CPSC 340: Machine Learning and Data Mining

Ordinary Least Squares

Fall 2017

Admin

- You can submit A1 with late day on Monday night.
- You can submit A2 with 2 late days on Wednesday night.
- Mark’s office hours will be cancelled on Tuesday (since he’s away).

Supervised Learning Round 2: Regression

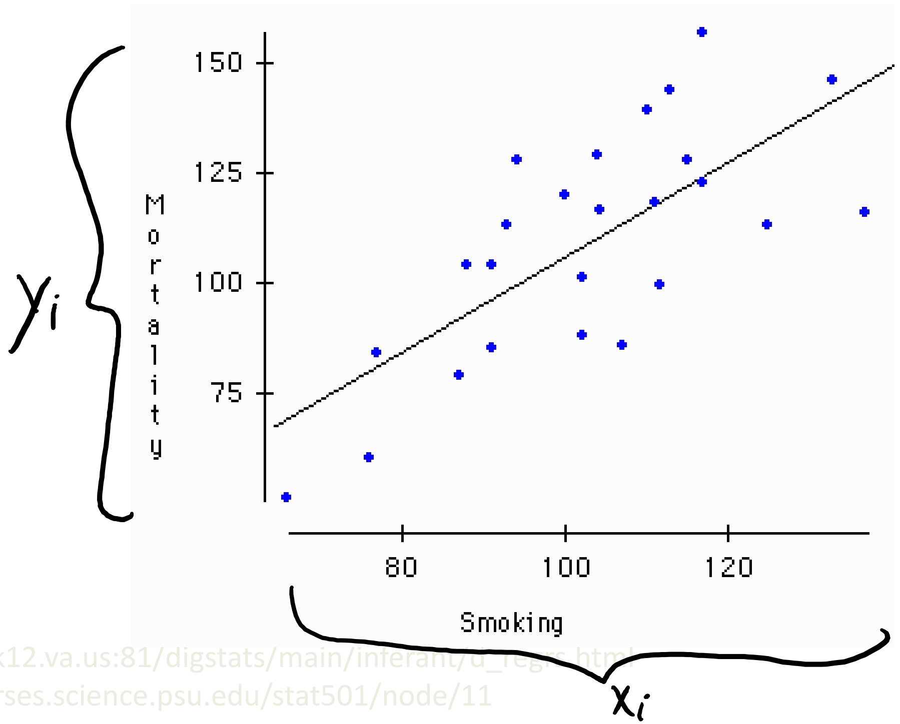
- We're going to revisit supervised learning:

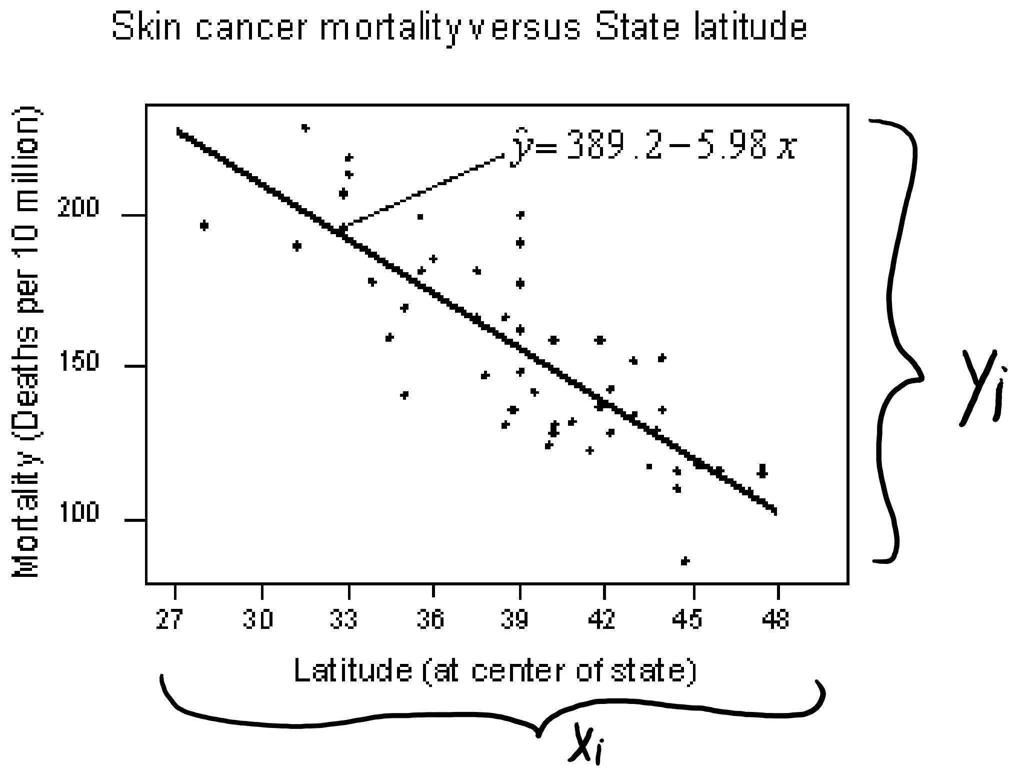
$$X = \begin{bmatrix} & & & \\ & & & \end{bmatrix} \quad y = \begin{bmatrix} \\ \end{bmatrix}$$

- Previously, we considered classification:
  - We assumed y_i was discrete: y_i = 'spam' or y_i = 'not spam'.
- Now we're going to consider regression:
  - We allow y_i to be numerical: y_i = 10.34cm.

Example: Dependent vs. Explanatory Variables

- We want to discover relationship between numerical variables:
  - Does number of lung cancer deaths change with number of cigarettes?
  - Does number of skin cancer deaths change with latitude?

