10-301/601: Introduction to Machine Learning Lecture 7 – Linear Regression

Henry Chai

5/30/23

When survey is active, respond at **pollev.com/301601polls**

Lecture 7 Polls

0 done

0 underway

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Which of the following best describes your use of SG2?

- Did not review SG2 at all
- Read some or all of SG2 but did not attempt any problems
- Completed or attempted some of the problems
- Completed or attempted all of the problems

To the nearest integer, how many hours did you spend reviewing SG2? Please respond using digits [0-9], e.g., "12" instead of "twelve".

Join by Web

1. Go to **PollEv.com**
2. Enter **301601POLLS**
3. Respond to activity

Instructions not active. **Log in** to activate

Front Matter

- Announcements:
  - PA2 released 5/25, due 6/01 at 11:59 PM
- Recommended Readings:
  - Murphy, Chapters 7.1-7.3

Recall: Regression

- Learning to diagnose heart disease as a (supervised) **<u>regression</u> task**

features / targets / data points

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	\$0
No	Medium	Normal	\$20
No	Low	Abnormal	\$30
Yes	Medium	Normal	\$100
Yes	High	Abnormal	\$5000

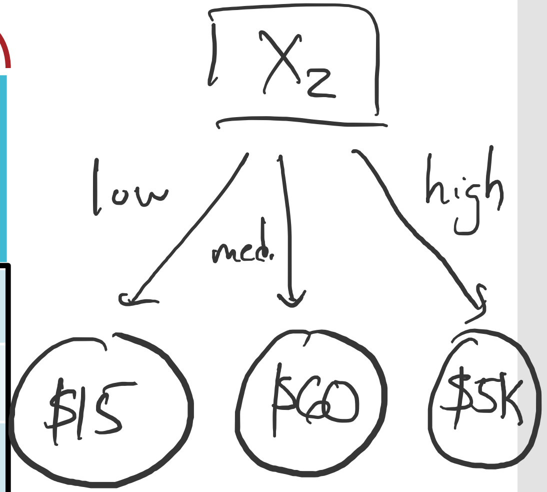
Decision ~~Tree~~ Stump Regression

- Learning to diagnose heart disease

  as a **(supervised)** **<u>regression</u> task**

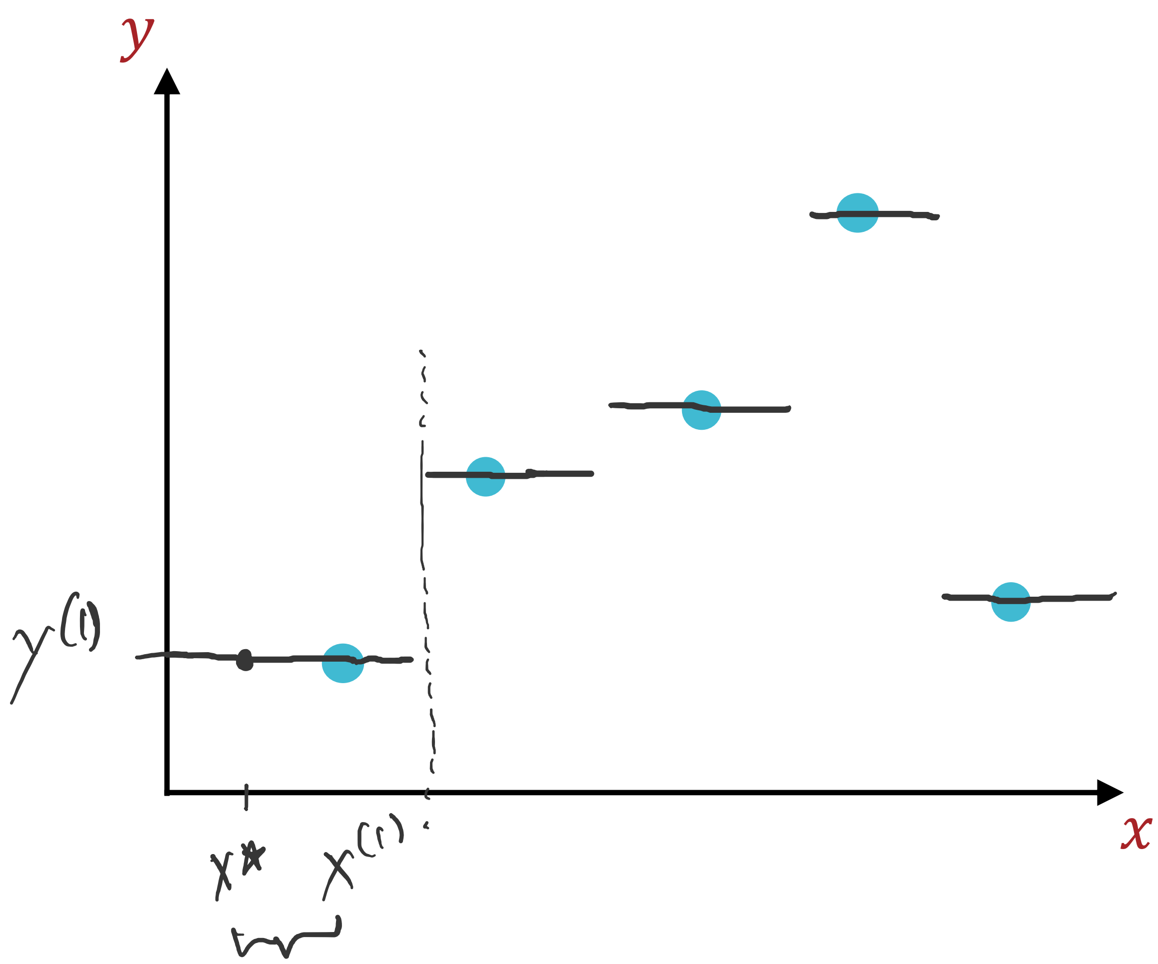
x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	$0
No	Medium	Normal	$20
No	Low	Abnormal	$30
Yes	Medium	Normal	$100
Yes	High	Abnormal	$5000

