10-301/601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Linear Regression

+

Optimization for ML

Matt Gormley
Lecture 8
Feb. 11, 2022

Q&A

Q: Could we just get rid of that pesky step size hyperparameter $\gamma^{(t)}$ in gradient descent?

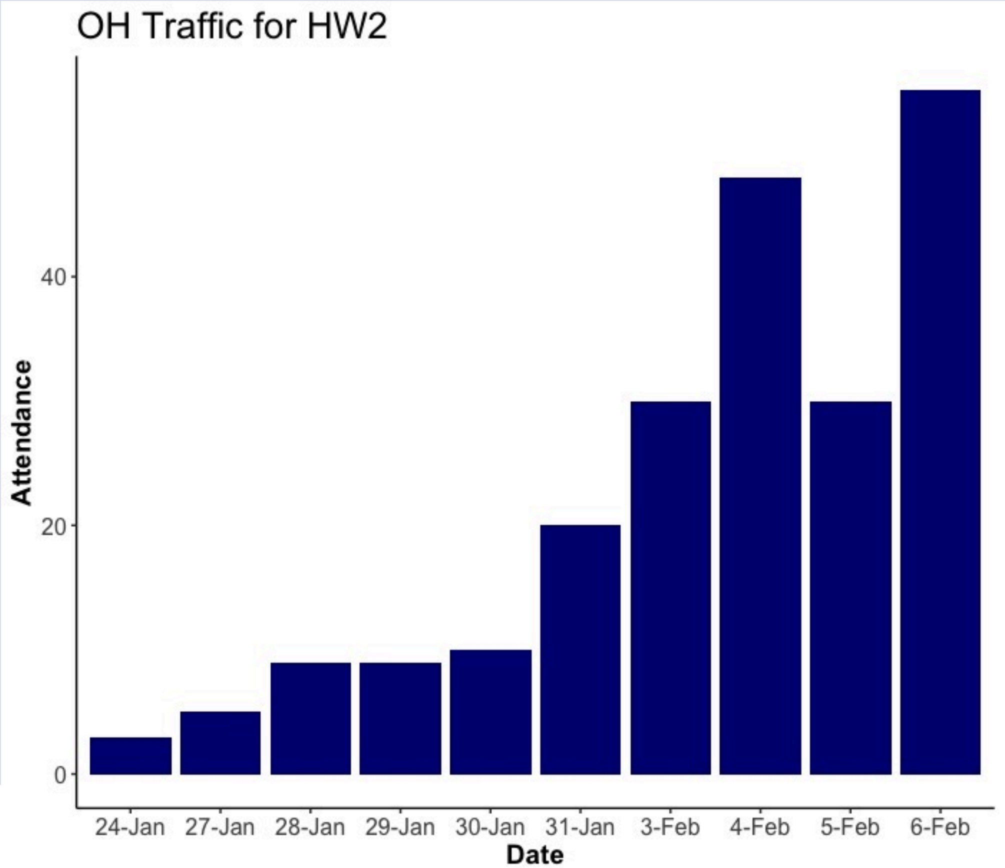
A: No!

In order to **prove** that gradient descent converges to a local minimum of a function, we need to **assume** gamma is properly defined.

Q&A

Q: How can I get more one-on-one interaction with the course staff?

A: Attend office hours as soon after the homework release as possible!



Q&A

Q: Can I email, tweeter, instasnap, or facetok my favorite TA directly about the course?

A: No. All course communication should be directed through one of the following channels:

- Piazza (public post)
- Piazza (private instructor post)
- Email to EAs eas-10-601@cs.cmu.edu
- Email to Matt (delays likely)
- In-person communication at OHs

Q&A

Q: I just asked a question in OH and now my TA is crying quietly -- what did I do wrong?

A: You've just committed the worst of crimes: asking a question that was directly answered in a recitation.

The TA you asked spent hours carefully writing careful recitation notes and solutions, practicing their recitation, responding to criticism / changes from me, etc.

To increase OH efficiency, please review the HW recitation before asking HW questions in OHs.

Reminders

- Practice for Exam 1
  - Mock Exam 1
    - Due: Wed, Feb. 16 at 11:59pm
    - See @683 for participation point details
  - Practice Problems 1 released on course website
- Exam 1: Thu, Feb. 17
  - Time: 6:30 – 8:30pm
  - Location: Your room/seat assignment will be announced on Piazza

EXAM 1 LOGISTICS

Exam 1

- **Time / Location**
  - **Time: Thu, Feb 17, at 6:30pm - 8:30pm**
  - **Location & Seats:** You have all been split across multiple rooms. Everyone has an assigned seat in one of these room.
  - Please watch Piazza carefully for announcements.
- **Logistics**
  - Covered material: Lecture 1 – Lecture 7
  - Format of questions:
    - Multiple choice
    - True / False (with justification)
    - Derivations
    - Short answers
    - Interpreting figures
    - Implementing algorithms on paper

Exam 1

- **How to Prepare**
  - Attend the midterm review lecture (right now!)
  - Participate in the Mock Exam
  - Review exam practice problems
  - Review this year’s homework problems
  - Consider whether you have achieved the “learning objectives” for each lecture / section
  - Write your one-page cheat sheet (back and front)

Midterm Exam

- **Advice (for during the exam)**
  - Solve the easy problems first
    (e.g. multiple choice before derivations)
    - if a problem seems extremely complicated you're likely missing something
  - Don't leave any answer blank!
  - If you make an assumption, write it down
  - If you look at a question and don't know the answer:
    - we probably haven't told you the answer
    - but we've told you enough to work it out
    - imagine arguing for some answer and see if you like it

Topics for Exam 1

- Foundations
  - Probability, Linear Algebra, Geometry, Calculus
  - Optimization
- Important Concepts
  - Overfitting
  - Experimental Design
- Classification
  - Decision Tree
  - KNN
  - Perceptron
- Regression
  - KNN Regression
  - Decision Tree Regression
  - Linear Regression

SAMPLE QUESTIONS

Sample Questions

5.2 Constructing decision trees

Consider the problem of predicting whether the university will be closed on a particular day. We will assume that the factors which decide this are whether there is a snowstorm, whether it is a weekend or an official holiday. Suppose we have the training examples described in the Table 5.2.

Snowstorm	Holiday	Weekend	Closed
T	T	F	F
T	T	F	T
F	T	F	F
T	T	F	F
F	F	F	F
F	F	F	T
T	F	F	T
F	F	F	T

Table 1: Training examples for decision tree

- **[2 points]** What would be the effect of the Weekend attribute on the decision tree if it were made the root? Explain in terms of information gain.

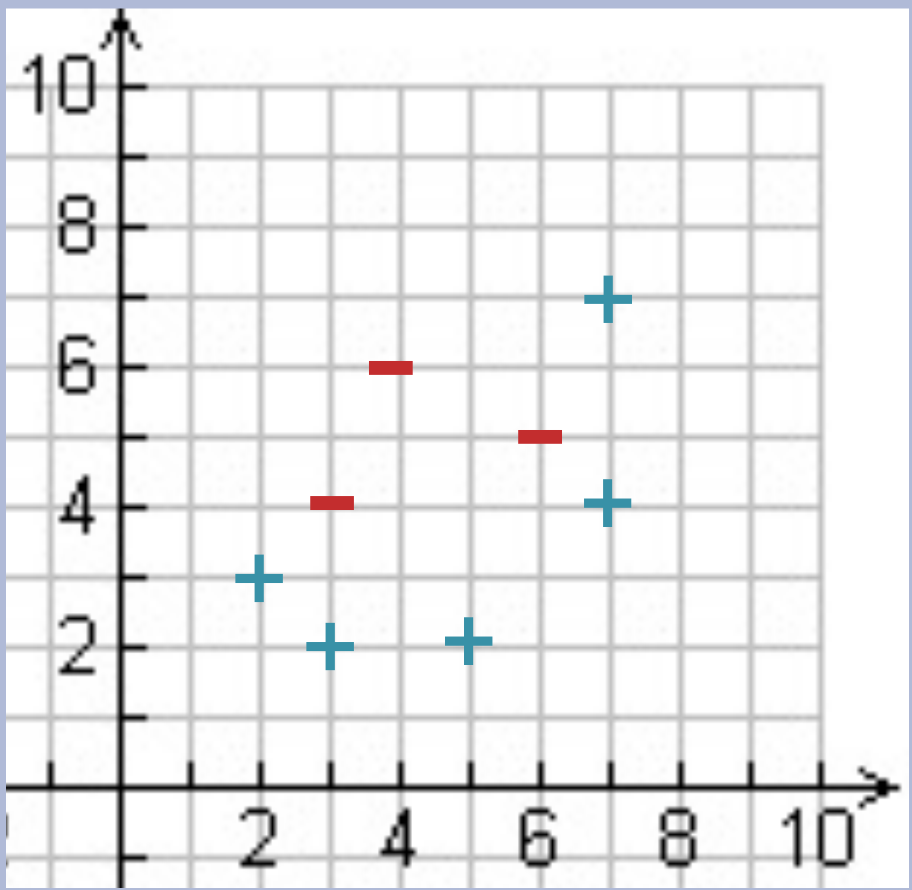
Q

- **[8 points]** If we cannot make Weekend the root node, which attribute should be made the root node of the decision tree? Explain your reasoning and show your calculations. (You may use $\log_2 0.75 = -0.4$ and $\log_2 0.25 = -2$)

Sample Questions

4 K-NN [12 pts]

Now we will apply K-Nearest Neighbors using Euclidean distance to a binary classification task. We assign the class of the test point to be the class of the majority of the k nearest neighbors.



3. [2 pts] What is the N-fold cross-validation error for the dataset shown in Figure 5? Assume k=1.

Sample Questions

4.1 True or False

Answer each of the following questions with **T** or **F** and **provide a one line justification**.

