CPSC 340:
Machine Learning and Data Mining

Least Squares

Summer 2021

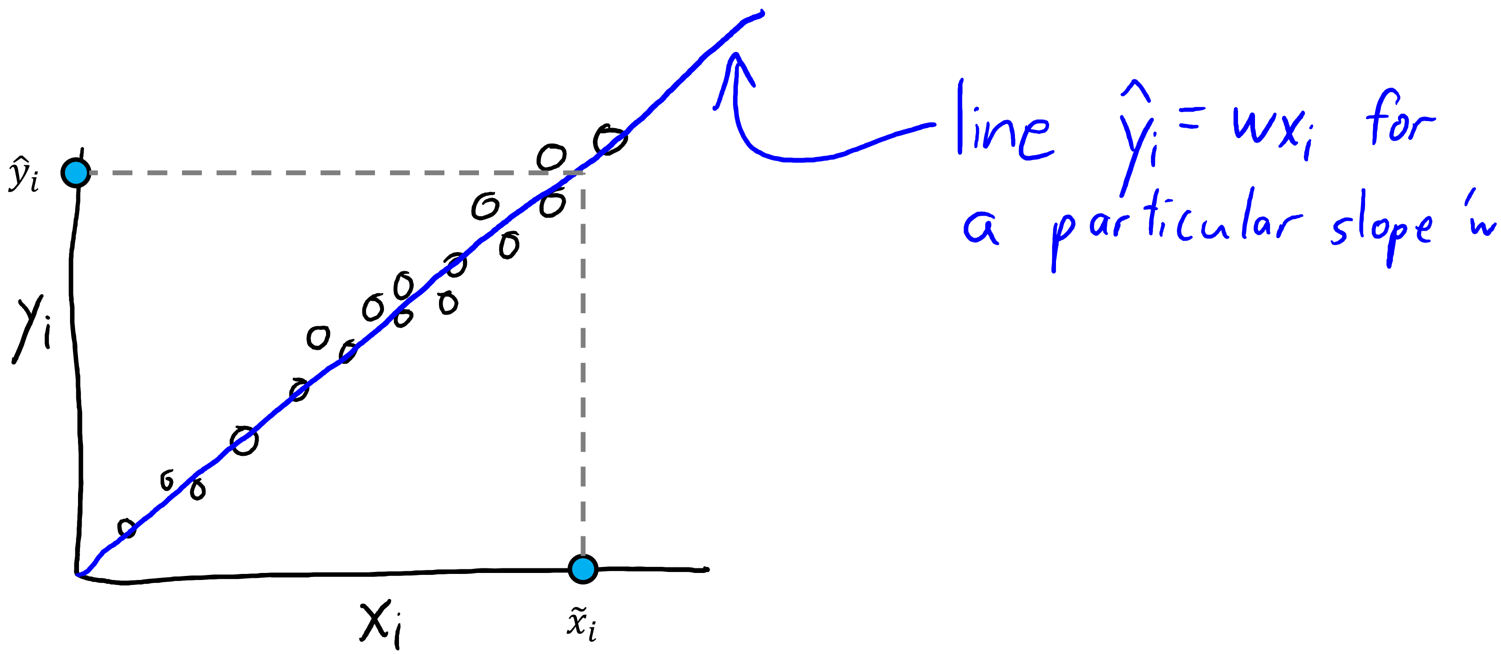
Admin

- Assignment 2:
  - Due 9:25am Monday!

- Assignment 3 is up.
  - Due 9:25am Friday!
  - Should be able to do most problems after today's lecture

- Until now, we described algorithms plainly
- Starting now, we will describe algorithms more technically
- We're going to start using calculus and linear algebra a lot
  - Start reviewing these ASAP if you are rusty.
  - Mark's calculus notes: here.
  - Mark's linear algebra notes: here.

