6.036: Introduction to Machine Learning

Lecture start: Tuesdays 9:35am

Lecture team: Prof. Tamara Broderick & TA Elizabeth Zou

Questions? Ask on Piazza

Materials: slides, video will all be available on Canvas

Live Zoom feed: https://mit.zoom.us/j/94238622313

Today's Plan

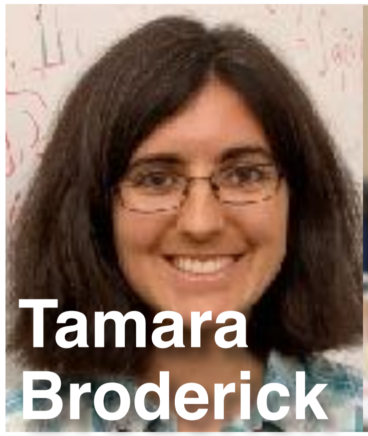
I. Meet the team

II. Machine learning setup

III. Linear regression

IV. Regularization

6.036: Introduction to Machine Learning



Lecture start: Tuesdays 9:35am

Lecture team: Prof. Tamara Broderick & TA Elizabeth Zou

Questions? Ask on Piazza

Materials: slides, video will all be available on Canvas

Live Zoom feed: https://mit.zoom.us/j/94238622313

Today's Plan

I. Meet the team
II. Machine learning setup
III. Linear regression
IV. Regularization

6.036: Introduction to Machine Learning

Lecture start: Tuesdays 9:35am

Lecture team: Prof. Tamara Broderick & TA Elizabeth Zou

Questions? Ask on Piazza: “lecture (week) 2” folder

Materials: slides, video will all be available on Canvas

Live Zoom feed: https://mit.zoom.us/j/94238622313

Today’s Plan

I. Meet the team

II. Machine learning setup

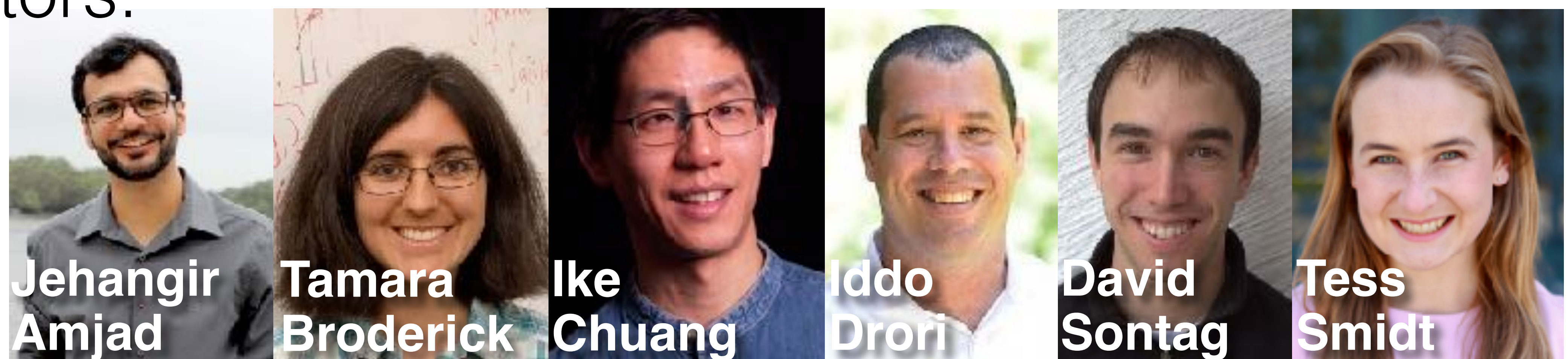
III. Linear regression

IV. Regularization

6.036: Introduction to Machine Learning, Staff

6.036: Introduction to Machine Learning, Staff

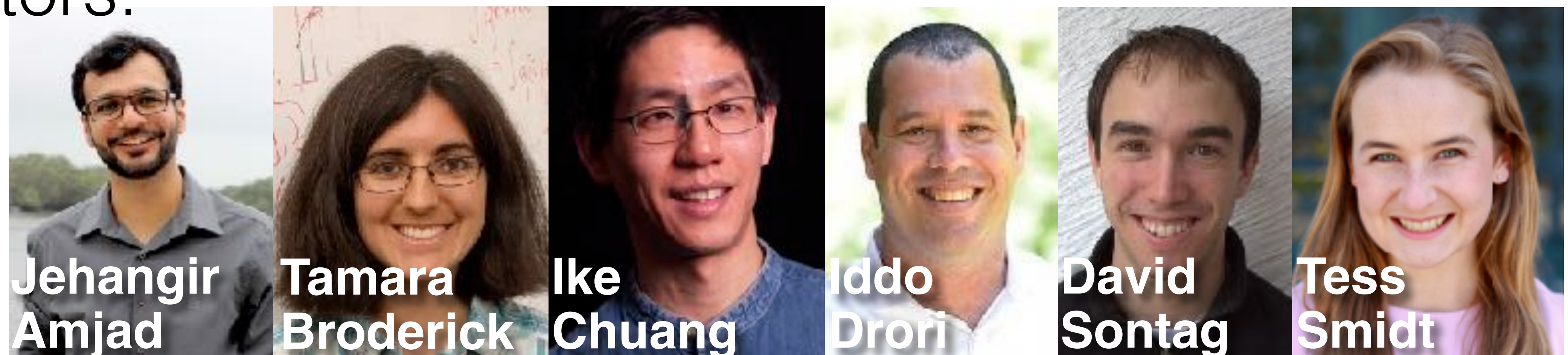
Instructors:



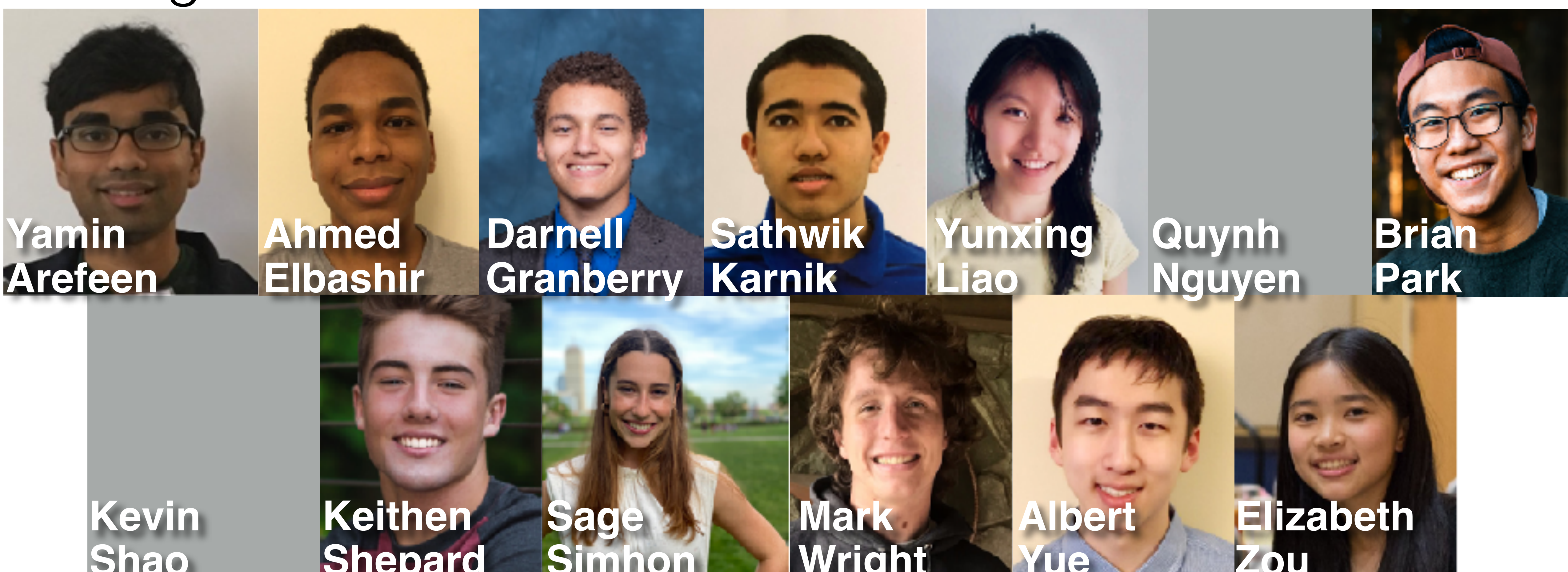
MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.036: Introduction to Machine Learning, Staff

Instructors:

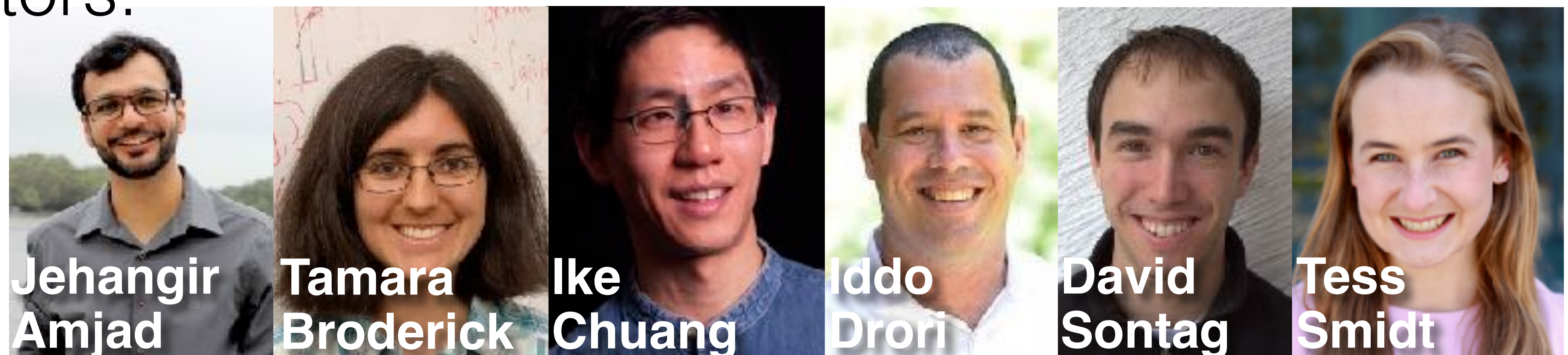


Teaching Assistants:

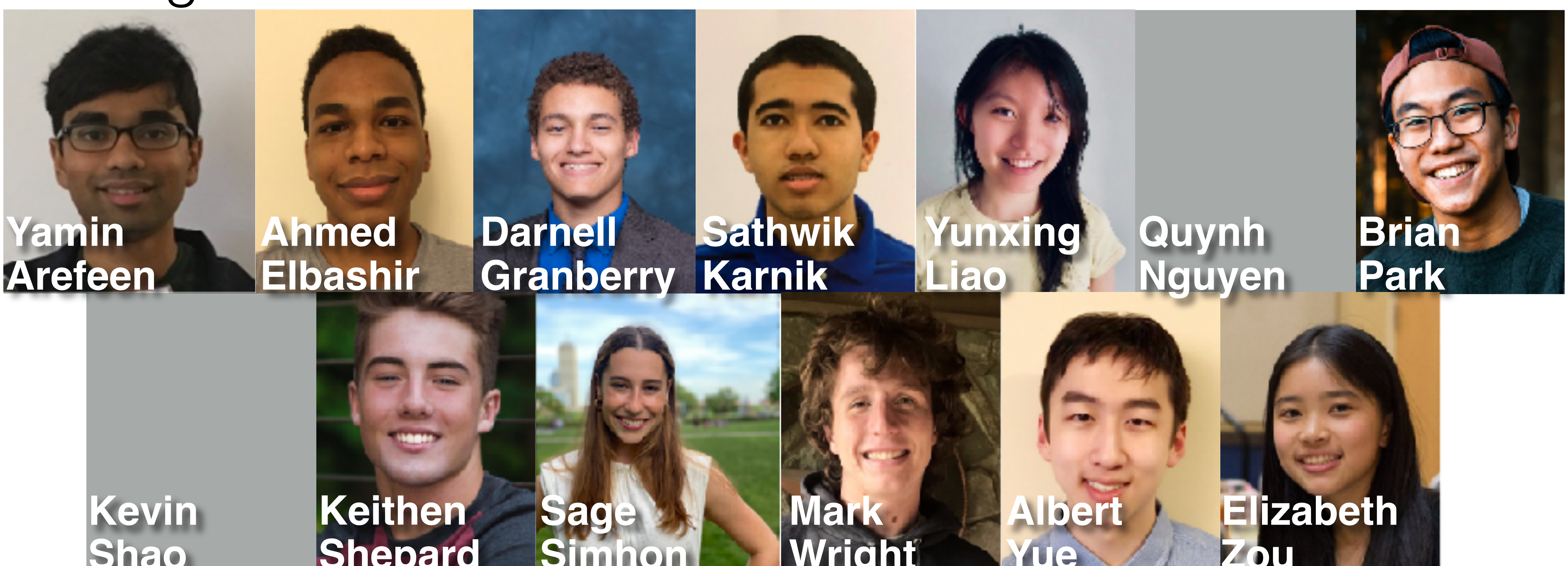


6.036: Introduction to Machine Learning, Staff

Instructors:



Teaching Assistants:



And Lab Assistants!

Machine learning (ML): why & what

Machine learning (ML): why & what

nature

Search Login

Content About Publish

NEWS | 22 July 2021

DeepMind's AI predicts structures for a vast trove of proteins

AlphaFold neural network produced a 'totally transformative' database of more than 350,000 structures from *Homo sapiens* and 20 model organisms.

Ewen Callaway

Underpinning the latest version of AlphaFold is a novel machine learning approach that incorporates physical and biological knowledge about protein structure, leveraging multi-sequence alignments, into the design of the deep learning algorithm.

[https://www.nature.com/articles/d41586-021-02025-4; https://www.nature.com/articles/s41586-021-03819-2]

Machine learning (ML): why & what

nature

Search Login

Conter

NEWS

De

str

of

Alpha

datab

sapie

Ewen

TechRepublic.