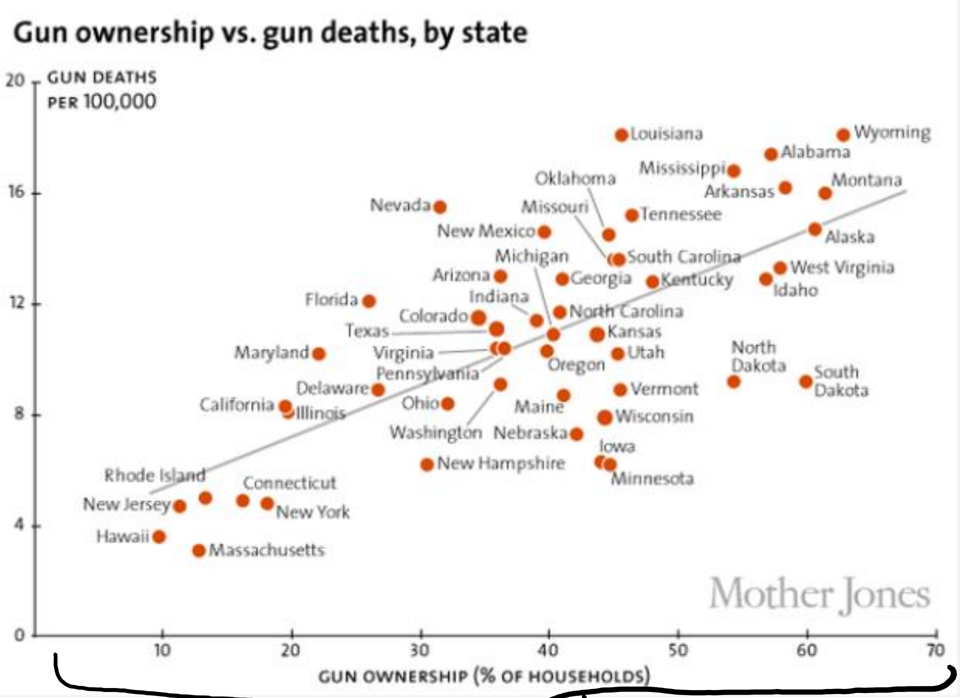




http://www.cvgs.k12.va.us:81/digstats/main/inferant/d_regrs.html
https://onlinecourses.science.psu.edu/stat501/node/11

Example: Dependent vs. Explanatory Variables

- We want to discover relationship between numerical variables:
  - Does number of lung cancer deaths change with number of cigarettes?
  - Does number of skin cancer deaths change with latitude?
  - Does number of gun deaths change with gun ownership?



y_i

x_i

http://www.vox.com/2015/10/3/9444417/gun-violence-united-states-america

Handling Numerical Labels

- One way to handle numerical y_i: discretize.
  - E.g., for ‘age’ could we use {‘age ≤ 20’, ‘20 < age ≤ 30’, ‘age > 30’}.
  - Now we can apply methods for classification to do regression.
  - But coarse discretization loses resolution.
  - And fine discretization requires lots of data.
- There exist regression versions of classification methods:
  - Regression trees, probabilistic models, non-parametric models.
- Today: one of oldest, but still most popular/important methods:
  - Linear regression based on squared error.
  - Very interpretable and the building block for more-complex methods.

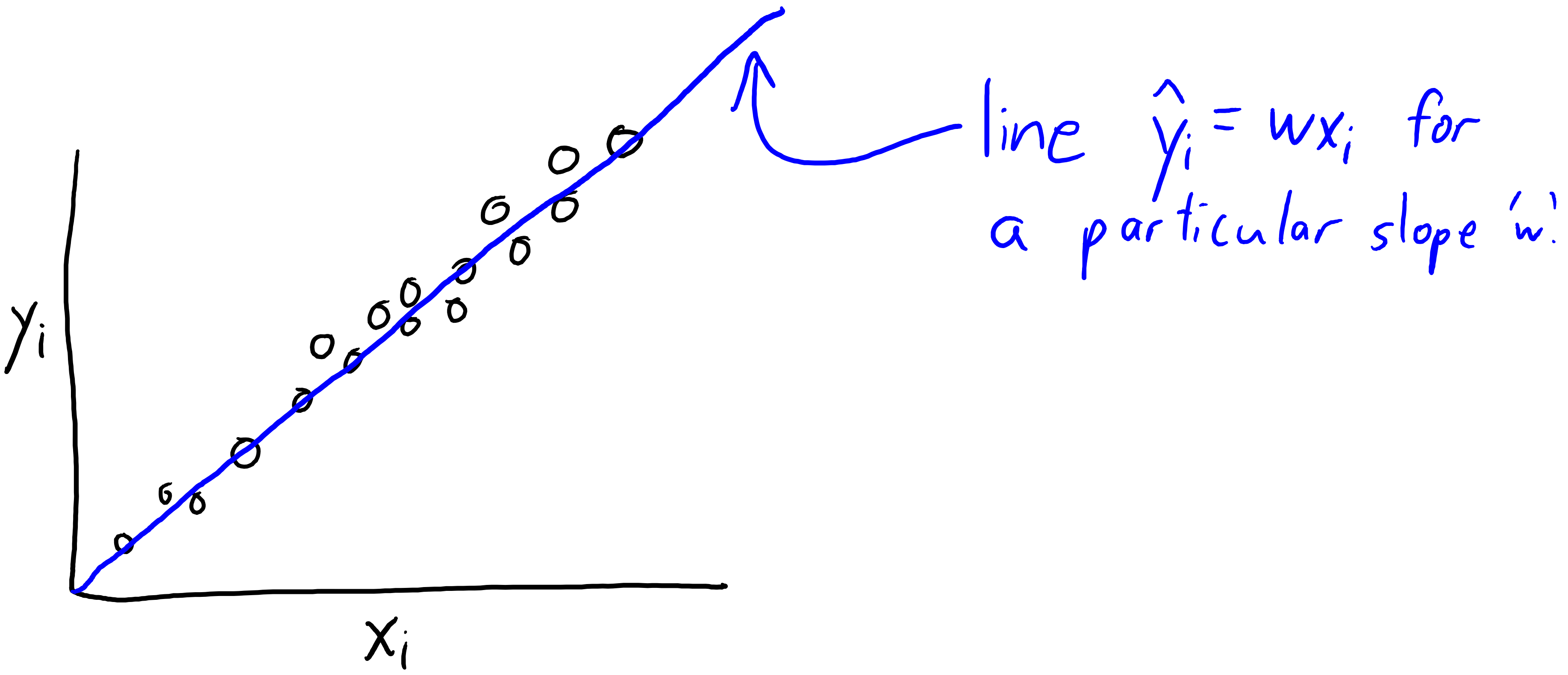
Linear Regression in 1 Dimension

- Assume we only have 1 feature (d = 1):
  - E.g., x_i is number of cigarettes and y_i is number of lung cancer deaths.
- Linear regression makes predictions $\hat{y}_i$ using a linear function of x_i:

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

- The parameter 'w' is the weight or regression coefficient of x_i.
- As x_i changes, slope 'w' affects the rate that $\hat{y}_i$ increases/decreases:
  - Positive 'w': $\hat{y}_i$ increase as x_i increases.
  - Negative 'w': $\hat{y}_i$ decreases as x_i increases.

