Computer Science

CSC380: Principles of Data Science

Linear Models 1

Credit:
- Jason Pacheco,
- Kwang-Sung Jun,
- Chicheng Zhang
- Xinchen yu

1

Outline

2

- Linear Regression
- Least Squares Estimation
- Regularized Least Squares
- Logistic Regression

first focus on **what** is a linear function

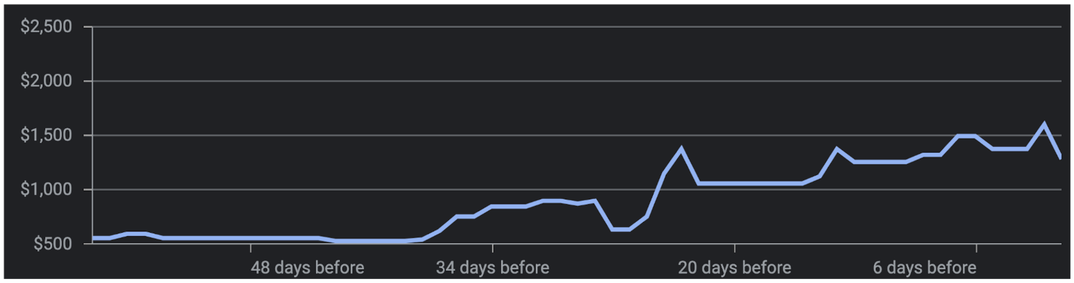
then learn **how to train** a linear function

2

Linear Regression

3

When is linear regression useful?



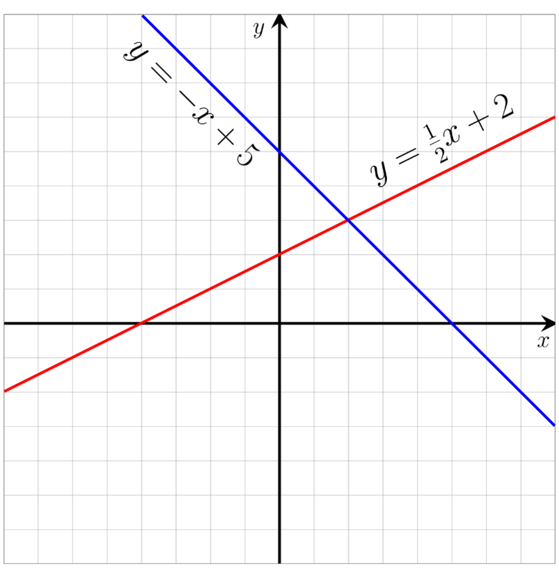
Price of an airline ticket

Used anywhere a linear relationship is assumed between inputs / (real-valued) outputs

3

Line Equation

4



Recall the equation for a line has a *slope* and an *intercept*,

$$y = w \cdot x + b$$

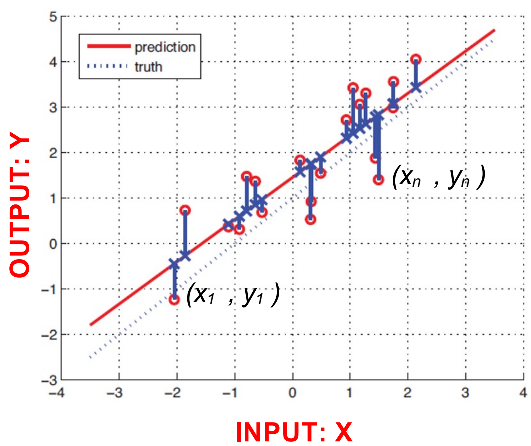
Slope Intercept

- Intercept (b) indicates where line crosses y-axis
- Slope controls angle of line
- Positive slope (w) → Line goes up left-to-right
- Negative slope → Line goes down left-to-right

4

Linear Regression

5



Regression Learn a function that predicts outputs from inputs.