(a) [2 pts.] Consider two datasets $D^{(1)}$ and $D^{(2)}$ where $D^{(1)} = \{(x_1^{(1)}, y_1^{(1)}), ..., (x_n^{(1)}, y_n^{(1)})\}$ and $D^{(2)} = \{(x_1^{(2)}, y_1^{(2)}), ..., (x_m^{(2)}, y_m^{(2)})\}$ such that $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$. Suppose $d_1 > d_2$ and $n > m$. Then the maximum number of mistakes a perceptron algorithm will make is higher on dataset $D^{(1)}$ than on dataset $D^{(2)}$.

Q

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

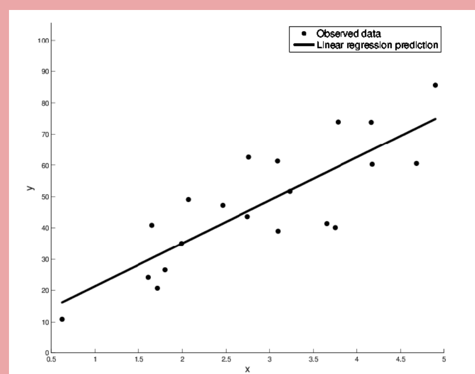


Figure 1: An observed data set and its associated regression line.

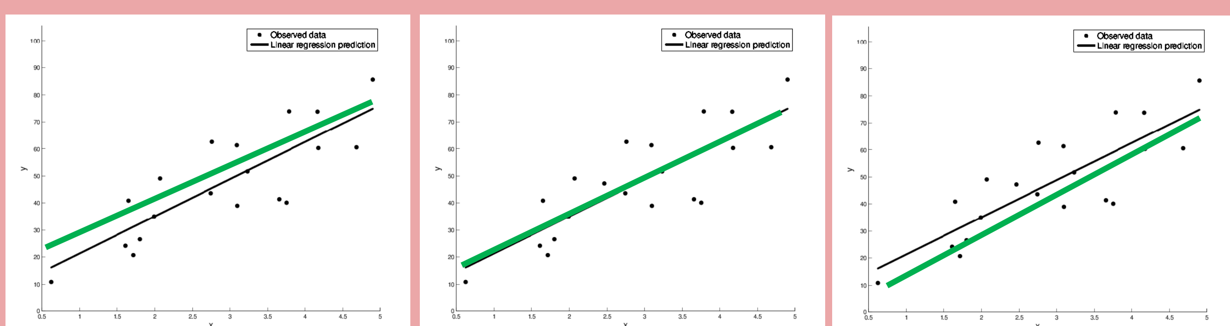
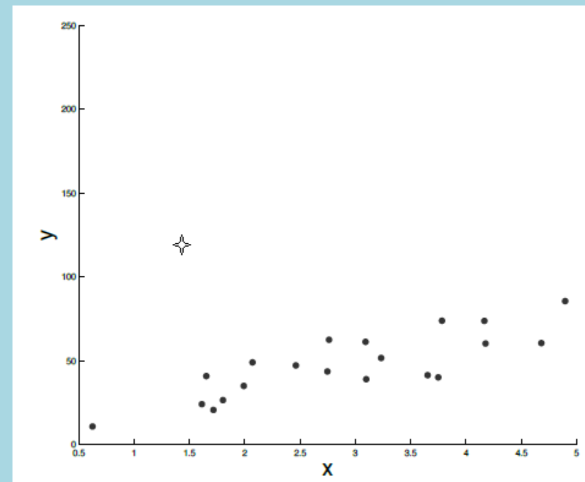


Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(a) Adding one outlier to the original data set.

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

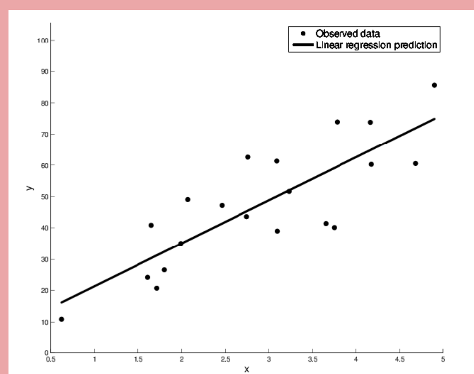


Figure 1: An observed data set and its associated regression line.

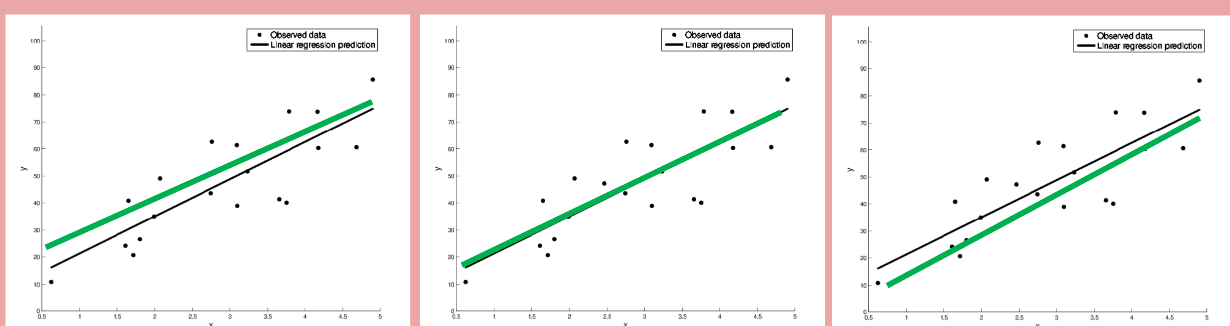
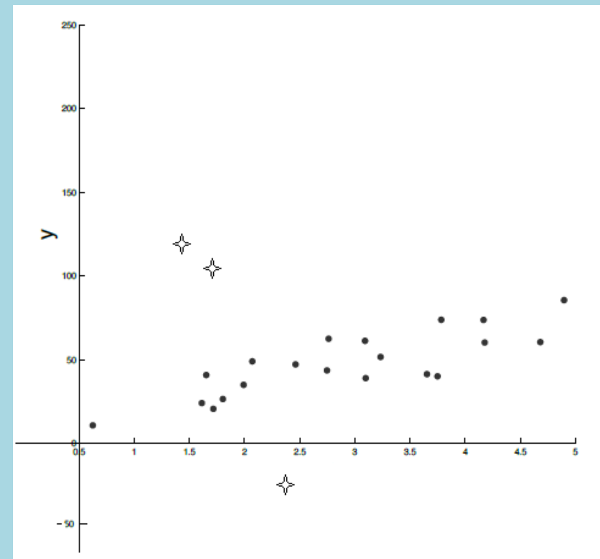


Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(c) Adding three outliers to the original data set. Two on one side and one on the other side.

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

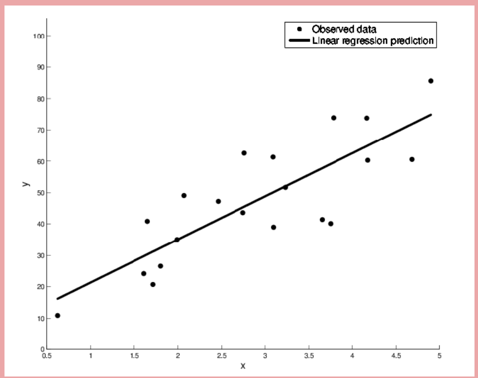


Figure 1: An observed data set and its associated regression line.

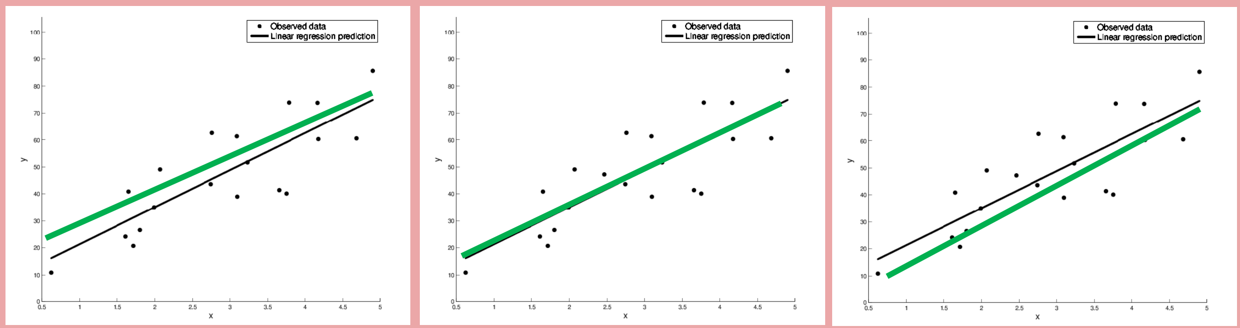
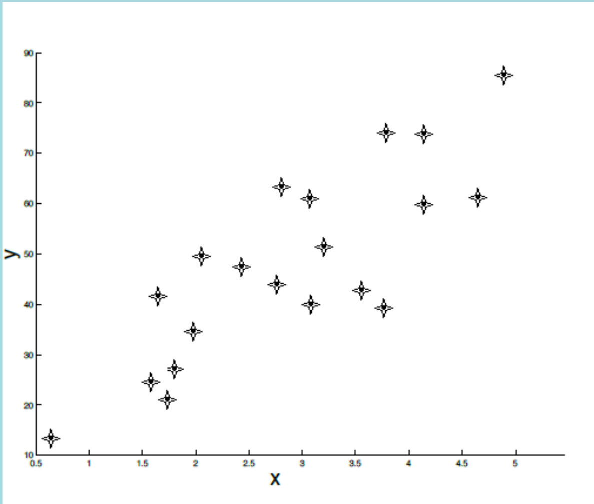


Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(d) Duplicating the original data set.

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

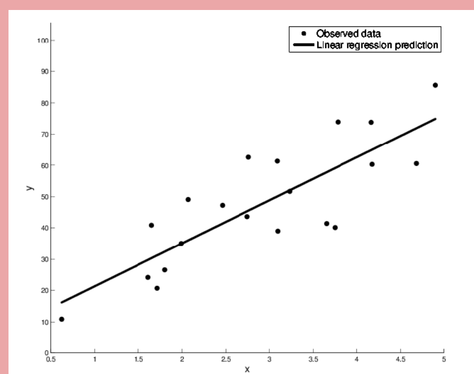


Figure 1: An observed data set and its associated regression line.

