CPSC 340: Machine Learning and Data Mining

Least Squares

Spring 2022 (2021W2)

Admin

- Assignment 2:
  - Due this Friday! (Or, with late days, Saturday or Sunday.)

- We're going to start using calculus and linear algebra a lot.
  - You should start reviewing these ASAP if you are rusty.
  - A review of relevant calculus concepts is here.
  - A review of relevant linear algebra concepts is here.

Supervised Learning Round 2: Regression

- We’re going to revisit supervised learning:

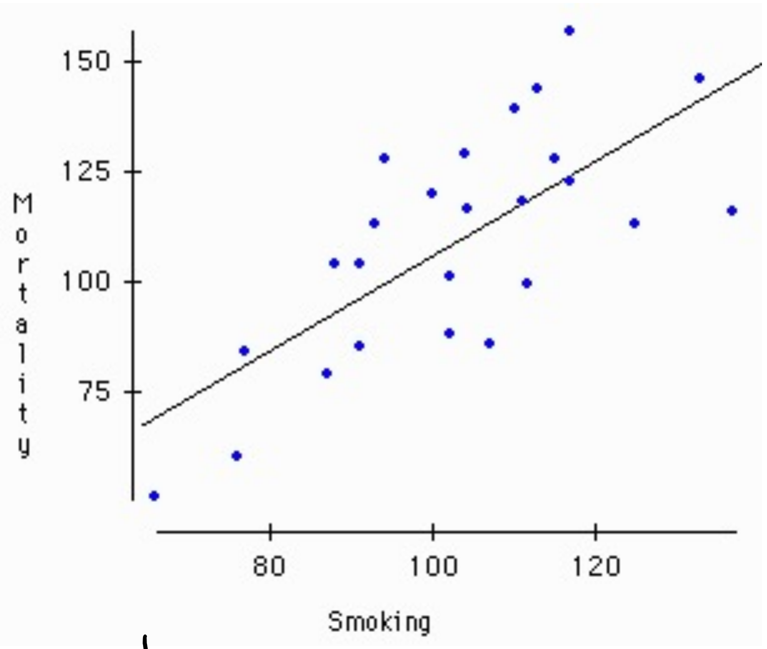
$$X = \begin{bmatrix} \quad & \quad & \quad \\ & & \\ & & \end{bmatrix} \qquad y = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix}$$

- 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.

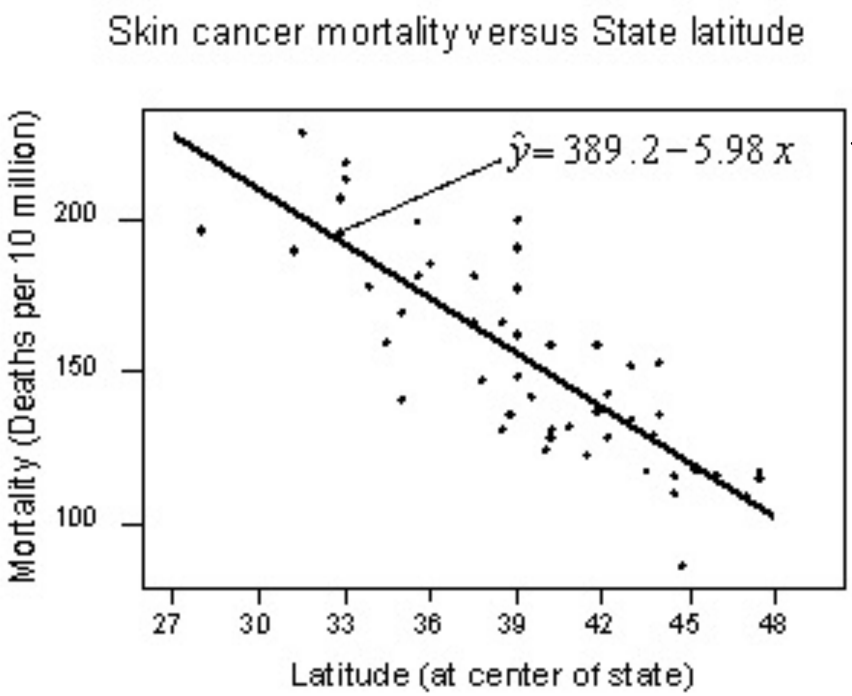
Example: Dependent vs. Explanatory Variables

- 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?

y_i



x_i



y_i

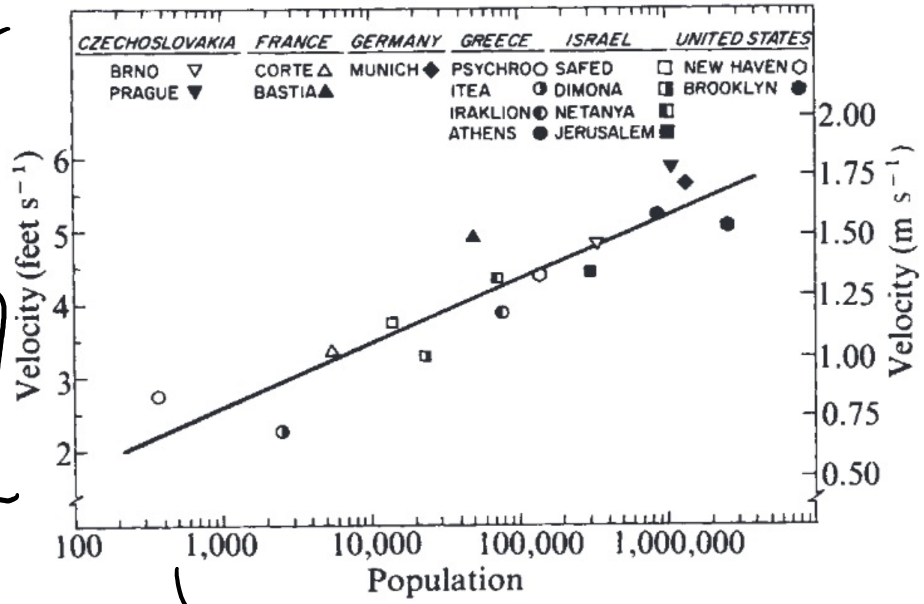
x_i

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

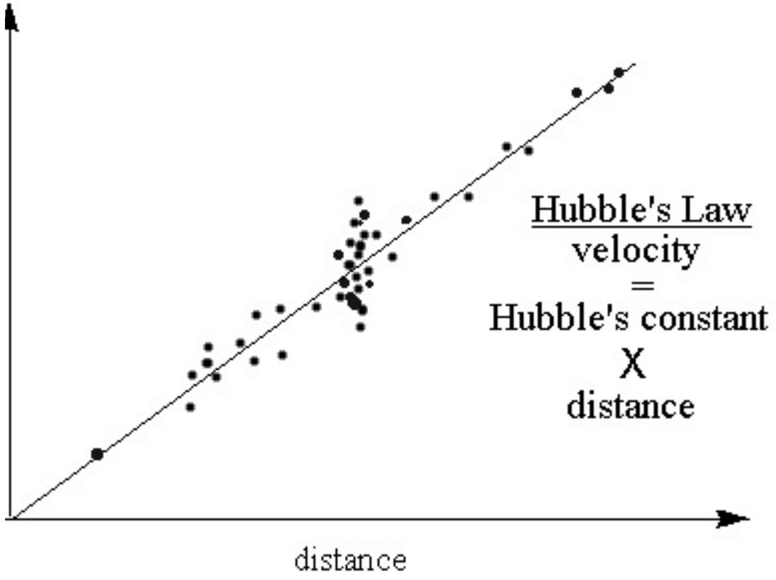
Example: Dependent vs. Explanatory Variables