Linear Regression in 1 Dimension



Aside: terminology woes

- Different fields use different terminology and symbols.
  - Data points = **objects** = **examples** = rows = observations.
  - **Inputs** = predictors = **features** = explanatory variables= regressors = independent variables = covariates = columns.
  - **Outputs** = outcomes = targest = response variables = dependent variables (also called a “label” if it’s categorical).
  - Regression coefficients = **weights** = parameters = betas.
- With linear regression, the symbols are inconsistent too:
  - In ML, the data is X and the weights are w.
  - In statistics, the data is X and the weights are β.
  - In optimization, the data is A and the weights are x.

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:

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

- But we can't use the same error as before:
  - Even if data comes from a linear model but has noise, we can have $\hat{y}_i \neq y_i$ for all training examples 'i' for the "best" model

Least Squares Objective

- We need a way to evaluate numerical error.
- Classic way is setting slope 'w' to minimize sum of squared errors:

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

Sum up the squared differences over all training examples.

Our prediction $\hat{y}_i$

True value of y_i

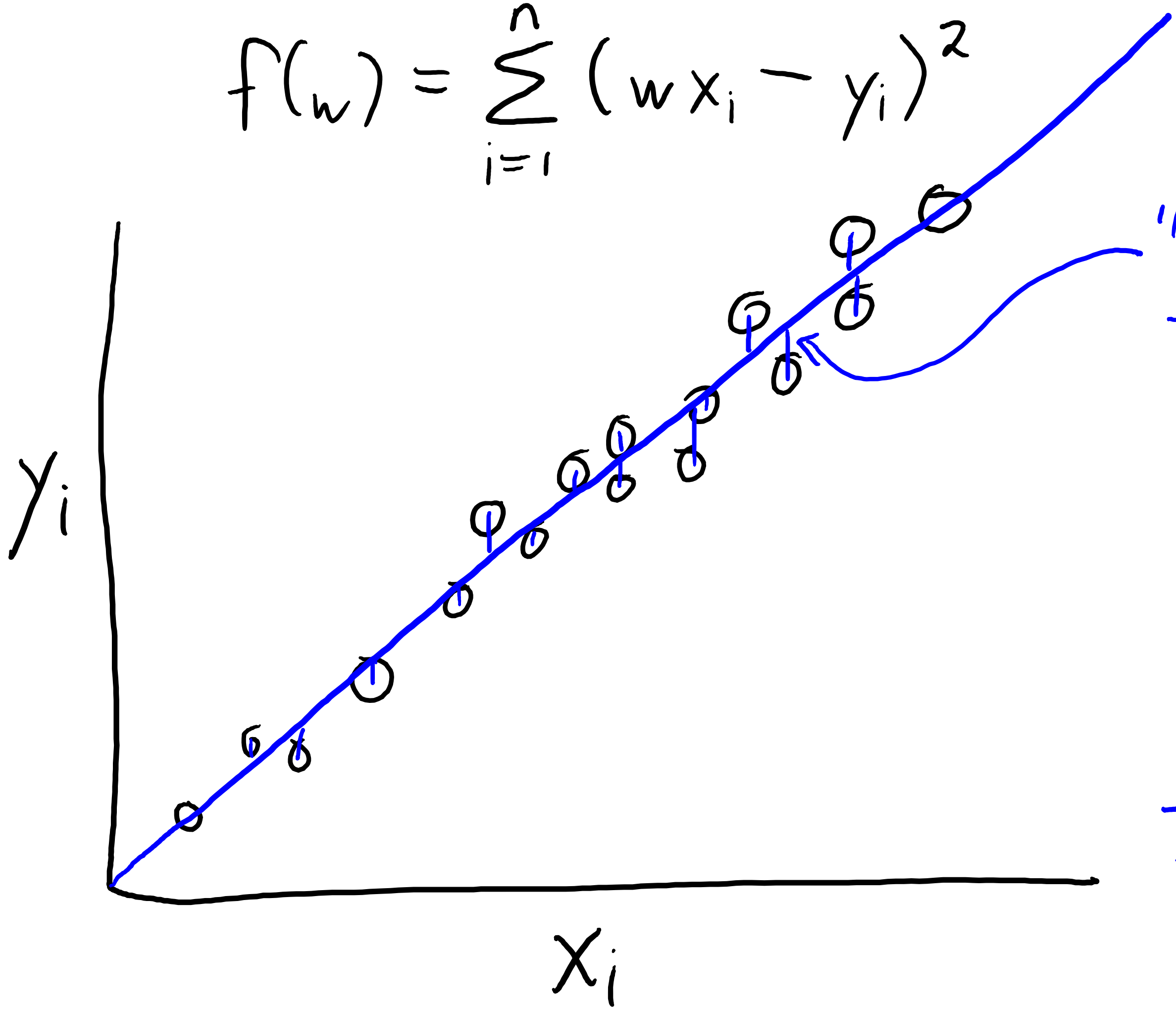
Difference between prediction and true value for example 'i'.

- 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.