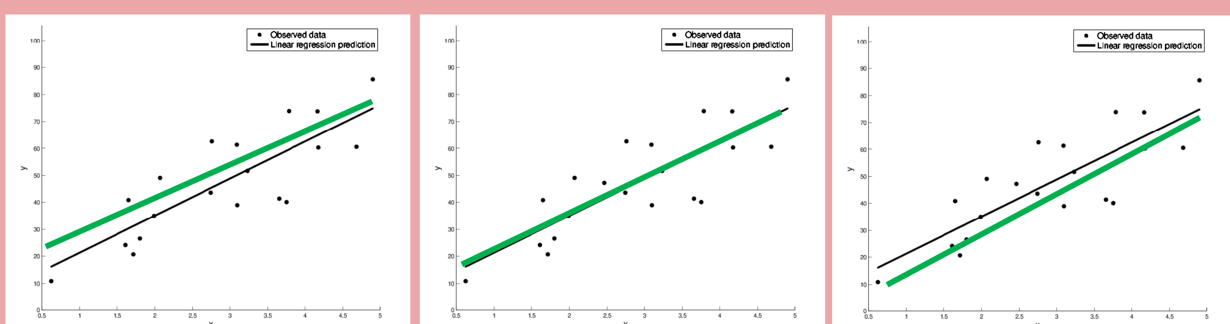
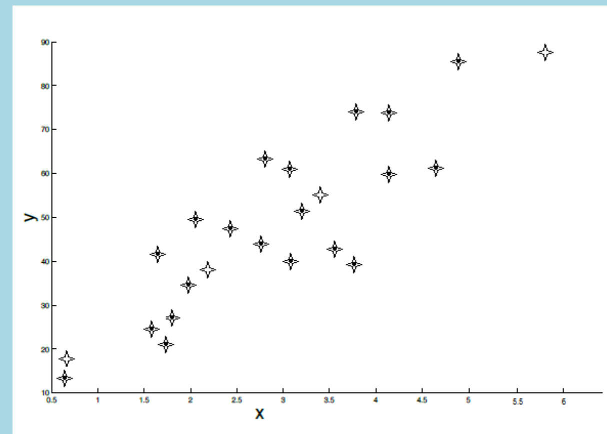


Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.

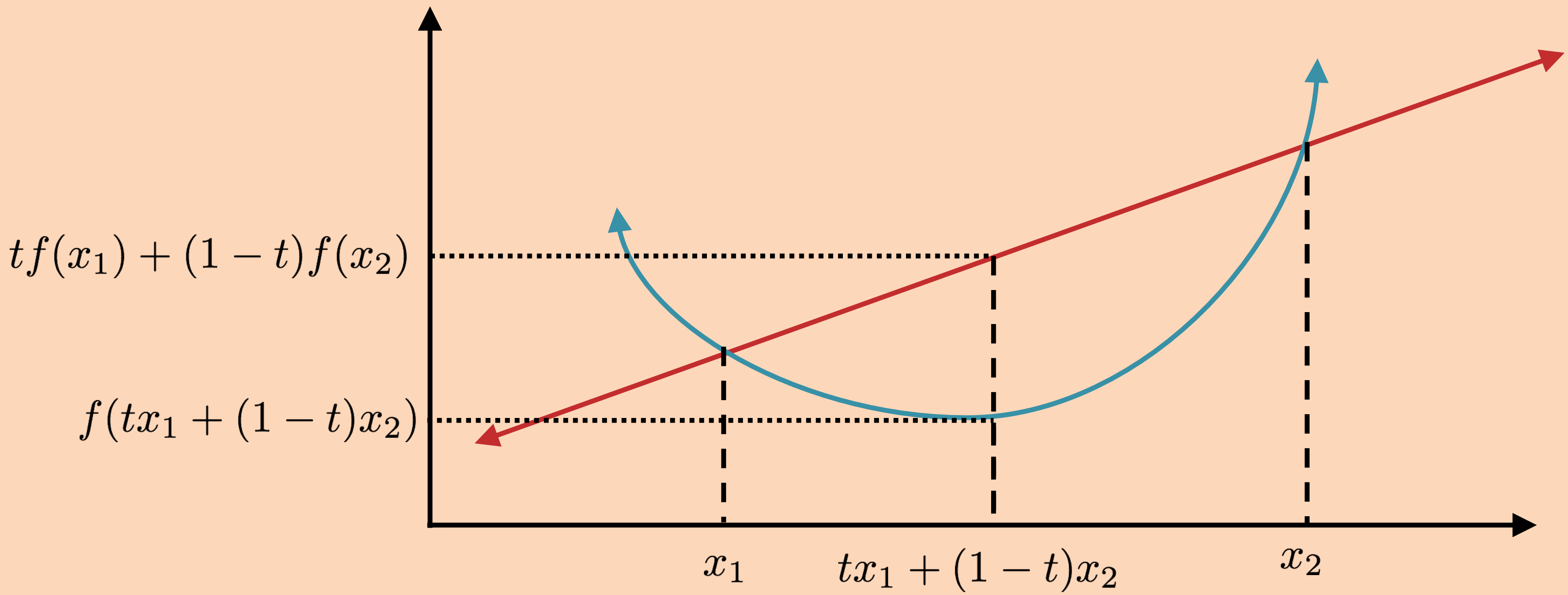
Q&A

CONVEXITY

Convexity

Function $f : \mathbb{R}^M \rightarrow \mathbb{R}$ is **convex**
if $\forall\ \mathbf{x}_1 \in \mathbb{R}^M, \mathbf{x}_2 \in \mathbb{R}^M, 0 \leq t \leq 1$:

$$f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2)$$

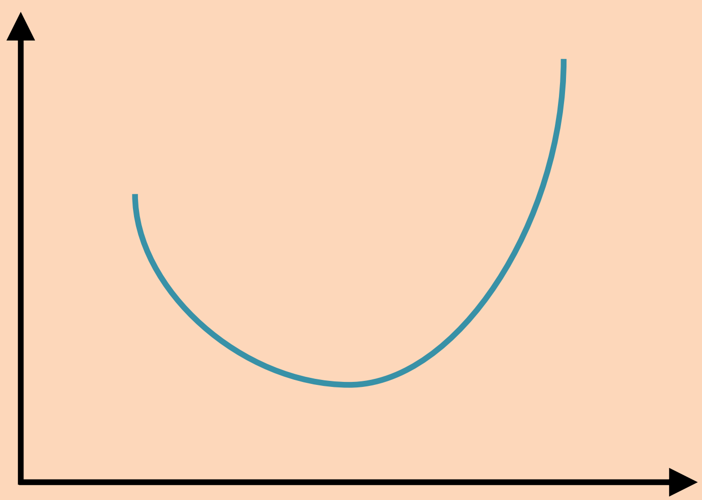


Convexity

Suppose we have a function $f(x) : \mathcal{X} \rightarrow \mathcal{Y}$.

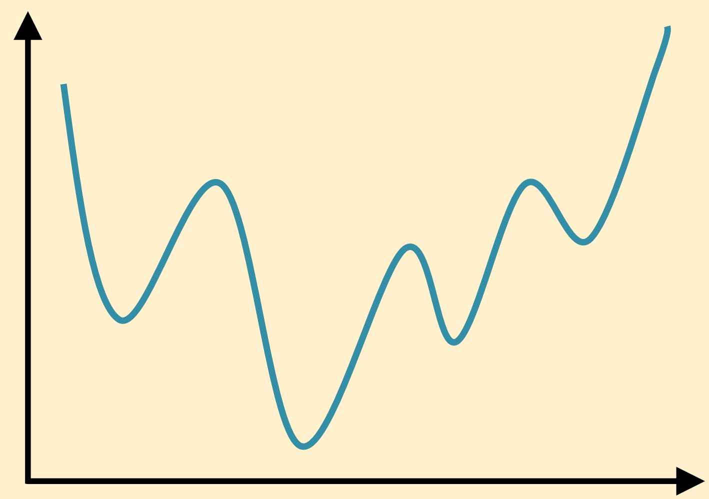
- The value x^* is a **global minimum** of f iff $f(x^*) \leq f(x), \forall x \in \mathcal{X}$.
- The value x^* is a **local minimum** of f iff $\exists \epsilon$ s.t. $f(x^*) \leq f(x), \forall x \in [x^* - \epsilon, x^* + \epsilon]$.

Convex Function



- Each **local minimum** is a **global minimum**

Nonconvex Function

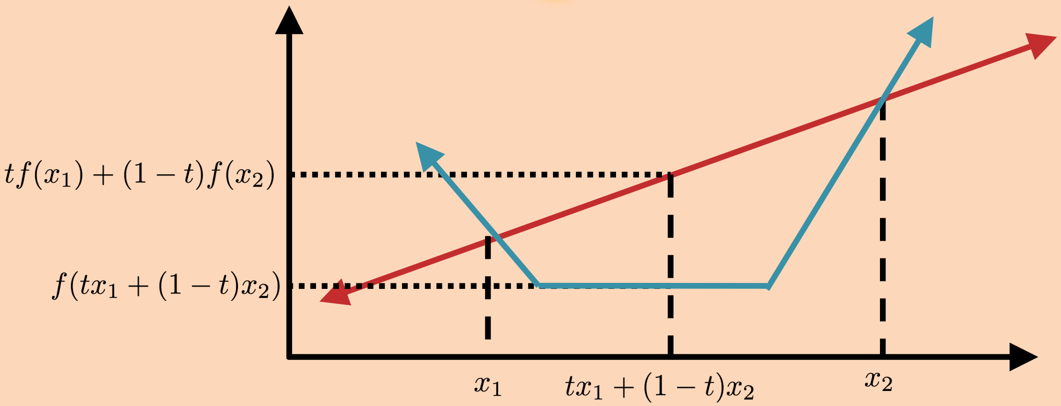


- A *nonconvex* function is **not convex**
- Each **local minimum** is **not** necessarily a **global minimum**

Convexity

Function $f : \mathbb{R}^M \to \mathbb{R}$ is **convex**
if $\forall\ \mathbf{x}_1 \in \mathbb{R}^M, \mathbf{x}_2 \in \mathbb{R}^M, 0 \leq t \leq 1$:

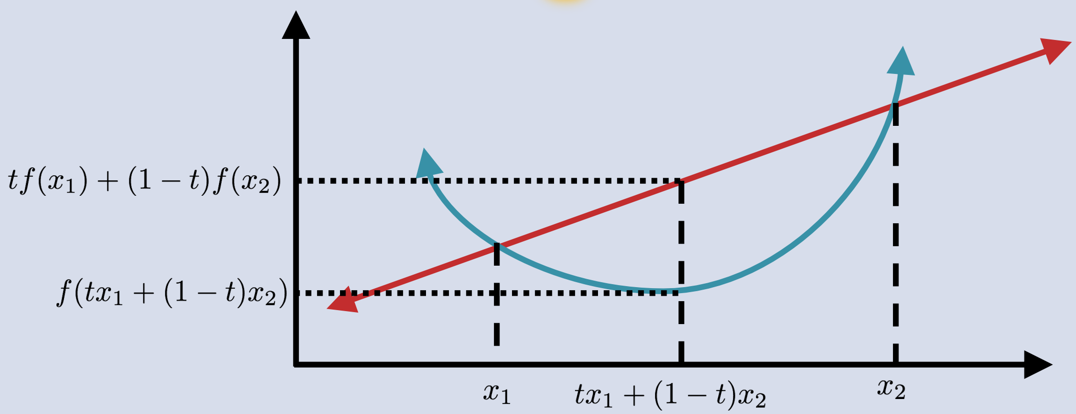
$$f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2)$$



Each **local minimum** of a **convex** function is also a **global minimum.**

Function $f : \mathbb{R}^M \to \mathbb{R}$ is **strictly convex**
if $\forall\ \mathbf{x}_1 \in \mathbb{R}^M, \mathbf{x}_2 \in \mathbb{R}^M, 0 \leq t \leq 1$:

$$f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) < tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2)$$



A **strictly convex** function has a **unique global minimum.**

CONVEXITY AND LINEAR REGRESSION

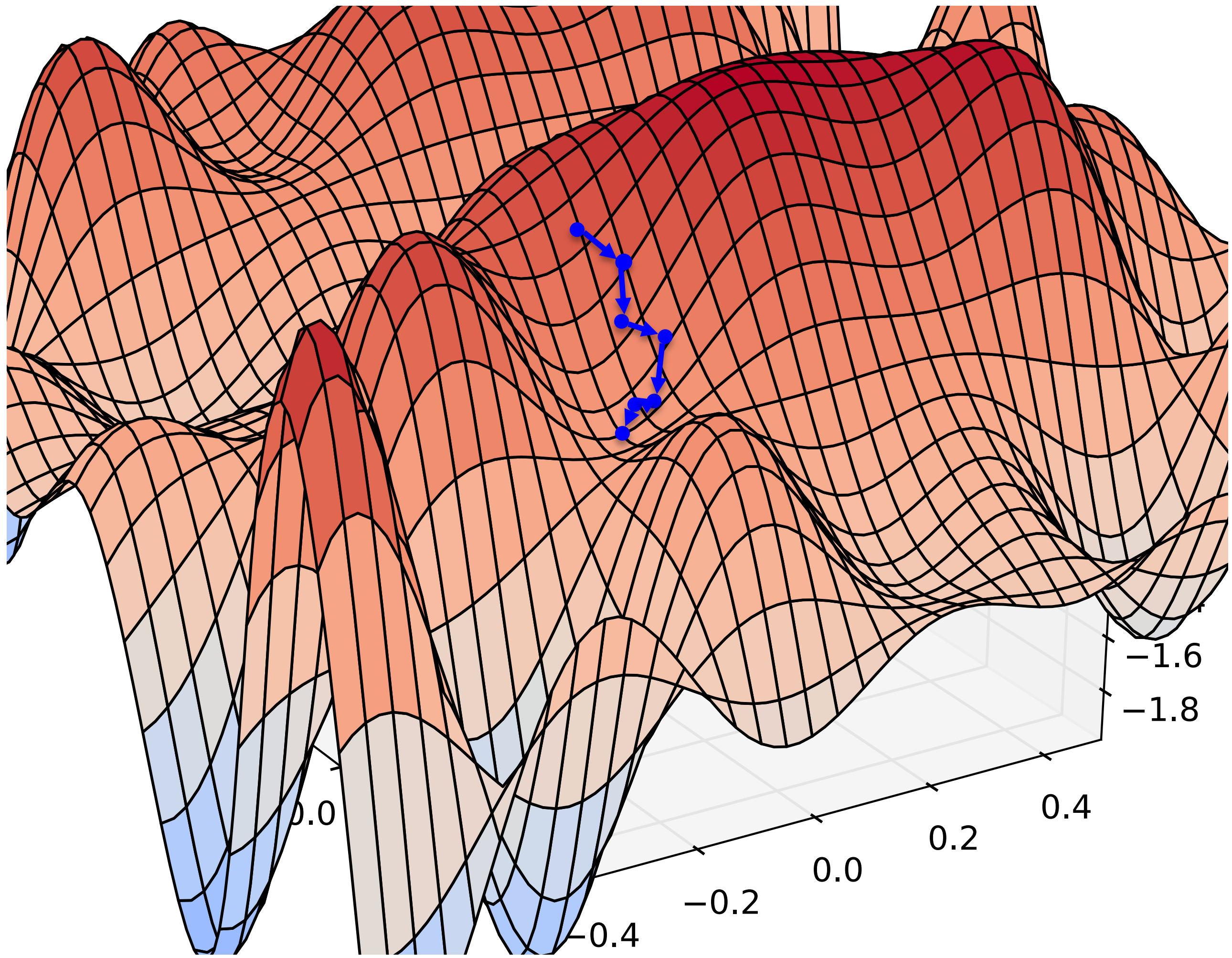
Convexity and Linear Regression

The **Mean Squared Error** function, which we minimize for learning the parameters of Linear Regression, **is convex**!

…but in the general case it is **not strictly convex.**

Gradient Descent & Convexity

- Gradient descent is a **local optimization algorithm**
- If the function is **nonconvex**, it will find a local minimum, not necessarily a global minimum
- If the function is **convex**, it will find a global minimum



Regression Loss Functions

In-Class Exercise:

Which of the following could be used as loss functions for training a linear regression model?

Select all that apply.

A. $\ell(\hat{y}, y) = ||\hat{y} - y||_2$

B. $\ell(\hat{y}, y) = |\hat{y} - y|$

C. $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$

D. $\ell(\hat{y}, y) = \frac{1}{4}(\hat{y} - y)^4$

E. $\ell(\hat{y}, y) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2 & \text{if } |\hat{y} - y| \leq \delta \\ \delta|\hat{y} - y| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$

F. $\ell(\hat{y}, y) = \log(\cosh(\hat{y} - y))$

OPTIMIZATION METHOD #2: CLOSED FORM SOLUTION

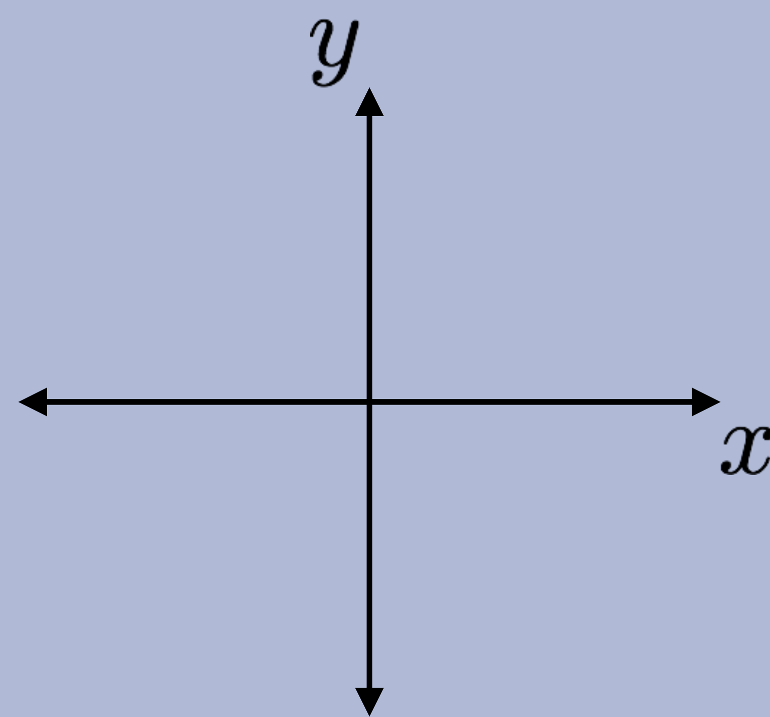
Calculus and Optimization

In-Class Exercise

Plot three functions:

1. $f(x) = x^3 - x$
2. $f'(x) = \frac{\partial y}{\partial x}$
3. $f''(x) = \frac{\partial^2 y}{\partial x^2}$

Answer Here:



Optimization: Closed form solutions

Chalkboard

- Zero Derivatives
- Example: 1-D function
- Example: higher dimensions

CLOSED FORM SOLUTION FOR LINEAR REGRESSION

Linear Regression as Function Approximation

$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
where $\mathbf{x} \in \mathbb{R}^M$ and $y \in \mathbb{R}$

1. Assume $\mathcal{D}$ generated as:

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$
$$y^{(i)} = h^*(\mathbf{x}^{(i)})$$

2. Choose hypothesis space, $\mathcal{H}$:
   all linear functions in M-dimensional space

$$\mathcal{H} = \{h_{\boldsymbol{\theta}} : h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}, \boldsymbol{\theta} \in \mathbb{R}^M\}$$

3. Choose an objective function:
   mean squared error (MSE)

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{N} \sum_{i=1}^N e_i^2 \\ &= \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})\right)^2 \\ &= \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2 \end{aligned}$$

4. Solve the unconstrained optimization problem via favorite method:
   - *gradient descent*
   - *closed form*
   - *stochastic gradient descent*
   - $\ldots$