- We want to discover relationship between numerical variables:
  - Do people in big cities walk faster?
  - Is the universe expanding or shrinking or staying the same size?

y_i



x_i



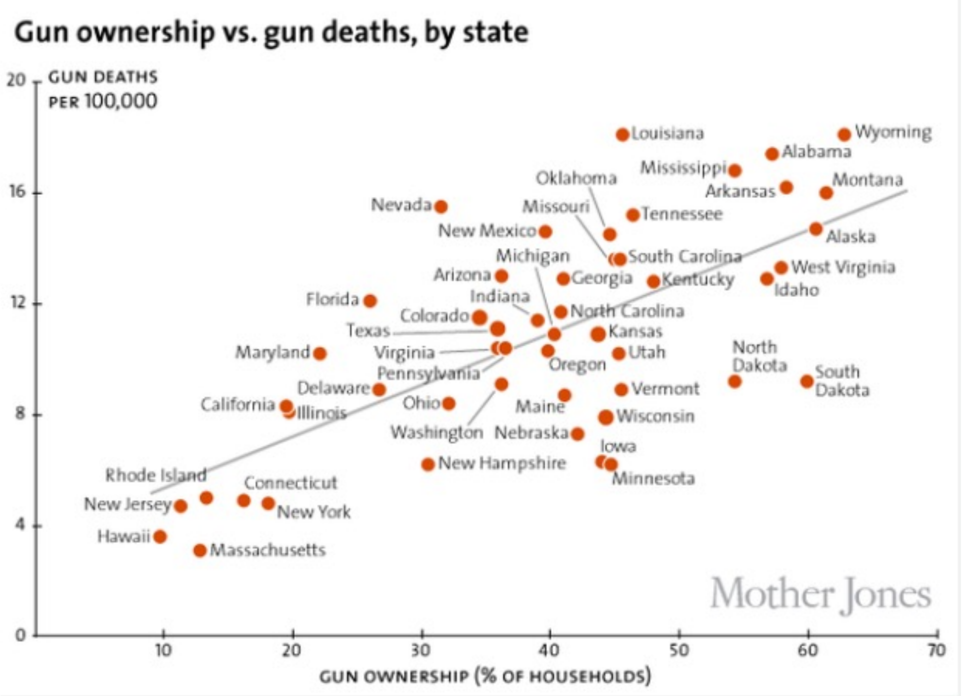
y_i

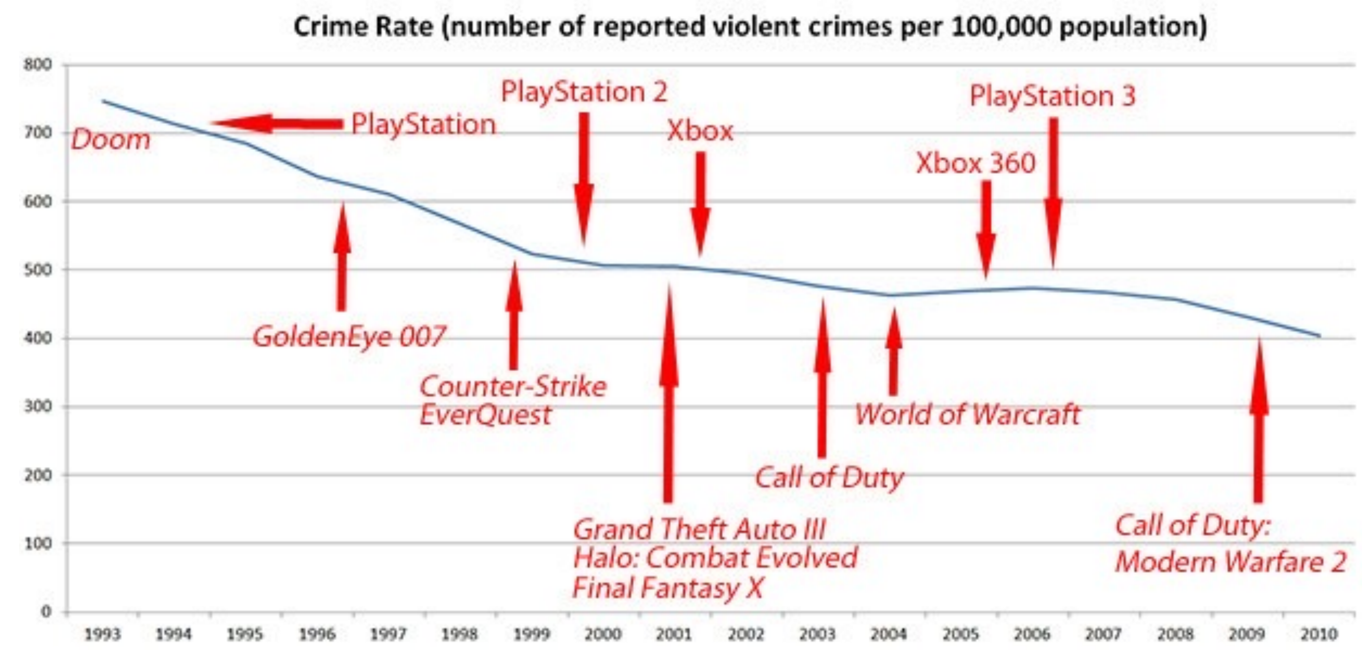
x_i

http://hosting.astro.cornell.edu/academics/courses/astro201/hubbles_law.htm
https://www.nature.com/articles/259557a0.pdf

Example: Dependent vs. Explanatory Variables

- We want to discover relationship between numerical variables:
  - Does number of gun deaths change with gun ownership?
  - Does number violent crimes change with violent video games?