Machine learning algorithm confirms 50 new exoplanets in historic first

f in

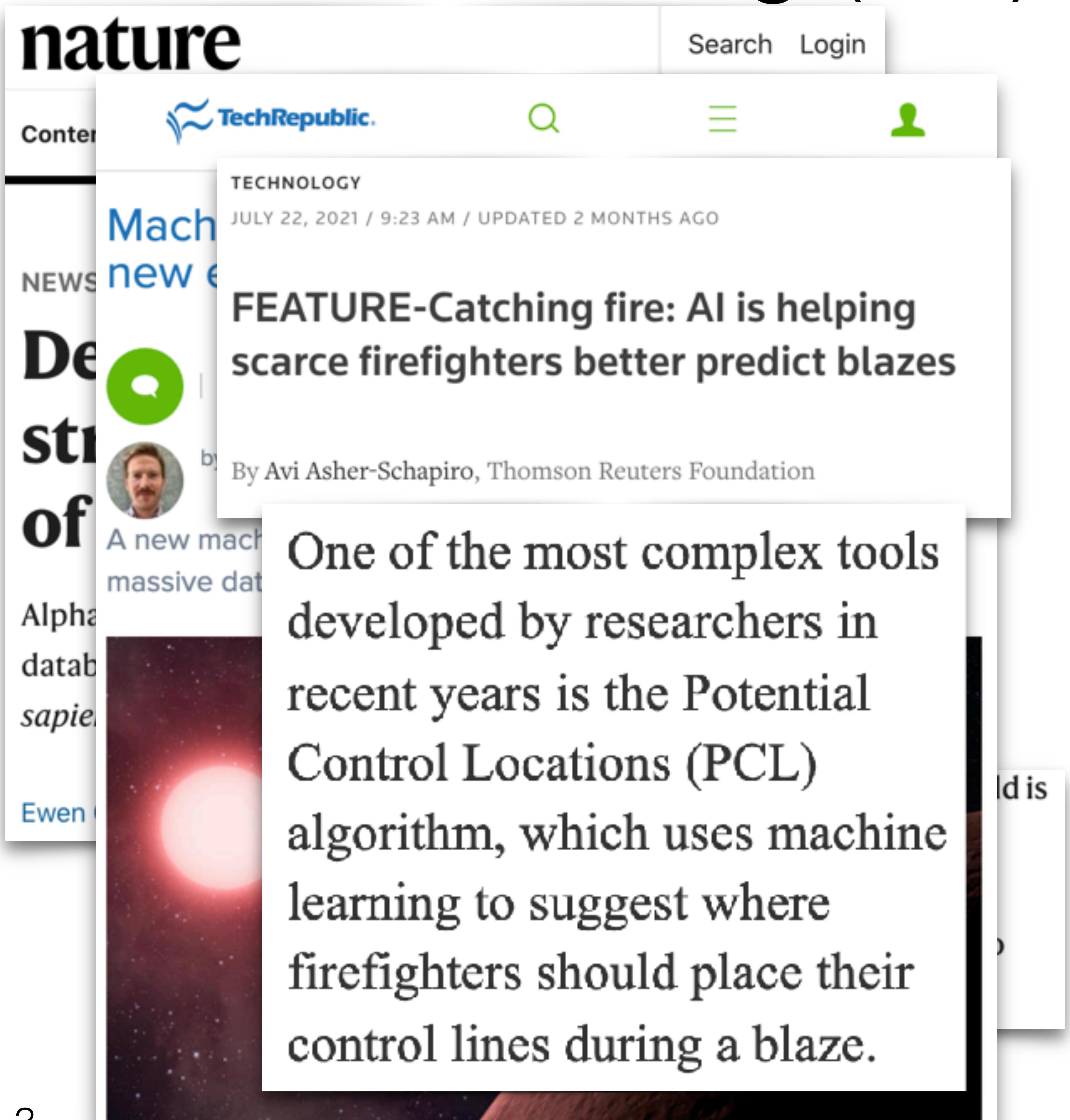
by **R. Dallon Adams** in **Innovation** on August 26, 2020, 9:07 AM PST

A new machine learning technique can be used to sift through massive datasets to discern exoplanets from false positives.

ld is

[https://www.techrepublic.com/article/machine-learning-algorithm-confirms-50-new-exoplanets-in-historic-first/]