Given points *(x_i , y_i)*, i=1….n, find a line *y=wx+b* that is close to the points

$$\arg \min_{w,b} \sum_{i=1}^{n} \Big(y_i - \overbrace{(w \cdot x_i + b)}^{y\text{-value of line at } x_i}\Big)^2$$

The vertical distance from each point to the line is the **residual**

Linear Regression As the name suggests, uses a *linear function*:

5

Review: inner product

Two vectors:

$$\vec{x} = \langle 2, -3 \rangle \qquad \mathbf{x} = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$$

$$\vec{y} = \langle 5, 1 \rangle \qquad \mathbf{y} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

Multiply corresponding entries and add:

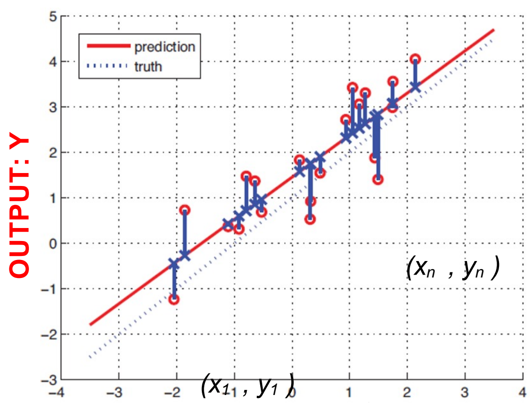
$$\vec{x} \cdot \vec{y} = \langle 2, -3 \rangle \cdot \langle 5, 1 \rangle = (2)(5) + (-3)(1) = 7$$

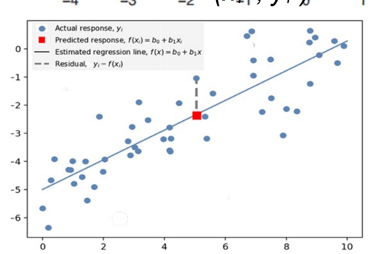
$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} 2 & -3 \end{bmatrix} \begin{bmatrix} 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \end{bmatrix} \quad \text{(or just 7)} \quad (\text{so } \vec{x} \cdot \vec{y} \text{ becomes } \mathbf{x}^T\mathbf{y})$$

6

Linear Regression

7





Regression Learn a function that predicts outputs from inputs. Given points *(x_i , y_i)*, i=1….n, find a line *y=wx+b* that is close to the points

$$\arg\min_{w,b} \sum_{i=1}^{n} \Big(y_i - \overbrace{(w \cdot x_i + b)}^{y\text{-value of line at } x_i} \Big)^2$$

We can use vector notation:

$$\vec{w} = (w_{slope},\ b_{intercept})$$

$$\vec{x_i} = (x_i, 1)$$

$$w_{slope} \cdot x_i \ + b_{intercept} = \vec{w}^T \cdot \vec{x_i}$$

$$\arg\min_w \sum (y_i - w^T x_i)^2$$

7

Moving to higher dimensions…

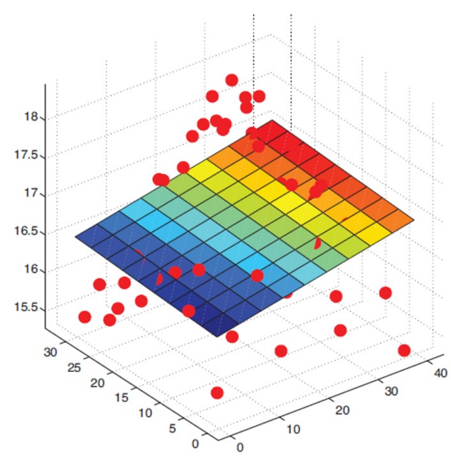
8

- **<u>1d regression</u>**: regression with 1d input:
  $$y = wx + b$$
- **<u>D-dimensional regression</u>**: input vector is $x \in \mathbb{R}^D$.

Recall the definition of an *inner product*:

$$w^T x = w_1 x_1 + w_2 x_2 + \ldots + w_D x_D \ = \sum_{d=1}^{D} w_d x_d$$

The model is $\quad y = w^T x + b$



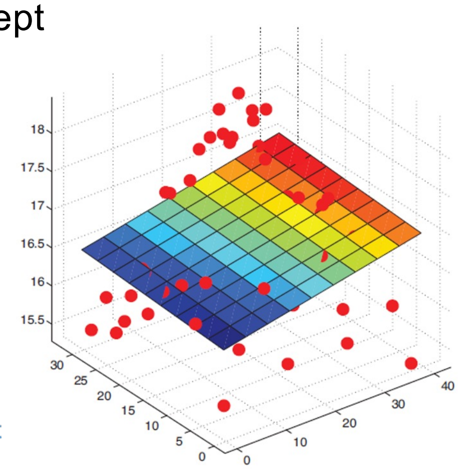
[Image: Murphy, K. (2012)]

8

Moving to higher dimensions… 9

Often we simplify this by including the intercept into the weight vector,

$$\widetilde{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_D \\ b \end{pmatrix} \qquad \widetilde{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_D \\ 1 \end{pmatrix} \qquad y = \widetilde{w}^T \widetilde{x}$$



Since:

$$\widetilde{w}^T \widetilde{x} = \sum_{d=1}^{D} w_d x_d + b \cdot 1 = w^T x + b$$

from now on, we assume that $w \in \mathbb{R}^D$ and $x \in \mathbb{R}^D$ already has b and 1 in the last coordinate respectively.