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)

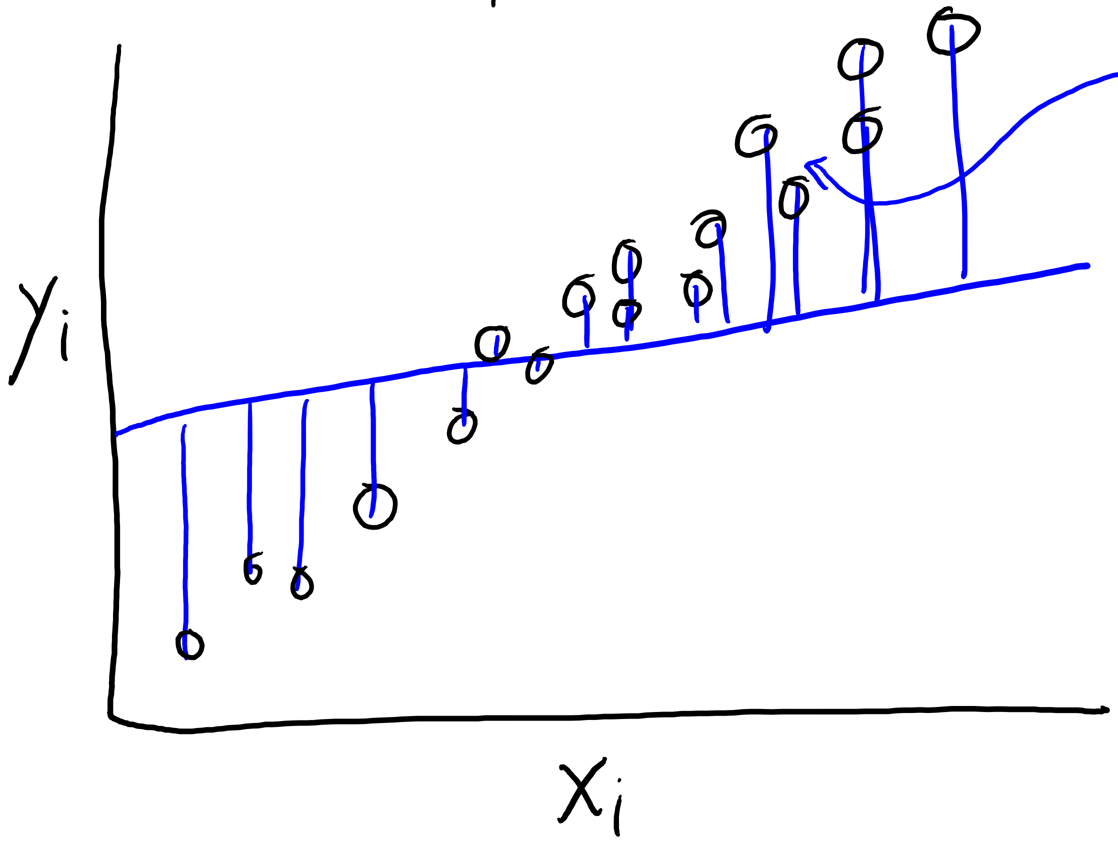
↓

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:

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

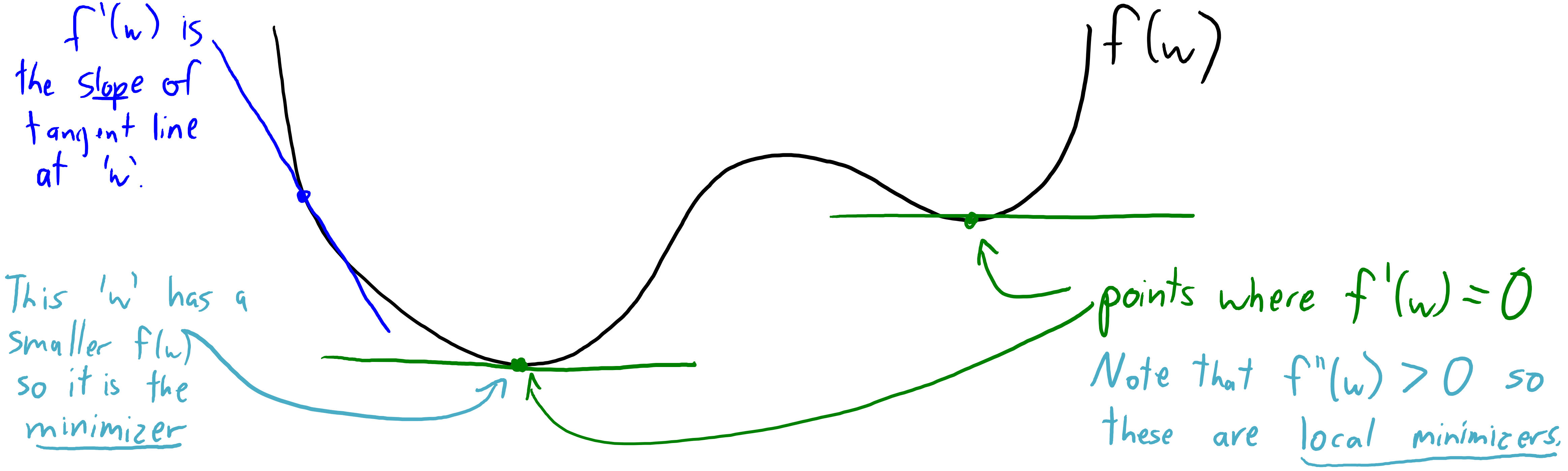


"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

If this error is 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 (but 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.
  - Gradient will still be zero at the same locations.
  - We'll use this trick a lot!

Finding Least Squares Solution

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

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(wx_i - y_i)^2 = \frac{1}{2}(wx_1 - y_1)^2 + \frac{1}{2}(wx_2 - y_2)^2 + \cdots + \frac{1}{2}(wx_n - y_n)^2$$

$$f'(w) = \sum_{i=1}^{n}(wx_i - y_i)x_i = (wx_1 - y_1)x_1 + (wx_2 - y_2)x_2 + \cdots + (wx_n - y_n)x_n$$

Set $f'(w) = 0$: $\sum_{i=1}^{n}(wx_i - y_i)x_i = 0$ or $\sum_{i=1}^{n}[wx_i^2 - y_i x_i] = 0$

or $\sum_{i=1}^{n} wx_i^2 = \sum_{i=1}^{n} y_i x_i$

or $w\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i$

so $\boxed{w = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} x_i^2}}$

Is this a minimizer?

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

Since $(\text{anything})^2$ is non-negative, $f''(w) \geq 0$.

If at least one $x_i \neq 0$ then $f''(w) > 0$ and this is a minimizer.

Motivation: Combining Explanatory Variables

- Smoking is not the only contributor to lung cancer.
  - For example, 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.

