10-301/601: Introduction to Machine Learning Lecture 11 – Linear Regression

Henry Chai

5/20/25

Recall: Regression

- Learning to diagnose heart disease

  as a **(supervised)** **regression task**

features / targets

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

Decision Tree Regression

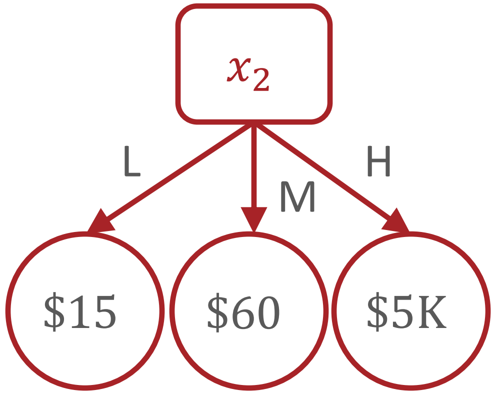
- Learning to diagnose heart disease

  as a **(supervised)** **regression task**

features | targets

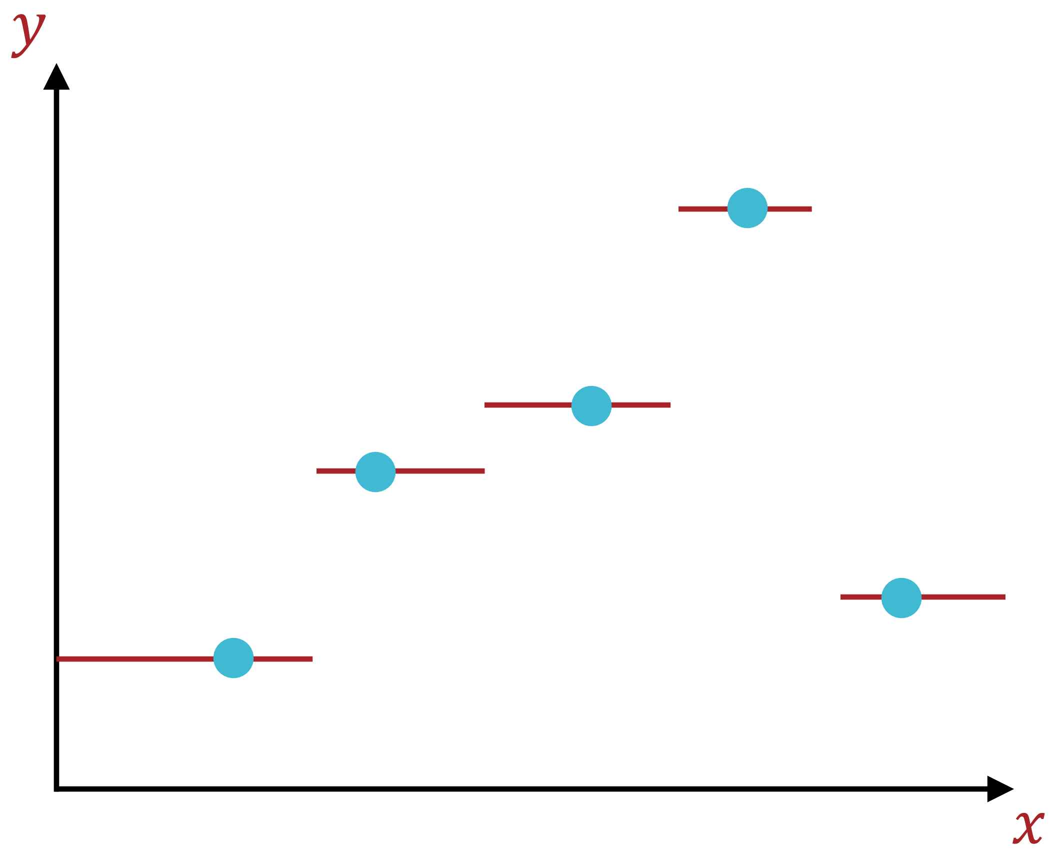
data points

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



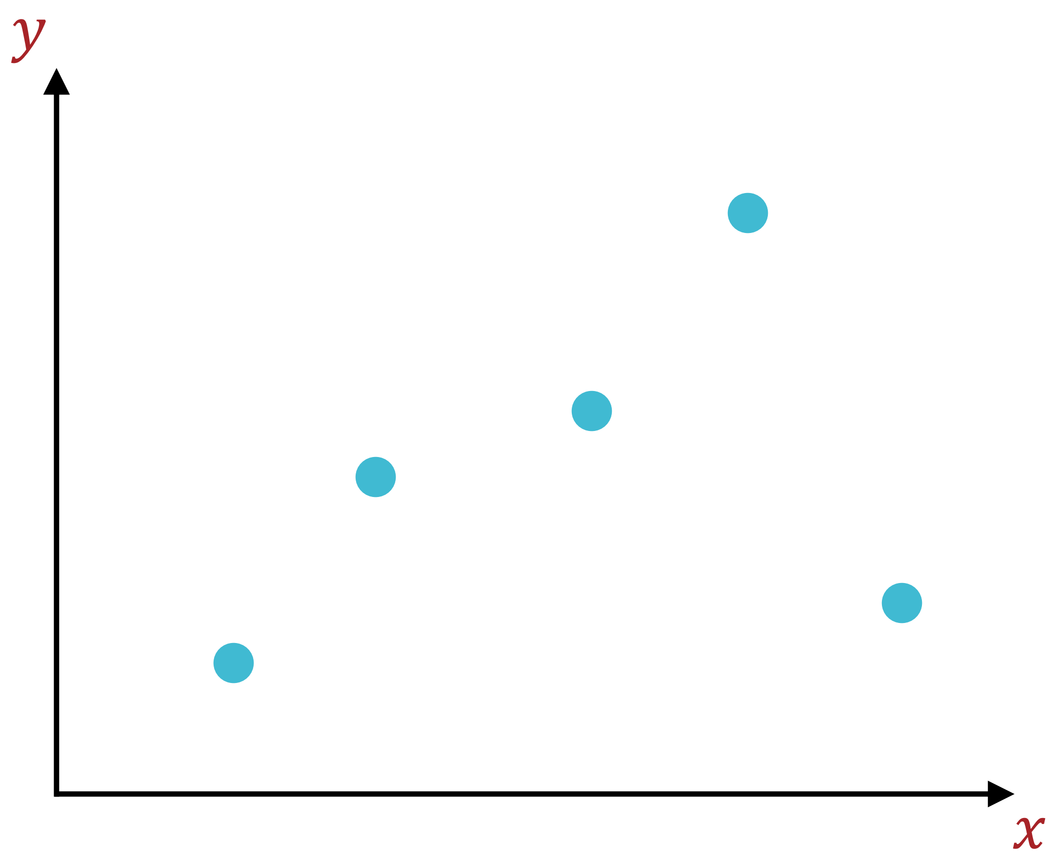
1-NN Regression

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



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 $\boldsymbol{x} = [1, x_1, \dots, x_D]^T \in \mathbb{R}^{D+1}$
- **Assume**

$$y = \boldsymbol{\theta}^T \boldsymbol{x} = [w_0 \; \boldsymbol{w}]^T \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}$$

- 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*

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 $y = \boldsymbol{\theta}^T \boldsymbol{x}$
  - Parameters: $\boldsymbol{\theta} = [w_0, w_1, \dots, w_D]$
- Write down an objective function
  - Minimize the mean squared error

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

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

Minimizing the Squared Error

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

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

$$= \frac{1}{N}(X\boldsymbol{\theta} - \boldsymbol{y})^T(X\boldsymbol{\theta} - \boldsymbol{y})$$

$$= \frac{1}{N}(\boldsymbol{\theta}^T X^T X\boldsymbol{\theta} - 2\boldsymbol{\theta}^T X^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y})$$

$$\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}) = \frac{1}{N}(2X^T X\boldsymbol{\theta} - 2X^T\boldsymbol{y})$$

Minimizing the Squared Error

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

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

$$= \frac{1}{N}(X\boldsymbol{\theta} - \boldsymbol{y})^T(X\boldsymbol{\theta} - \boldsymbol{y})$$

$$= \frac{1}{N}\left(\boldsymbol{\theta}^T X^T X\boldsymbol{\theta} - 2\boldsymbol{\theta}^T X^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}\right)$$

$$\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\widehat{\boldsymbol{\theta}}) = \frac{1}{N}\left(2X^TX\widehat{\boldsymbol{\theta}} - 2X^T\boldsymbol{y}\right) = 0$$