Machine learning (ML): why & what



TECHNOLOGY

JULY 22, 2021 / 9:23 AM / UPDATED 2 MONTHS AGO

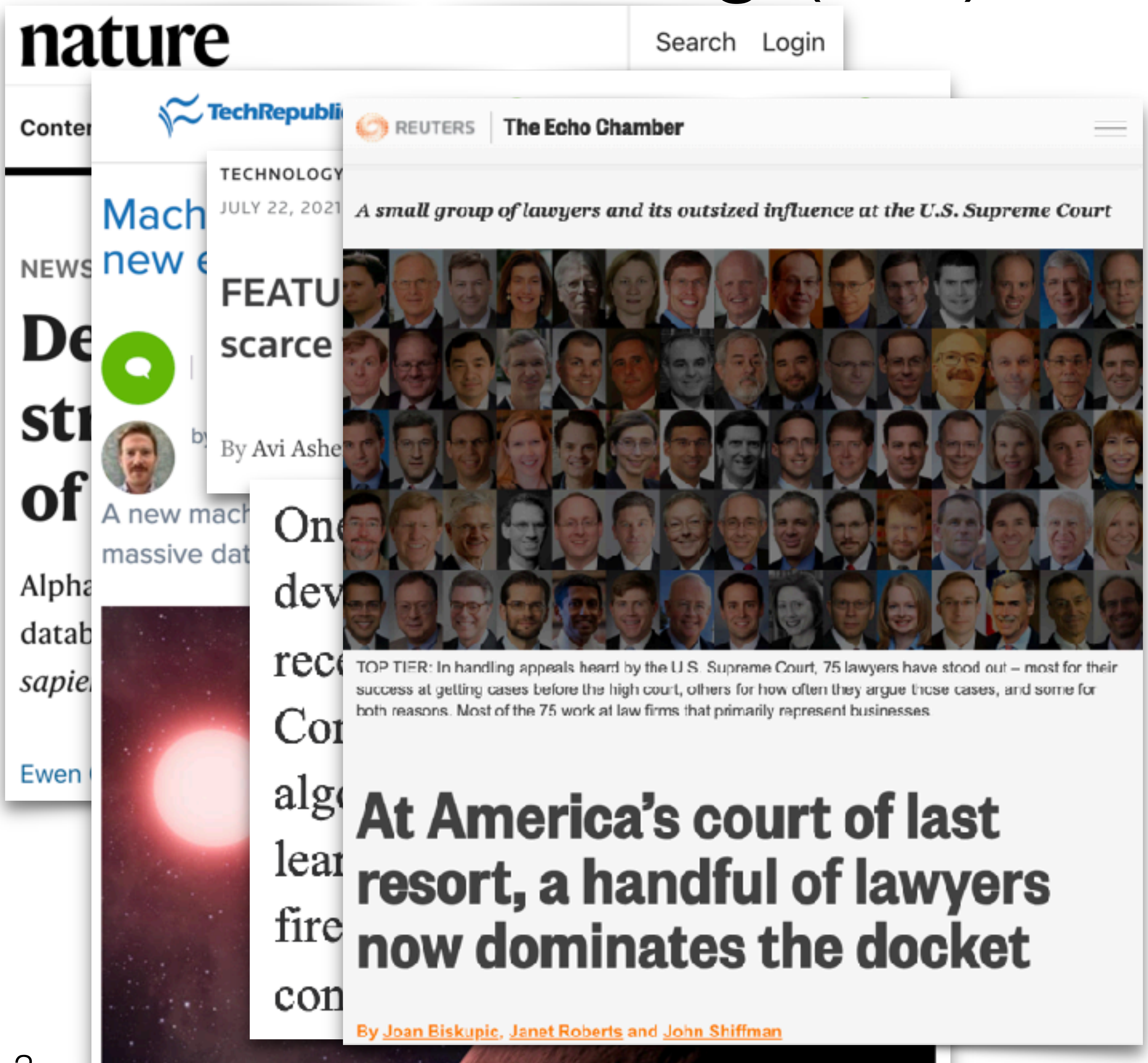
FEATURE-Catching fire: AI is helping scarce firefighters better predict blazes

By Avi Asher-Schapiro, Thomson Reuters Foundation

One of the most complex tools developed by researchers in recent years is the Potential Control Locations (PCL) algorithm, which uses machine learning to suggest where firefighters should place their control lines during a blaze.

[https://www.reuters.com/article/tech-environment-wildfire-climate-change-idUSL8N2OW008]