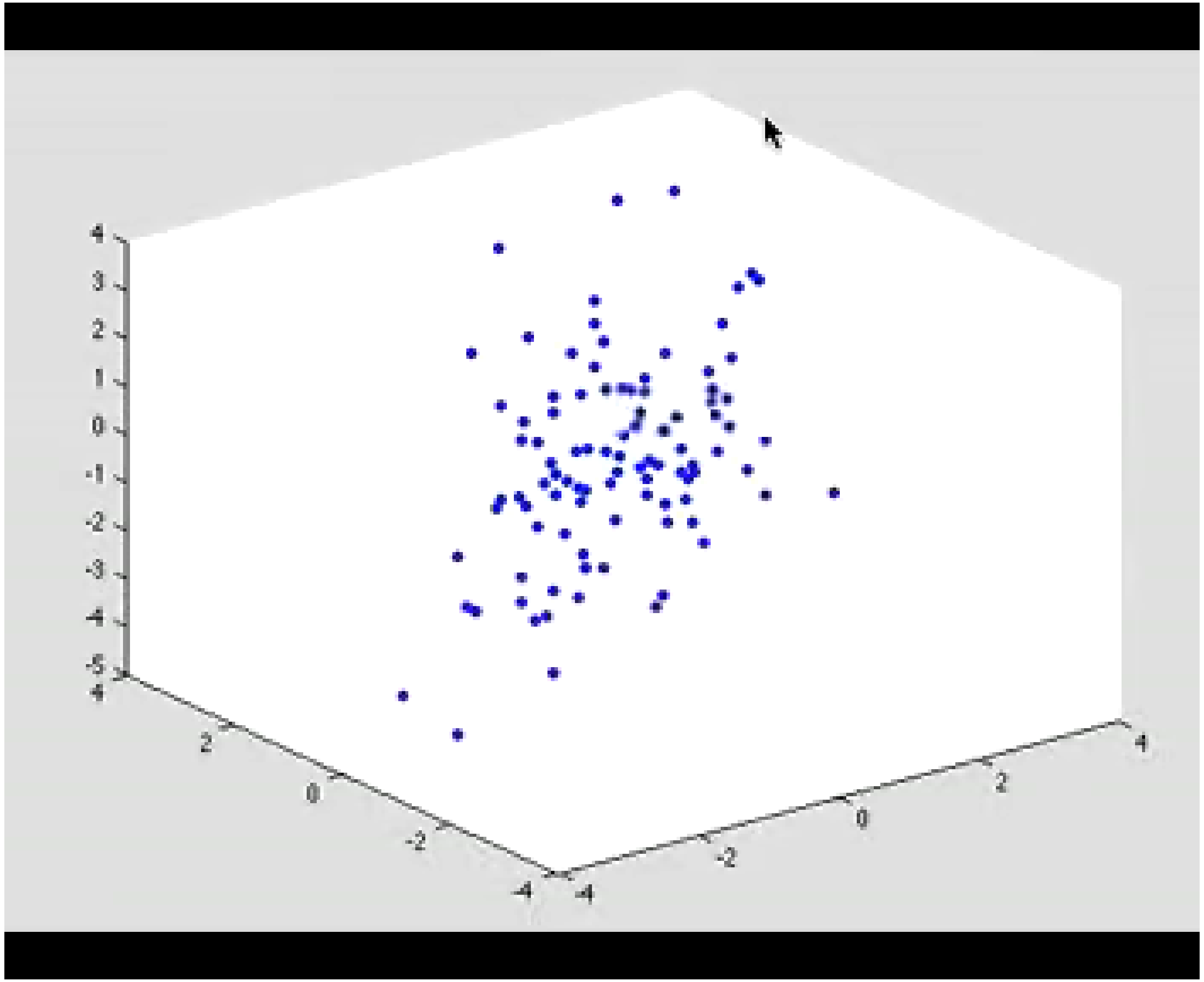
"weight" of feature 1

Value of feature 1 in example 'i'

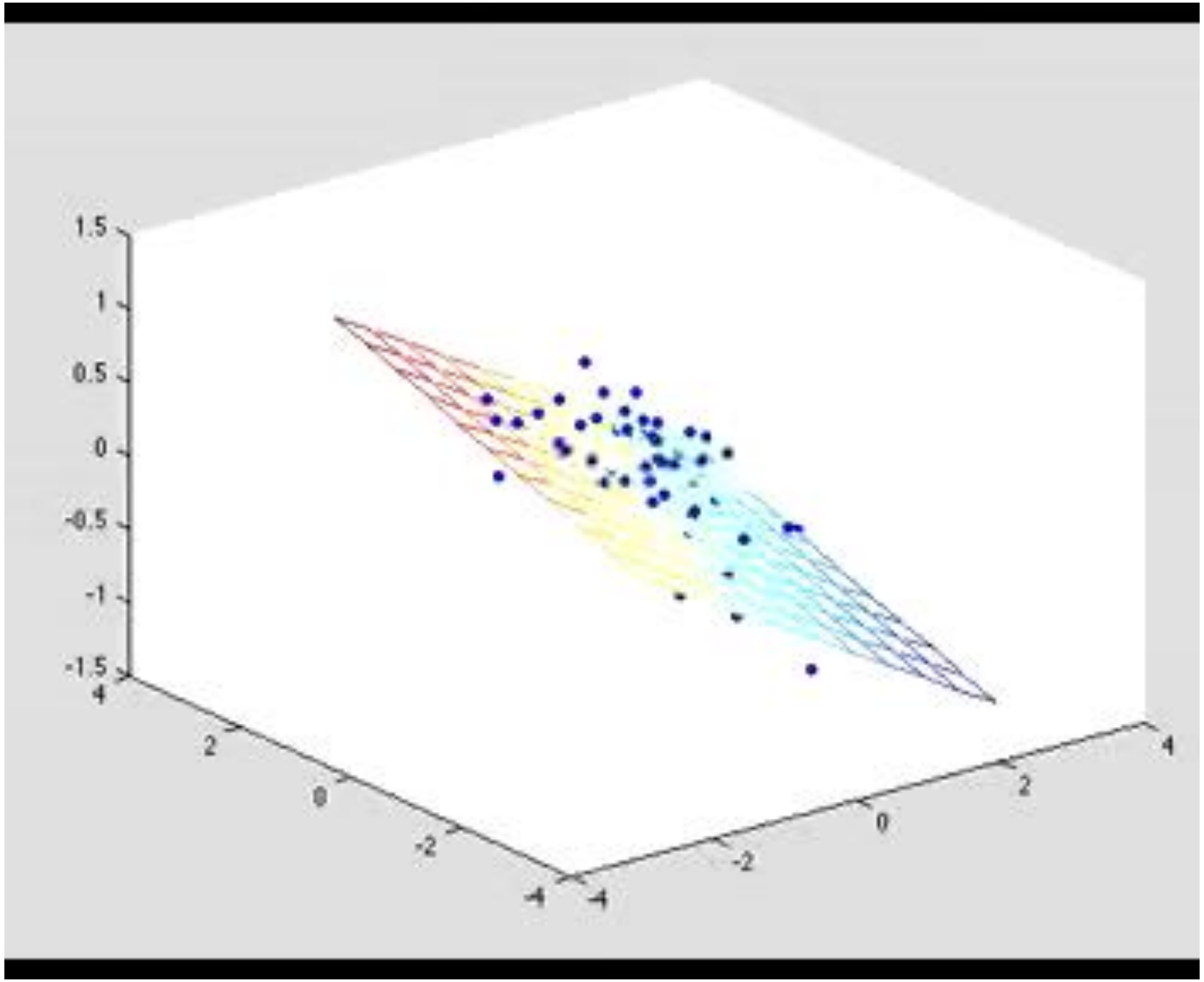
- We have a weight w_1 for feature '1' and w_2 for feature '2':