Last Time: Linear Regression

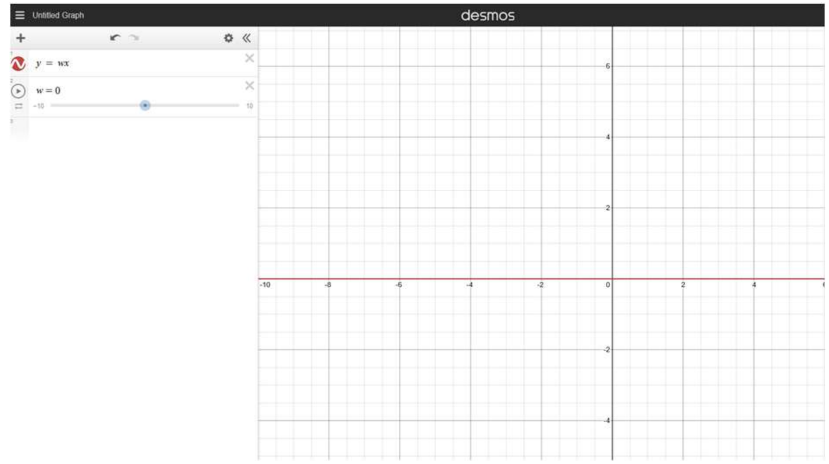


In This Lecture

1. Least Squares (20 minutes)
   – LOTS OF MATH
2. Normal Equations (25 minutes)
   – LOTS OF MATH

Coming Up Next

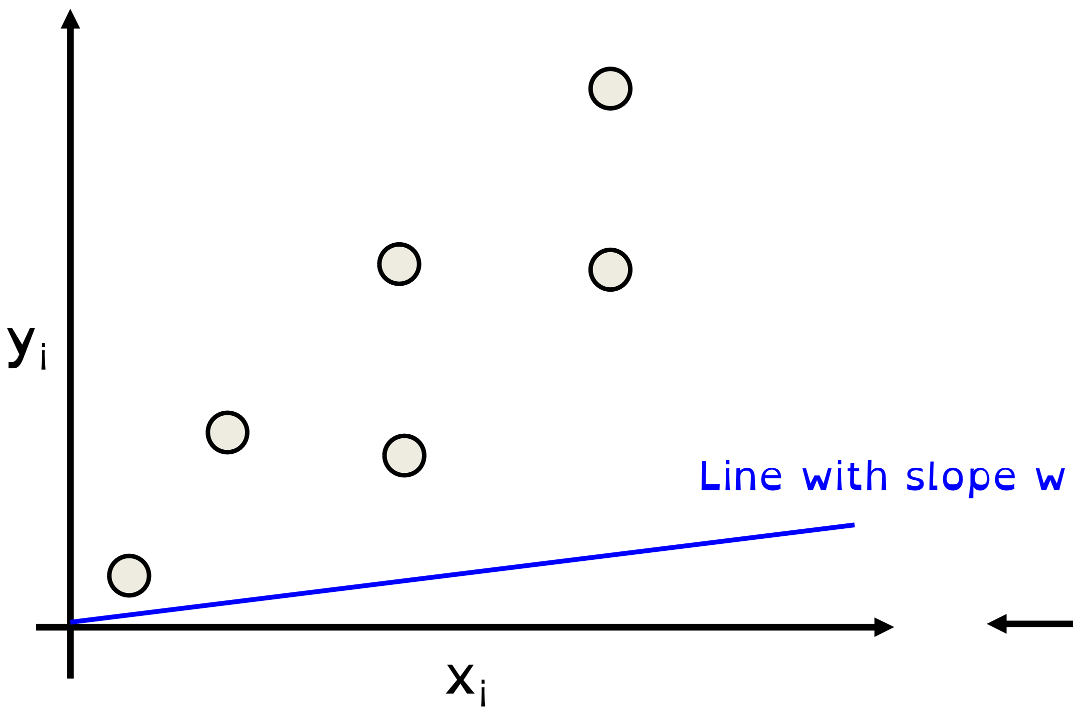
LEAST SQUARES

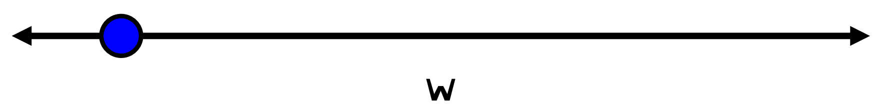


graphing calculator

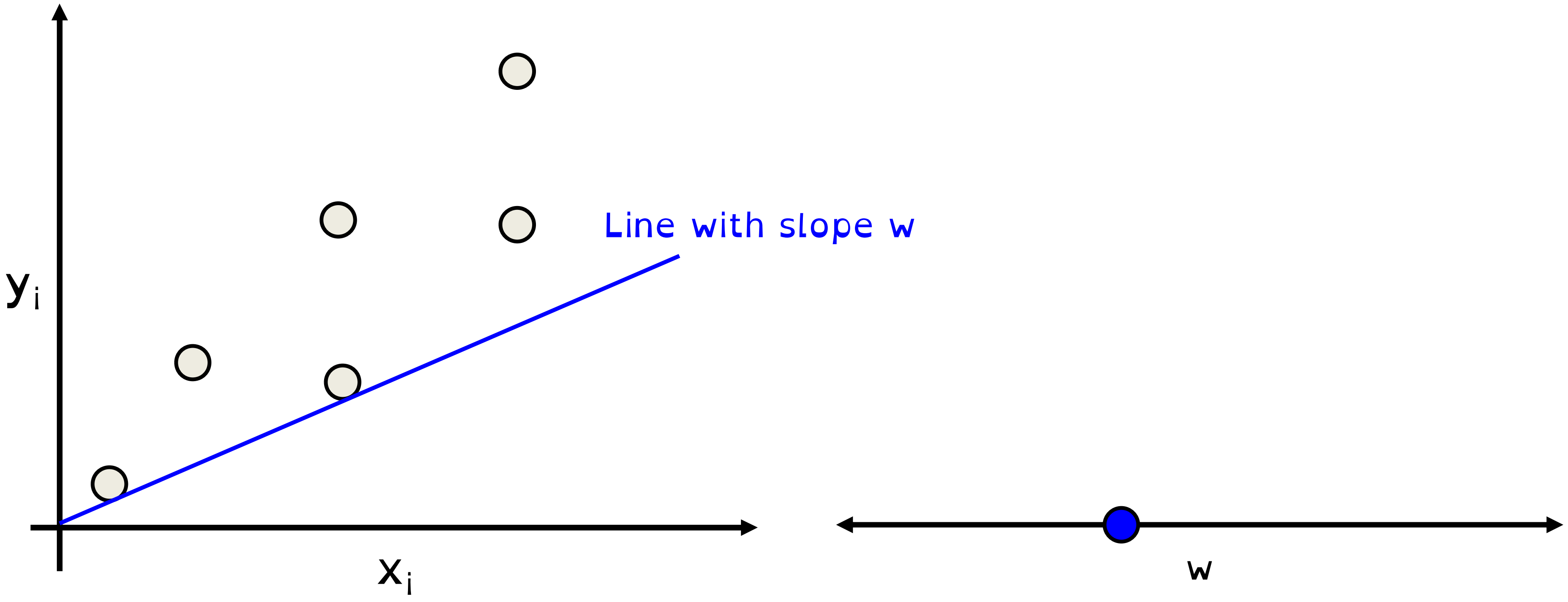
human-in-the-loop machine learning algorithm