features: x_1, x_2, x_3 — targets: y — rows: data points



1-NN Regression

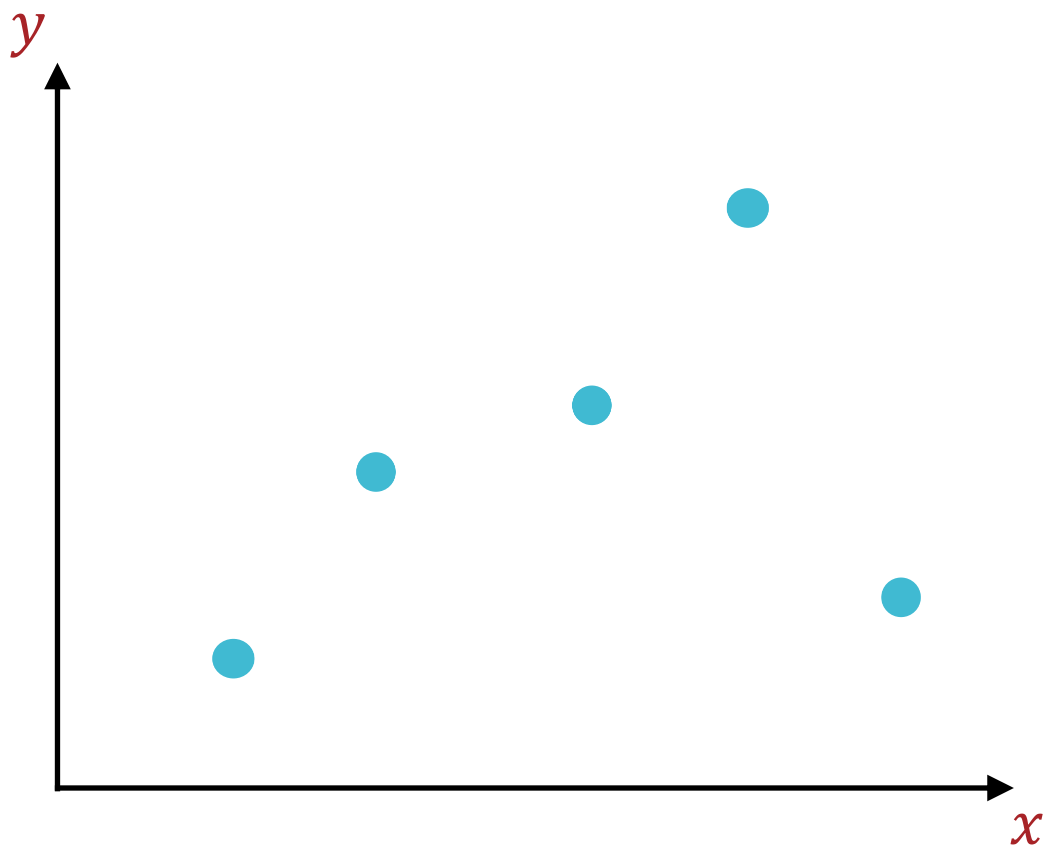
- Suppose we have real-valued targets $y \in \mathbb{R}$ and one-dimensional inputs $x \in \mathbb{R}$



Distance-weighted?

