61A Extra Lecture 13

Announcements

Prediction

Regression

Given a set of (x, y) pairs, find a function f(x) that returns good y values

```
pairs = [(64.75, 163.5), (63.75, 147.5), (72.5, 224), ...]
```

Height in inches: 5 feet $4\frac{3}{4}$ inches

Weight in pounds

Data from a health survey of 6342 adults

Measuring error: $|y-f(x)|$ or $(f(x)-y)^2$ are both typical

Over the whole set of (x, y) pairs, we can average this "squared error"

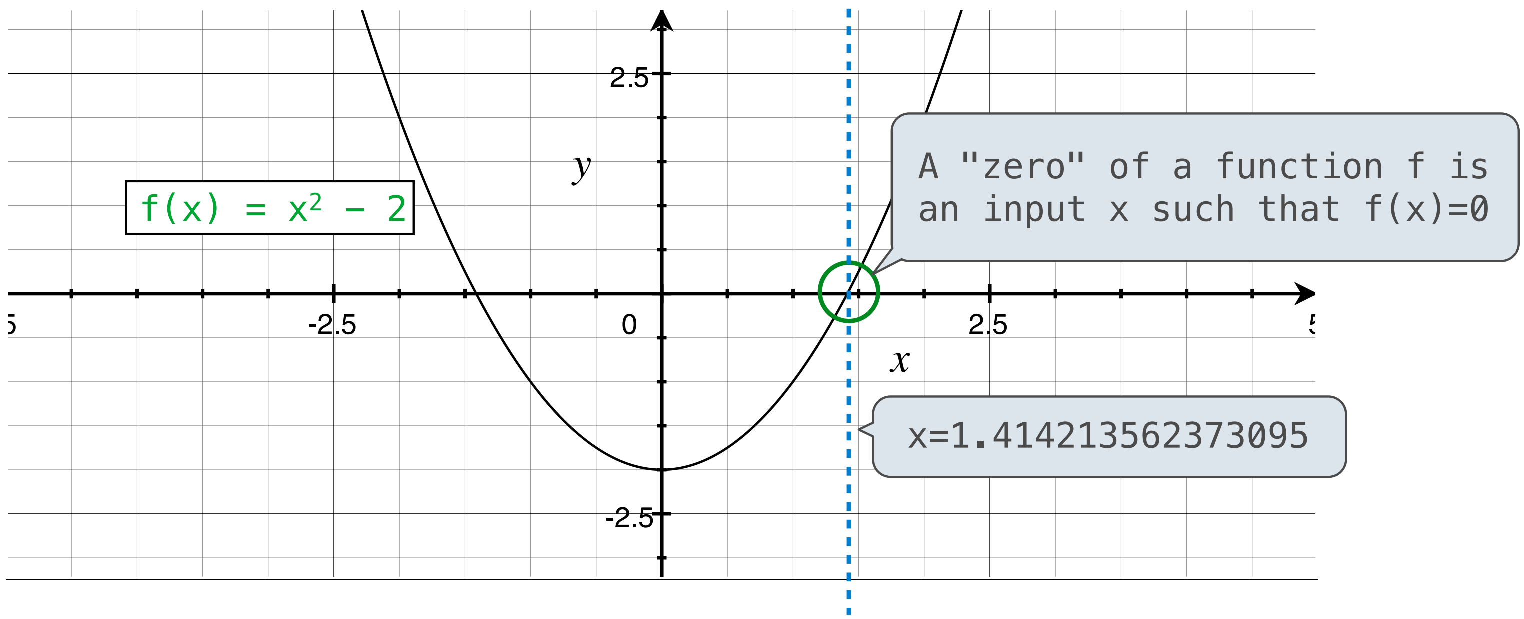
Squared error has the wrong units, so it's common to take the square root

The result is the "root mean squared error" of a predictor f on a set of (x, y) pairs

(Demo)

Purpose of Newton's Method

Quickly finds accurate approximations to zeroes of differentiable functions!