$$\rightarrow X^TX\widehat{\boldsymbol{\theta}} = X^T\boldsymbol{y}$$

$$\rightarrow \widehat{\boldsymbol{\theta}} = (X^TX)^{-1}X^T\boldsymbol{y}$$

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

Closed Form Solution

1. Is $X^T X$ invertible?

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

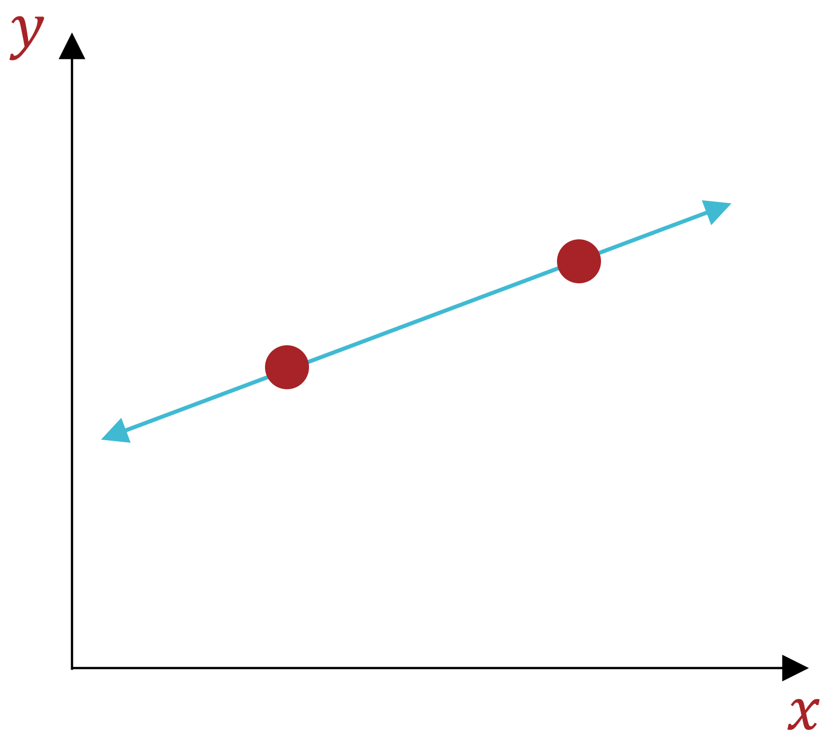
$$\hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

Closed Form Solution

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…
     - Computing $X^T X$ takes $O(ND^2)$ time
   - What alternative optimization method(s) can we use to minimize the mean squared error?

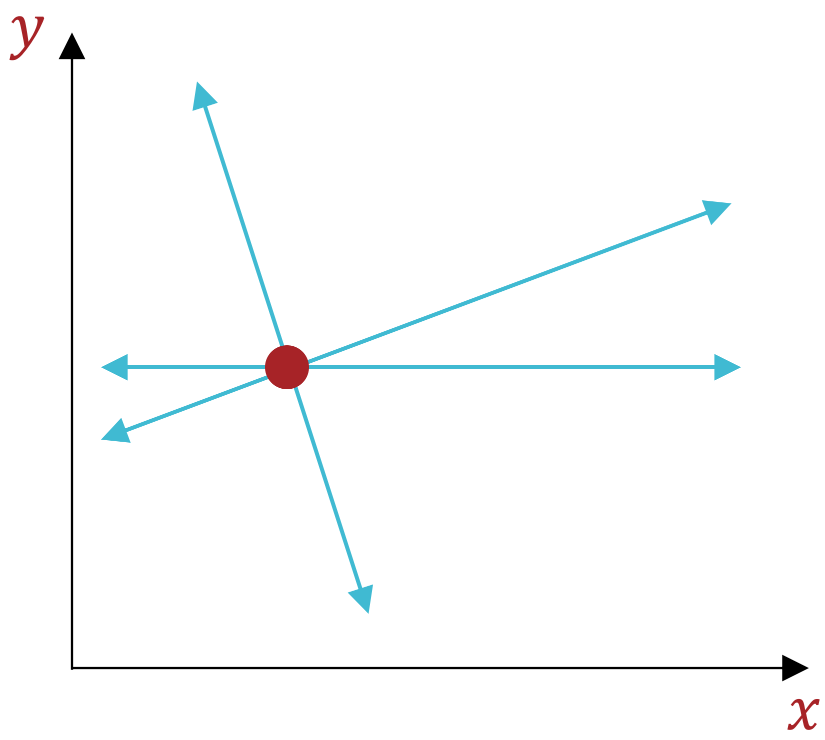
Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