Machine learning (ML): why & what

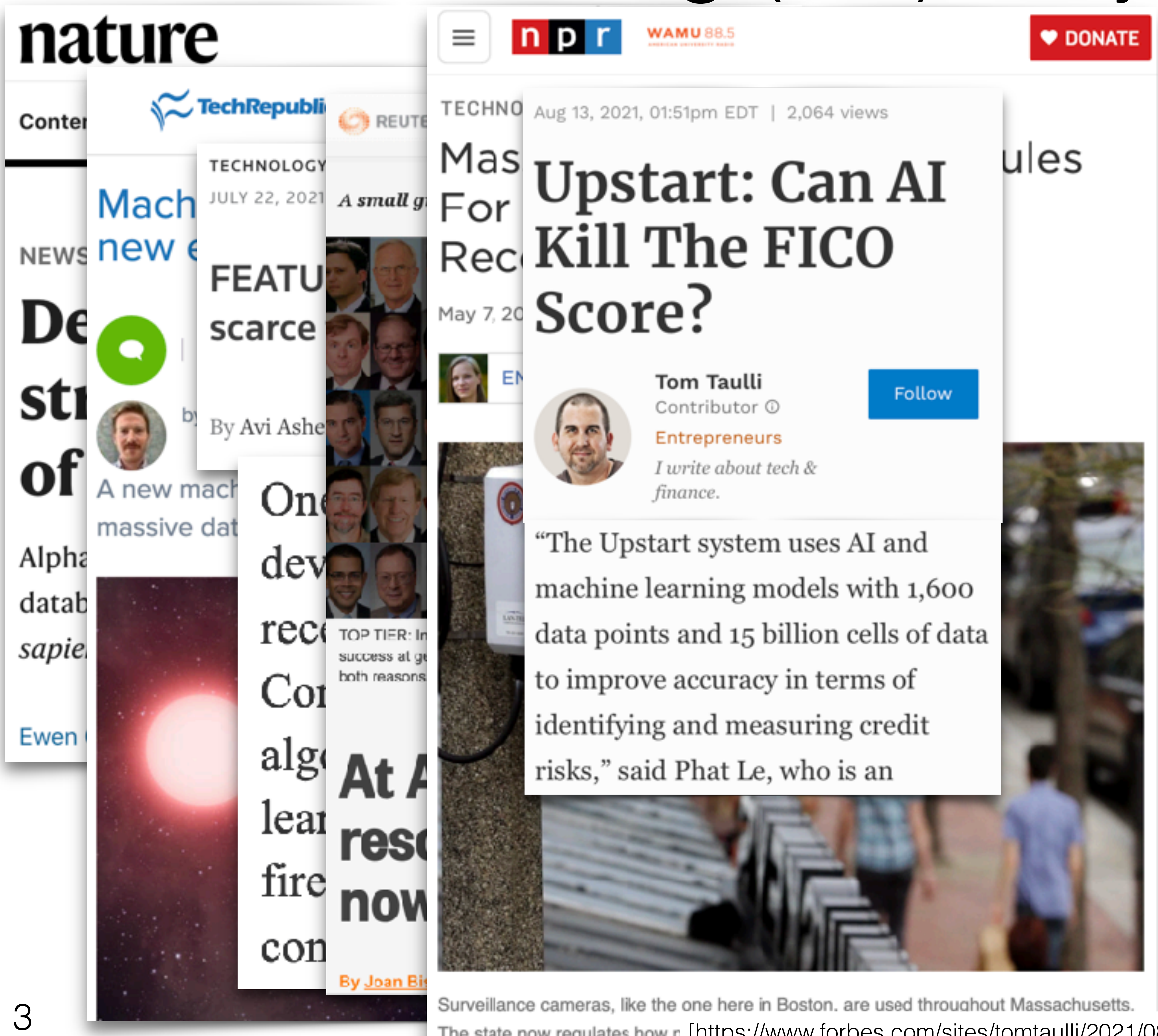


[https://www.reuters.com/investigates/special-report/scotus/]

Machine learning (ML): why & what

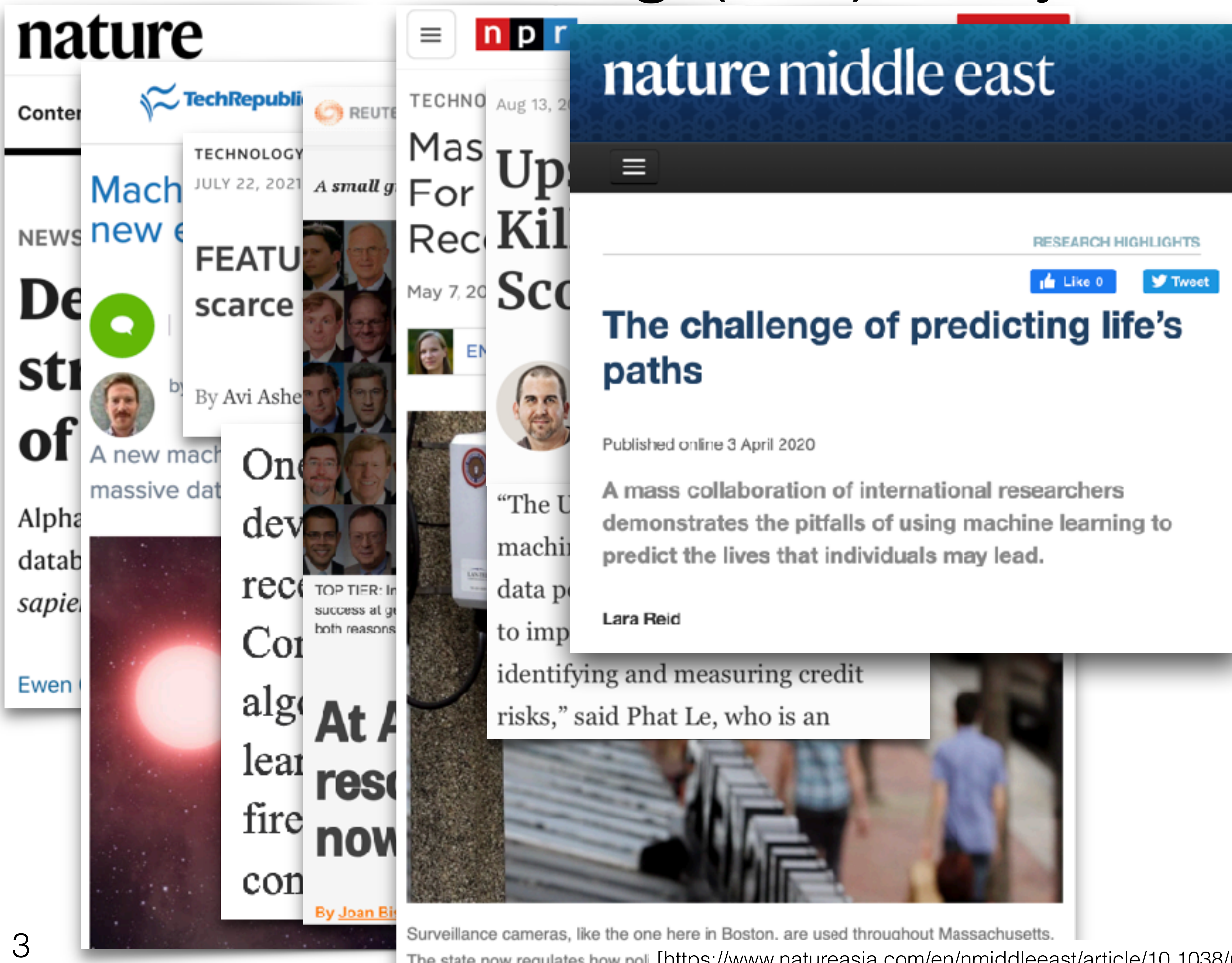
[https://www.npr.org/2021/05/07/982709480/massachusetts-pioneers-rules-for-police-use-of-facial-recognition-tech]