Application: Find the minimum of a function by finding the zero of its derivative

Approximate Differentiation

```
Differentiation can be performed
symbolically or numerically
```

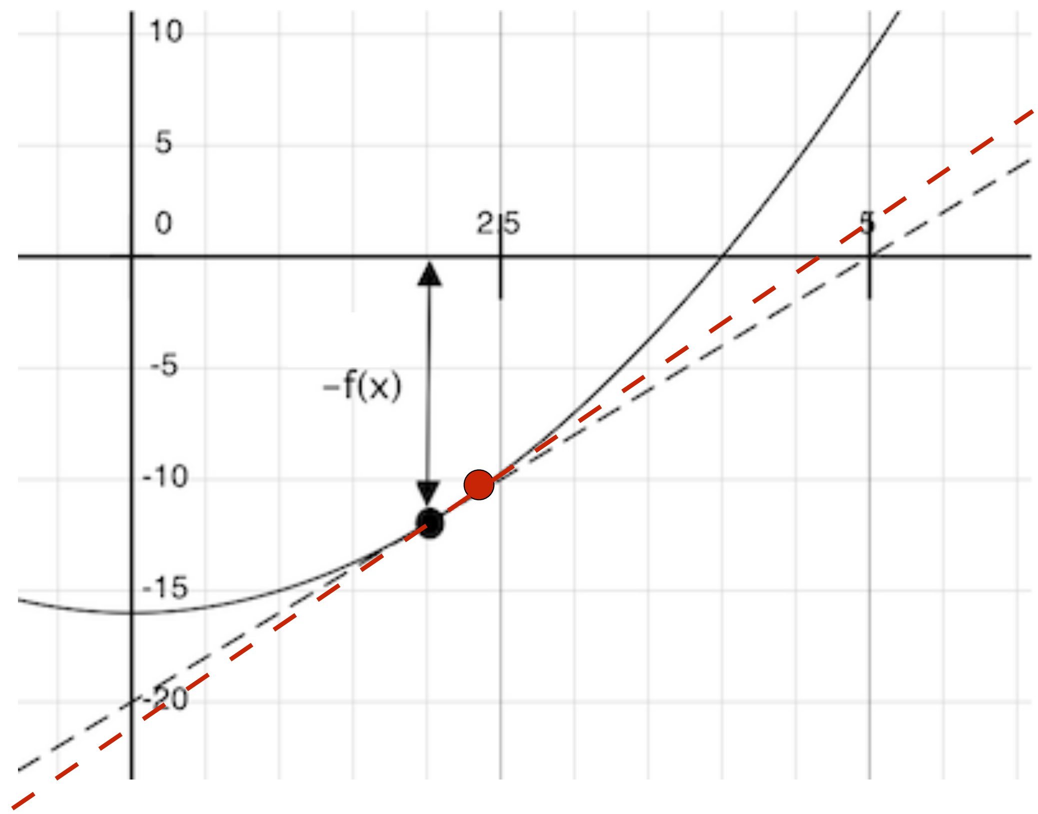
f(x) = x^2 − 16

f'(x) = 2x

f'(2) = 4

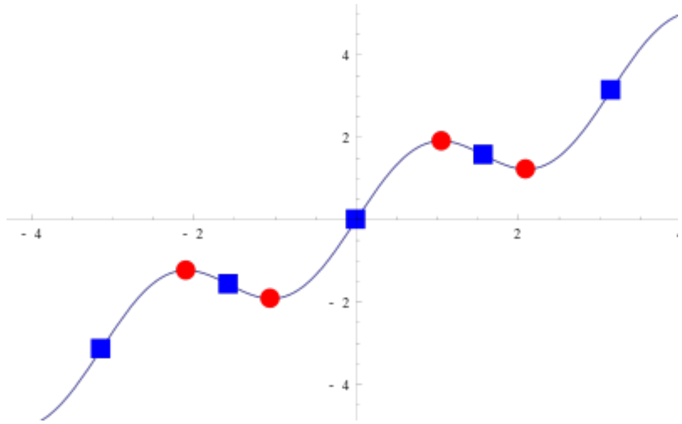
$$f'(x) = \lim_{a \to 0} \frac{f(x+a) - f(x)}{a}$$

$$f'(x) \approx \frac{f(x+a) - f(x)}{a}$$ (if a is small)



Critical Points and Inverses

Maxima, minima, and inflection points of a differentiable function occur when the derivative is 0



The global minimum of convex functions that are (mostly) twice-differentiable can be computed numerically using techniques that are similar to Newton's method

(Demo)

http://upload.wikimedia.org/wikipedia/commons/f/fd/Stationary_vs_inflection_pts.svg

Multiple Linear Regression

Given a set of (xs, y) pairs, find a linear function f(xs) that returns good y values

A linear function has the form **w** • **xs** + *b* for vectors **w** and **xs** and scalar *b*

(Demo)

Note: Root mean squared error can be optimized through linear algebra alone, but numerical optimization works for a much larger class of related error measures

