Math 2270-6

Notes of 11/13/2019

Announcement

- **Undergraduate Colloquium**

  today, 12:45-1:45, LCB 225

- Ryleigh Moore will present:

  Adventures in the Arctic

  Mathematics Ph.D. student, Ryleigh Moore, was one of three American graduate students invited to participate in the Multidisciplinary drifting Observatory for the Study of Arctic Climate (MOSAiC) expedition out of Tromsoe, Norway, from September 20 - October 28, 2019. The flagship German icebreaker, RV Polarstern, will be frozen in ice and drift for a full year through the Central Arctic following in the footsteps of an earlier 19th century expedition under Norwegian explorer, Fridtjof Nansen. MOSAiC will mark the first time a modern research icebreaker will study near the North Pole throughout the polar winter; it is hailed as the largest Central Arctic expedition ever, with 19 countries, over 600 people, and a budget exceeding 155 million US dollars.

  In this talk, Ryleigh will discuss the science goals of MOSAiC and her experiences while on the Russian research vessel, Akademik Fedorov. She will discuss life on a research vessel, how the expedition identified ice floes that would be used for instrument deployments, and her role of leading the installation of three seasonal ice mass balance (SIMB3) buoys in the Central Arctic.

More on Least Squares

- Suppose we have the overdetermined linear system
$$A\mathbf{x} = \mathbf{b}$$
where A is $m \times n$ and $m > n$.

- We discussed solving instead the Least Squares Problem
$$\|A\mathbf{x} - \mathbf{b}\| = \min \tag{1}$$

Note that any solution $\mathbf{x}$ of (1) is also a solution of the problem

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \min \tag{2}$$

- We saw that we can find $\mathbf{x}$ by solving the **normal equations**
$$A^T A\mathbf{x} = A^T\mathbf{b}.$$

- Moreover, we saw that

0. Geometrically we get $A\mathbf{x}$ by projecting $\mathbf{b}$ into the column space of A and we get the normal equations by observing that $A\mathbf{x} - \mathbf{b}$ is in the orthogonal complement of the column space of A.

1. $A\mathbf{x}$ exists and is unique for every $\mathbf{b}$.

2. $\mathbf{x}$ is unique if the columns of A are linearly independent.

3. $A^T A$ is invertible if and only if the the columns of A are linearly independent.

4. In that case we get

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

although we would not normally actually compute the inverse of $A^T A$

- Example: Suppose you have a some data (x_i, y_i), $i = 1, 2, \ldots, n$ and you have reason to believe that the answer is some constant plus a 2π-periodic function that looks something like a sine curve. In other words,

$$y_i \approx f(x_i) = \alpha + \beta \sin x_i + \gamma \cos x_i.$$

- In equation form this is

$$A\mathbf{x} = \mathbf{b}$$

where

$$A = \begin{bmatrix} 1 & \sin x_1 & \cos x_1 \\ 1 & \sin x_2 & \cos x_2 \\ \vdots & \vdots & \vdots \\ 1 & \sin x_n & \cos x_n \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- This is an overdetermined linear system. As we discussed in class yesterday, its least squares solution $\mathbf{x}$ is given by the normal equations

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

where, in this case, with all sums running from 1 to n,

$$A^T A = \begin{bmatrix} \sum 1 & \sum \sin x_i & \sum \cos x_i \\ \sum \sin x_i & \sum \sin^2 x_i & \sum \sin x_i \cos x_i \\ \sum \cos x_i & \sum \cos x_i \sin x_i & \sum \cos^2 x_i \end{bmatrix}$$

and

$$A^T\mathbf{b} = \begin{bmatrix} \sum y_i \\ \sum y_i \sin x_i \\ \sum y_i \cos x_i \end{bmatrix}$$

- Example: Temperatures in Salt Lake City.
- You'd expect (low and high daily) temperatures T(t) in SLC to follow (roughly) some kind of sin curve such as

$$Y(t) = a + b \sin \frac{2\pi t}{12} + c \cos t \frac{2\pi t}{12} \qquad (3)$$

where t is the time of the year, measured in months. a is the mean temperature over the whole year. b and c are suitable coefficients. I found the following data at

`http://www.rssweather.com/climate/Utah/Salt%20Lake%20City/`

Month	Low	High
Jan	21.F	37.0F
Feb	25.5F	43.F
Mar	33.4F	52.8F
Apr	39.0F	60.9F
May	46.9F	70.6F
Jun	55.8F	82.2F
Jul	63.4F	90.6F
Aug	62.4F	88.7F
Sept	52.4F	77.6F
Oct	41.0F	64.0F
Nov	30.4F	48.7F
Dec	22.4F	38.0F

- Suppose you want to find the Least Squares approximation of the form (3) that best approximates the data.