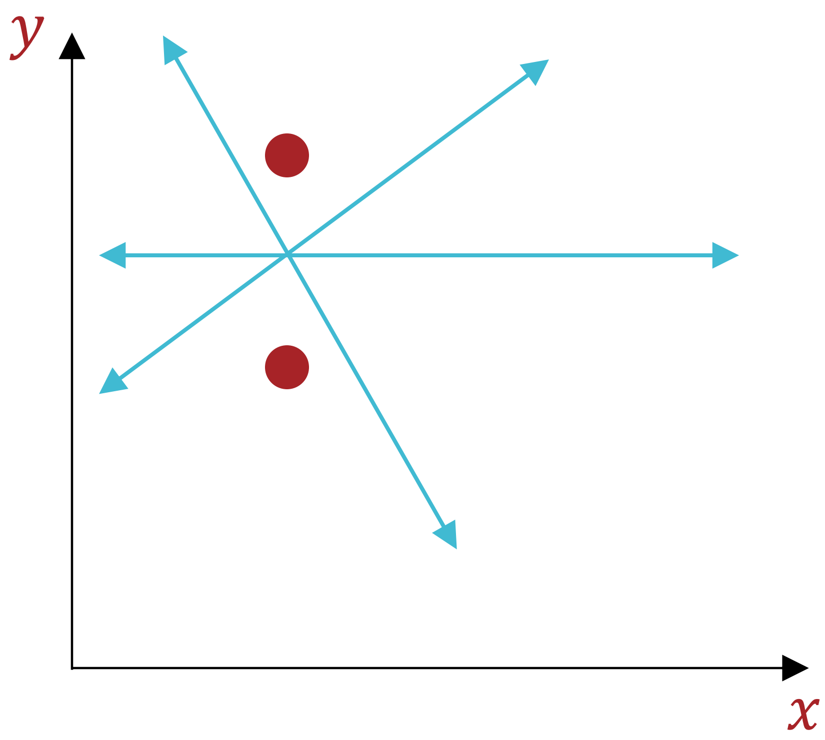
Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



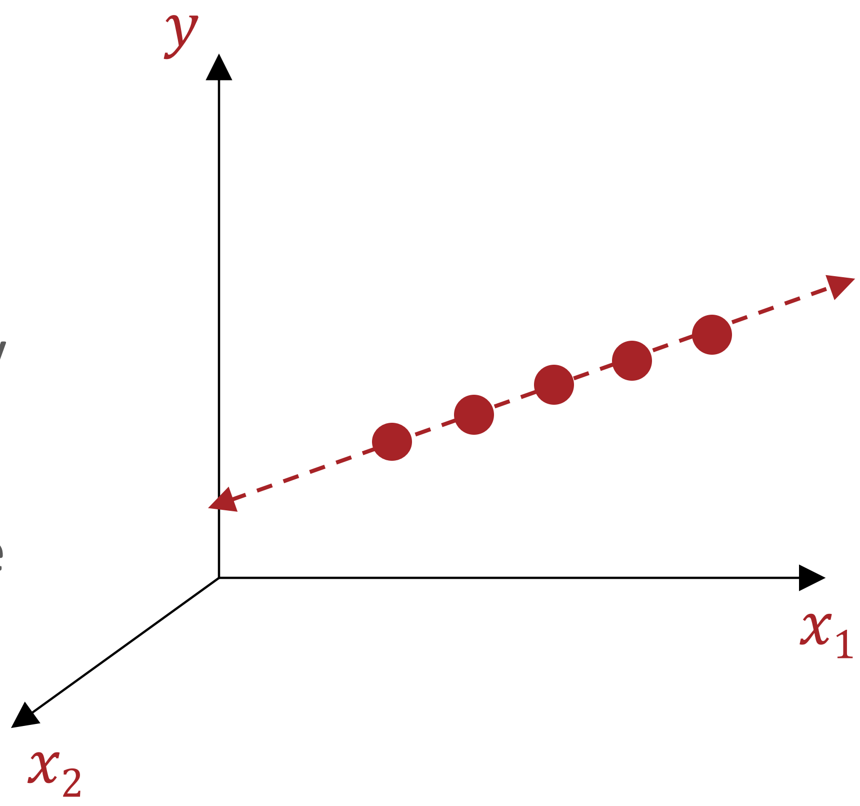
Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