Classification

Classification

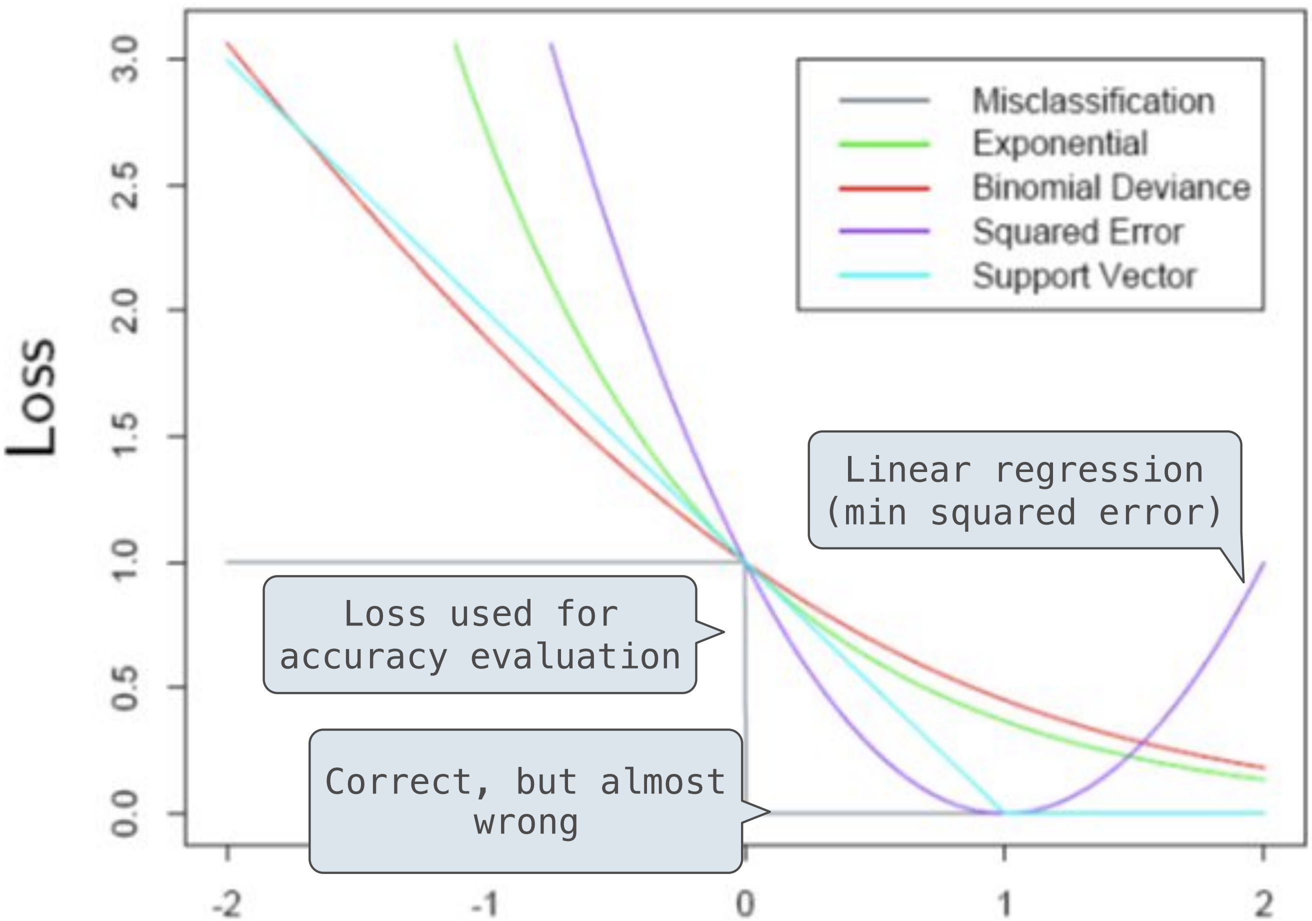
When the output of prediction is a category instead of a quantity, prediction is called *classification* instead of *regression*

The approach is basically the same, except that the output is coerced into a category

Error is measured by *accuracy*: the proportion of categorical predictions that were correct

(Demo)

Loss Functions



Changes (0, 1) to (−1, 1)

Shifts choice to 0 from 0.5

$(2*y-1) * (\mathbf{w} \bullet \mathbf{xs} + b - 0.5)$

How close to being right/wrong

Language Models

Natural Language Can Be Predicted

```
                                 for          to
Programs should be written ~~fur~~ people ~~two~~ read.

             X                    Y
```

Programs should be written ~~fur~~ (for) people ~~two~~ (to) read.

X: Programs should be written

Y: for

```
"should be written for": 1
    "be written for": 1
       "written for": 1
               "for": 1
    "nice computer": 0
```

(Demo)