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

Least Squares in 2-Dimensions

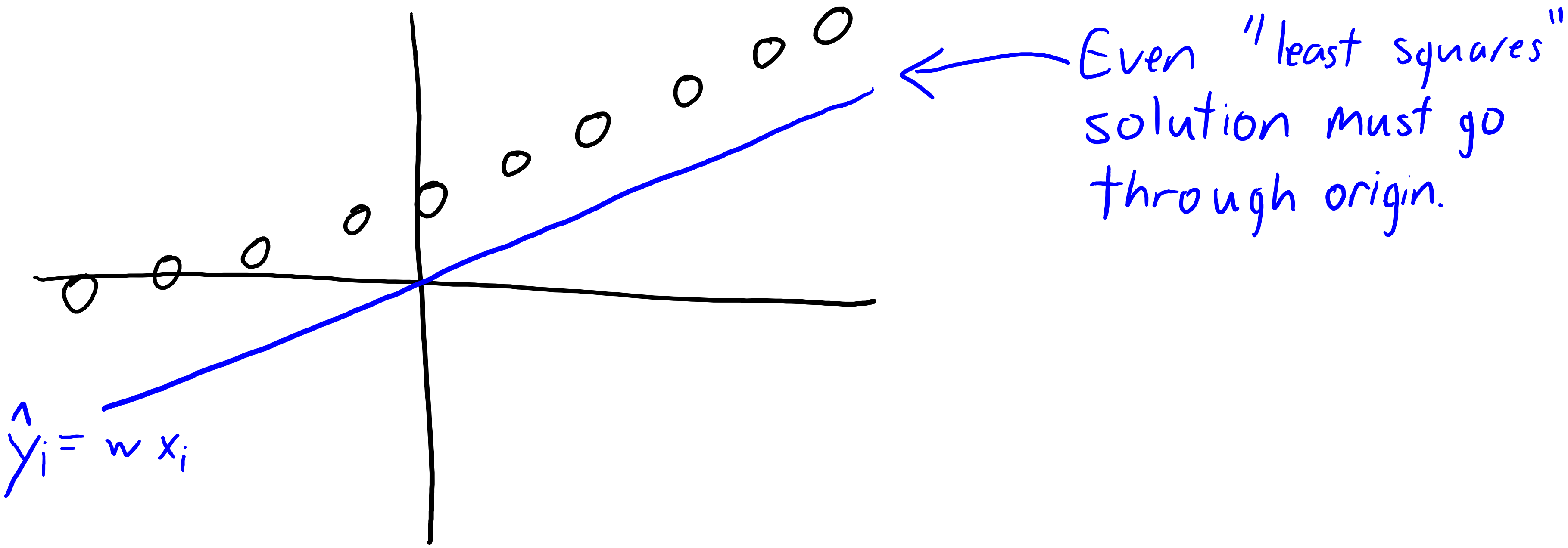


Least Squares in 2-Dimensions



Why don't we have a y-intercept?