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\, J(\boldsymbol{\theta})$$

5. Test time: given a new $\mathbf{x}$, make prediction $\hat{y}$

$$\hat{y} = h_{\hat{\boldsymbol{\theta}}}(\mathbf{x}) = \hat{\boldsymbol{\theta}}^T \mathbf{x}$$

Linear Regression: Closed Form

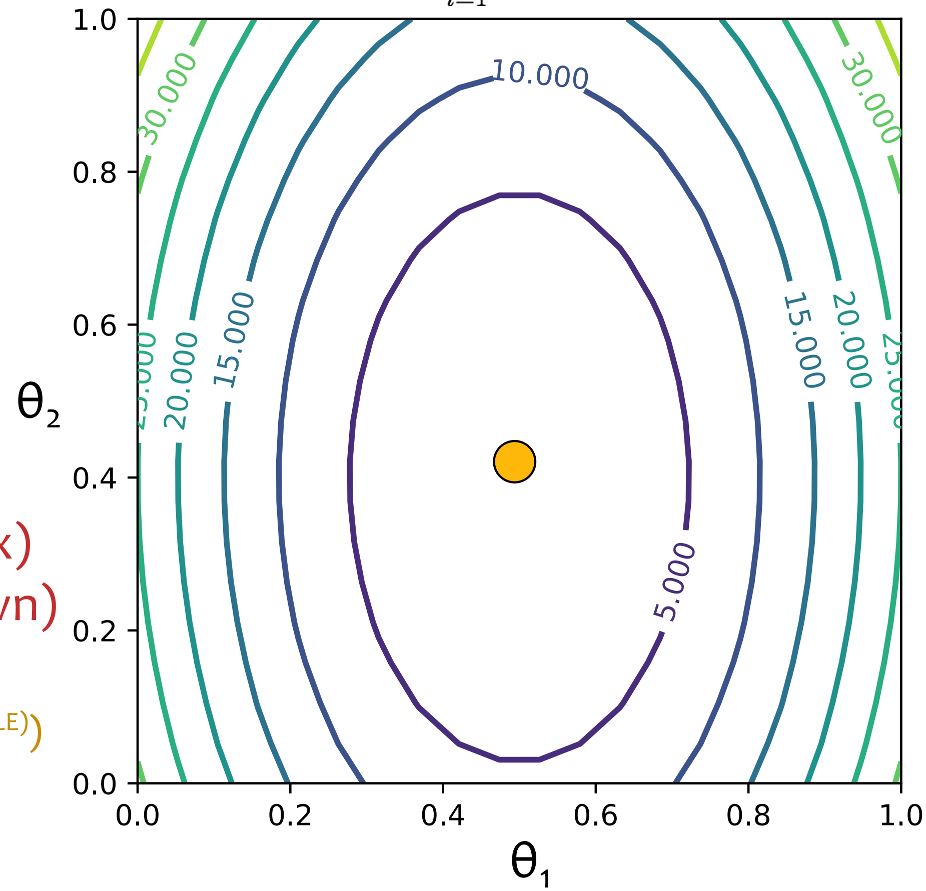
Optimization Method #2: Closed Form

1. Evaluate

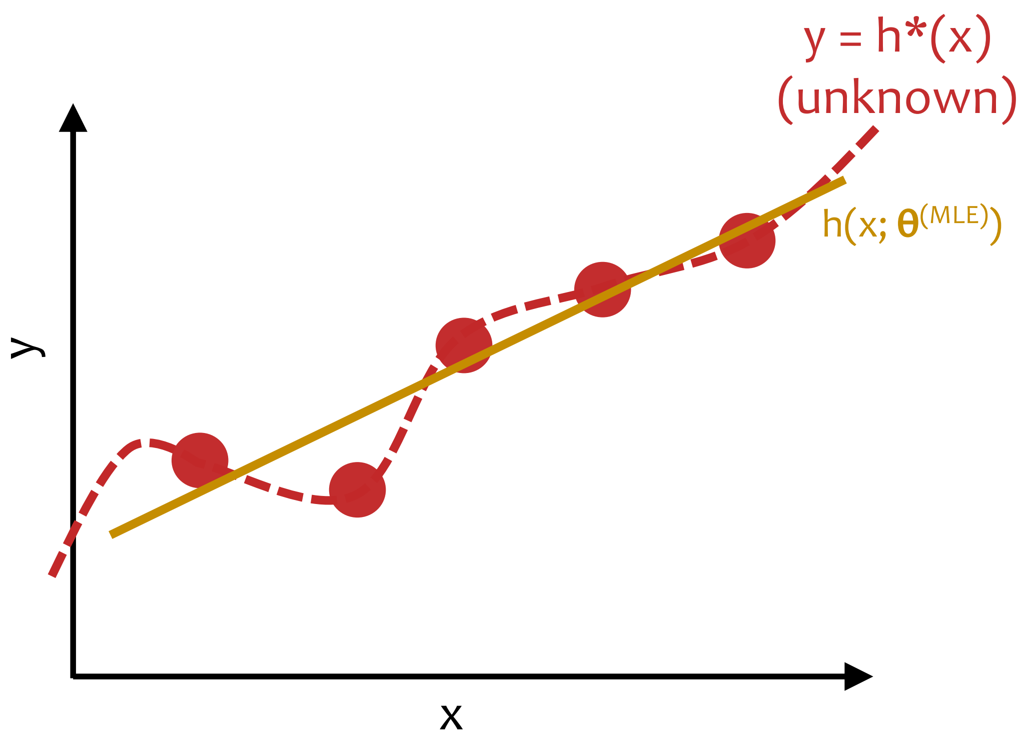
$$\boldsymbol{\theta}^{\text{MLE}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

2. Return $\boldsymbol{\theta}^{\text{MLE}}$

$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \boldsymbol{\theta}^T\mathbf{x}^{(i)}\right)^2$$



t	θ_1	θ_2	$J(\theta_1, \theta_2)$
MLE	0.59	0.43	0.2



Optimization for Linear Regression

Chalkboard

– Closed-form (Normal Equations)

COMPUTATIONAL COMPLEXITY

Computational Complexity of OLS

To solve the Ordinary Least Squares problem we compute:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(y^{(i)} - (\boldsymbol{\theta}^T\mathbf{x}^{(i)}))^2$$
$$= (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$$

The resulting shape of the matrices:

$$\underbrace{(\underbrace{\mathbf{X}^T}_{M\times N}\ \underbrace{\mathbf{X}}_{N\times M})^{-1}}_{M\times M}\ \underbrace{(\underbrace{\mathbf{X}^T}_{M\times N}\ \underbrace{\mathbf{Y}}_{N\times 1})}_{M\times 1}$$

Background: Matrix Multiplication Given matrices $\mathbf{A}$ and $\mathbf{B}$

- If $\mathbf{A}$ is $q \times r$ and $\mathbf{B}$ is $r \times s$, computing $\mathbf{AB}$ takes $O(qrs)$
- If $\mathbf{A}$ and $\mathbf{B}$ are $q \times q$, computing $\mathbf{AB}$ takes $O(q^{2.373})$
- If $\mathbf{A}$ is $q \times q$, computing A^{-1} takes $O(q^{2.373})$.

Computational Complexity of OLS:

$\mathbf{X}^T\mathbf{X}$	$O(M^2N)$
$(\quad\quad)^{-1}$	$O(M^{2.373})$
$\mathbf{X}^T\mathbf{Y}$	$O(MN)$
$(\quad\quad)^{-1}(\quad\quad)$	$O(M^2)$
total	$O(M^2N + M^{2.373})$

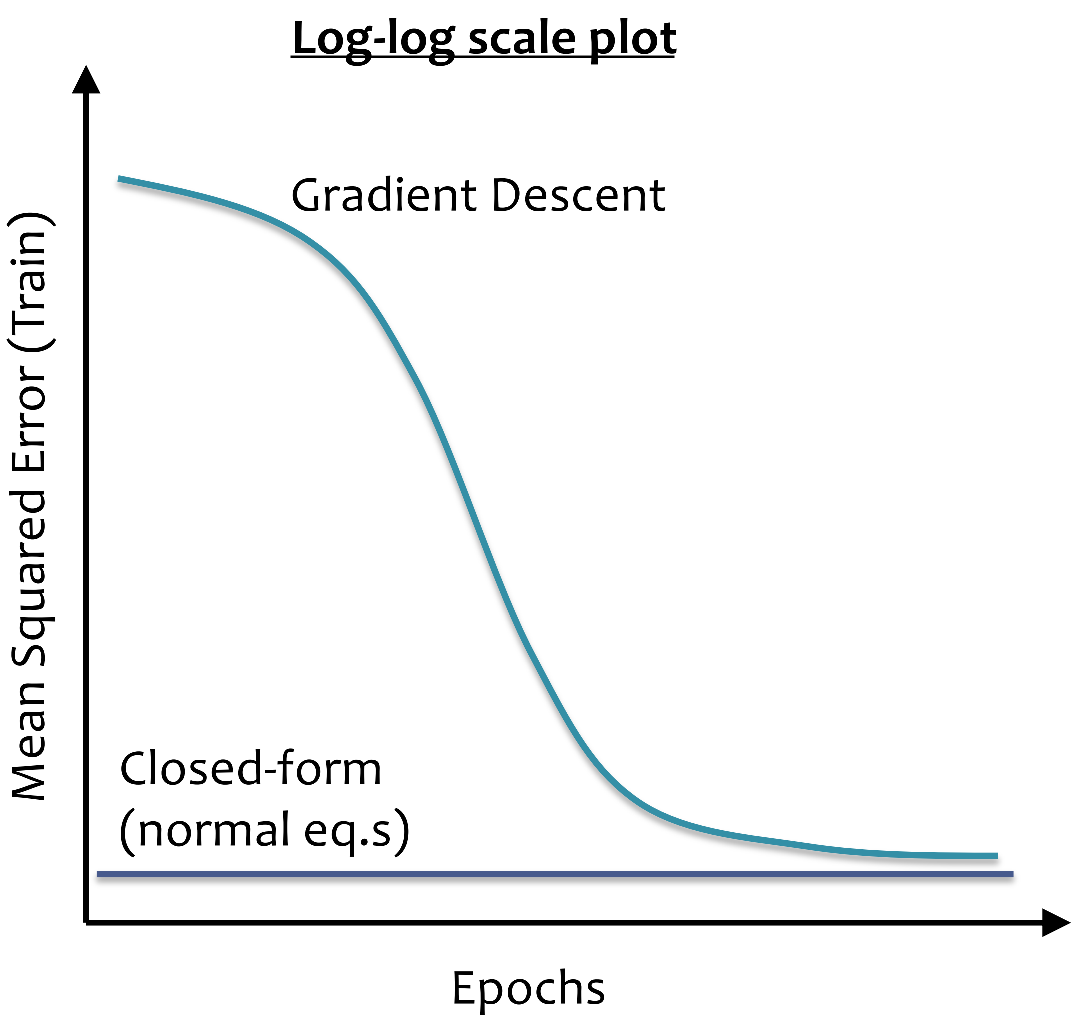
Linear in # of examples, N
Polynomial in # of features, M

Gradient Descent

Cases to consider gradient descent:

1. What if we **can not** find a closed-form solution?
2. What if we **can**, but it's inefficient to compute?
3. What if we **can**, but it's numerically unstable to compute?