- It would be unreasonable to do this problem by hand. Numerous facilities (e.g., matlab, maple, wolfram alpha, any number of programming languages) exist to get a computer to do the necessary calculations.
- The following matlab code and its outputs show the results of doing a Least Squares Approximation.

```
% approximate low and high temperatures in SLC by sin and cos
% curve
% see http://www.rssweather.com/climate/Utah/Salt%20Lake%2

low=[21.3;25.5;33.4;39.0;46.9;55.8;63.4;62.4;52.4;41.0;30.

pi = 2.0*acos(0.0);

for i =1:12
  O(i,1) = 1.0;
  S(i,1) = sin(2*i/12*pi);
  C(i,1) = cos(2*i/12*pi);
end

A=[O'*O,O'*S,O'*C;S'*O,S'*S,S'*C;C'*O,C'*S,C'*C]
B=[O'*low;S'*low;C'*low];

X= A\B

for i = 1:12
  app = X(1) + X(2)*S(i,1) + X(3)*C(i,1);
  err = low(i)-app;
  res = [i,low(i), app, err];
  disp(res)
end
```

```
A =

   12.0000   -0.0000   -0.0000
   -0.0000    6.0000    0.0000
   -0.0000    0.0000    6.0000

X =

   41.1583
  -10.9147
  -16.9332

    1.0000   21.3000   21.0364    0.2636
    2.0000   25.5000   23.2393    2.2607
    3.0000   33.4000   30.2436    3.1564
    4.0000   39.0000   40.1725   -1.1725
    5.0000   46.9000   50.3655   -3.4655
    6.0000   55.8000   58.0915   -2.2915
    7.0000   63.4000   61.2803    2.1197
    8.0000   62.4000   59.0774    3.3226
    9.0000   52.4000   52.0731    0.3269
   10.0000   41.0000   42.1442   -1.1442
   11.0000   30.4000   31.9511   -1.5511
   12.0000   22.4000   24.2252   -1.8252
```

```
% approximate low and high temperatures in SLC by sin and cos
% curve
% see http://www.rssweather.com/climate/Utah/Salt%20Lake%2

high=[37.0;43.4;52.8;60.9;70.6;82.2;90.6;88.7;77.6;64.0;48

pi = 2.0*acos(0.0);

for i =1:12
  O(i,1) = 1.0;
  S(i,1) = sin(i/6*pi);
  C(i,1) = cos(i/6*pi);
end

A=[O'*O,O'*S,O'*C;S'*O,S'*S,S'*C;C'*O,C'*S,C'*C]
B=[O'*high;S'*high;C'*high];

X= A\B

for i = 1:12
  app = X(1) + X(2)*S(i,1) + X(3)*C(i,1);
  err = high(i)-app;
  res = [i,high(i), app, err];
  disp(res)
end
```

```

A =

   12.0000   -0.0000   -0.0000
   -0.0000    6.0000    0.0000
   -0.0000    0.0000    6.0000


X =

   62.8750
  -13.7609
  -21.7808

    1.0000   37.0000   37.1318   -0.1318
    2.0000   43.4000   40.0673    3.3327
    3.0000   52.8000   49.1141    3.6859
    4.0000   60.9000   61.8481   -0.9481
    5.0000   70.6000   74.8573   -4.2573
    6.0000   82.2000   84.6558   -2.4558
    7.0000   90.6000   88.6182    1.9818
    8.0000   88.7000   85.6827    3.0173
    9.0000   77.6000   76.6359    0.9641
   10.0000   64.0000   63.9019    0.0981
   11.0000   48.7000   50.8927   -2.1927
   12.0000   38.0000   41.0942   -3.0942

```

Perhaps more impressive is a graphic display of our data and its approximations:

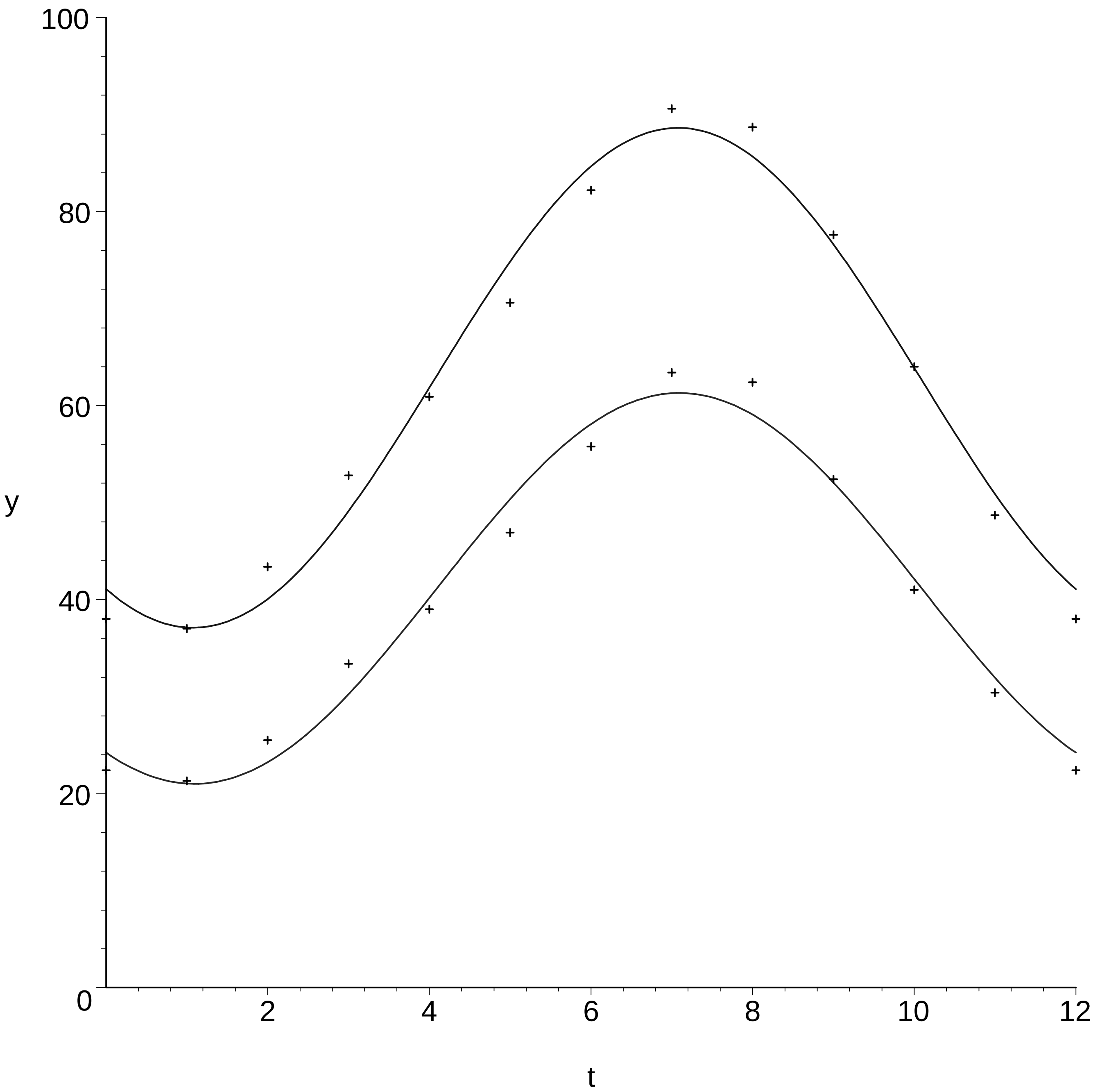

Figure 1. Temperature in Salt Lake City.

- As we've seen, things get more convenient if we have matrices with orthonormal columns. Let's first look at the case that A itself have orthonormal columns.

- Next let's see what happens if we have a QR factorization of A.
- Suppose
$$A = QR$$
where R is upper triangular and Q has orthonormal columns.