Machine learning (ML): why & what



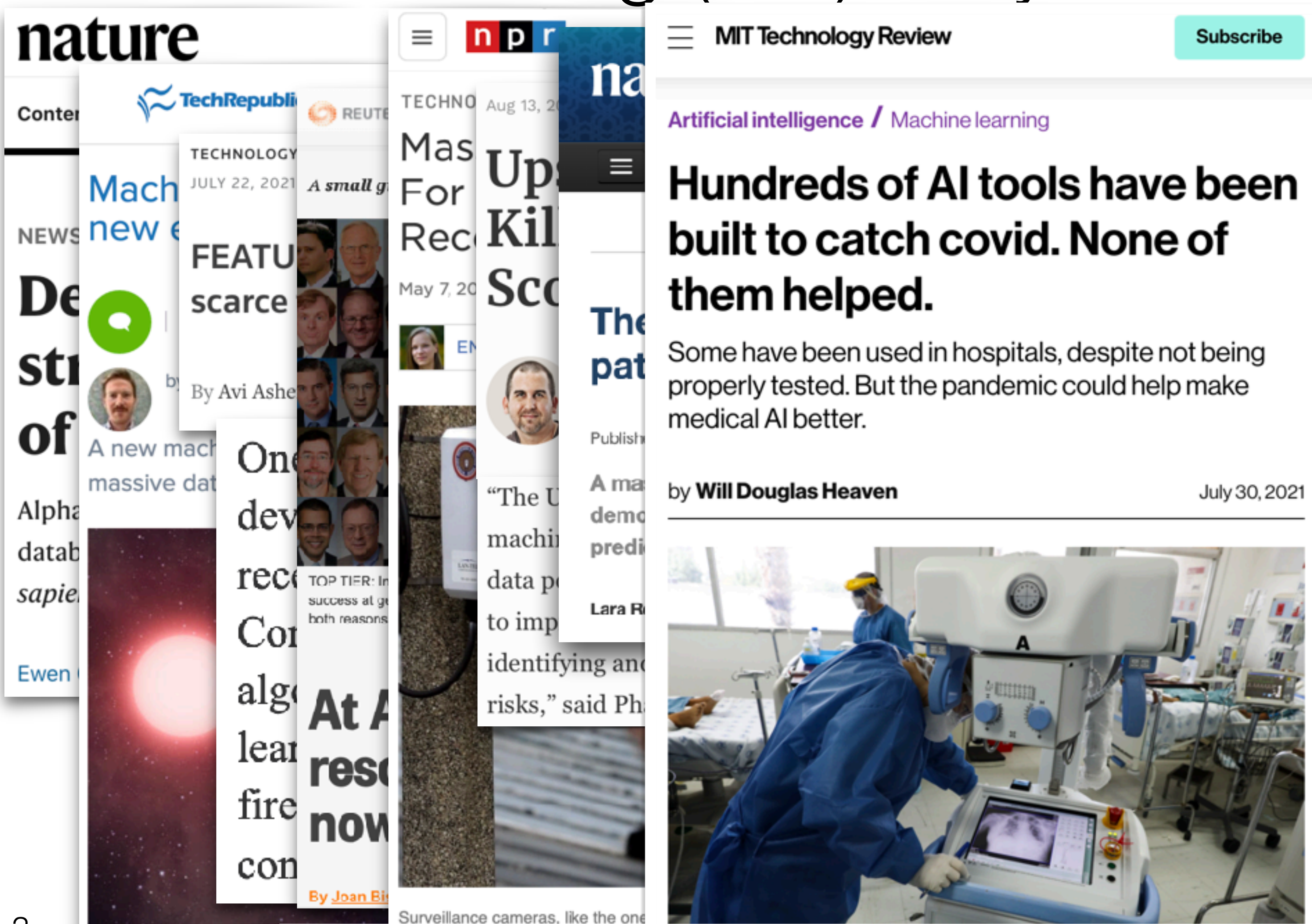
[https://www.forbes.com/sites/tomtaulli/2021/08/13/upstart-can-ai-kill-the-fico-score/]