9

Learning Linear Regression Models 10

There are several ways to think about fitting regression:

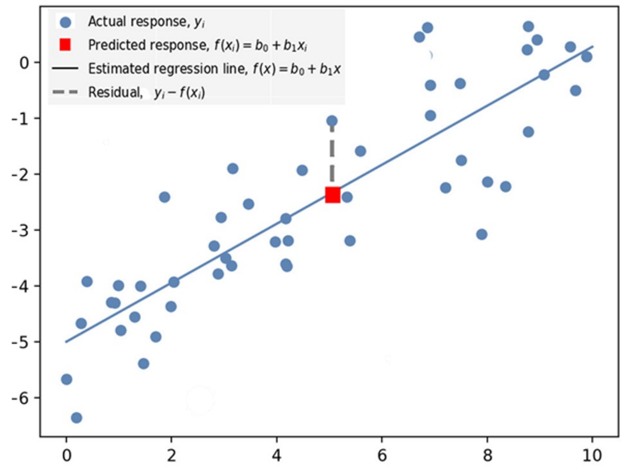
- **Intuitive** Find a plane/line that is close to data
- **Functional** Find a line that minimizes the *least squares* loss
- **Estimation** Find maximum likelihood estimate of parameters

They are all the same thing…

10

Fitting Linear Regression

11



Intuition Find a line that is as *close as possible* to every training data point

The distance from each point to the line is the **residual**

$$y - w^T x$$

Training Output **Prediction**

Let's find w that will minimize the residual!

https://www.activestate.com/resources/quick-reads/how-to-run-linear-regressions-in-python-scikit-learn/

11

Outline

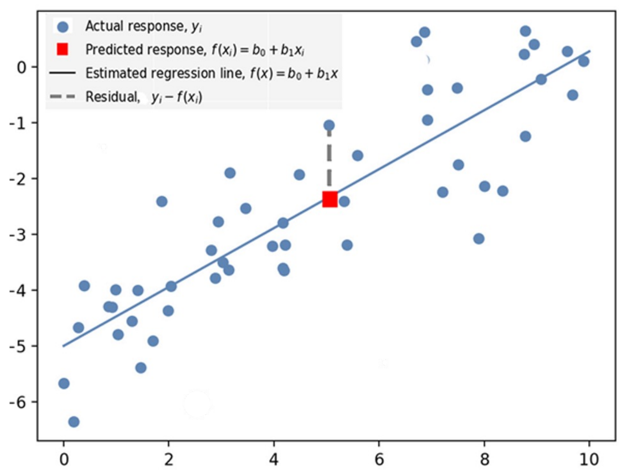
12

- Linear Regression
- Least Squares Estimation
- Regularized Least Squares
- Logistic Regression

12

Least Squares Solution

13



Functional Find a line that minimizes the sum of squared residuals!

Given: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$

Compute:

$$w^* = \arg\min_{w} \sum_{i=1}^{m} \left(y^{(i)} - w^T x^{(i)}\right)^2$$

Least squares regression

https://www.activestate.com/resources/quick-reads/how-to-run-linear-regressions-in-python-scikit-learn/

13

Least Squares

14

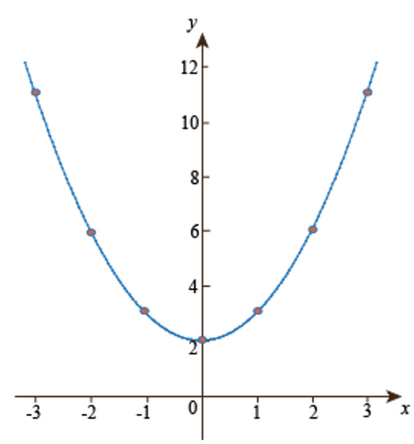
$$\min_{w} \sum_{i=1}^{N} (y^{(i)} - w^T x^{(i)})^2$$

