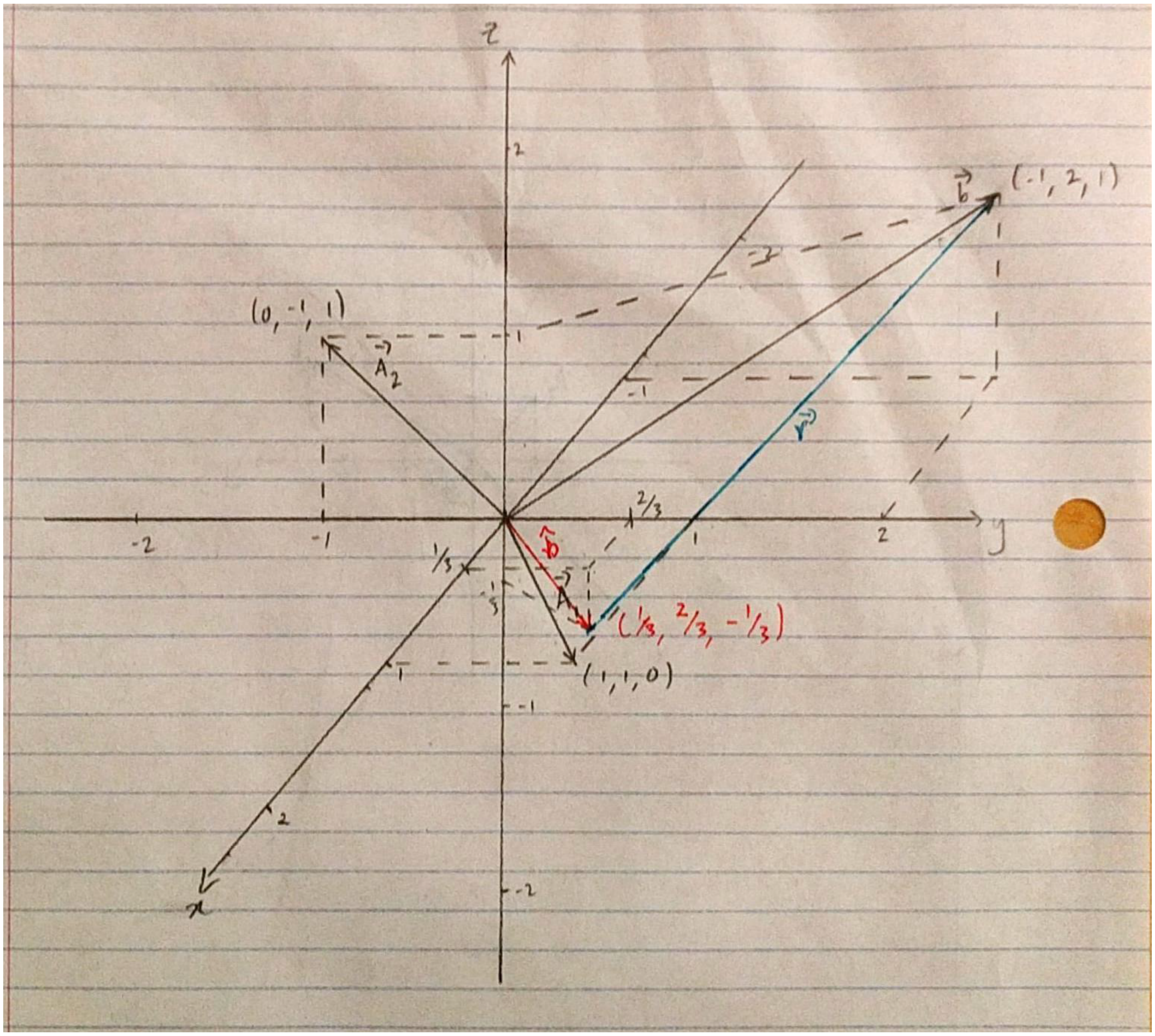
In [43]:
```
A = [1  0
     1 -1
     0  1]

b = [-1
      2
      1]

x_opt = inv(transpose(A)*A)*transpose(A)*b
```

Out[43]:
```
2-element Array{Float64,1}:
  0.333333
 -0.333333
```

b) Make a sketch of the geometry of this particular problem in $\mathbb{R}^3$, showing the columns of A, the plane they span, the target vector b, the residual vector and the solution $\hat{b} = A\hat{x}$.

In [44]:
```
b_opt = A * x_opt
```

Out[44]:
```
3-element Array{Float64,1}:
  0.333333
  0.666667
 -0.333333
```



Sorry if the image isn't clear enough; the vector A_1 lies in the xy-plane which from this perspective lies over the $\hat{b}$ vector, obscuring it.

Please have mercy; I spent the better part of 30 minutes drawing this as clearly as I could.