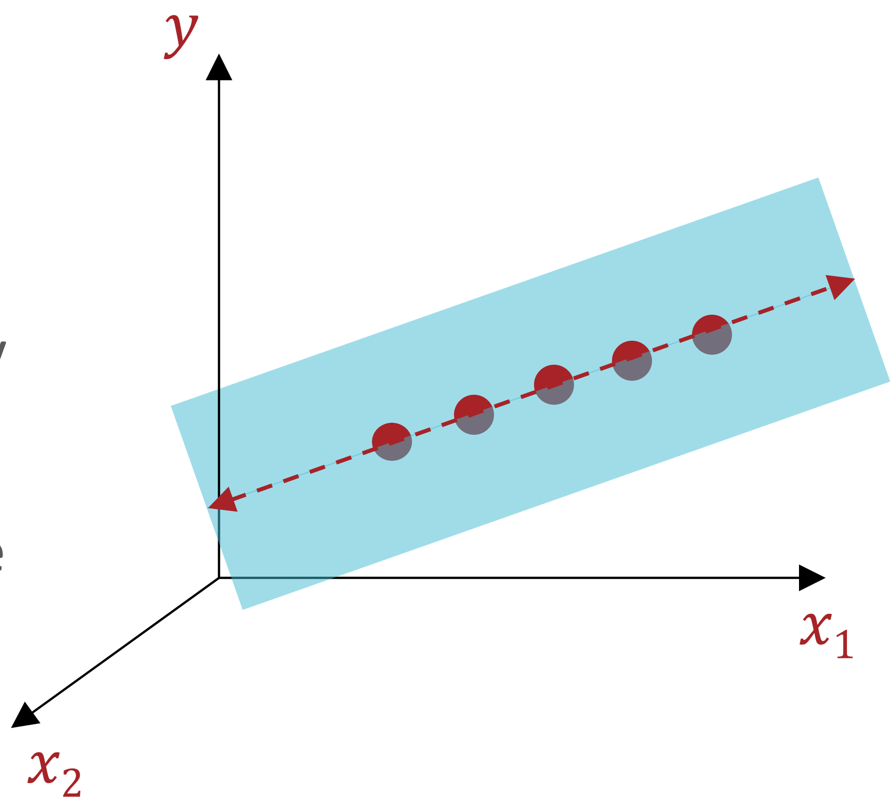
Linear Regression: Uniqueness

- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



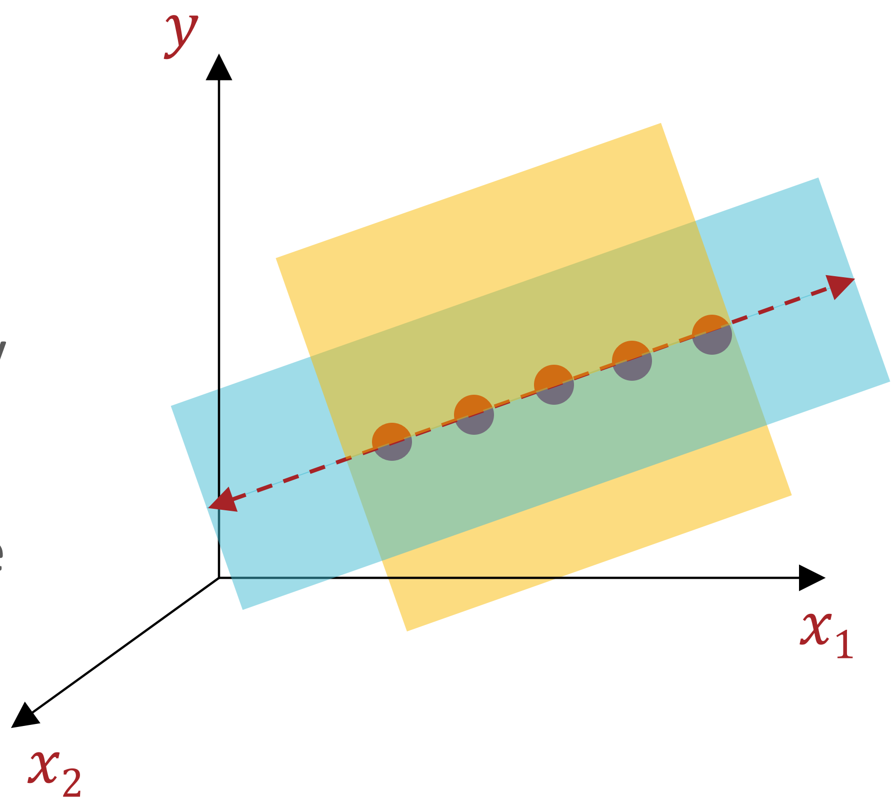
Linear Regression: Uniqueness

- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



Linear Regression: Uniqueness

- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



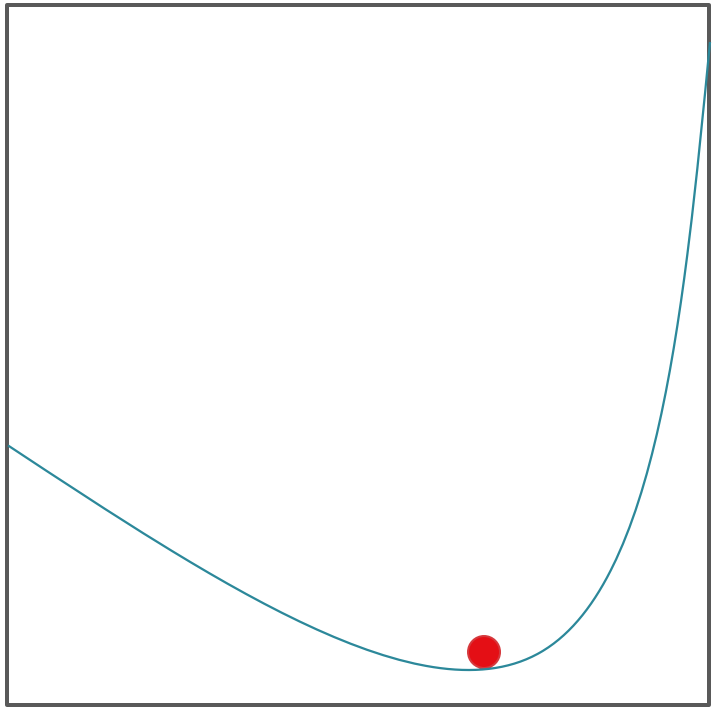
$$\hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