Manually Fitting Linear Model

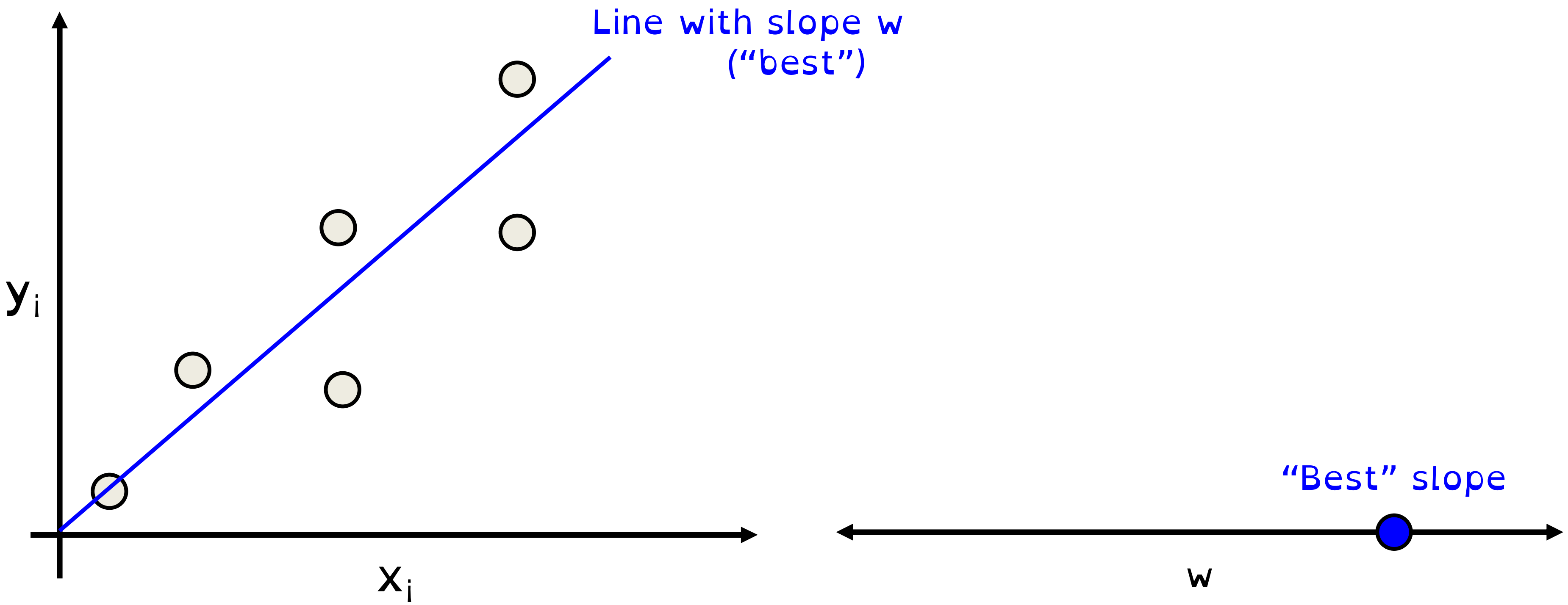




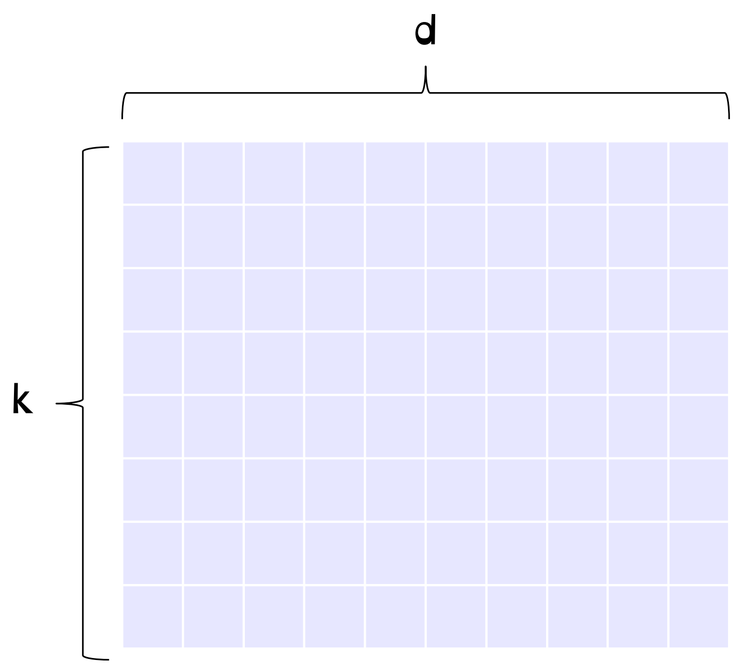
Manually Fitting Linear Model



Manually Fitting Linear Model

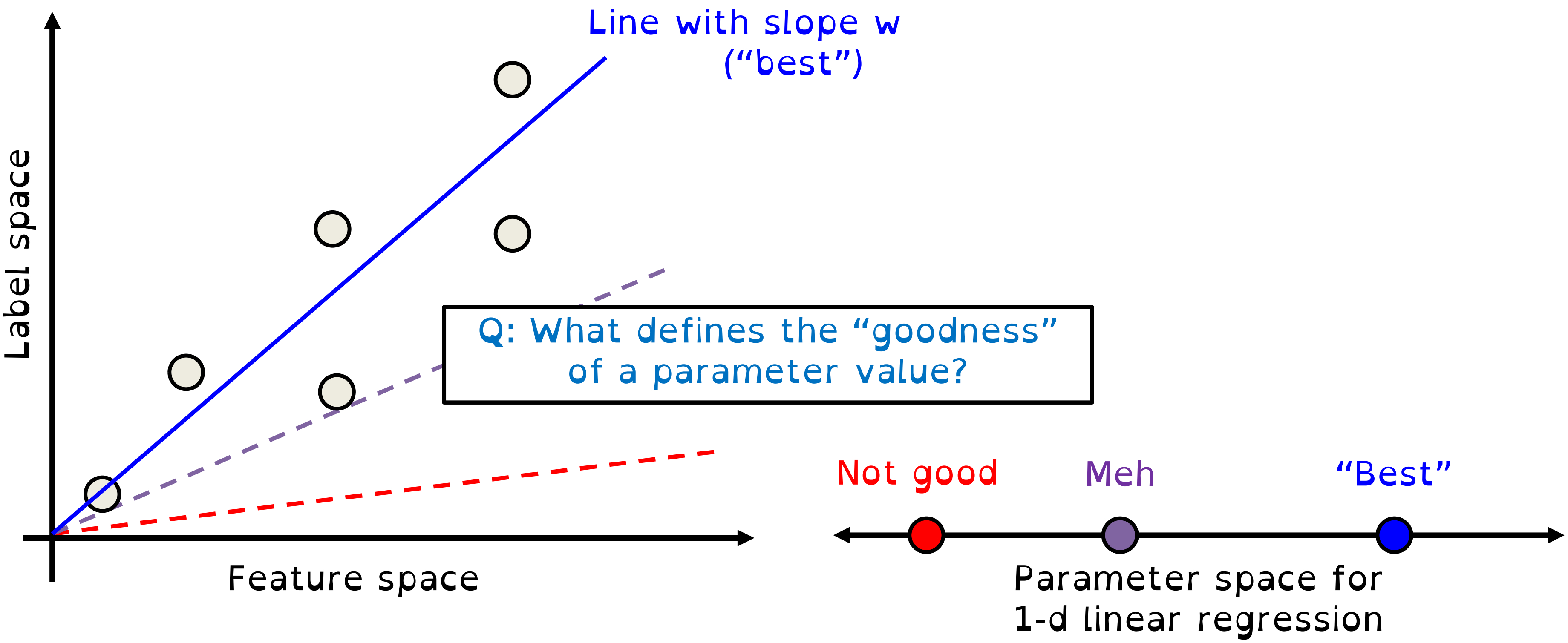


"Parameter Space"



Space of possible decision stumps
("parameter space" of a decision stump)

“Parameter Space”



Q: What defines the “goodness” of a parameter value?

Least Squares Objective

- Our linear model is given by:

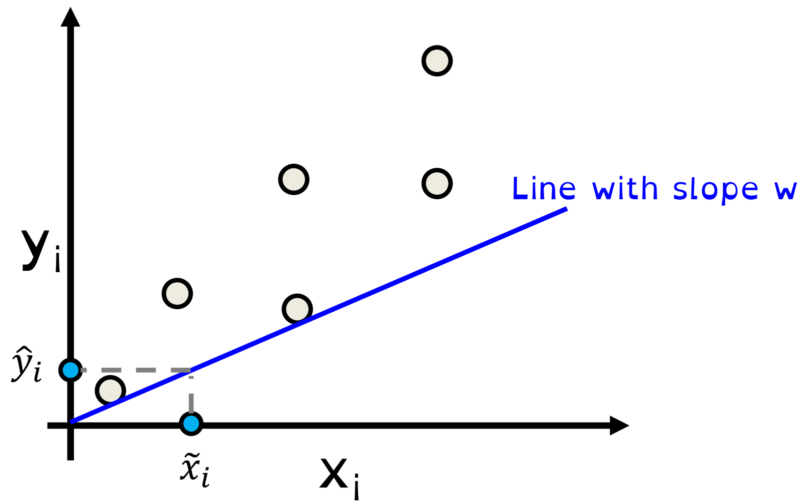
$$\hat{y}_i = w x_i$$

- So we make predictions for a new example by using:

$$\hat{y}_i = w \tilde{x}_i$$

- Our task is to find an optimal w in parameter space.

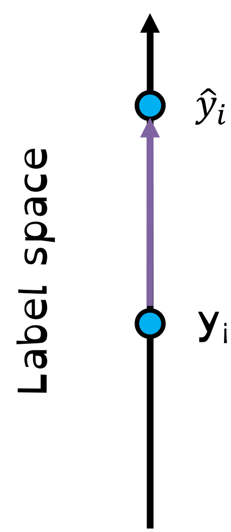
Which “Error” Should We Use?



- We can’t use the classification accuracy as before!
- ____________ never happens in practice
  - Two floating point numbers are never “equal”.
  - Even if two floating points can be “equal”, model will almost always give a slightly wrong prediction.
    - Due to noise or relationship not being quite linear

"Residual"

- Residual := difference between prediction and true label
  - Usually: prediction minus truth
  - Measure of "error" in continuous prediction



$r_i = \hat{y}_i - y_i$

aka "residual"

Q: What do residuals look like when my model is good?

Least Squares Objective

Q: What's wrong with this?

$$\text{error} = \sum_{i=1}^{n} \hat{y}_i - y_i$$

Q: How do we compute $\hat{y}_i$?

$$\text{error} = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$$= \sum_{i=1}^{n} (w x_i - y_i)^2$$

Least Squares Objective

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

$$f(w) = \sum_{i=1}^{n} (wx_i - y_i)^2$$

- The function f is called an "error" or "objective function"
  - Input: slope
  - Output: "error" of slope
- Best slope w minimizes f, the sum of squared errors (WHY squared?)
  - There are some justifications for this choice.
  - A probabilistic interpretation is coming later in the course.
- But usually, it is done because it is easy to minimize.

"Signature"

- Signature: specifies input and output "types" of function

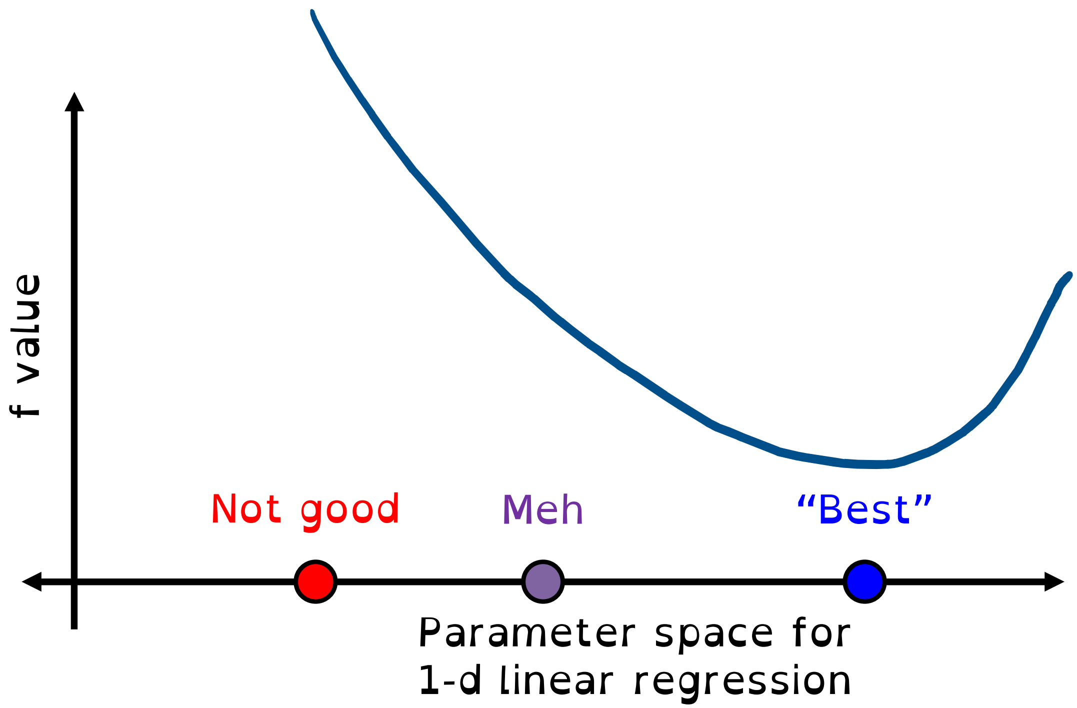
$$f : \underbrace{\mathbb{R}}_{\text{input: scalar}} \rightarrow \underbrace{\mathbb{R}}_{\text{output: scalar}}$$

