Least squares fitting

Prof. Ramin Zabih

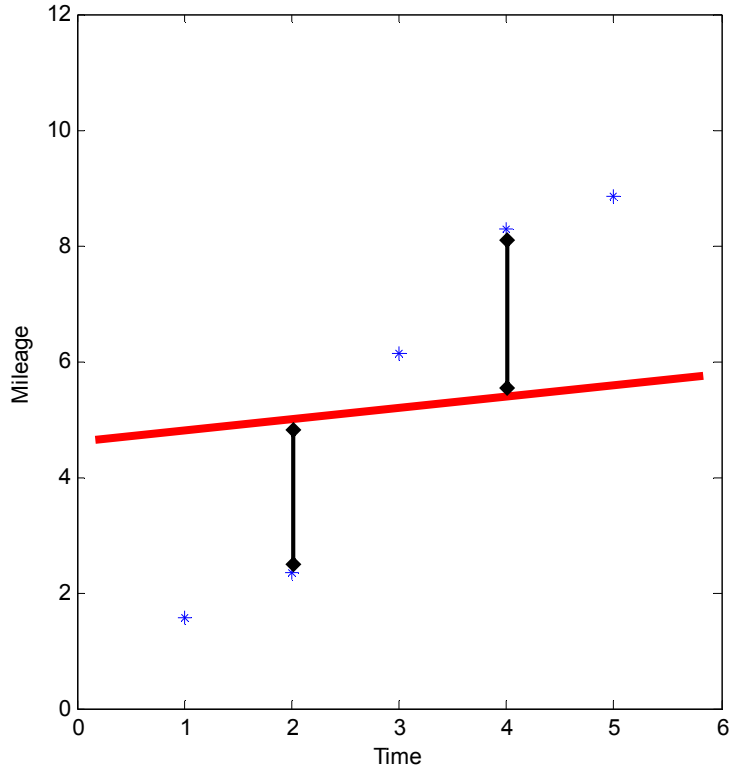
http://cs100r.cs.cornell.edu

Cornell University
Computer Science

Administrivia

- Assignment 4 out, due next Friday
- Quiz 6 will be Thursday Oct 25

LS fitting and residuals



$$\text{Good}(m,b) = -\sum_i \left[y_i - (mx_i + b) \right]^2$$

How to find the LS line fit

- LS-hillclimbing algorithm:
- Start with some initial (m,b)
- Make a small change to (m,b)
  - See if the figure of merit Good(m,b) improves
    - Smaller "sum of squared errors"
  - If so, make that your new (m,b) and continue
- Done small changes don't improve (m,b)

Optimization methods

- This is a simple example of an optimization method
  - There are some possible answers (i.e., lines) and a way to rate each answer (i.e., Good)
  - We want the best answer
    - Equivalently, minimize sum of squared errors
- The algorithm we just described is sometimes called "hill climbing"
  - There are many smart variants, including "gradient descent", that we won't cover
    - Need to leave a few things for the rest of CS…

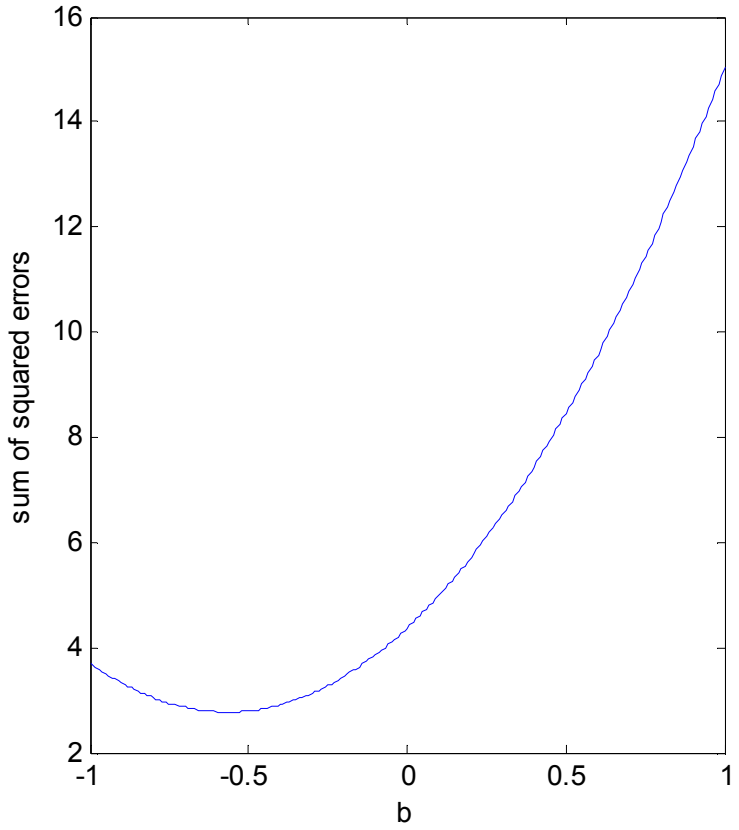
Why least squares?