6. Polynomial fitting. Suppose we observe pairs of points (a_i, b_i), $i = 1, \dots, m$. Imagine these points are measurements from a scientific experiment. The variables a_i are the experimental conditions and the b_i correspond to the measured response in each condition. Suppose we wish to fit a degree $d < m$ polynomial to these data. In other words, we want to find the coefficients of a degree d polynomial p so that $p(a_i) \approx b_i$ for $i = 1, 2, \dots, m$. We will set this up as a least-squares problem.

a) Suppose p is a degree d polynomial. Write the general expression for $p(a_i) = b_i$.

Solution: A polynomial p with degree d is of the form: $p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_d x^d$, where c_i are the coefficients of the degree terms.

Therefore, the general expression of $p(a_i) = b_i$ is

$$p(a_i) = c_0 + c_1 a_i + c_2 a_i^2 + \cdots + c_d a_i^d = b_i$$

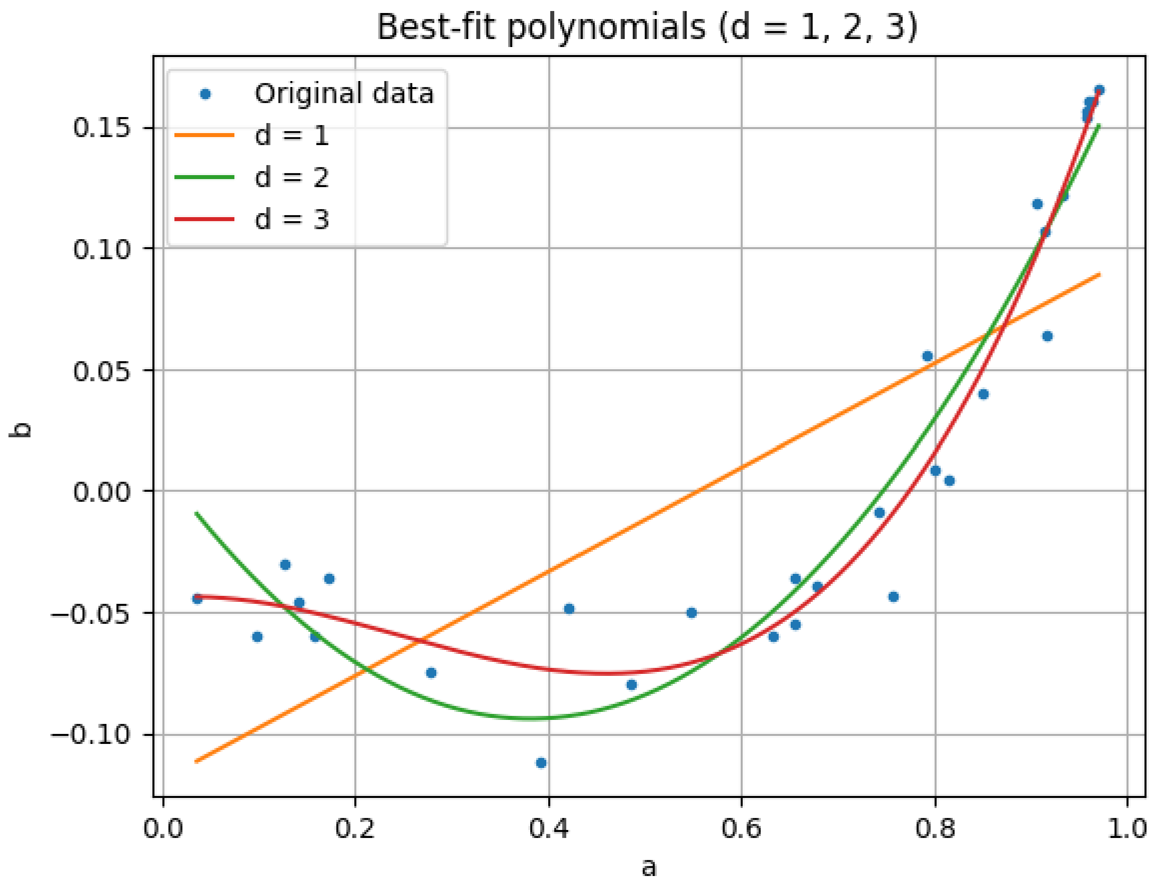
b) Express the $i = 1, \ldots, m$ equations as a system in matrix form $Ax = b$. Specifically, what is the form/structure of A in terms of the given a_i.

$$\begin{bmatrix} 1 & a_1 & a_1^2 & \ldots & a_1^d \\ 1 & a_2 & a_2^2 & \ldots & a_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_m & a_m^2 & \ldots & a_m^d \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_d \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix}$$

c) Write a Matlab or Python script to find the least-squares fit to the $m = 30$ data points in $\texttt{polydata.mat}$. Plot the points and the polynomial fits for $d = 1, 2, 3$.

Solution: This was a hard problem!

