PS 9 - Least squares and curve-fitting

Math087 - George McNinch

due 2025-04-13

1. *Mass estimation* Suppose you are assembling components for a piece of heavy machinery. The final product involves 5 components, `a`, `b`, `c`, `d`, and `e`. You require estimates for the total mass of each of the components.

   You have take measurements of the mass of *groups* of the components:

   ```
   mass_estimates = {('a', 'b', 'c'): 551.03,
                     ('a', 'b', 'd'): 353.19,
                     ('a', 'b', 'e'): 574.36,
                     ('a', 'c', 'd'): 506.1,
                     ('a', 'c', 'e'): 724.92,
                     ('a', 'd', 'e'): 531.9,
                     ('b', 'c', 'd'): 478.21,
                     ('b', 'c', 'e'): 701.98,
                     ('b', 'd', 'e'): 504.75,
                     ('c', 'd', 'e'): 653.07
                     }
   ```

   For example, the equality

   ```
   mass_estimates[('a','b','c')] == 551.03
   ```

   means that the sum of the masses of components `a`, `b`, and `c` was measured to be `551.03`.

   Let's write `ma`,~mb~,~mc~,~md~,~me~ for the masses of the components.

   The estimates amount to linear equations in the variables `ma`, `mb`, … e.g. we have

   ```
   ma + mb + mc == 551.03
   ma + mb + md == 353.19
   ma + mb + me == 576.36
   # ... etc ...
   ```

   Write

   ```
   b = array([551.03, 353.19, 574.36, 506.1, 724.92,
              531.9, 478.21, 701.98, 504.75, 653.07])
   ```

a. Find an estimate for the vector

```python
x = array([ ma, mb, mc, md, me ])
```

which is the *least squares solution* to the equation

```python
M @ x = b
```

Put another way, find the vector `x` as above such that the length of the vector `python b - M @ x` is minimized.

b. Using your *least-squares solution* from a., estimate the total mass of the assembled machine; i.e. estimate the sum

`ma + mb + mc + md + me`

c. Explain why you know that the equation

```python
M @ x == b
```

has no solution `x`.

2. *Curve fitting*

An object was catapulted vertically into the air from atop a tall building – it traveled upwards, then fell straight to the ground.

The object was equipped with an altimeter and a recorder, so you have some data about its height above ground at various points during its flight.

Neglecting air resistance, we know from physics that the height above ground of the object is a *quadratic* function

$$f(t) = \alpha t^2 + \beta t + \delta.$$

The following dictionary `height_esimates` indicates the height `height_estimates[t]` in meters above ground of the object after `t` seconds:

```python
height_estimates = { 0.0: 199.6,
                     0.25: 202.96,
                     0.5: 207.23,
                     0.75: 208.29,
                     1.0: 207.47,
                     1.25: 203.96,
                     1.5: 199.18,
                     1.75: 202.91,
                     2.0: 204.29,
                     2.25: 196.31,
                     2.5: 195.71,
                     2.75: 187.89,
                     3.0: 187.61,
                     3.25: 177.12,
                     3.5: 171.07,
                     3.75: 171.89,
                     4.0: 158.68,
                     4.25: 152.64,
                     4.5: 146.7,
                     4.75: 138.52,
                     5.0: 127.27,
                     5.25: 122.38,
                     5.5: 103.97,
                     5.75: 96.91,
                     6.0: 83.08,
                     6.25: 67.34,
                     6.5: 55.75,
                     6.75: 45.42,
                     7.0: 25.33,
                     7.25: 14.67,
                     7.5: -1.45
                     }
```

We can see a graph of these values using the following code:

```python
import matplotlib.pyplot as plt

def plot_data(x,y):
  fig, ax = plt.subplots(figsize=(12,6))
  return ax.plot(x,y,"o")

x1 = height_estimates.keys()
y1 = list(height_estimates.values())
plot_data(x1,y1)
```

a. Find the values of the coefficients α, β, δ so that

$$f(t) = \alpha t^2 + \beta t + \delta.$$

is the *best fit* for the available data. Using this best-fit, what is the initial height above ground of the object (i.e. the height of the tall building?) What is the initial velocity of the object?

b. Produce via `matplotlib` a graph of the function `f` you obtained as the best-fit in a., superimposed on the data plot above. (You can mimic the code from the lecture notebook).

c. Since we know that acceleration due to gravity is (approximately) -9.8 m/s near the Earth's surface, we actually know that the function f should be given by

$$(\clubsuit) \quad f(t) = \frac{-9.8}{2}t^2 + \beta t + \delta.$$

Now find the values of the coefficients β, δ so that $(\clubsuit)$ is the *best fit* for the available data.

Are your estimates for the height of the building or the initial velocity affected by this simplification?

d. Produce via `matplotlib` a graph of the function `f` you obtained as the best-fit in c., superimposed on the data plot above. (Again, you can use the code from the lecture notebook).