- As many of you observed, there are lots of sensible ways to compute a figure of merit
- Why do we always use least squares?
- This is too deep a topic to cover completely in a freshman course
  - But we will point out two things that are special about least squares
  - The full story probably needs to wait for graduate-level courses
    - Or you can ask RDZ some evening…

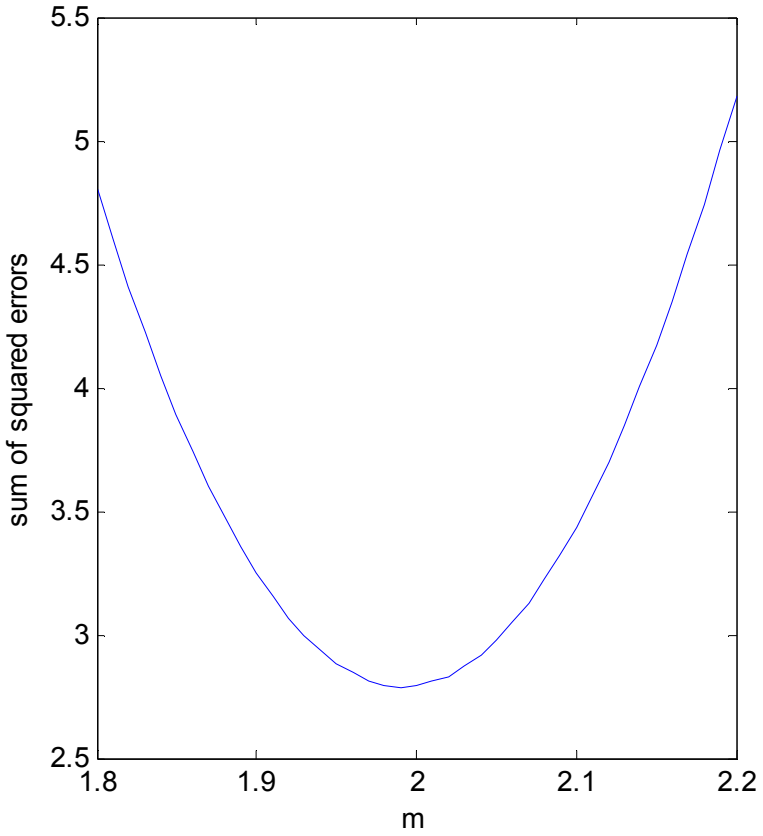
The LS-hillclimbing method

- The method we have described for finding the best (m,b) is very general
  - But it isn't very smart
- It's only slightly smarter than snailsort
  - Snailsort: guess an answer; prove it correct
  - LS hillclimbing: guess an initial answer; try to improve your guess
  - It's not at all clear how we could prove that we had the optimal (m,b)!
    - Only that small changes won't improve it