In [114]:
```
using MAT, PyPlot

polydata = matread("polydata.mat")
b = polydata["b"]
a = polydata["a"]

D = 3 # Number of degrees desired
N = 10000 # Number of points to interpolate
M = min(length(a), length(b)) # Only consider the minimum number of datapoints of
z = linspace(minimum(a), maximum(a), N) # N equally spaced points to evaluate the
y = zeros(max_d, N) # Initializing the storage vector of interpolated values

for d = 1:D

    A = zeros(m, d+1) # generate the Vandermonde matrix A
    for m = 1:M
        for k = 1:d+1
            A[m,k] = a[m]^(k-1)
        end
    end

    x = A\b # solve for the optimal x using least squares

    for n = 1:N # generate the max_d best-fit lines
        for k = 1:d+1
            y[d,n] = y[d,n] + x[k]*z[n]^(k-1)
        end
    end

end

plot(a, b, ".")
for d = 1:D
    plot(z, y[d,:])
end
grid()
title("Best-fit polynomials (d = 1, 2, 3)")
xlabel("a")
ylabel("b")
legend(["Original data", "d = 1", "d = 2", "d = 3"]);
```



7. Recall the cereal calorie prediction problem discussed in class. The data matrix for this problem is

$$A = \begin{bmatrix} 25 & 0 & 1 \\ 20 & 1 & 2 \\ 40 & 1 & 6 \end{bmatrix}$$

Each row contains the grams/serving of carbohydrates, fat, and protein, and each row corresponds to a different cereal *(Frosted Flakes, Grape-Nuts, Teenage Mutant Ninja Turtles)*. The total calories for each cereal are

$$b = \begin{bmatrix} 110 \\ 110 \\ 210 \end{bmatrix}$$

a) Write a small program (e.g., in Matlab or Python) that solves the system of equations $Ax = b$. Recall the solution x gives the calories/gram of carbohydrate, fat, or protein. What is the solution?

In [115]:
```
A = [25 0 1
     20 1 2
     40 1 6]

b = [110
     110
     210]

x = A\b
```

Out[115]:
```
3-element Array{Float64,1}:
  4.25
 17.5
  3.75
```

b) The solution may not agree with the known calories/gram, which are 4 for carbs, 9 for fat and 4 for protein. We suspect this may be due to rounding the grams to integers, especially the fat grams. Assuming the true value for calories/gram is

$$x^* = \begin{bmatrix} 4 \\ 9 \\ 4 \end{bmatrix}$$

and that the total calories, grams of carbs, and grams of protein are correctly reported above, determine the “correct” grams of fat in each cereal.

Solution: With this new information, the equation to solve becomes:

$$\begin{bmatrix} 25 & f_1 & 1 \\ 20 & f_2 & 2 \\ 40 & f_3 & 6 \end{bmatrix} \begin{bmatrix} 4 \\ 9 \\ 4 \end{bmatrix} = \begin{bmatrix} 110 \\ 110 \\ 210 \end{bmatrix}$$

So we have 3 new equations:

(1) $25(4) + f_1(9) + 1(4) = 110 \implies f_1 = \frac{6}{9}$

(2) $20(4) + f_2(9) + 2(4) = 110 \implies f_2 = \frac{22}{9}$

(3) $40(4) + f_3(9) + 6(4) = 210 \implies f_3 = \frac{26}{9}$

c) Now suppose that we predict total calories using a more refined breakdown of carbohydrates, into total carbohydrates, complex carbohydrates and sugars (simple carbs). So now we will have 5 features to predict calories (the three carb features + fat and protein). So let’s suppose we measure the grams of these features in 5 different cereals to obtain this data matrix

$$A = \begin{bmatrix} 25 & 15 & 10 & 0 & 1 \\ 20 & 12 & 8 & 1 & 2 \\ 40 & 30 & 10 & 1 & 6 \\ 30 & 15 & 15 & 0 & 3 \\ 35 & 20 & 15 & 2 & 4 \end{bmatrix}$$

and the total calories in each cereal

$$b = \begin{bmatrix} 104 \\ 97 \\ 193 \\ 132 \\ 174 \end{bmatrix}$$

Can you solve $Ax = b$? Carefully examine the situation in this case. Is there a solution that agrees with the true calories/gram?

In [116]:
```
A = [25 15 10 0 1
     20 12  8 1 2
     40 30 10 1 6
     30 15 15 0 3
     35 20 15 2 4]

b = [104
      97
     193
     132
     174]

rank(A)
```

Out[116]:
```
4
```

Solution: Since the $rank(A) = 4$, we know that there won't be a unique solution to this problem.

However, we can find a matrix that contains only linearly independent column vectors of A. By inspection, we can see that A_1 (the first column vector of A) is the sum of A_2 and A_3, and so we can discard A_1. (This makes sense, by the way, since A_1 represented total carbohydrates, the sum of A_2 and A_3)

If we perform least-squares with this new matrix B:

In [117]:
```
B = [15 10 0 1
     12  8 1 2
     30 10 1 6
     15 15 0 3
     20 15 2 4]

y = B\b
```

Out[117]:
```
4-element Array{Float64,1}:
 4.0
 4.0
 9.0
 4.0
```

We see that we can obtain an exact solution, one that agrees with our known exact solution. But it is just one possible solution to our original problem $Ax = b$.