Empirical Convergence



- *Def*: an **epoch** is a single pass through the training data

1. For GD, only **one update** per epoch
2. For SGD, ***N* updates** per epoch
   N = (# train examples)

- SGD reduces MSE much more rapidly than GD
- For GD / SGD, training MSE is initially large due to uninformed initialization

LINEAR REGRESSION: SOLUTION UNIQUENESS

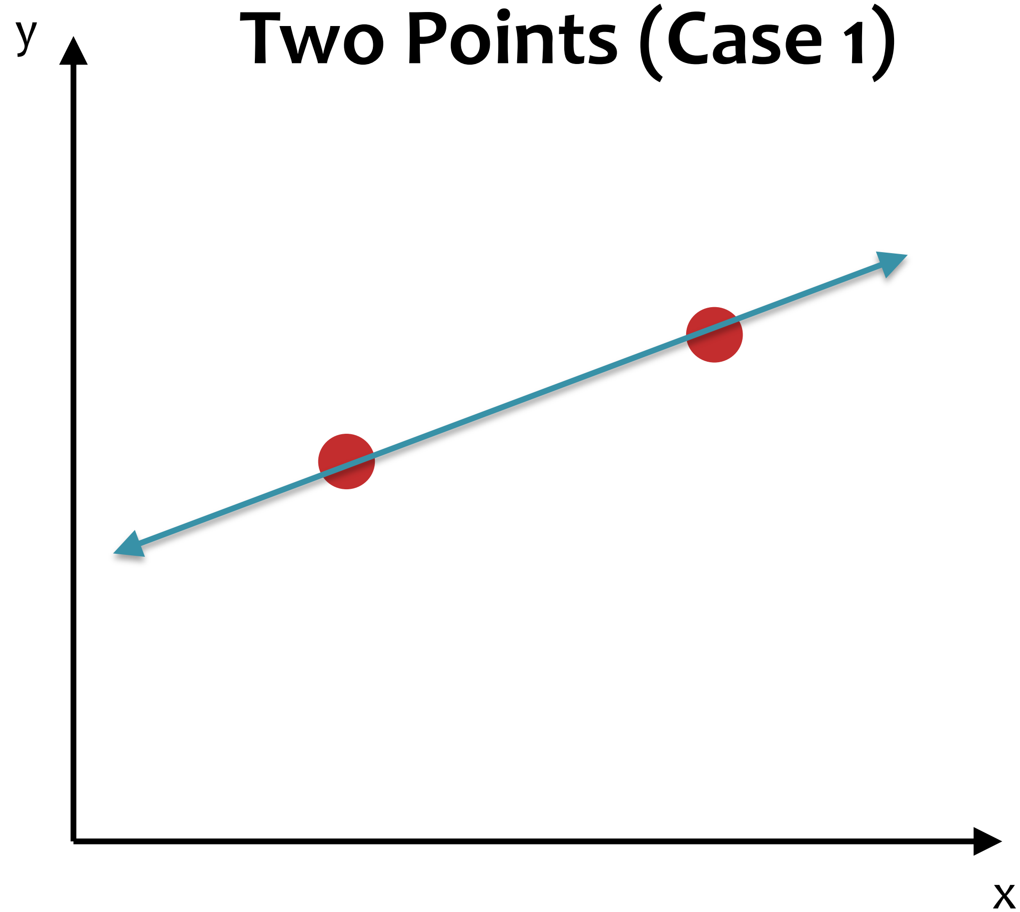
Linear Regression: Uniqueness

Question:

Consider a 1D linear regression model trained to minimize MSE.

How many solutions (i.e. sets of parameters w,b) are there for the given dataset?

Two Points (Case 1)

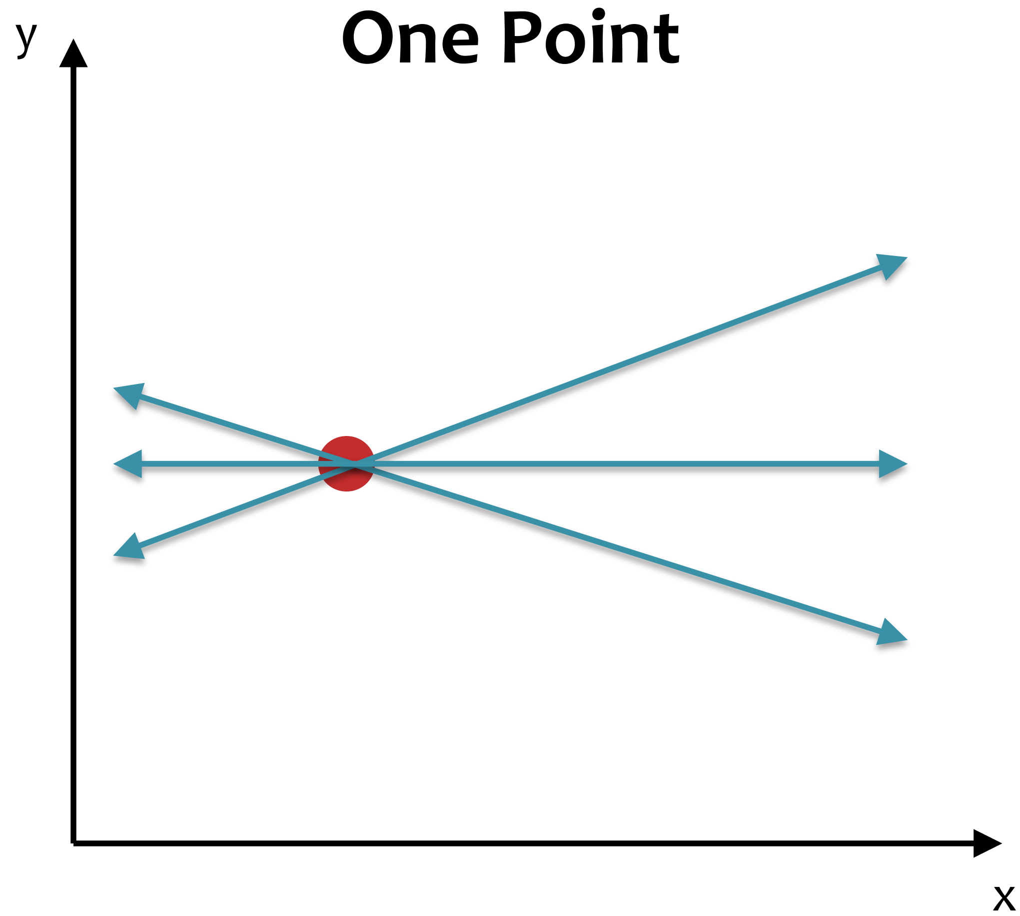


Linear Regression: Uniqueness

Question:

Consider a 1D linear regression model trained to minimize MSE.

How many solutions (i.e. sets of parameters w,b) are there for the given dataset?

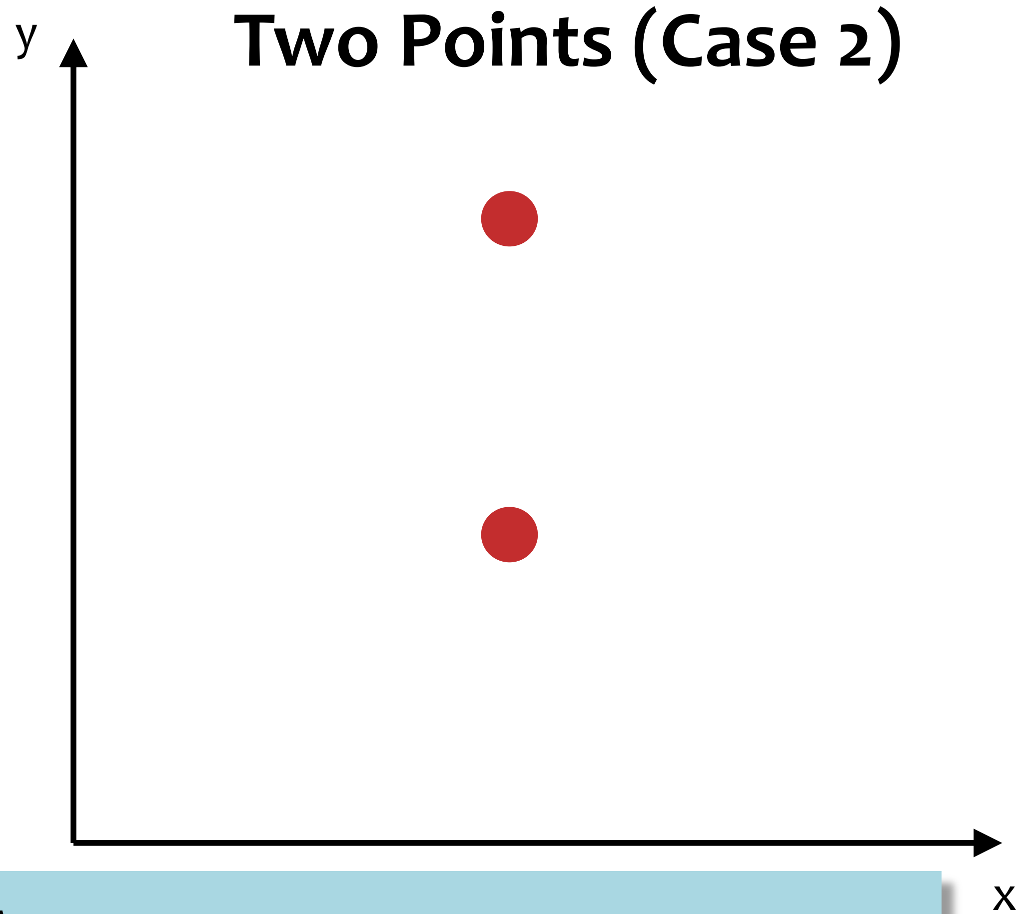


Linear Regression: Uniqueness

Question:

Consider a 1D linear regression model trained to minimize MSE.