Closed Form Solution

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) then there are infinitely many solutions.
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…
     - Computing $X^T X$ takes $O(ND^2)$ time
   - Can use gradient descent to (potentially) speed things up when N and D are large!

Gradient Descent: Intuition

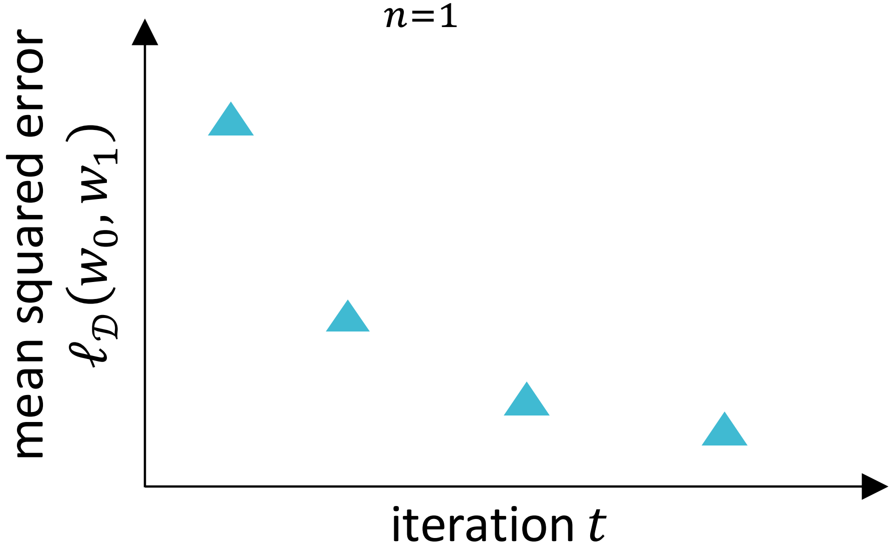
- An iterative method for minimizing functions
- Requires the gradient to exist everywhere