This is just a quadratic function…

- *Convex,* unique minimum
- Minimum given by zero-derivative
- Can find a closed-form solution

Let's see for scalar case with no bias,

$$y = wx$$



14

Least Squares : Simple Case

15

$$\frac{d}{dw}\sum_{i=1}^{N}(y^{(i)} - wx^{(i)})^2 =$$

Derivative (+ chain rule)

$$= \sum_{i=1}^{N} 2(y^{(i)} - wx^{(i)})(-x^{(i)}) = 0 \Rightarrow$$

Distributive Property
(and multiply -1 both sides)

$$0 = \sum_{i=1}^{N} y^{(i)}x^{(i)} - w\sum_{j=1}^{N}(x^{(j)})^2$$

Algebra

$$w = \frac{\sum_i y^{(i)}x^{(i)}}{\sum_j (x^{(j)})^2}$$

15

Least Squares: Higher Dimensions

16

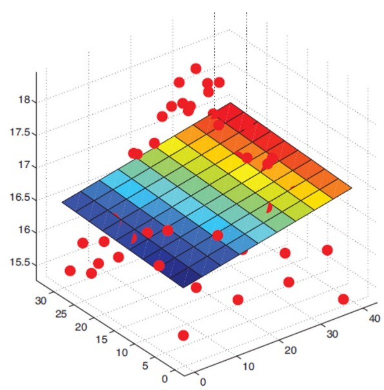
Things are a bit more complicated in higher dimensions and involve more linear algebra,

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \dots & x_D^{(1)} & 1 \\ x_1^{(2)} & \dots & x_D^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(m)} & \dots & x_D^{(m)} & 1 \end{pmatrix}$$

Design Matrix
(each row is a data point)

$$\mathbf{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix}$$

Vector of labels



Can write regression over *all training data* more compactly…

$$\mathbf{y} \approx \mathbf{X}w$$ ← **mx1 Vector**

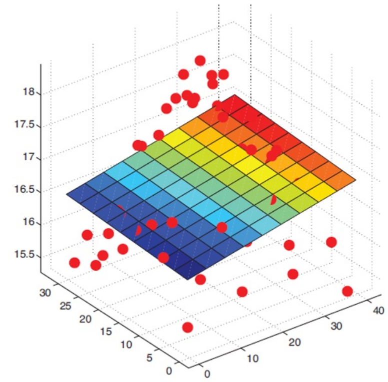
$$= \begin{pmatrix} (x^{(1)})^\top w \\ \dots \\ (x^{(m)})^\top w \end{pmatrix}$$

16

Special Case: D = #data points

17

Things are a bit more complicated in higher dimensions and involve more linear algebra,



$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \dots & x_D^{(1)} & 1 \\ x_1^{(2)} & \dots & x_D^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(m)} & \dots & x_D^{(m)} & 1 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix}$$

Design Matrix
(each row is a data point)

Vector of labels

As before, writing

$$\mathbf{y} \approx \mathbf{X}w$$

mx1 Vector

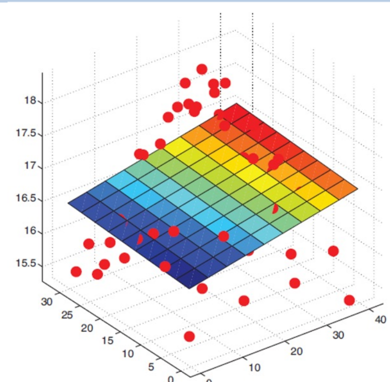
Minimizing **||*y-Xw*||**
But now we (sometimes) could computer X^{-1} and write
$\boldsymbol{X^{-1}y = w}$

17

Least Squares: Higher Dimensions

18

Least squares can also be written more compactly, $\boxed{\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}\cdot\boldsymbol{x}}.}$



$$\min_w \sum_{i=1}^{N} (y^{(i)} - w^T x^{(i)})^2 = \|\mathbf{y} - \mathbf{X}w\|^2$$

Some slightly more advanced linear algebra gives us a solution,

$$w = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

compare with the 1d version: $w = \frac{\sum_i y^{(i)} x^{(i)}}{\sum_j (x^{(j)})^2}$

Ordinary Least Squares *(OLS)* solution

Derivation a bit advanced for this class, but enough to know
- it has a closed-form and why
- we can evaluate it
- generally know where it comes from.

18