- Here, function f takes a scalar value and outputs a scalar value
- Later, we will generalize this to

$$f : \underbrace{\mathbb{R}^d}_{\text{input: } d \times 1 \text{ vector}} \rightarrow \underbrace{\mathbb{R}}_{\text{output: scalar}}$$

Objective in 1D Parameter Space

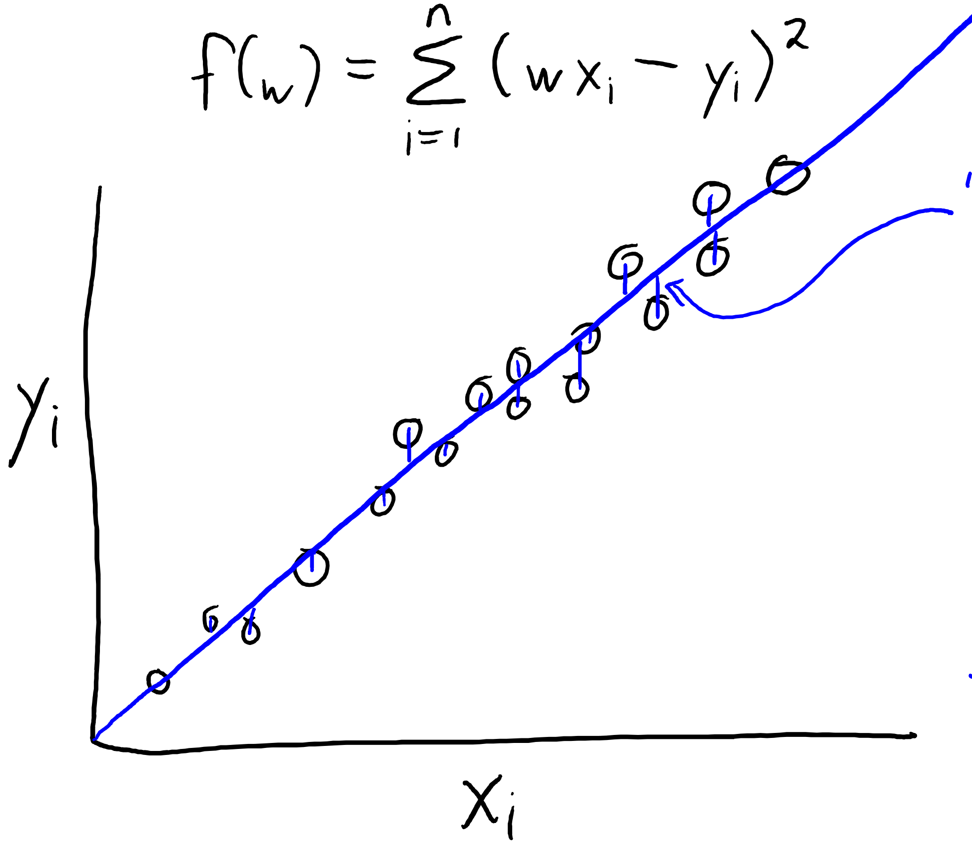
$f: \mathbb{R} \to \mathbb{R}$



Least Squares Objective

- Classic way to set slope 'w' is minimizing sum of squared errors:

$$f(w) = \sum_{i=1}^{n} (w x_i - y_i)^2$$



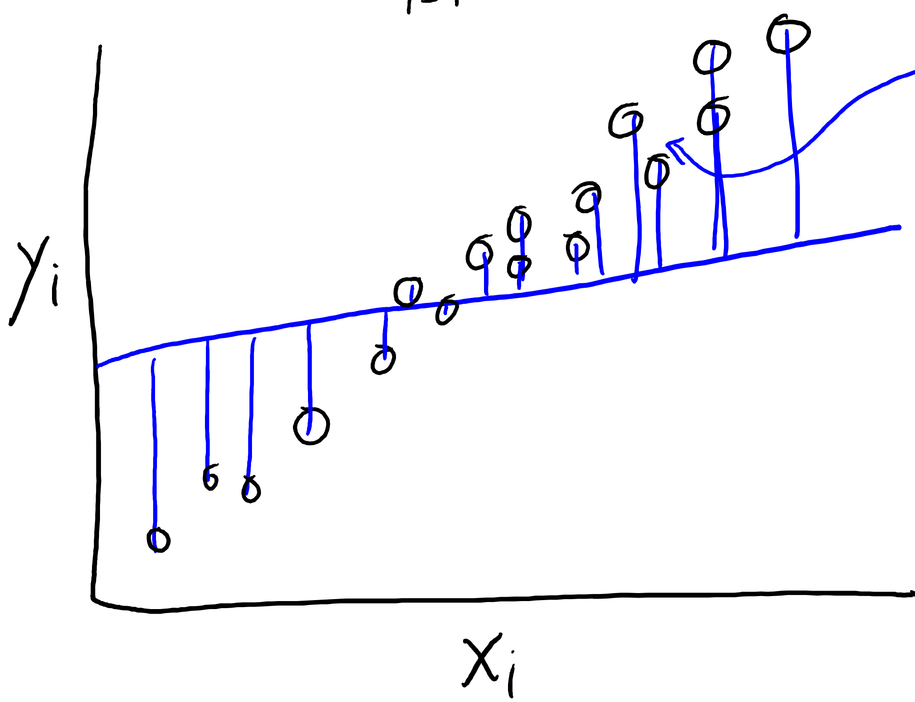
"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

If this error is small, then our predictions are close to the targets.

Least Squares Objective

- Classic way to set slope 'w' is minimizing sum of squared errors:

$$f(w) = \sum_{i=1}^{n} (wx_i - y_i)^2$$

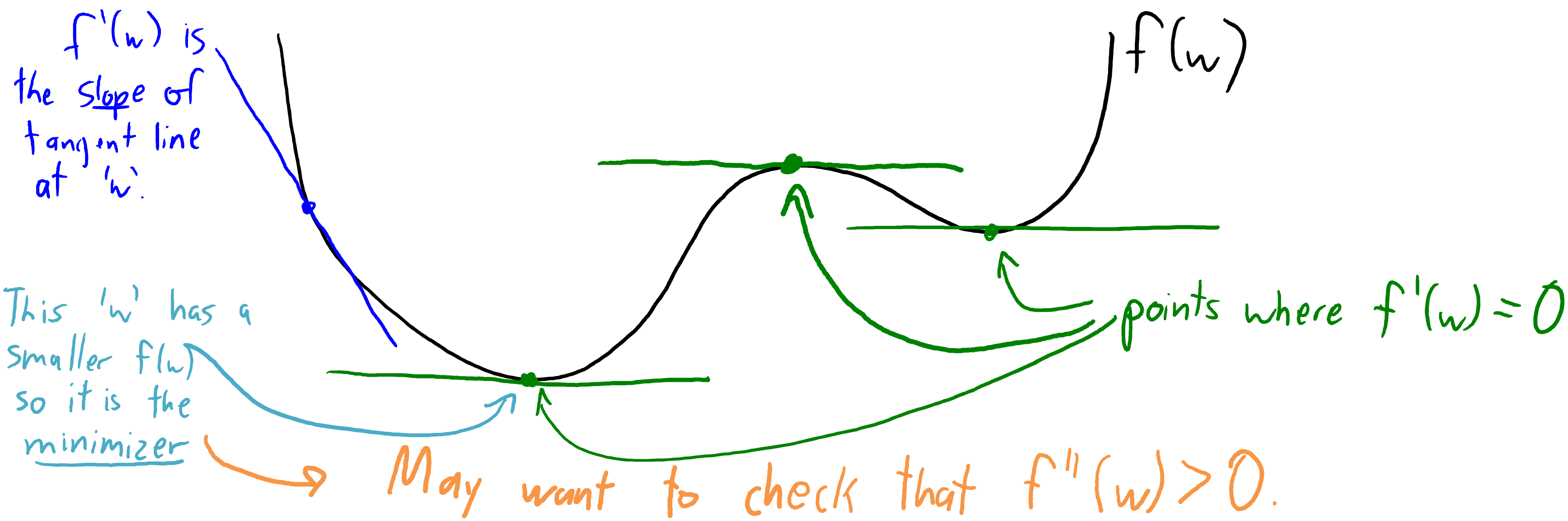


"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

If this error is large, then our predictions are far from the targets.

Minimizing a Differential Function

- Math 101 approach to minimizing a differentiable function ‘f’:
  1. Take the derivative of ‘f’.
  2. Find points ‘w’ where the derivative f’(w) is equal to 0.
  3. Choose the smallest one (and check that f’’(w) is positive).