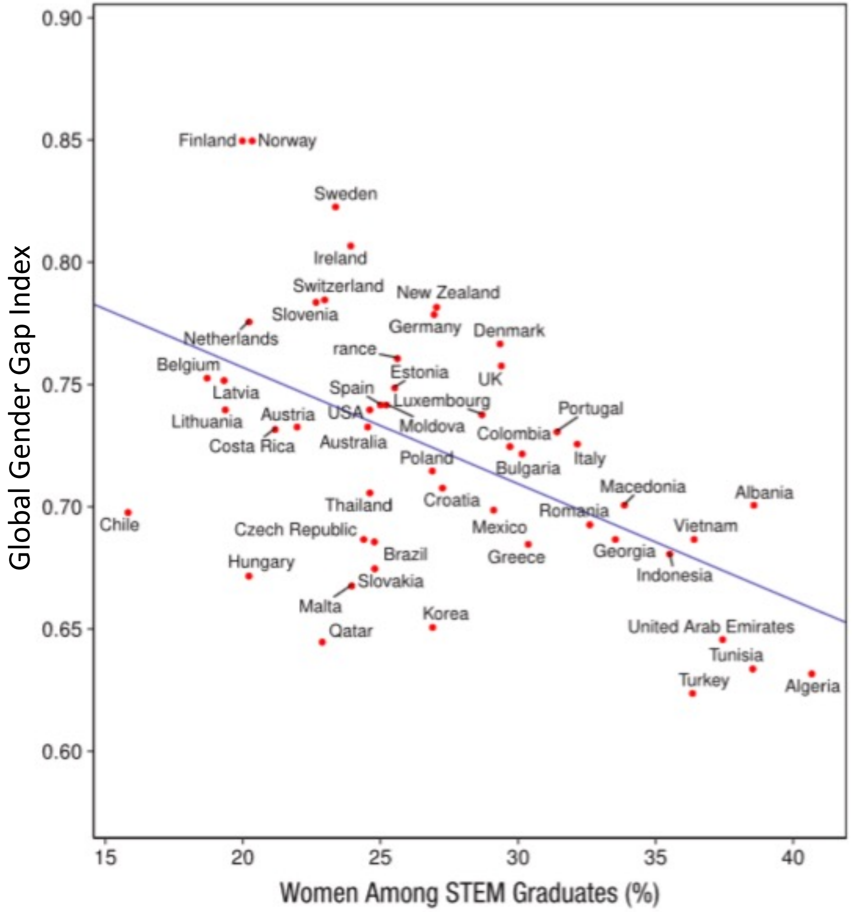


http://www.vox.com/2015/10/3/9444417/gun-violence-united-states-america
https://www.soundandvision.com/content/violence-and-video-games

Example: Dependent vs. Explanatory Variables

- We want to discover relationship between numerical variables:
  - Does higher gender equality index lead to more women STEM grads?

- Not that we're doing supervised learning:
  - Trying to predict value of 1 variable (the 'y_i' values). (instead of measuring correlation between 2).
- Supervised learning does not give causality:
  - OK: "Higher index is correlated with lower grad %".
  - OK: "Higher index helps predict lower grad %".
  - BAD: "Higher index leads to lower grads %".
    - People/media get these confused all the time, be careful!
    - There are lots of potential reasons for this correlation.



https://www.weforum.org/agenda/2018/02/does-gender-equality-result-in-fewer-female-stem-grads/

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.
  - Interpretable and the building block for more-complex methods.

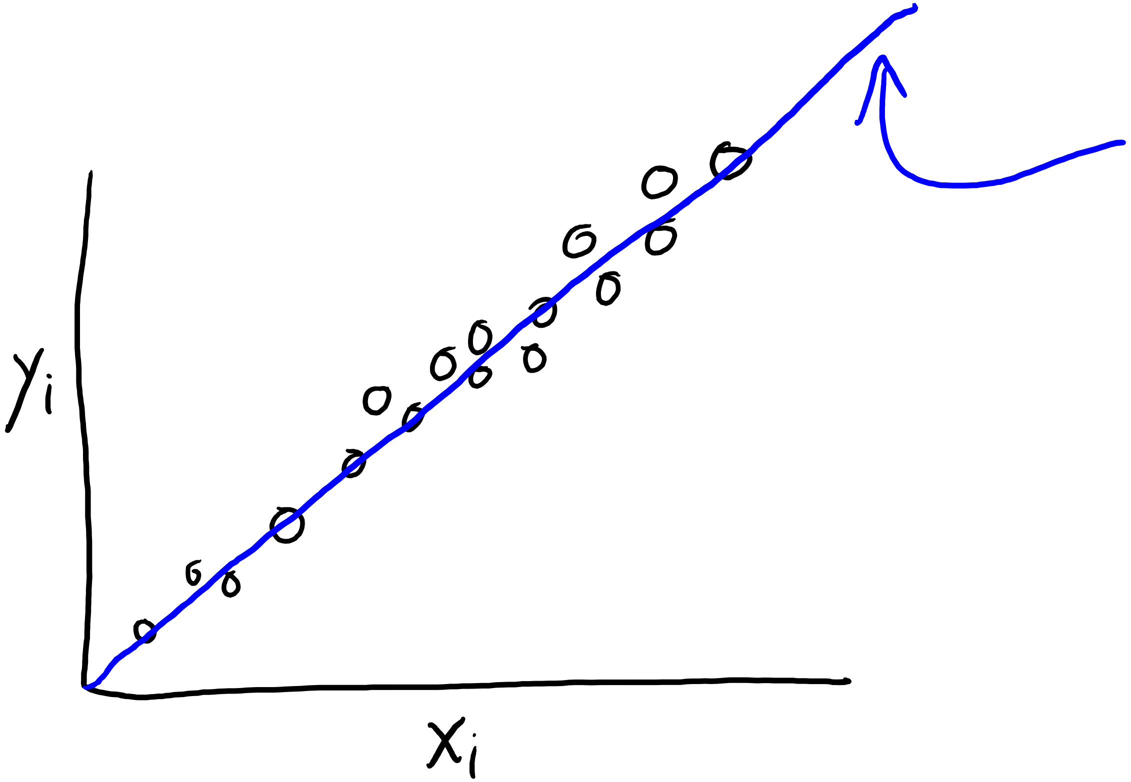
Linear Regression in 1 Dimension

- 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.
  - We're temporarily ignoring the y-intercept.
- 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



line $\hat{y}_i = wx_i$ for a particular slope 'w'.

bonus!

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 = targets = response variables = dependent variables = labels (especially if it's categorical).
  - Regression coefficients = **weights** = parameters = betas.
- With linear regression, the symbols are inconsistent too:
  - In ML, the data is X and y, and the weights are w; X is n by d.
  - In statistics, the data is X and y, and the weights are β; X is n by p.
  - In optimization, the data is A and b, and the weights are x; X is m by n.

Is linear regression "really" machine learning?

bonus!

Statisticians might hate it....

Darren Dahly, PhD @statsepi · May 4
This nonsense is everywhere now.

> Darren Dahly, PhD @statsepi · Apr 8
> THERE ISN'T ANY F**KING "AI" IN THIS PAPER. nature.com/articles/s4158...
> Show this thread

1 7

Lior Pachter @lpachter

Replying to @statsepi

Oh I know... logistic regression is "AI". Linear regression is "machine learning".

...but by any reasonable definition of ML, yes.