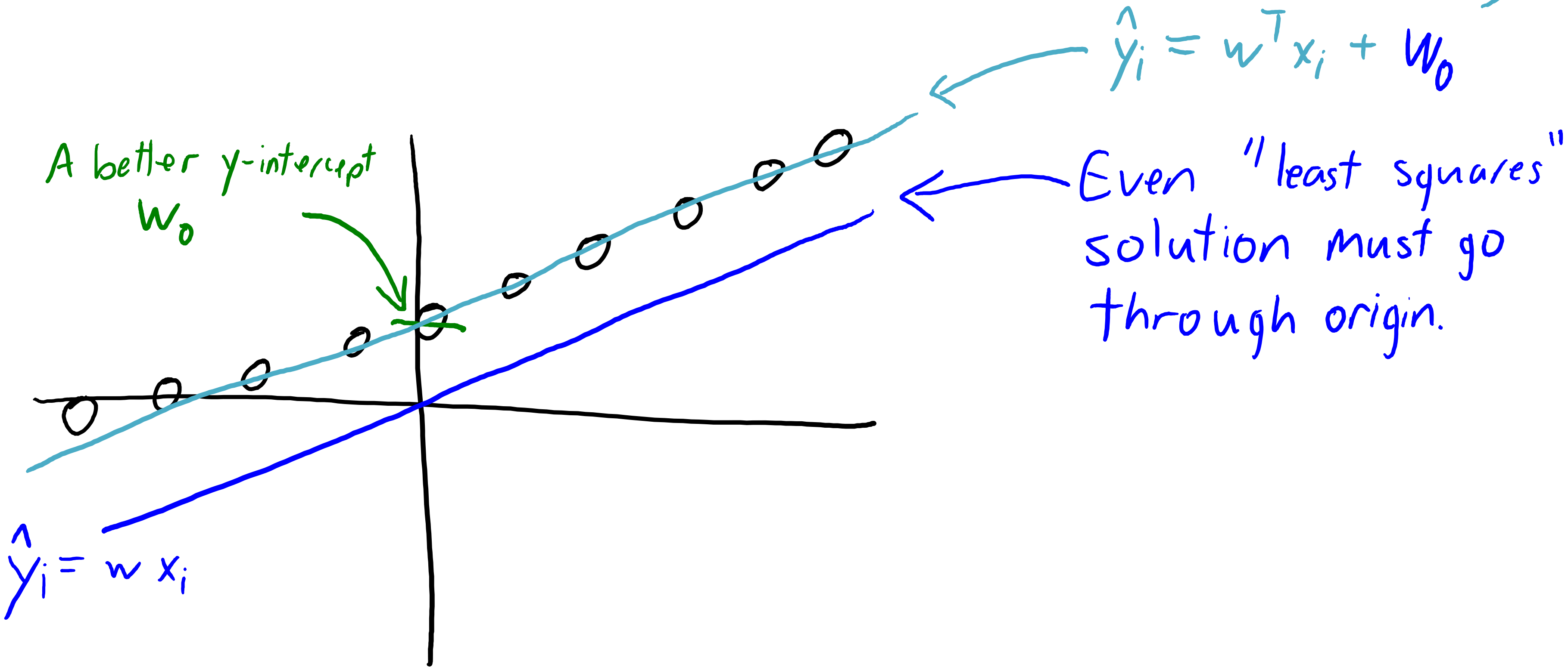
– Linear model is $\hat{y}_i = wx_i$ instead of $\hat{y}_i = wx_i + w_0$ with y-intercept w_0.
– Without an intercept, if $x_i = 0$ then we must predict $\hat{y}_i = 0$.



Why don't we have a y-intercept?

– Linear model is $\hat{y}_i = wx_i$ instead of $\hat{y}_i = wx_i + w_0$ with y-intercept w_0.
– Without an intercept, if $x_i = 0$ then we must predict $\hat{y}_i = 0$.

Adding y-intercept fixes this.



Adding a Bias Variable

- Simple trick to add a y-intercept ("bias") variable:
  - Make a new matrix "Z" with an extra feature that is always "1".

$$X = \begin{bmatrix} -0.1 \\ 0.3 \\ 0.2 \end{bmatrix} \qquad Z = \begin{bmatrix} \underbrace{\begin{matrix} 1 \\ 1 \\ 1 \end{matrix}}_{\text{"always 1"}} & \underbrace{\begin{matrix} -0.1 \\ 0.3 \\ 0.2 \end{matrix}}_{X} \end{bmatrix}$$

- Now use "Z" as your features in linear regression.
  - We'll use 'v' instead of 'w' as regression weights when we use features 'Z'.

$$\hat{y}_i = \underset{\downarrow\, w_0}{v_1}\, \underset{\downarrow\, 1}{z_{i1}} + \underset{\downarrow\, w_1}{v_2}\, \underset{\downarrow\, x_{i1}}{z_{i2}} = w_0 + w_1 x_{i1}$$

- So we can have a non-zero y-intercept by changing features.
  - This means we can ignore the y-intercept in our derivations, which is cleaner.

Least Squares in d-Dimensions

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

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

- 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:

$$\hat{y}_i = \underbrace{w^T x_i}_{\text{"inner product" between vectors}}$$

$$w^T x = \overbrace{\begin{bmatrix} w_1 & w_2 & \cdots & w_d \end{bmatrix}}^{w^T} \overbrace{\begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}}^{x_i} = \sum_{j=1}^{d} w_j x_{ij}$$

- In words, our model is that the output is a weighted sum of the inputs.

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:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(\underbrace{w^T x_i}_{\text{prediction is inner product of 'w' and 'x_i'}} - y_i)^2$$