Digression: Multiplying by a Positive Constant

- Note that this problem:

$$f(w) = \sum_{i=1}^{n} (wx_i - y_i)^2$$

- Has the same set of minimizers as this problem:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n} (wx_i - y_i)^2$$

- And these also have the same minimizers:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} (wx_i - y_i)^2 \qquad f(w) = \frac{1}{2n}\sum_{i=1}^{n} (wx_i - y_i)^2 + 1000$$

- I can multiply 'f' by any positive constant and not change solution.
  - Derivative will still be zero at the same locations.
  - We'll use this trick a lot!

(Quora trolling on ethics of this)

Finding Least Squares Solution

If you're reviewing: try this on your own first!

- Find 'w' that minimizes sum of squared errors:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(wx_i - y_i)^2$$

Finding Least Squares Solution

- Find ‘w’ that minimizes sum of squared errors:

[1] $$f(w) = \frac{1}{2}\sum_{i=1}^{n}(wx_i - y_i)^2 \overset{\text{Expand}}{=} \frac{1}{2}w^2\sum_{i=1}^{n}x_i^2 - w\sum_{i=1}^{n}x_i y_i + \underbrace{\frac{1}{2}\sum_{i=1}^{n}y_i^2}_{\text{NO w here.}}.$$

[2] $$f'(w) \overset{\text{derivatives}}{=} w\sum_{i=1}^{n}x_i^2 - \sum_{i=1}^{n}x_i y_i.$$

[3] $$f'(w) = 0, \text{ when } \quad w \overset{\text{rearrange [2]}}{=} \frac{\sum_{i=1}^{n}x_i y_i}{\sum_{i=1}^{n}x_i^2}.$$

Q: What can go wrong here?

Finding Least Squares Solution

- Finding 'w' that minimizes sum of squared errors:

$$f'(w) = 0, \text{ when } w = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}.$$

Q: Are we done?

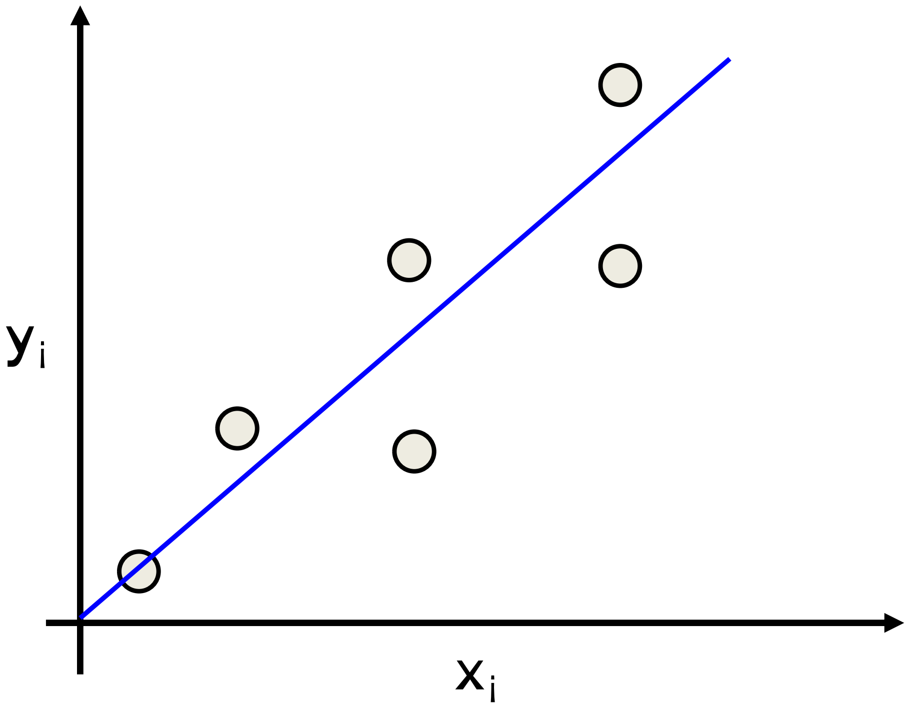
- Check that this is a minimizer by checking second derivative:

$$f'(w) = w\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i y_i$$

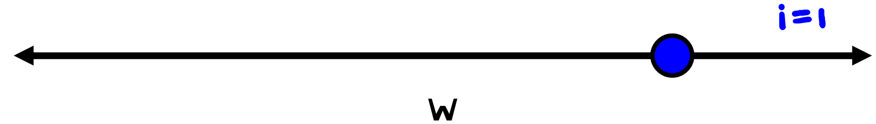
$$f''(w) = \sum_{i=1}^{n} x_i^2$$

  - Since (anything)^2 is non-negative and (anything non-zero)$^2 > 0$, if we have one non-zero feature then $f''(w) > 0$ and this is a minimizer.

Least Squares on 1D Parameter Space



$$W = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}.$$



Q: Does this generalize to higher-dimensional data?

Coming Up Next

HIGHER-DIMENSIONAL LEAST SQUARES

Motivation: Combining Explanatory Variables

- Smoking is not the only contributor to lung cancer.
  - For example, there environmental factors like exposure to asbestos.
- How can we model the combined effect of smoking and asbestos?
- A simple way is with a 2-dimensional linear function:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2}$$

Value of feature 2 in example 'i'

"weight" on feature 2.

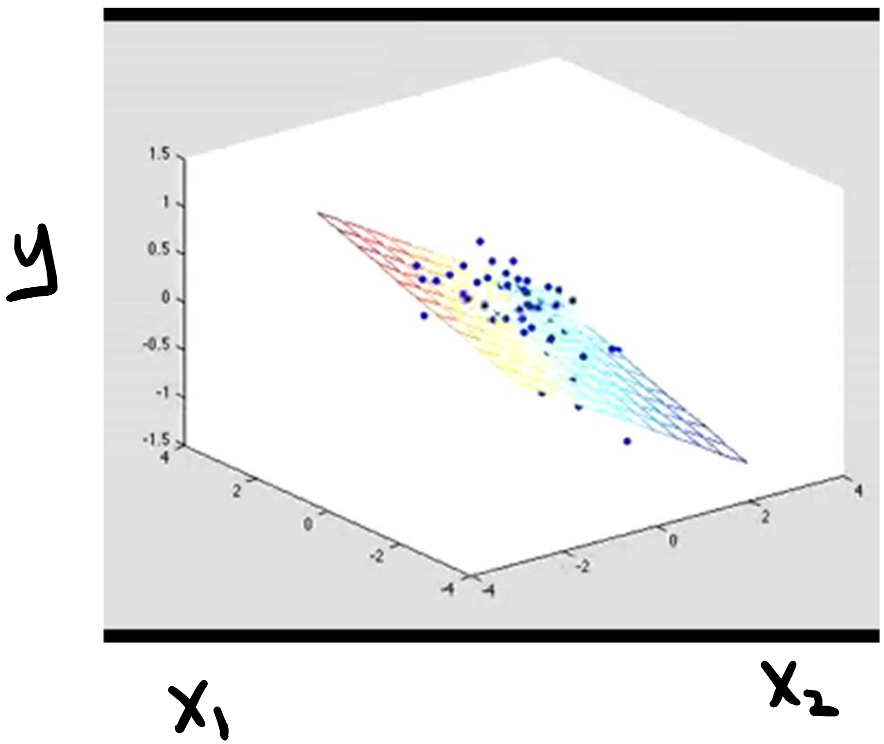
"weight" of feature 1

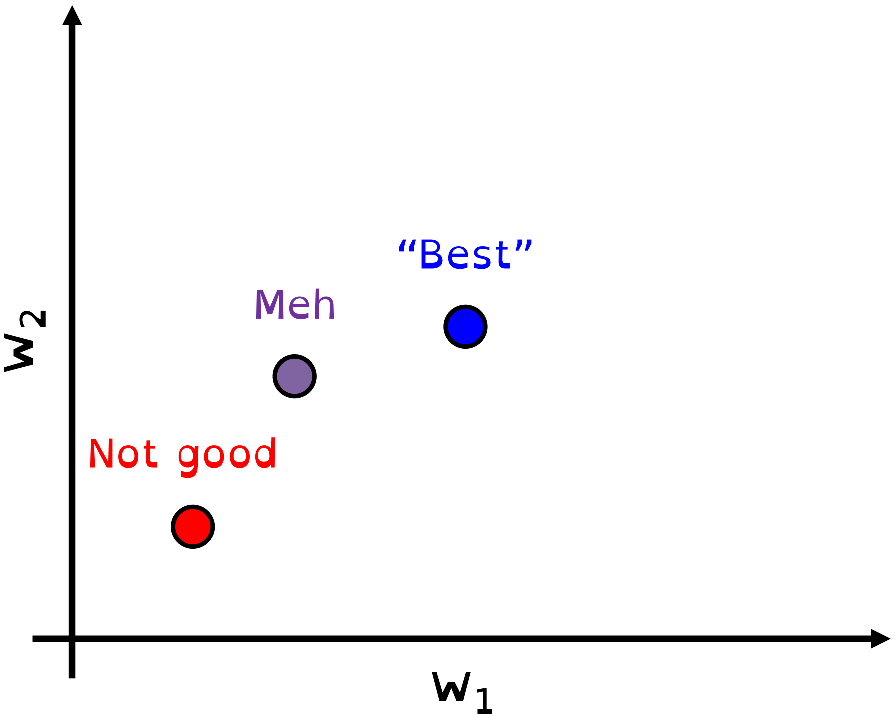
Value of feature 1 in example 'i'