One rough "definition" of ML:
you can publish about it at NeurIPS

On Uniform Convergence and Low-Norm Interpolation Learning

[NeurIPS 2020]

Lijia Zhou University of Chicago zlj@uchicago.edu

Danica J. Sutherland TTI-Chicago danica@ttic.edu

Nathan Srebro TTI-Chicago nati@ttic.edu

Abstract

We consider an underdetermined noisy linear regression model where the minimum-norm interpolating predictor is known to be consistent, and ask: can

Uniform Convergence of Interpolators: Gaussian Width, Norm Bounds and Benign Overfitting

[NeurIPS 2021]

Frederic Koehler* MIT fkoehler@mit.edu

Lijia Zhou* University of Chicago zlj@uchicago.edu

Danica J. Sutherland UBC and Amii dsuth@cs.ubc.ca

Nathan Srebro TTI-Chicago nati@ttic.edu

Collaboration on the Theoretical Foundations of Deep Learning (deepfoundations.ai)

Abstract

We consider interpolation learning in high-dimensional linear regression with Gaussian data, and prove a generic uniform convergence guarantee on the general-

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:
  - Usually don't have a line where $\hat{y}_i = yi$ exactly for many points n.
    - Sampling noise, relationship not being quite linear, or even just floating-point issues.
  - "Best" model may have $|\hat{y}_i - y_i|$ small but not exactly 0.

Least Squares Objective

- Instead of "exact y_i", we evaluate "size" of the error in prediction.
- Classic way is setting slope 'w' to minimize sum of squared errors:

$$f(w) = \sum_{i=1}^{n} (\underbrace{w x_i}_{\text{Our prediction } \hat{y}_i} - \underbrace{y_i}_{\text{True value of } y_i})^2$$

True value of y_i

Our prediction $\hat{y}_i$

Sum up the squared differences over all training examples.

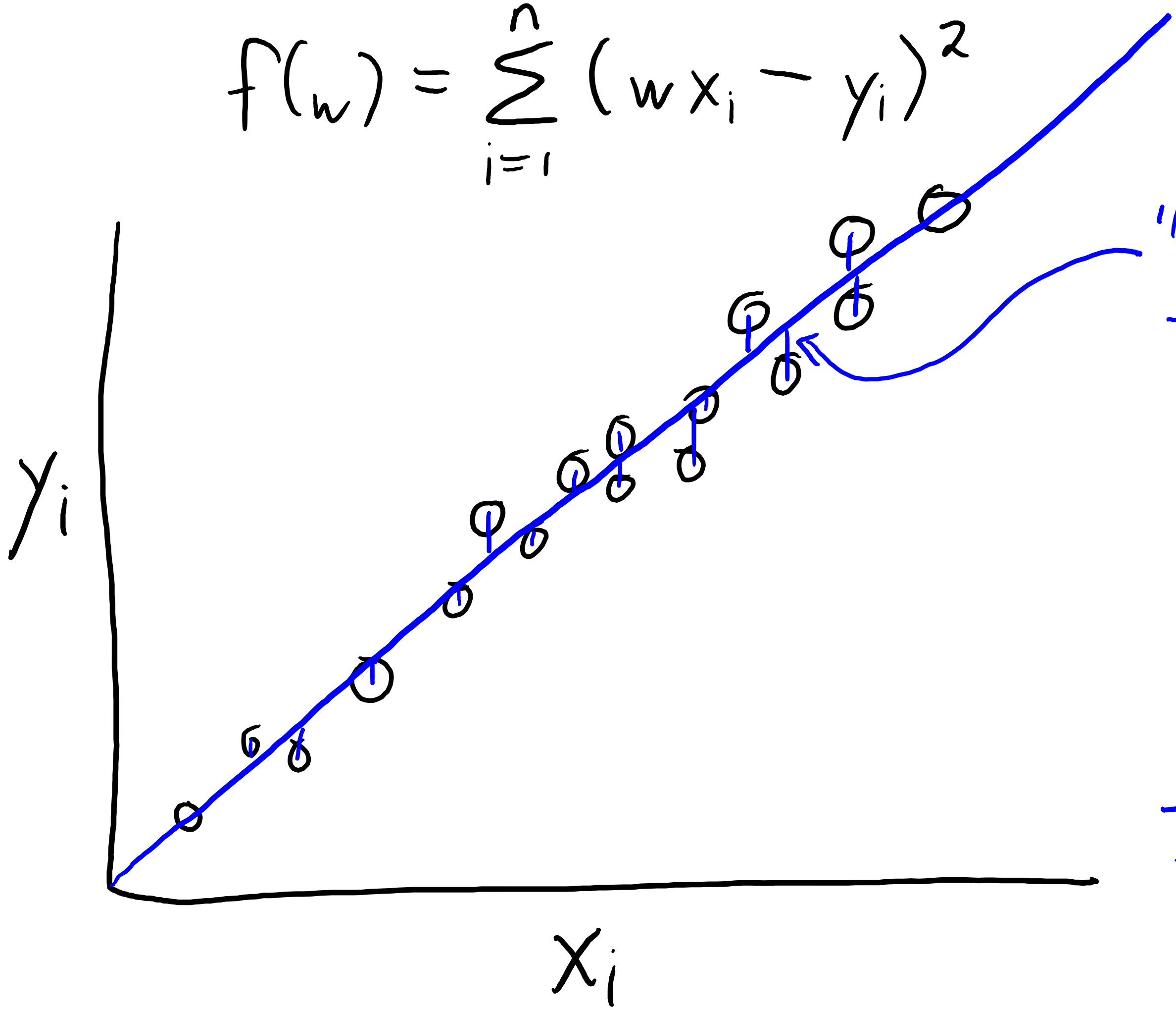
Difference between prediction and true value for example 'i'.

- There are some justifications for this choice.
  - A probabilistic interpretation is coming later in the course.
- But one strong reason is it is easy to minimize.

Least Squares Objective

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

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



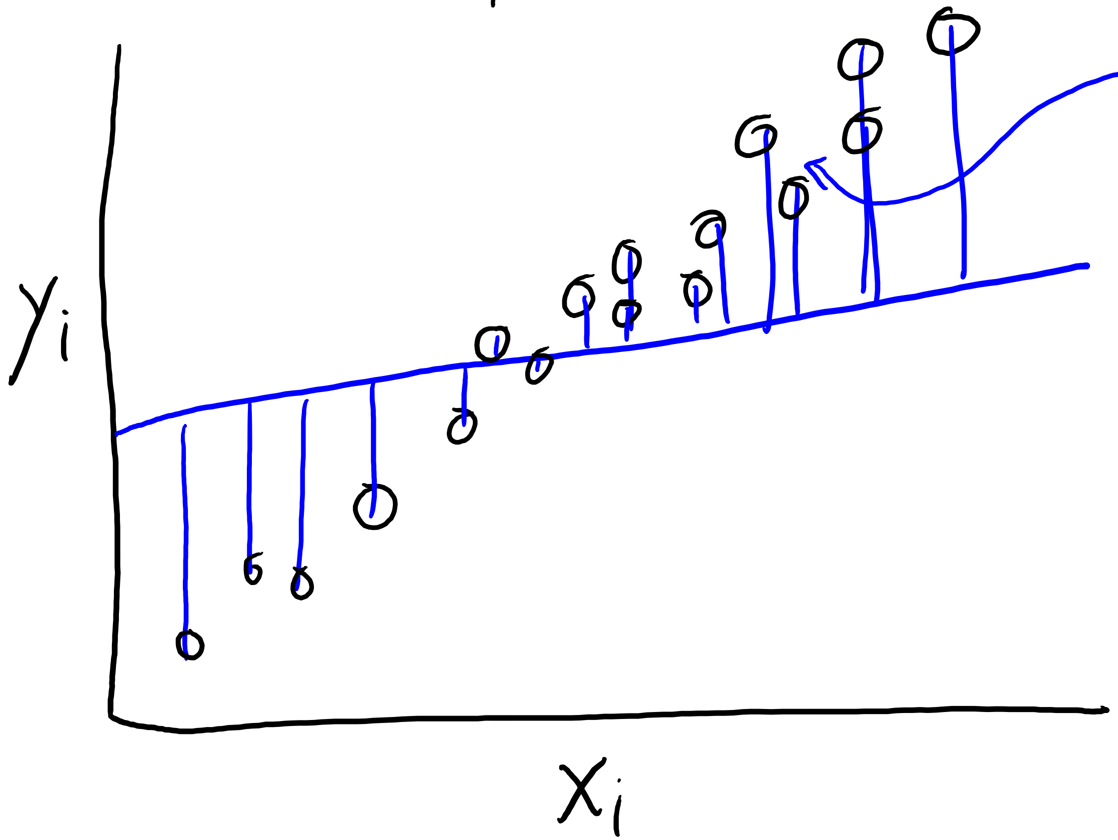
"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