Fix m=2, change b

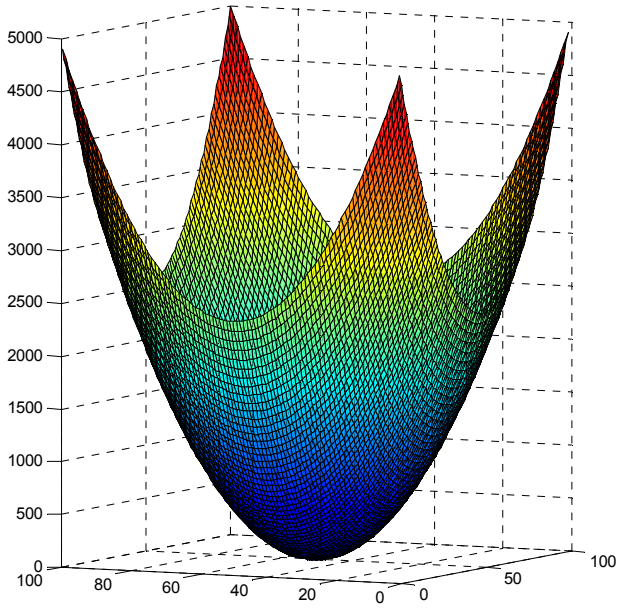


Fix b=-0.5, change m



Some error functions are easy

- You will notice that the squared error has a **single** minimum value
  - The plots I just showed you demonstrate this is true in 1D
    - But it's true in 2D also

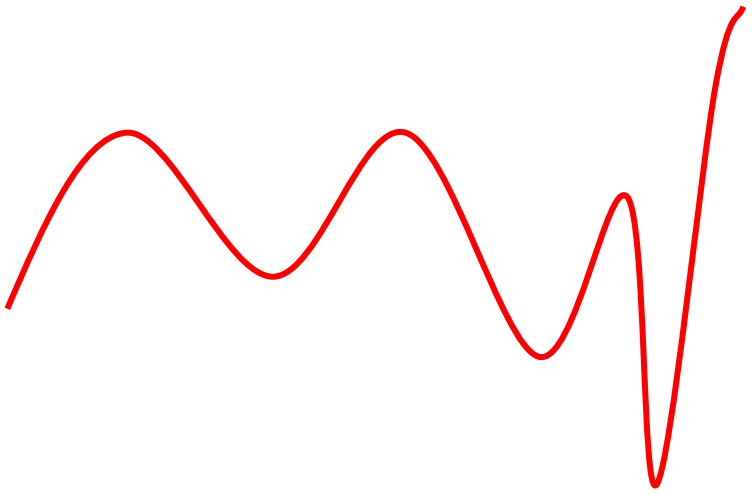


Consequences

- For error functions like this, if we get an answer where a small change doesn't improve it, we know we are right!
  - There are some issue with what "small" means that will need to wait for CS322
- Moreover, our LS-hillclimbing method will always converge to the right answer
  - By slowly rolling downhill
  - It's actually quite hard to analyze how long it will take (see CS322 and beyond)

Easy and hard error measures

- What makes this error function easy?
  - It has a single minimum, and we can get to it from anywhere by rolling downhill
  - Not all error functions have this property
    - But, least squares does (proof: CS322)



Why is an error function hard?

- An error function where we can get stuck if we roll downhill is a hard one
  - Where we get stuck depends on where we start (i.e., initial guess/conditions)
  - An error function is hard if the area "above it" has a certain shape
    - Nooks and crannies
    - In other words, CONVEX!
  - Non-convex error functions are hard to minimize

What else about LS?

- Least squares has an even more amazing property than convexity
  - Suppose we have a collection of data points (x_i,y_i) and we want to find the (m,b) that minimizes the squared error
  - If (m,b) were precisely right, we would have:

$$\begin{aligned} m \cdot x_1 + b &= y_1 \\ m \cdot x_2 + b &= y_2 \\ \vdots &= \vdots \\ m \cdot x_n + b &= y_n \end{aligned}$$

Closed form solution!

- There is a magic formula for the optimal choice of (m,b)
  - You don't need to search, you can simply compute the right answer
    - Shades of snailsort versus quicksort…
- This is a huge part of why everyone uses least squares
  - The other piece is too hard to explain

♦ Closed form LS formula

- The derivation requires linear algebra
  - Most books use calculus also, but it's not required (see the "Links" section on the course web page)

$$S_{xx} \equiv \sum_i (x_i - \bar{x})^2$$

$$S_{xy} \equiv \sum_i (x_i - \bar{x})(y_i - \bar{y})$$

$$m = \frac{S_{xy}}{S_{xx}}$$

$$b = \bar{y} - m\bar{x}$$