2-NN Regression?

- Suppose we have real-valued targets $y \in \mathbb{R}$ and one-dimensional inputs $x \in \mathbb{R}$



Linear Regression

- Suppose we have real-valued targets $y \in \mathbb{R}$ and D-dimensional inputs $\boldsymbol{x} = [\quad x_1, \dots, x_D]^T \in \mathbb{R}^D$
- **Assume**

$$y = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

Linear Regression

- Suppose we have real-valued targets $y \in \mathbb{R}$ and D-dimensional inputs $\boldsymbol{x} = [1, x_1, \dots, x_D]^T \in \mathbb{R}^{D+1}$
- **Assume**

$$y = \boldsymbol{w}^T \boldsymbol{x}$$

General Recipe for Machine Learning

- Define a model and model parameters

- Write down an objective function

- Optimize the objective w.r.t. the model parameters

Recipe for Linear Regression

- Define a model and model parameters

Assume $y = w^T x$

Parameters: $w = [w_0, w_1, \ldots, w_D]^T$

- Write down an objective function

Minimize the squared error

$$\ell_D(w) = \sum_{n=1}^{N} \ell^{(n)}(w) = \sum_{n=1}^{N} \left(w^T x^{(n)} - y^{(n)}\right)^2$$

- Optimize the objective w.r.t. the model parameters

Solve in "closed-form" by taking partial derivatives and finding "critical points"

Minimizing the Squared Error

$$\ell_D(w) = \sum_{n=1}^{N} \left(w^T x^{(n)} - y^{(n)}\right)^2$$

$$= \sum_{n=1}^{N} \left(\left(\sum_{d=0}^{D} w_d x_d^{(n)}\right) - y^{(n)}\right)^2$$

$$\frac{\partial \ell_D}{\partial w_d} = \sum_{n=1}^{N} \frac{\partial}{\partial w_d}\left(\left(\sum_{d=0}^{D} w_d x_d^{(n)}\right) - y^{(n)}\right)^2$$

$$= \sum_{n=1}^{N} 2\left(\left(\sum_{d=0}^{D} w_d x_d^{(n)}\right) - y^{(n)}\right)\frac{\partial}{\partial w_d}\left(\left(\sum_{d=0}^{D} w_d x_d^{(n)}\right) - y^{(n)}\right)$$

$$= \sum_{n=1}^{N} 2\left(\left(\sum_{d=0}^{D} w_d x_d^{(n)}\right) - y^{(n)}\right) x_d^{(n)}$$

↳ set equal to 0 & solve!

Recipe for Linear Regression

- Define a model and model parameters
  - Assume $y = \boldsymbol{w}^T \boldsymbol{x}$
  - Parameters: $\boldsymbol{w} = [w_0, w_1, \dots, w_D]$
- Write down an objective function
  - Minimize the squared error

$$\ell_{\mathcal{D}}(\boldsymbol{w}) = \sum_{n=1}^{N} \ell^{(n)}(\boldsymbol{w}) = \sum_{n=1}^{N} \left(\boldsymbol{w}^T \boldsymbol{x}^{(n)} - y^{(n)}\right)^2$$

- Optimize the objective w.r.t. the model parameters
  - Solve in *closed form*: take ~~partial derivatives~~ gradient, set to 0 and solve

Linear Regression

- Suppose we have real-valued targets $y \in \mathbb{R}$ and D-dimensional inputs $\boldsymbol{x} = [1, x_1, \dots, x_D]^T \in \mathbb{R}^{D+1}$
- **Assume**

$$y = \boldsymbol{w}^T \boldsymbol{x}$$

- Notation: given training data $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}$
  - $X = \begin{bmatrix} 1 & \boldsymbol{x}^{(1)^T} \\ 1 & \boldsymbol{x}^{(2)^T} \\ \vdots & \vdots \\ 1 & \boldsymbol{x}^{(N)^T} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_D^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & \cdots & x_D^{(N)} \end{bmatrix} \in \mathbb{R}^{N \times D+1}$

    is the *design matrix*
  - $\boldsymbol{y} = \left[y^{(1)}, \dots, y^{(N)}\right]^T \in \mathbb{R}^N$ is the *target vector*