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

Least Squares Objective

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

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

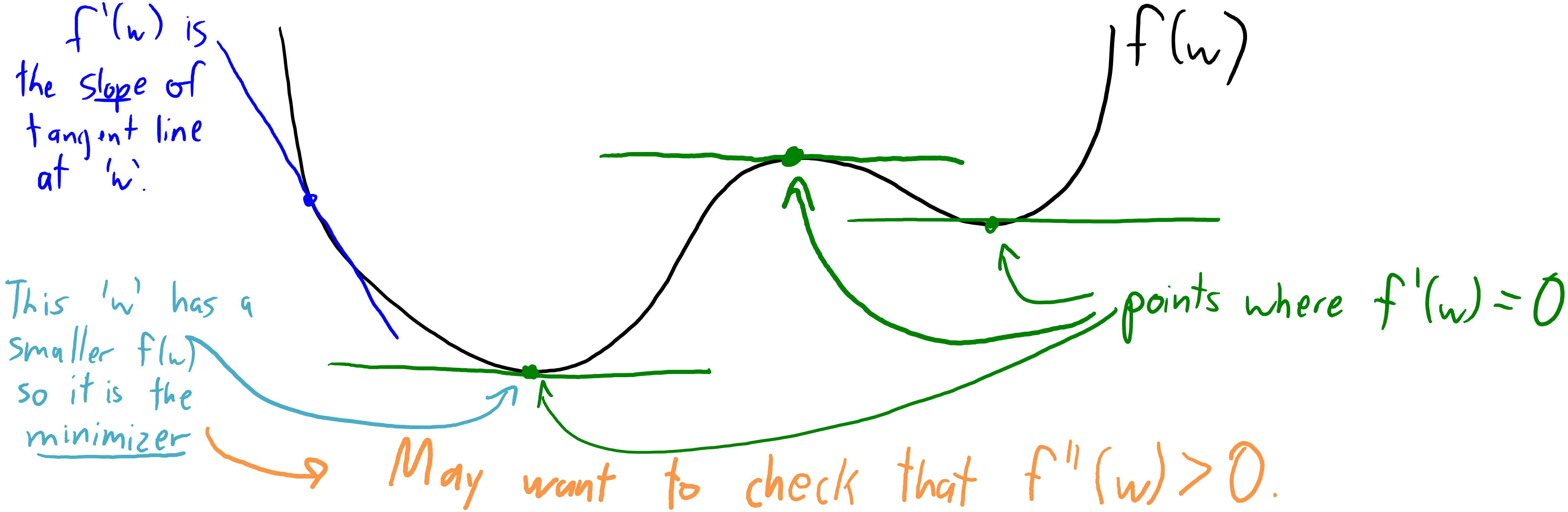


"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

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

Minimizing a Differentiable Function

- Math 100 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!

[(an extremely serious Reddit post on ethics of this)]()

Finding Least Squares Solution

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

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(wx_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{n}\left[w^2x_i^2 - 2wx_iy_i + y_i^2\right]$$ (expand square)

$$= \frac{w^2}{2}\underbrace{\sum_{i=1}^{n}x_i^2}_{\text{constant 'a'}} - w\underbrace{\sum_{i=1}^{n}x_iy_i}_{\text{constant 'b'}} + \underbrace{\frac{1}{2}\sum_{i=1}^{n}y_i^2}_{\text{constant 'c'}}$$ (split sums, take 'w' outside)

$$= \frac{w^2}{2}a - wb + c$$

Take derivative: $f'(w) = wa - b + 0$

Setting $f'(w)=0$ and solving gives $w = \frac{b}{a} = \frac{\sum_{i=1}^{n}x_iy_i}{\sum_{i=1}^{n}x_i^2}$ (exists if we have a non-zero feature)

Finding Least Squares Solution

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

Setting $f'(w) = 0$ and solving gives $w = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}$ (exists if we have one non-zero x_{ij})

- Let’s check that this is a minimizer by checking second derivative:

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

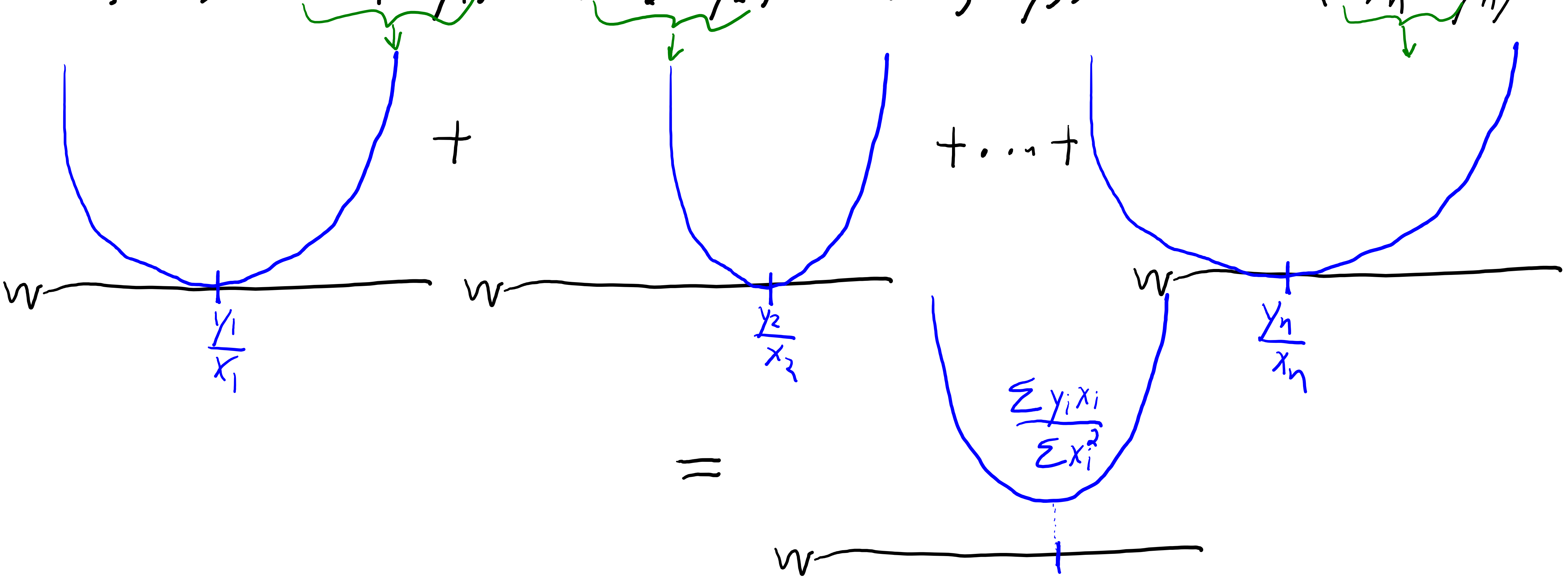
$$f''(w) = \sum_{i=1}^n x_i^2$$

  – 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 Objective/Solution (Another View)

- Least squares minimizes a quadratic that is a sum of quadratics:

$$f(w) = (wx_1 - y_1)^2 + (wx_2 - y_2)^2 + (wx_3 - y_3)^2 + \cdots + (wx_n - y_n)^2$$



(pause)

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':

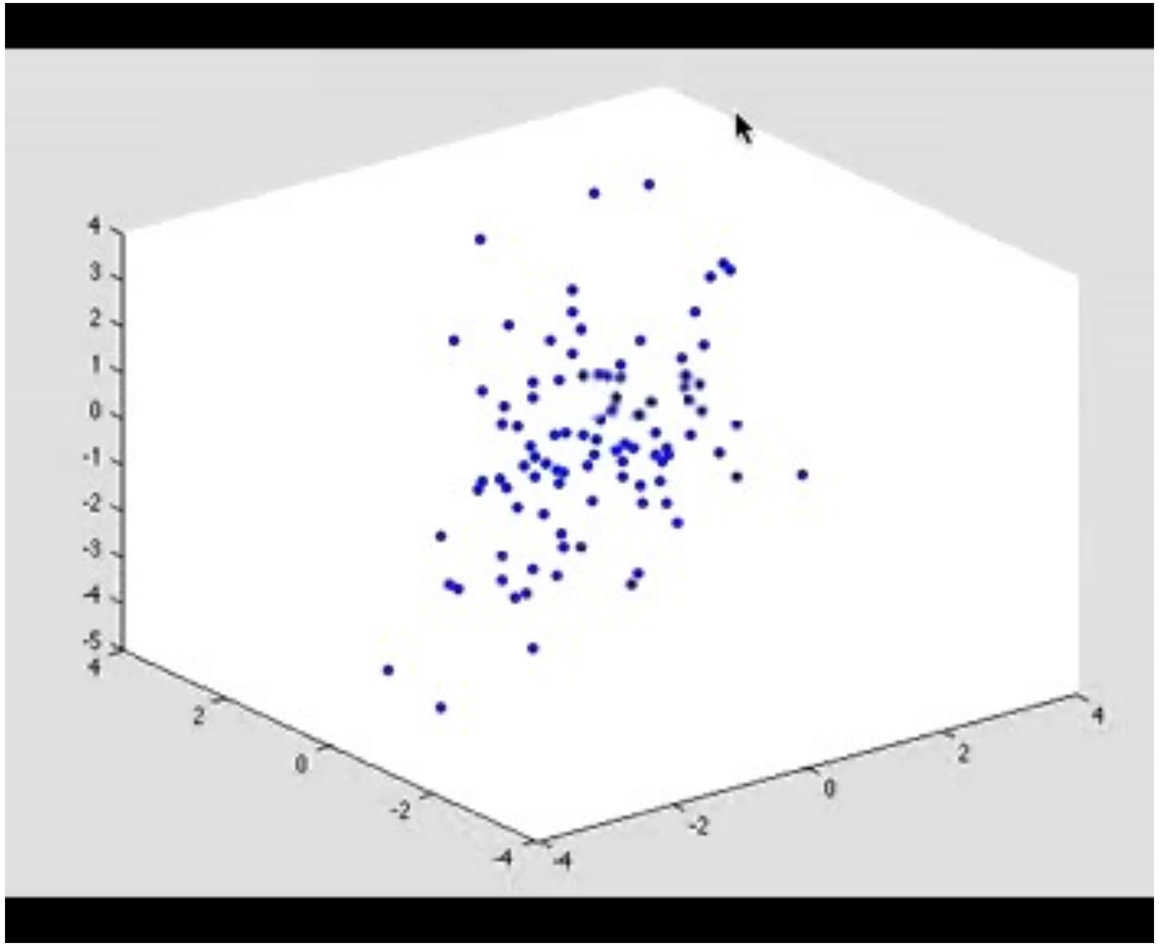
$$\hat{y}_i = 10(\#\text{cigarettes}) + 25(\#\text{asbetos})$$

Least Squares in 2-Dimensions

- Linear model:

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

- This defines a two-dimensional plane.