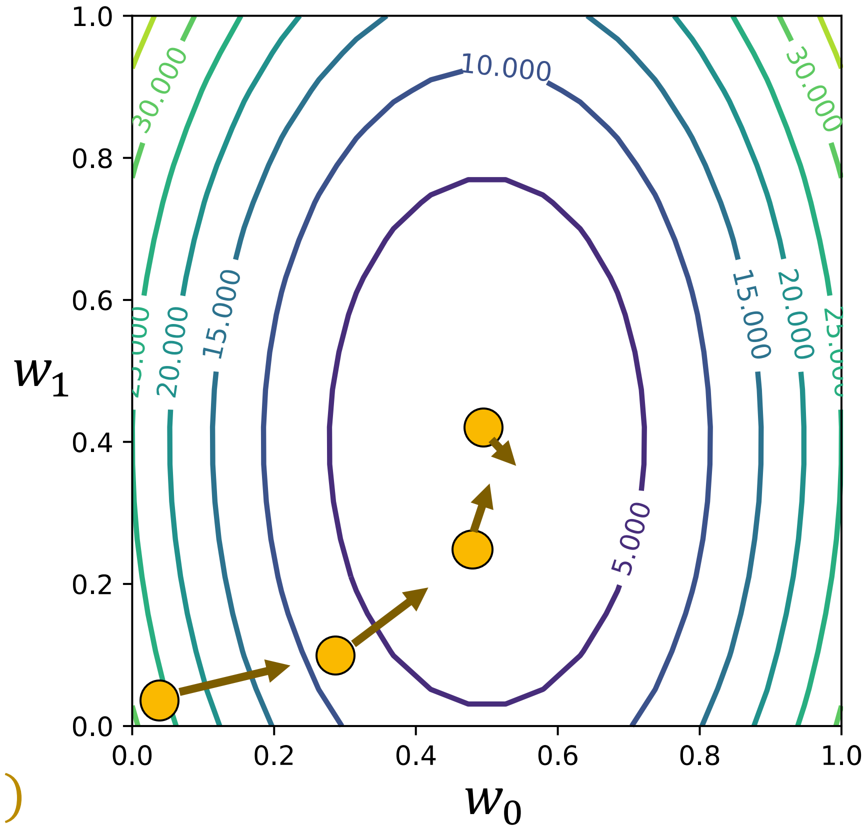


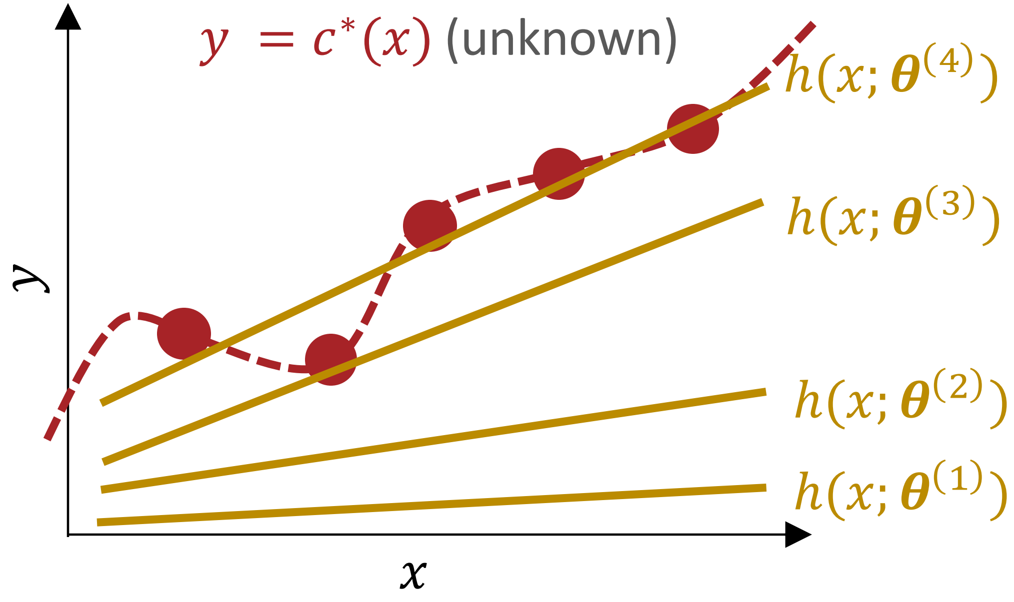
- Good news: the squared error is also convex!