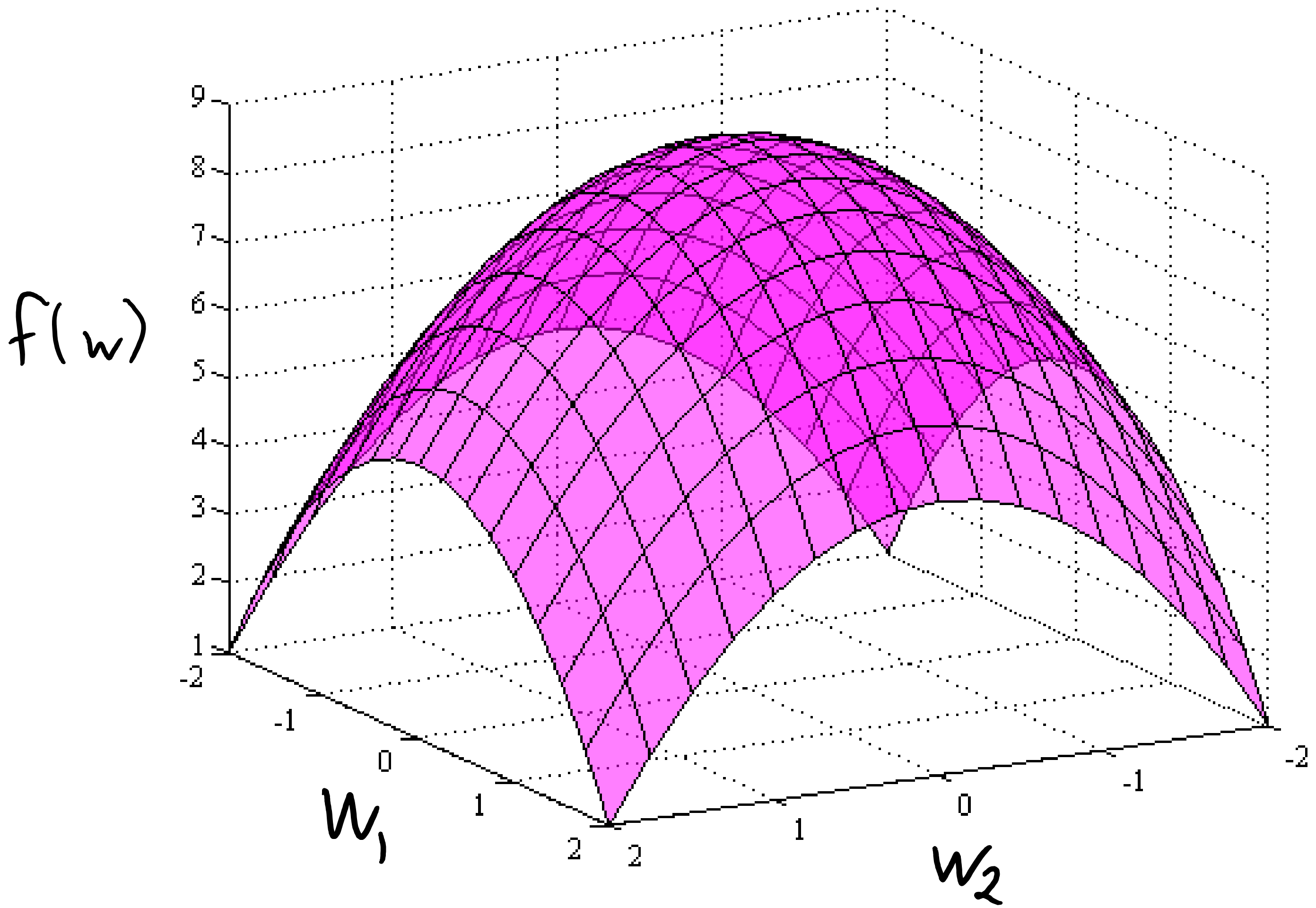
'w' is now a vector

prediction is inner product of 'w' and 'x_i' (linear combination of features)

"Error" is still the sum of squared differences between "true" y_i and our "prediction" $w^T x_i$

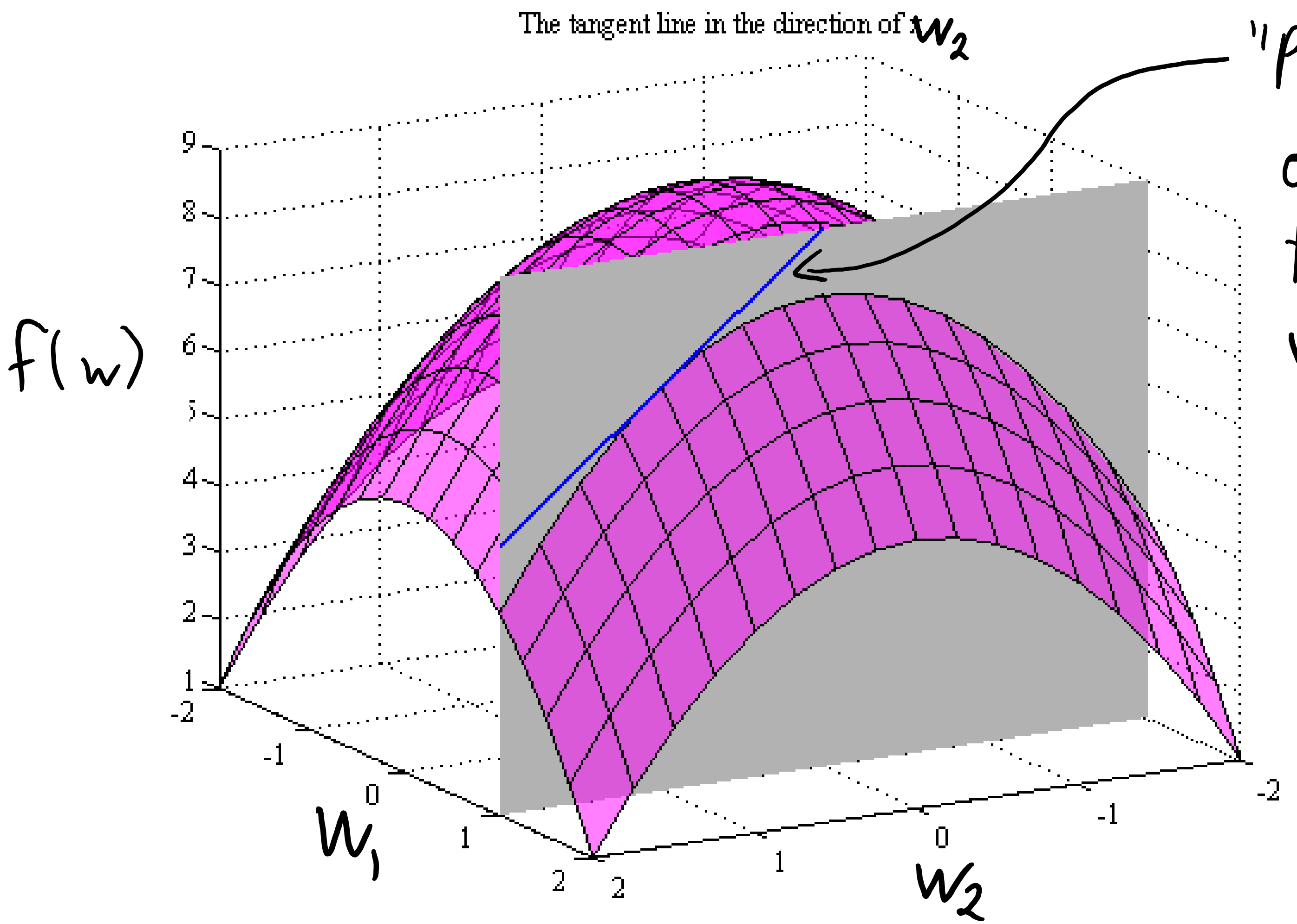
- How do we find the best vector 'w'?
  - Set the derivative of each variable ("partial derivative") to 0?

Partial Derivatives



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 when all other variables are held fixed.

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

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

Summary

- Regression considers the case of a numerical y_i.
- Least squares is a classic method for fitting linear models.
  - With 1 feature, it has a simple closed-form solution.
- Gradient is vector containing partial derivatives of all variables.

- Next time:

minimizing $\frac{1}{2}\sum_{i=1}^{n}(w^Tx_i - y_i)^2$ in terms of 'w' is:

$$w = (X'X)\backslash(X'y)$$

(in Julia)