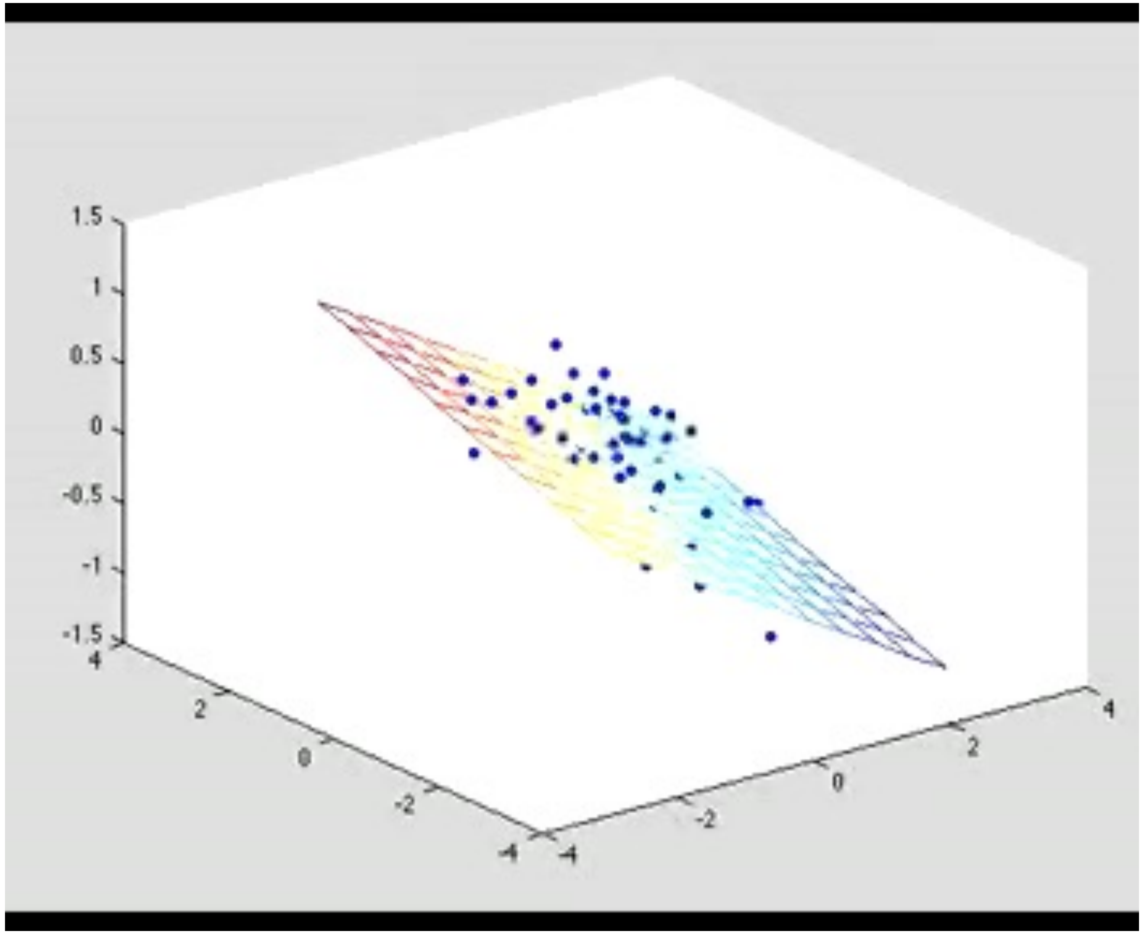
Least Squares in 2-Dimensions

- Linear model:

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

- This defines a two-dimensional plane.

- Not just a line!



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 weighted sum 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:

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

(assuming 'w' and x_i are column-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:

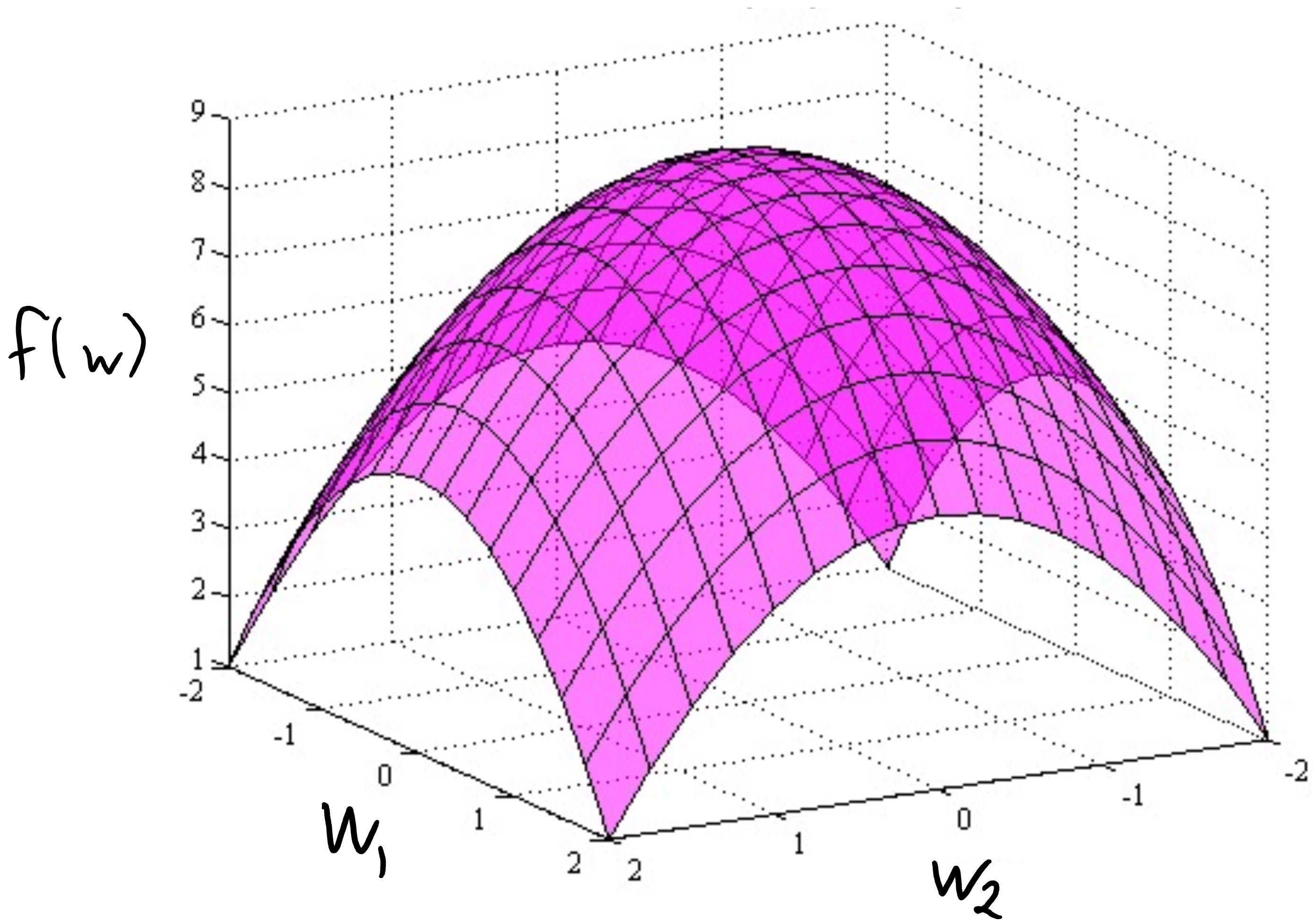
$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(\underbrace{w^T x_i}_{\text{prediction is inner product of 'w' and 'x_i' (linear combination of features)}} - y_i)^2$$