Gradient Descent for Linear Regression

$$\ell_{\mathcal{D}}(w_0, w_1) = \frac{1}{N}\sum_{n=1}^{N}\left(w_1 x^{(n)} + w_0 - y^{(n)}\right)^2$$



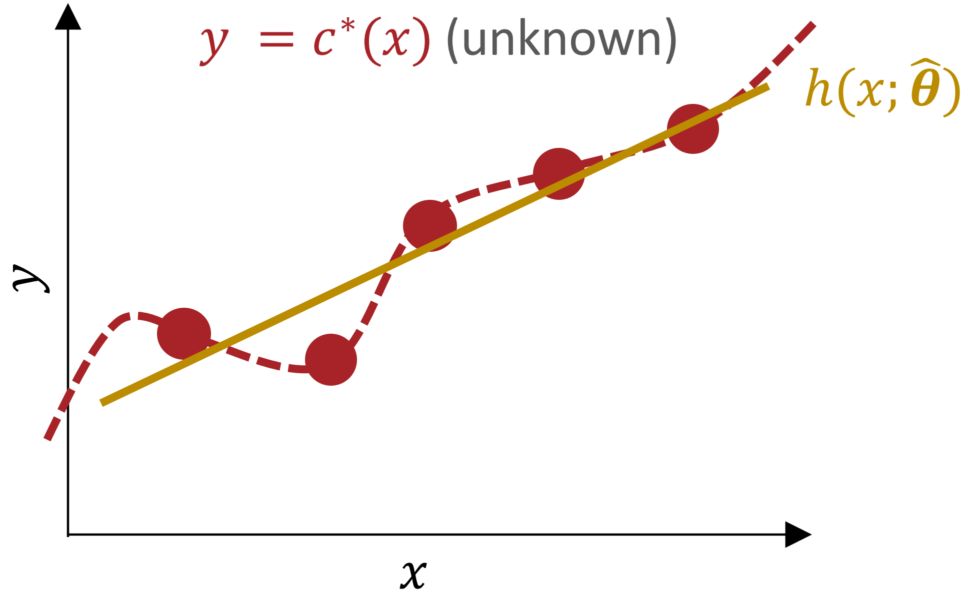


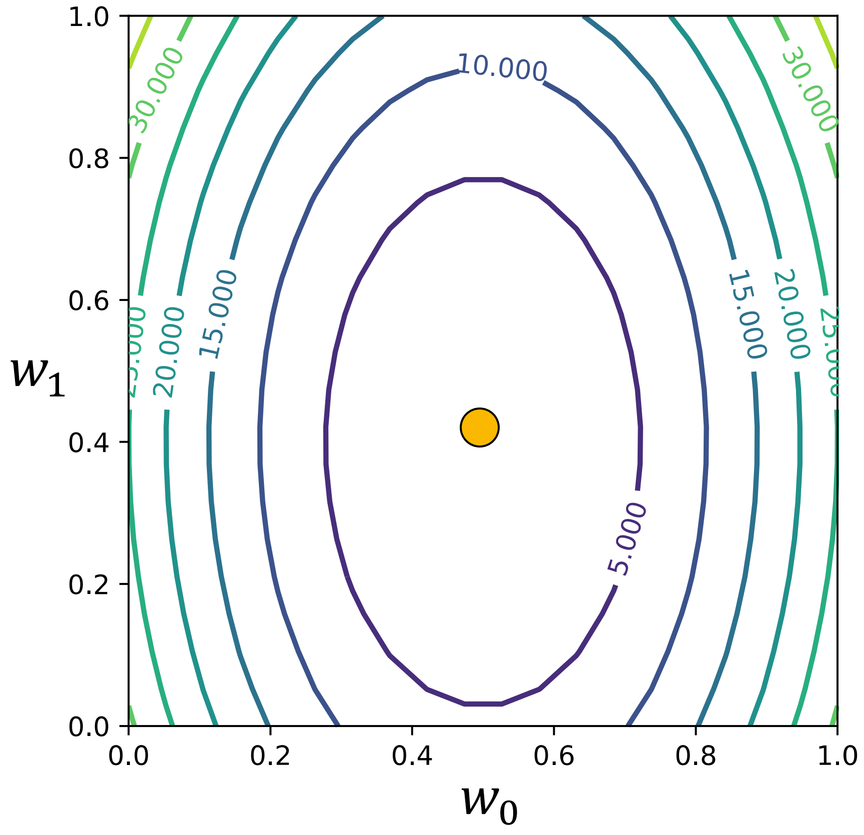


t	w_0	w_1	$\ell_{\mathcal{D}}(w_0, w_1)$
1	0.01	0.02	25.2
2	0.30	0.12	8.7
3	0.51	0.30	1.5
4	0.59	0.43	0.2

Closed Form Optimization

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





t	w_0	w_1	$\ell_{\mathcal{D}}(w_0, w_1)$
1	0.59	0.43	0.2

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 costs