- We have a weight w_1 for feature '1' and w_2 for feature '2':

$$\hat{y}_i = 10(\#\text{cigarettes}) + 25(\#\text{asbetos})$$

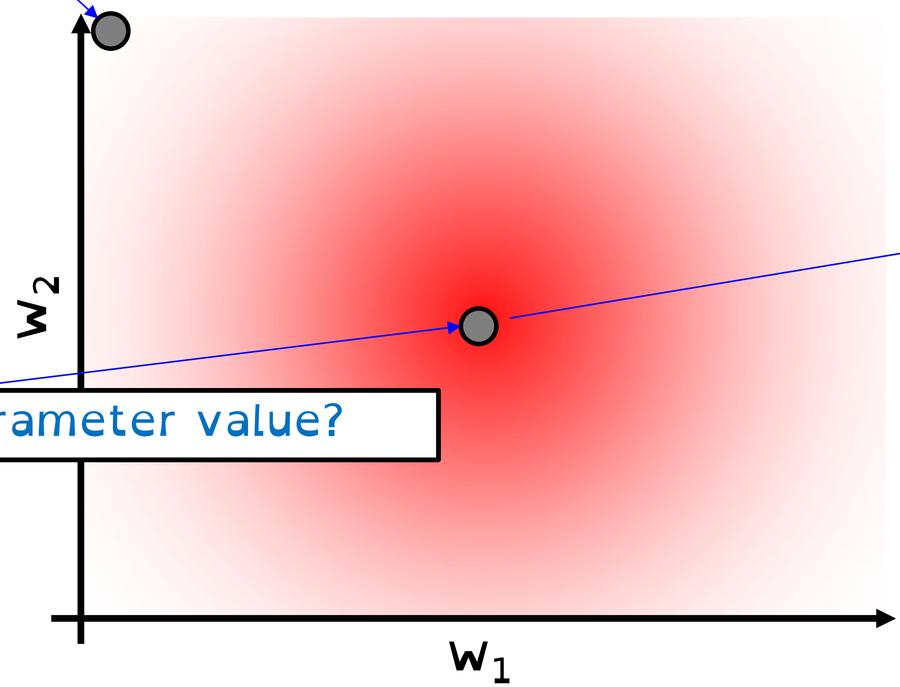
Parameter Space in 2D





Objective in 2D Parameter Space

Q: Is this a good parameter value?



$f: \mathbb{R}^2 \to \mathbb{R}$

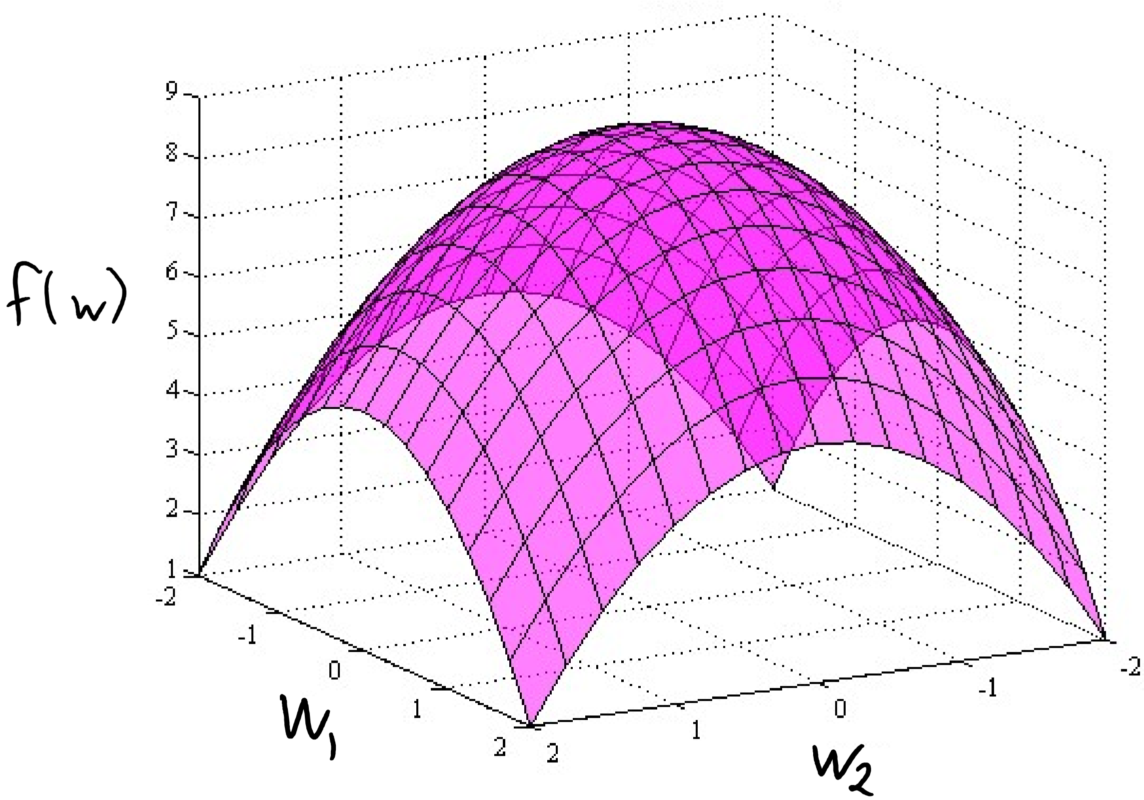
Thicker colour:
Lower objective value

Q: Is this a good parameter value?

Q: What makes it the "best" parameter value?