'w' is now a vector

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

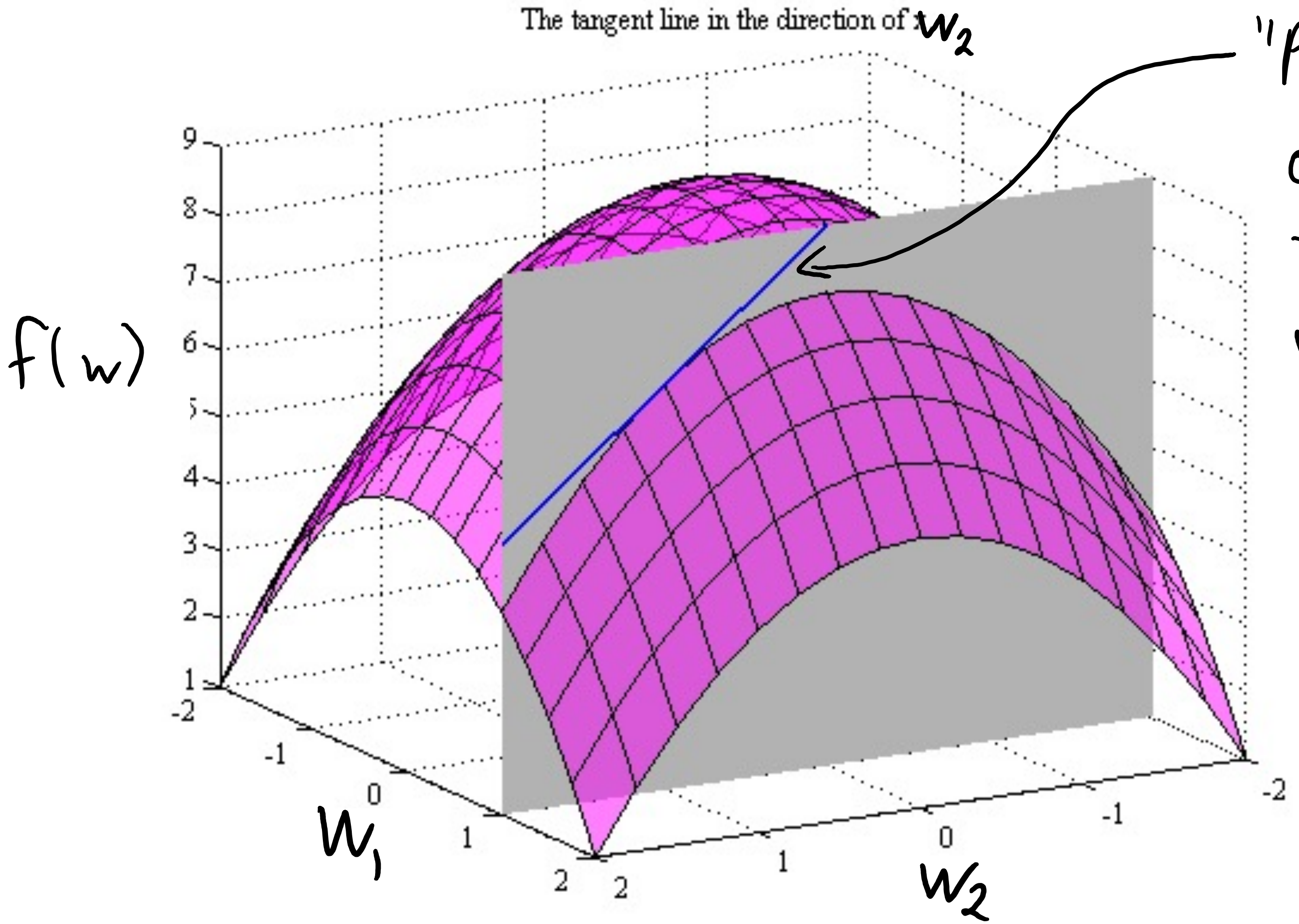
- Dates back to 1801: Gauss used it to predict location of Ceres.
- How do we find the best vector 'w' in 'd' dimensions?
  - Can we set the "partial derivative" of each variable to 0?

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_2 when all other variables are held <u>fixed</u>.

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

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

Least Squares Partial Derivatives (1 Example)

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

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

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

- Computing the partial derivative for variable '1':

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

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

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

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:

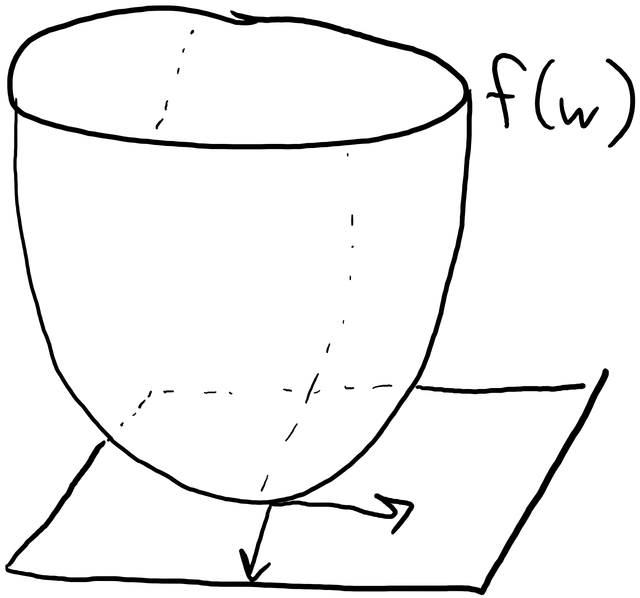
$$\frac{\partial}{\partial w_j} f(w_1, w_2, \dots, w_d) = \sum_{i=1}^{n} (w^T x_i - y_i) x_{ij}$$

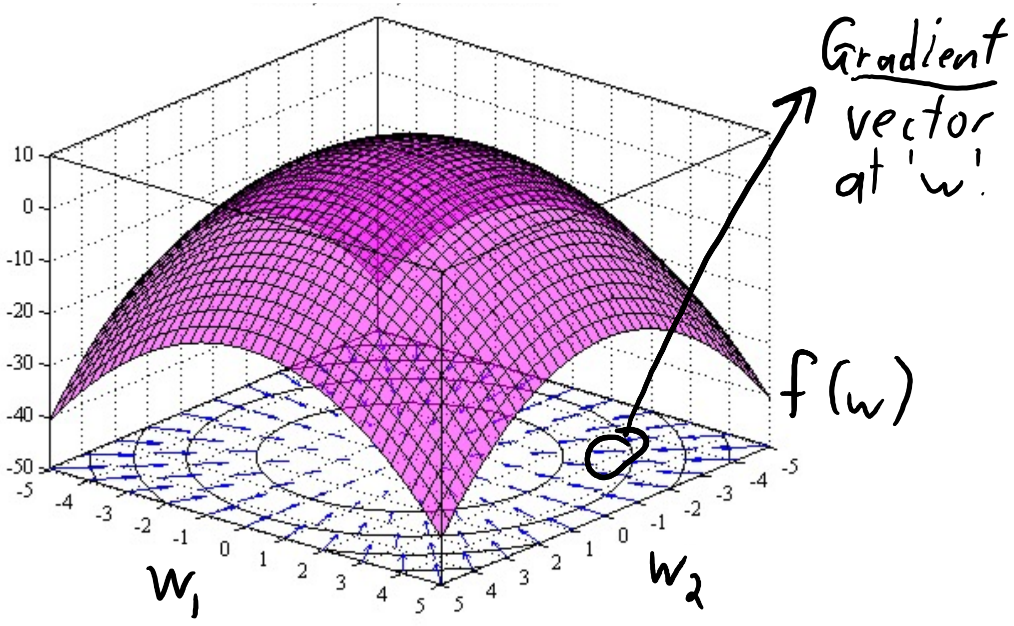
- Unfortunately, the partial derivative for w_j depends on all $\{w_1, w_2, \dots, w_d\}$
  - I can't just "set equal to 0 and solve for w_j".

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 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 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>.

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.
  - Can be generalized to ‘d’ features.
- Gradient is vector containing partial derivatives of all variables.
- Next time:

minimizing $\frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$ in terms of 'w' is:

```
w = np.linalg.solve(X.T @ X, X.T @ y)
```

bonus!

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