How many solutions (i.e. sets of parameters w,b) are there for the given dataset?



Answer:

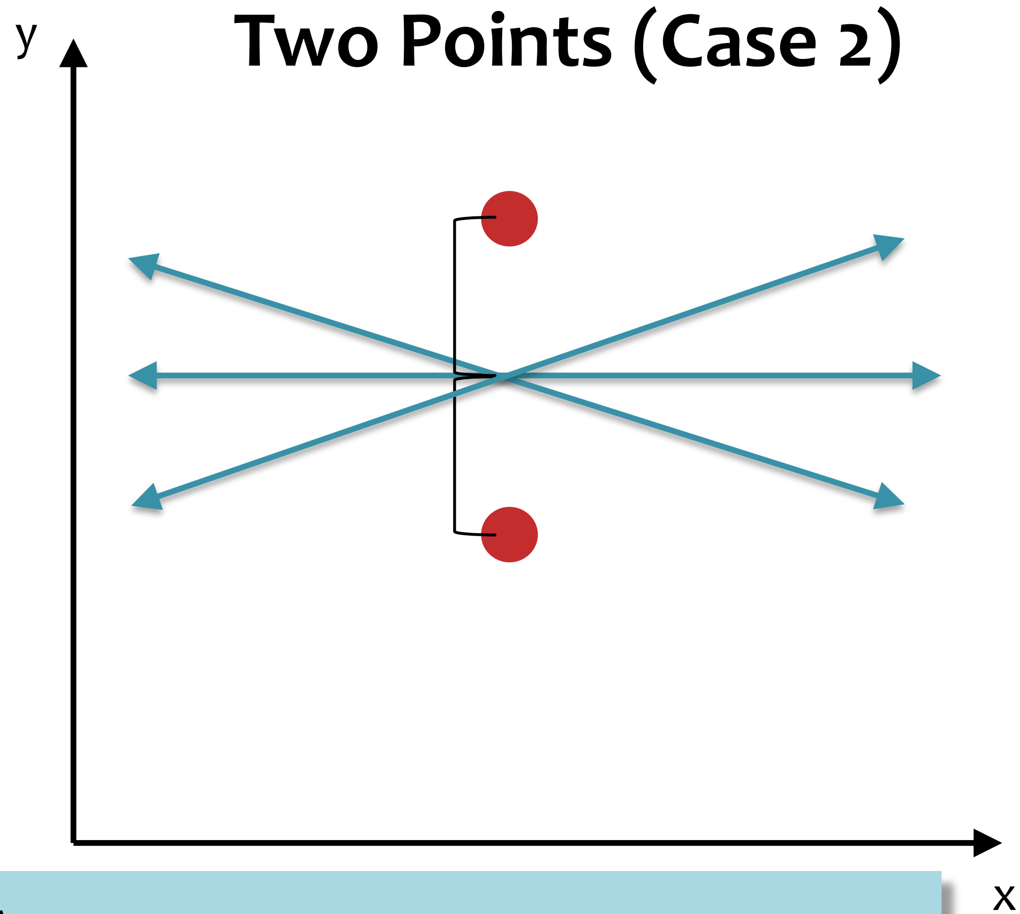
A: 0 B: 1 C: 2 D: +∞

Linear Regression: Uniqueness

Question:

Consider a 1D linear regression model trained to minimize MSE.

How many solutions (i.e. sets of parameters w,b) are there for the given dataset?



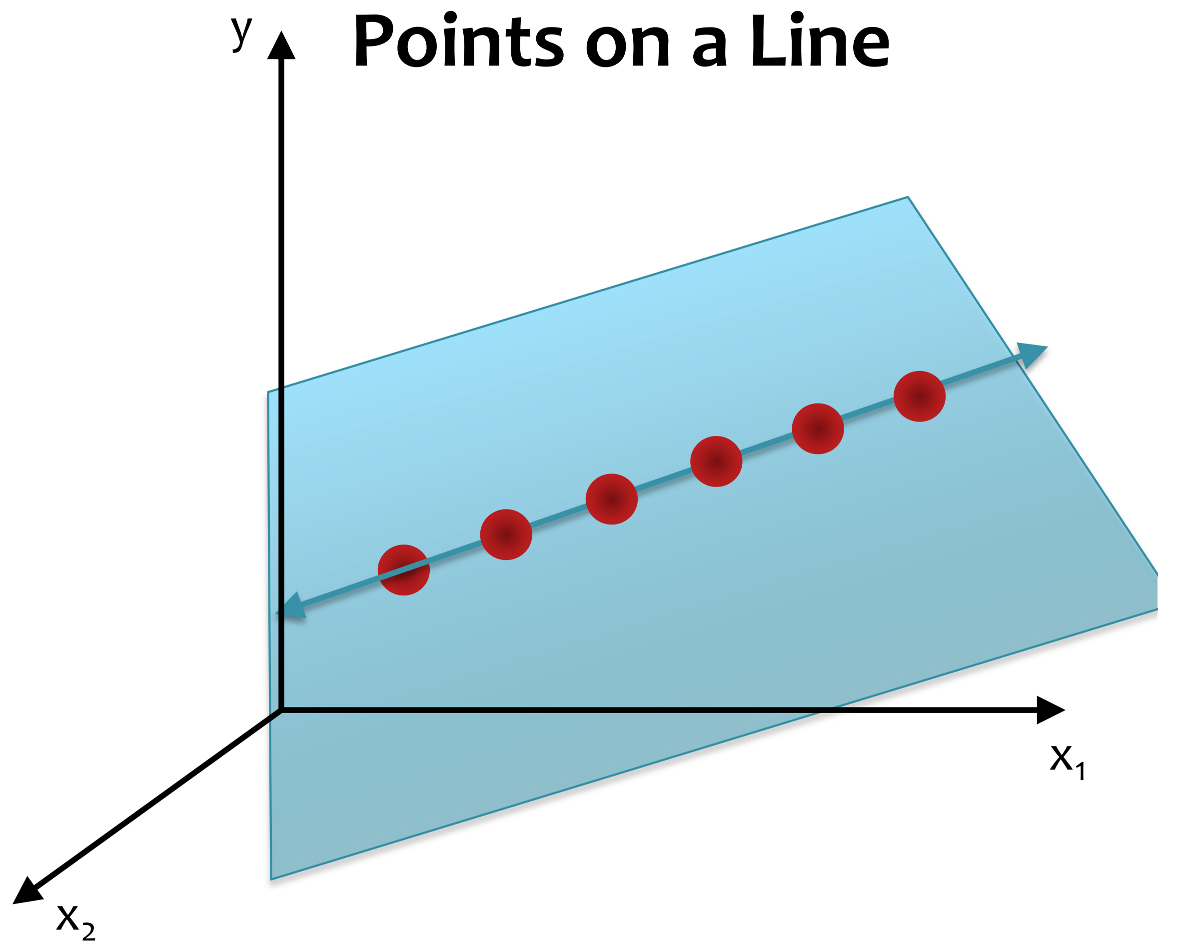
Answer:

A: 0   B: 1   C: 2   D: $+\infty$

Linear Regression: Uniqueness

Question:

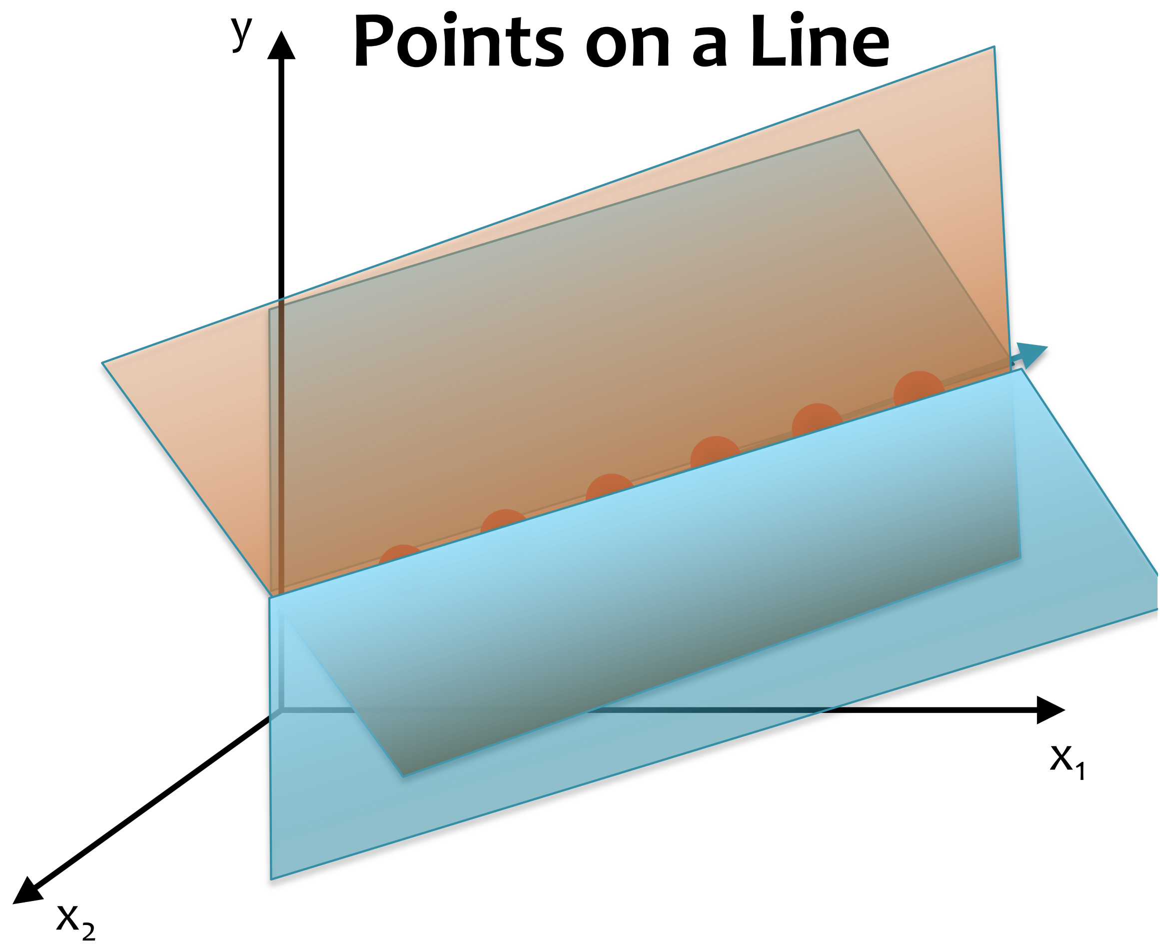
- Consider a **2D** linear regression model trained to minimize MSE
- How many solutions (i.e. sets of parameters w_1, w_2, b) are there for the given dataset?