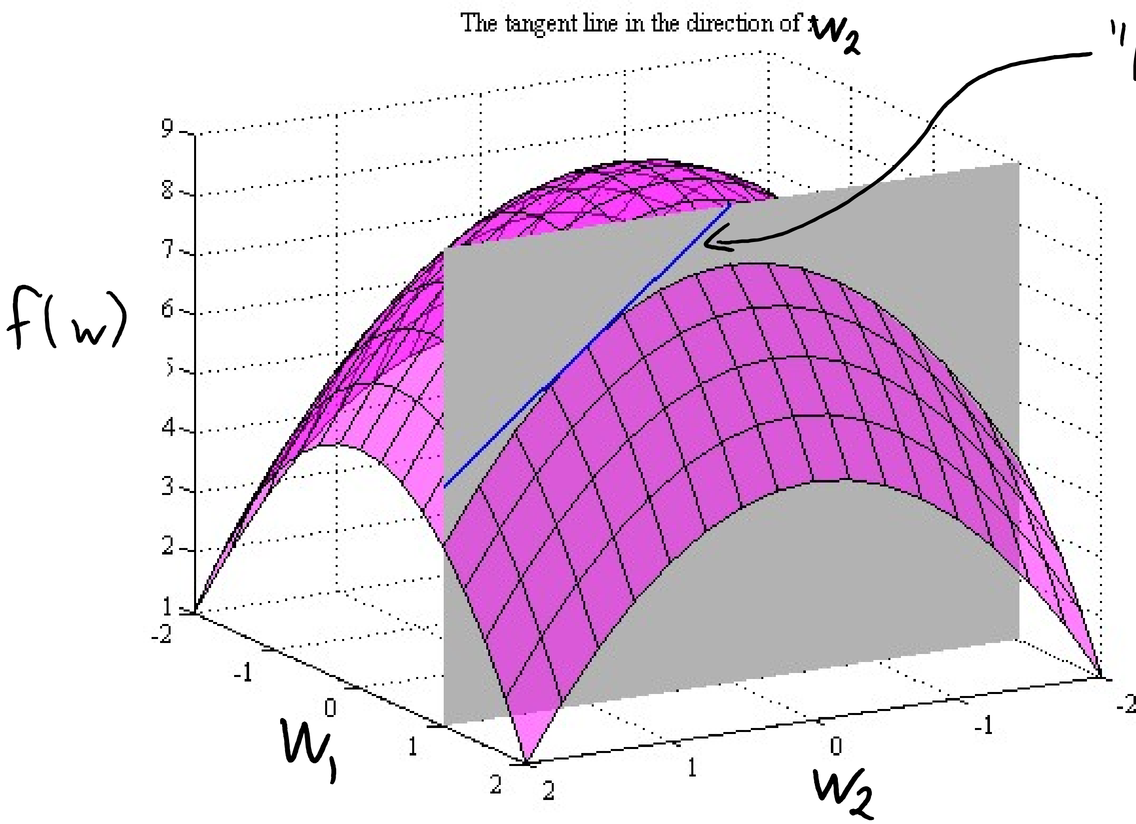
Partial Derivatives

Q: If I "fix" w_1, what does $f(w_2)$ look like?



http://msemac.redwoods.edu/~darnold/math50c/matlab/pderiv/index.xhtml

Partial Derivatives



"Partial" derivative of 'f' with respect to w_2 is the derivative with respect to w_2 when all other variables are held <u>fixed</u>.

Denoted by $\frac{\partial}{\partial w_2}$ for variable w_2

Different Notations for Least Squares

- If we have 'd' features, the d-dimensional linear model is:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + \cdots + w_d x_{id}$$

  - In words, our model is that the output is a ________________ of the inputs.
- We can re-write this in summation notation:

$$\hat{y}_i = \sum_{j=1}^{d} w_j x_{ij}$$

- We can also re-write this in vector notation:

$$\hat{y}_i = \underbrace{w^T x_i}$$

(assuming 'w' and x_i are column-vectors)

"inner product" between vectors

Notation Alert (again)

- In this course, all vectors are assumed to be column-vectors:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}$$

- So $w^T x_i$ is a scalar:

$$w^T x_i = \begin{bmatrix} w_1 & w_2 & \cdots & w_d \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id}$$

$$= \sum_{j=1}^{d} w_j x_{id}$$

- So rows of 'X' are actually transpose of column-vector x_i:

$$X = \begin{bmatrix} — & x_1^T & — \\ — & x_2^T & — \\ & \vdots & \\ — & x_n^T & — \end{bmatrix}$$

Least Squares in d-Dimensions

- The linear least squares model in d-dimensions minimizes:

$$f: \mathbb{R}^d \rightarrow \mathbb{R}$$

$$f(\qquad\qquad\qquad) = \sum_{i=1}^{n} \left(\underline{\qquad\quad} - y_i \right)^2$$

- Dates back to 1801: Gauss used it to predict location of Ceres.
- How do we find the best vector 'w' in 'd' dimensions?
  - Can we set the partial derivative of each variable to 0?

Least Squares Partial Derivatives (1 Example)

If you're reviewing: try this on your own first!

- The linear least squares model in d-dimensions for 1 example:

$$f(w_1, w_2, \dots, w_d) = \frac{1}{2}(\hat{y}_i - y_i)^2$$

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + \dots w_d x_{id}$$

- Computing the partial derivative for variable '1':

$$\frac{\partial}{\partial w_1} f(w_1, w_2, \dots, w_d) =$$

Least Squares Partial Derivatives (1 Example)

- The linear least squares model in d-dimensions for 1 example:

[1] $f(w_1, w_2, \ldots, w_d) = \frac{1}{2}(\hat{y}_i - y_i)^2 = \frac{1}{2}\hat{y}_i^2 - \hat{y}_i y_i + \frac{1}{2}y_i^2$

[2] $\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + \ldots w_d x_{id}$ $= \frac{1}{2}\left(\sum_{j=1}^{d} w_j x_{ij}\right)^2 + \left(\sum_{j=1}^{d} w_j x_{ij}\right) y_i + \frac{1}{2} y_i^2$

- Computing the partial derivative for variable '1':

[3] $\frac{\partial}{\partial w_1} f(w_1, w_2, \ldots, w_d) = \left(\sum_{j=1}^{d} w_j x_{ij}\right) x_{i1} - y_i x_{i1} + 0$

[4] $= \left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right) x_{i1}$

[5] $= (w^T x_i - y_i) x_{i1}$

Least Squares Partial Derivatives ('n' Examples)

- Linear least squares partial derivative for variable 1 on example 'i':

$$\frac{\partial}{\partial w_1} f(w_1, w_2, \dots, w_d) = (w^T x_i - y_i) x_{i1}$$

- For a generic variable 'j' we would have:

$$\frac{\partial}{\partial w_j} f(w_1, w_2, \dots, w_d) = (w^T x_i - y_i) x_{ij}$$

- And if 'f' is summed over all 'n' examples we would have:

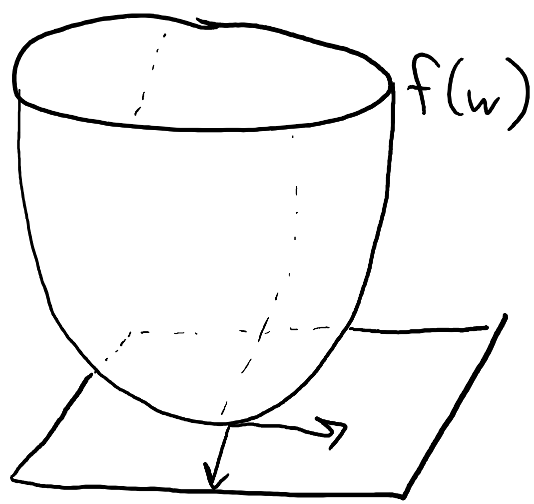
$$\frac{\partial}{\partial w_j} f(w_1, w_2, \dots, w_d) = \sum_{i=1}^{n} (w^T x_i - y_i) x_{ij}$$

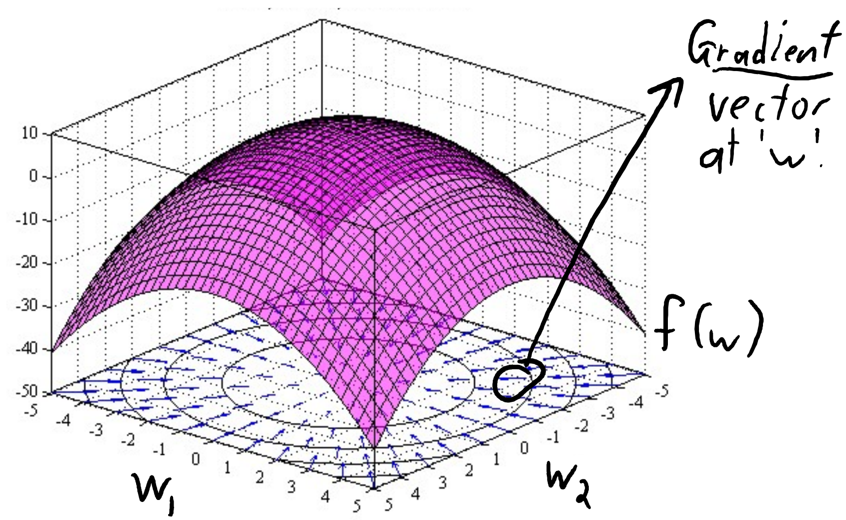
- Unfortunately, the partial derivative for w_j depends on all $\{w_1, w_2, \dots, w_d\}$
  - I can't just "set equal to 0 and solve for w_j".

Gradient and Critical Points in d-Dimensions

- Generalizing "set the derivative to 0 and solve" in d-dimensions:
  - Find 'w' where the gradient vector equals the zero vector.
- Gradient is a _-dimensional vector with partial derivative 'j' in position 'j':

$$\nabla f(w) = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix}$$





Tangent slope is 0 in every direction at minimizer.