Minimizing the Squared Error

$$\ell_D(w) = \sum_{n=1}^{N} \left(w^T x^{(n)} - y^{(n)}\right)^2 = \sum_{n=1}^{N} \left(\underbrace{x^{(n)^T} w - y^{(n)}}\right)^2$$

$$= (Xw - y)^T (Xw - y) = \|Xw - y\|_2^2$$

$$= (w^T X^T - y^T)(Xw - y)$$

kind of like $(Xw)^2$

$$= w^T X^T X w \underbrace{- y^T X w - w^T X^T y} + y^T y$$

like $(y)^2$

$$= \underline{w^T X^T X w} - 2 w^T X^T y + \underline{y^T y}$$

$$\nabla_w \ell_D(w) = \begin{bmatrix} \frac{\partial \ell_D}{\partial w_0} \\ \vdots \\ \frac{\partial \ell_D}{\partial w_D} \end{bmatrix} = 2X^T X w - 2X^T y + 0$$

Minimizing the Squared Error

$$\ell_{\mathcal{D}}(\boldsymbol{w}) = \sum_{n=1}^{N} \left(\boldsymbol{w}^T \boldsymbol{x}^{(n)} - y^{(n)}\right)^2 = \sum_{n=1}^{N} \left({\boldsymbol{x}^{(n)}}^T \boldsymbol{w} - y^{(n)}\right)^2$$

$$= \|X\boldsymbol{w} - \boldsymbol{y}\|_2^2 \text{ where } \|\boldsymbol{z}\|_2 = \sqrt{\sum_{d=1}^{D} z_d^2} = \sqrt{\boldsymbol{z}^T\boldsymbol{z}}$$

$\in \mathbb{R}^{D+1}$

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

$$= (\boldsymbol{w}^T X^T X \boldsymbol{w} - 2\boldsymbol{w}^T X^T \boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{y})$$

$$\nabla_{\boldsymbol{w}} \ell_{\mathcal{D}}(\hat{\boldsymbol{w}}) = (2X^T X \hat{\boldsymbol{w}} - 2X^T \boldsymbol{y}) = \vec{0}$$

$$\rightarrow X^T X \hat{\boldsymbol{w}} = X^T \boldsymbol{y}$$

$$\rightarrow \hat{\boldsymbol{w}} = \underbrace{(X^T X)^{-1} X^T}\boldsymbol{y}$$

Minimizing the Squared Error

$$\ell_{\mathcal{D}}(\boldsymbol{w}) = \sum_{n=1}^{N}\left(\boldsymbol{w}^T\boldsymbol{x}^{(n)} - y^{(n)}\right)^2 = \sum_{n=1}^{N}\left({\boldsymbol{x}^{(n)}}^T\boldsymbol{w} - y^{(n)}\right)^2$$

$= \|X\boldsymbol{w} - \boldsymbol{y}\|_2^2$ where $\|\boldsymbol{z}\|_2 = \sqrt{\sum_{d=1}^{D} z_d^2} = \sqrt{\boldsymbol{z}^T\boldsymbol{z}}$

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

$$= (\boldsymbol{w}^T X^T X\boldsymbol{w} - 2\boldsymbol{w}^T X^T \boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y})$$

$$\nabla_{\boldsymbol{w}}\ell_{\mathcal{D}}(\boldsymbol{w}) = (2X^T X\boldsymbol{w} - 2X^T\boldsymbol{y})$$

$$H_{\boldsymbol{w}}\ell_{\mathcal{D}}(\boldsymbol{w}) = 2X^T X$$

$H_{\boldsymbol{w}}\ell_{\mathcal{D}}(\boldsymbol{w})$ is positive semi-definite $\Rightarrow$ at a minimum

Beyond the scope of the class

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

Closed Form Solution

1. Is $X^T X$ invertible?

2. If so, how computationally expensive is inverting $X^T X$?

Is $X^T X$ always invertible?

- Yes
- No
- Unsure

If $X^T X$ is invertible, how computationally expensive is it to invert?

- $O(N^2)$
- $O(D^2)$
- $O(ND)$
- $O(N^3)$
- $O(D^3)$

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

Closed Form Solution

$X \in \mathbb{R}^{N \times (D+1)}$

1. Is $X^T X$ invertible?
   - When $N \gg D + 1$, $X^T X$ is (almost always) full rank and therefore, invertible!
   - If $X^T X$ is not invertible (occurs when one of the features is a linear combination of the others), what does that imply about our problem?
2. If so, how computationally expensive is inverting $X^T X$?
   - $X^T X \in \mathbb{R}^{D+1 \times D+1}$ so inverting $X^T X$ takes $O(D^3)$ time… can get $O(D^{2.781})$
     - Computing $X^T X$ takes $O(ND^2)$ time
   - What alternative optimization method(s) can we use to minimize the mean squared error?

Key Takeaways

- Decision tree and kNN regression
- Closed form solution for linear regression
  - Setting partial derivative/gradients to 0 and solving for critical points
  - Potential issues with the closed form solution: invertibility and computational c.sts