Linear Regression: Uniqueness

Question:

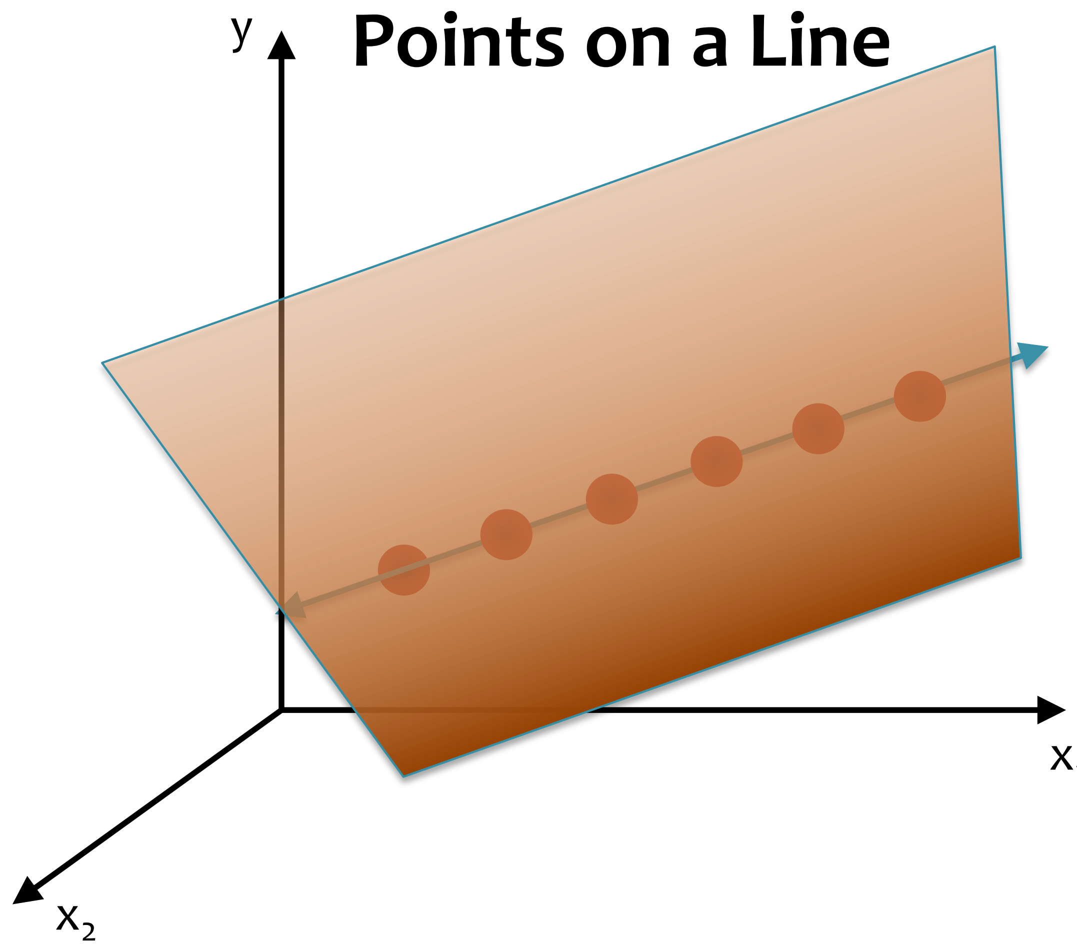
- Consider a **2D** linear regression model trained to minimize MSE
- How many solutions (i.e. sets of parameters w_1, w_2, b) are there for the given dataset?



Linear Regression: Uniqueness

Question:

- Consider a **2D** linear regression model trained to minimize MSE
- How many solutions (i.e. sets of parameters w_1, w_2, b) are there for the given dataset?



Linear Regression: Uniqueness

To solve the Ordinary Least Squares problem we compute:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(y^{(i)} - (\boldsymbol{\theta}^T\mathbf{x}^{(i)}))^2$$
$$= (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$$

These geometric intuitions align with the linear algebraic intuitions we can derive from the normal equations.

1. If $(\mathbf{X}^T\mathbf{X})$ **is invertible,** then there is exactly one solution.
2. If $(\mathbf{X}^T\mathbf{X})$ **is not invertible,** then there are either no solutions or infinitely many solutions.

Linear Regression: Uniqueness

To solve the Ordinary Least Squares problem we compute:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(y^{(i)} - (\boldsymbol{\theta}^T\mathbf{x}^{(i)}))^2$$
$$= (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$$

These geometric intuitions align with the linear algebraic intuitions we can derive from the normal equations.

1. If $(\mathbf{X}^T\mathbf{X})$ **is invertible,** then there is exactly one solution.
2. If $(\mathbf{X}^T\mathbf{X})$ **is not inv**
   no solutions or inf

Invertability of $(\mathbf{X}^T\mathbf{X})$ is equivalent to X being **full rank.** That is, there is **no feature that is a linear combination of the other features.**

Solving Linear Regression

Question:

True or False: If Mean Squared Error (i.e. $\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - h(\mathbf{x}^{(i)}))^2$) has a unique minimizer (i.e. argmin), then Mean Absolute Error (i.e. $\frac{1}{N}\sum_{i=1}^{N}|y^{(i)} - h(\mathbf{x}^{(i)})|$) must also have a unique minimizer.

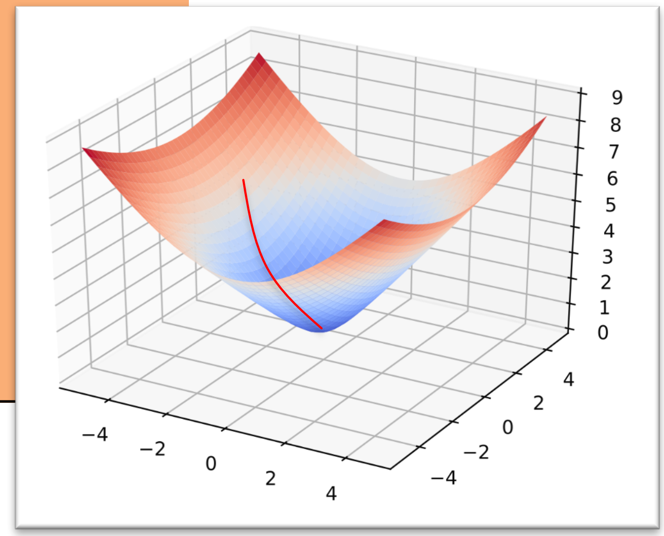
Answer:

OPTIMIZATION METHOD #3: STOCHASTIC GRADIENT DESCENT

Gradient Descent

Algorithm 1 Gradient Descent

1: **procedure** GD($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
2: $\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
3: $\quad$**while** not converged **do**
4: $\quad\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
5: $\quad$**return** $\boldsymbol{\theta}$

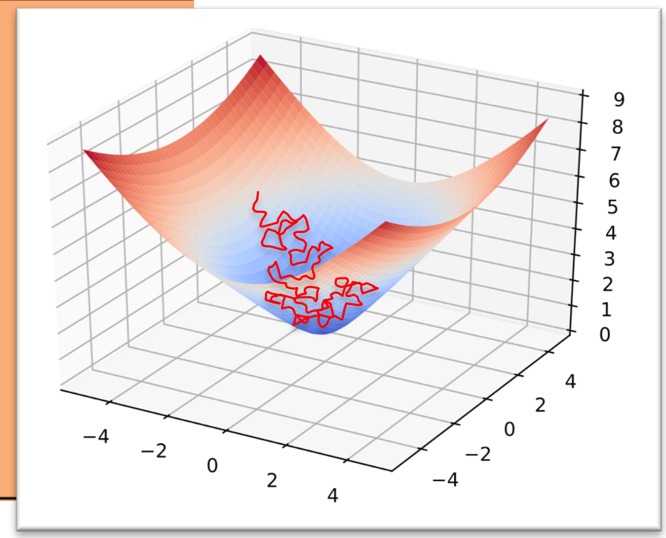


M

Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

procedure SGD($\mathcal{D}, \boldsymbol{\theta}^{(0)}$)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
    while not converged **do**
        $i \sim$ Uniform($\{1, 2, \ldots, N\}$)
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta})$
    return $\boldsymbol{\theta}$



per-example objective:

$$J^{(i)}(\boldsymbol{\theta})$$

original objective:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} J^{(i)}(\boldsymbol{\theta})$$