Gradient and Critical Points in d-Dimensions

- Generalizing "set the derivative to 0 and solve" in d-dimensions:
  - Find 'w' where the gradient vector equals the zero vector.
- Gradient is a d-dimensional vector with partial derivative 'j' in position 'j':

$$\nabla f(w) = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix}$$

For linear least squares:

$$\nabla f(w) = \begin{bmatrix} \sum_{i=1}^n (w^T x_i - y_i) x_{i1} \\ \sum_{i=1}^n (w^T x_i - y_i) x_{i2} \\ \vdots \\ \sum_{i=1}^n (w^T x_i - y_i) x_{id} \end{bmatrix}$$

Claims for linear least square:

1. Finding a 'w' where $\nabla f(w) = 0$ can be done by solving a <u>system of linear equations</u>.
2. <u>All</u> 'w' where $\nabla f(w) = 0$ are <u>minimizers</u>.

Coming Up Next

NORMAL EQUATIONS

Matrix/Norm Notation (MEMORIZE/STUDY THIS)

- To solve the d-dimensional least squares, we use matrix notation:
  - We use 'w' as a "d by 1" vector containing weight 'w_j' in position 'j'.
  - We use 'y' as an "n by 1" vector containing target 'y_i' in position 'i'.
  - We use 'x_i' as a "d by 1" vector containing features 'j' of example 'i'.
    - We're now going to be careful to make sure these are column vectors.
  - So 'X' is a matrix with x_i^T in row 'i'.

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} = \begin{bmatrix} — & x_1^T & — \\ — & x_2^T & — \\ & \vdots & \\ — & x_n^T & — \end{bmatrix}$$

Matrix/Norm Notation (MEMORIZE/STUDY THIS)

- To solve the d-dimensional least squares, we use matrix notation:
  - Our prediction for example 'i' is given by the scalar $w^T x_i$.
  - Our predictions for all 'i' (n by 1 vector) is the matrix-vector product Xw.

$$\hat{y}_i = w^T x_i$$

Also, because $w^T x_i$ is a scalar,
we have $w^T x_i = x_i^T w$.
(e.g., $[5]^T = [5]$)

$$Xw = \underbrace{\begin{bmatrix} — & x_1^T & — \\ — & x_2^T & — \\ & \vdots & \\ — & x_n^T & — \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} | \\ w \\ | \end{bmatrix}}_{w} = \begin{bmatrix} x_1^T w \\ x_2^T w \\ \vdots \\ x_n^T w \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \hat{y}$$

Prediction for example 'i' in row 'i'

Matrix/Norm Notation (MEMORIZE/STUDY THIS)

- To solve the d-dimensional least squares, we use matrix notation:
  - Our prediction for example 'i' is given by the scalar $w^T x_i$.
  - Our predictions for all 'i' (n by 1 vector) is the matrix-vector product Xw.
  - Residual vector 'r' gives difference between predictions and y_i (n by 1).
  - Least squares can be written as the squared L2-norm of the residual.

$$f(w) = \sum_{i=1}^{n} (\underbrace{w^T x_i - y_i}_{r_i})^2 = \sum_{i=1}^{n} (r_i)^2$$

$$= \sum_{i=1}^{n} r_i r_i$$

$$= r^T r$$

$$= \|r\|^2 = \|Xw - y\|^2$$

$$r = \hat{y} - y = \underbrace{Xw}_{\hat{y}} - y = \begin{bmatrix} w^T x_1 \\ w^T x_2 \\ \vdots \\ w^T x_n \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} w^T x_1 - y_1 \\ w^T x_2 - y_2 \\ \vdots \\ w^T x_n - y_n \end{bmatrix}$$

r_2 is difference for example 2.

Back to Deriving Least Squares for d > 2…

- We can write vector of predictions $\hat{y}_i$ as a matrix-vector product:

$$\hat{y} = Xw = \begin{bmatrix} w^T x_1 \\ w^T x_2 \\ \vdots \\ w^T x_n \end{bmatrix}$$

- And we can write linear least squares in matrix notation as:

$$f(w) = \frac{1}{2}\|Xw - y\|^2 = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$$

- We'll use this notation to derive d-dimensional least squares 'w'.
  - By setting the gradient $\nabla f(w)$ equal to the zero vector and solving for 'w'.

Digression: Matrix Algebra Review

- Quick review of linear algebra operations we'll use:
  - If 'a' and 'b' be vectors, and 'A' and 'B' be matrices then:

$$a^T b = b^T a$$

$$\|a\|^2 = a^T a$$

$$(A+B)^T = A^T + B^T$$

$$(AB)^T = B^T A^T$$

$$(A+B)(A+B) = AA + BA + AB + BB$$

$$a^T \underbrace{Ab}_{\text{vector}} = \underbrace{b^T A^T}_{\text{vector}} a$$

Sanity check:

ALWAYS CHECK THAT DIMENSIONS MATCH

(if not, you did something wrong)

Linear and Quadratic Gradients

If you're reviewing: try this on your own first!

- From these rules we have (see post-lecture slide for steps):

[1] $f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$

Linear and Quadratic Gradients

- From these rules we have (see post-lecture slide for steps):

[1] $$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2 \xrightarrow{\text{matrix notation}} = \frac{1}{2}\|Xw - y\|^2 \xrightarrow{\text{1. dot producting self, 2. expand}} = \frac{1}{2} w^T \underbrace{X^T X}_{A} w - w^T \underbrace{X^T y}_{b} + \underbrace{\frac{1}{2} y^T y}_{c}$$

matrix notation

1. dot producting self
2. expand

[2] $$\nabla_w f(w) = \frac{1}{2}\nabla w^T A w - \nabla w^T b + \nabla c$$

∇ to each term

[3] $$= \frac{1}{2}\cdot 2Aw - b + 0 = Aw - b = X^T X w - X^T y$$

Calculate gradients (see notes on website)

Q: Do the dimensions make sense?

Normal Equations

- Set gradient equal to ____________________ to find the "critical" points:

$$\nabla_w f(w) = X^T X w - X^T y = 0$$

- We now move terms not involving 'w' to the other side:

$$X^T X w = X^T y$$

- This is a set of 'd' linear equations called the "normal equations".
  - This a linear system like "Ax = b".
  - You can use Gaussian elimination to solve for 'w'.
  - In Python, you solve linear systems in 1 line using numpy.linalg.solve (A3)

Q: What are A and b in this linear system?

Incorrect Solutions to Least Squares Problem

The least squares objective is $f(w) = \frac{1}{2}\|Xw - y\|^2$

The minimizers of this objective are solutions to the linear system:

$$X^T X w = X^T y$$

The following are not the solutions to the least squares problem:

$w = (X^T X)^{-1}(X^T y)$ (only true if $X^T X$ is invertible)

$w X^T X = X^T y$ (matrix multiplication is not commutative, dimensions don't even match)

$w = \dfrac{X^T y}{X^T X}$ (you cannot divide by a matrix)

Summary

- Least squares: a classic method for fitting linear models.
  - With 1 feature, it has a simple closed-form solution.
  - Can be generalized to 'd' features.
- Normal equations: system of equations for solving least squares
- Next time: doing linear regression with a million features
  - We will talk about gradient descent!

Review Questions

- Q1: Why can't we use classification accuracy for regression?
- Q2: What is the input and the output of an objective function?
- Q3: Why is a system of linear equations necessary for computing the stationary point of an objective function?
- Q4: Why can't we always use $(X^T X)^{-1}$ to find w in normal equations?

Linear Least Squares: Expansion Step

Want 'w' that minimizes

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^Tx_i - y_i)^2 = \underbrace{\frac{1}{2}\|Xw - y\|_2^2}_{\text{Let's expand then compute gradient.}} = \frac{1}{2}(Xw - y)^T(Xw - y)$$

$$= \frac{1}{2}((Xw)^T - y^T)(Xw - y)$$

$$= \frac{1}{2}(w^TX^T - y^T)(Xw - y)$$

$$= \frac{1}{2}(w^TX^T(Xw - y) - y^T(Xw - y))$$

$$= \frac{1}{2}(w^TX^TXw - w^TX^Ty - y^TXw + y^Ty)$$

$$= \underbrace{\frac{1}{2}w^TX^TXw - w^TX^Ty + \frac{1}{2}y^Ty}_{\text{Sanity check: all of these are scalars.}}$$

Rule:

$\|a\|^2 = a^Ta$

$(A+B^T) = (A^T + B^T)$

$(AB)^T = B^TA^T$

$(A+B)C = AC + BC$

$A(B+C) = AB + BC$

$\underbrace{a^T}_{\text{vector}}A\underbrace{b}_{\text{vector}} = b^TA^Ta$

- In Smithsonian National Air and Space Museum (Washington, DC):