Machine learning (ML): why & what



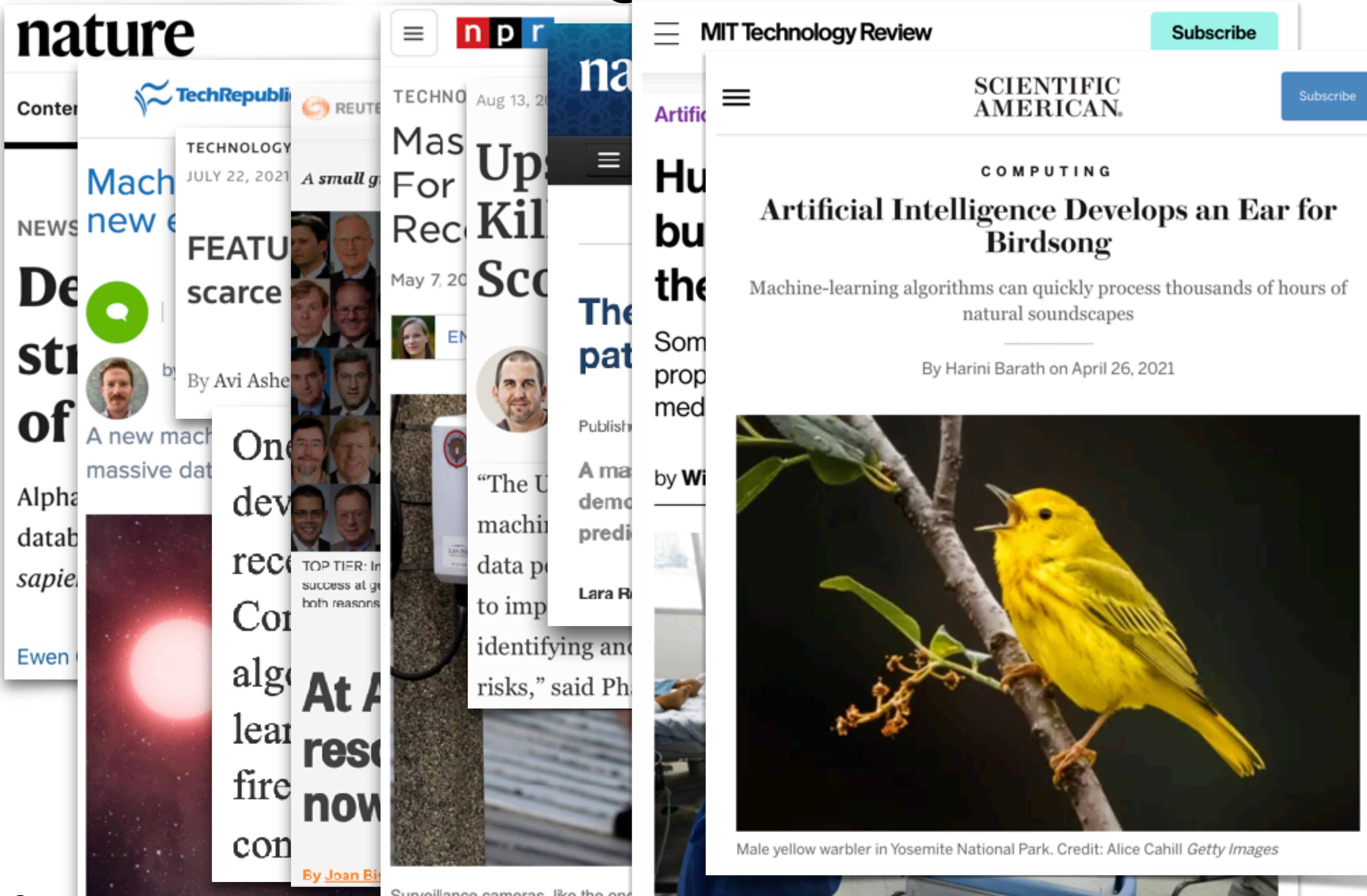
[https://www.natureasia.com/en/nmiddleeast/article/10.1038/nmiddleeast.2020.46]

Machine learning (ML): why & what



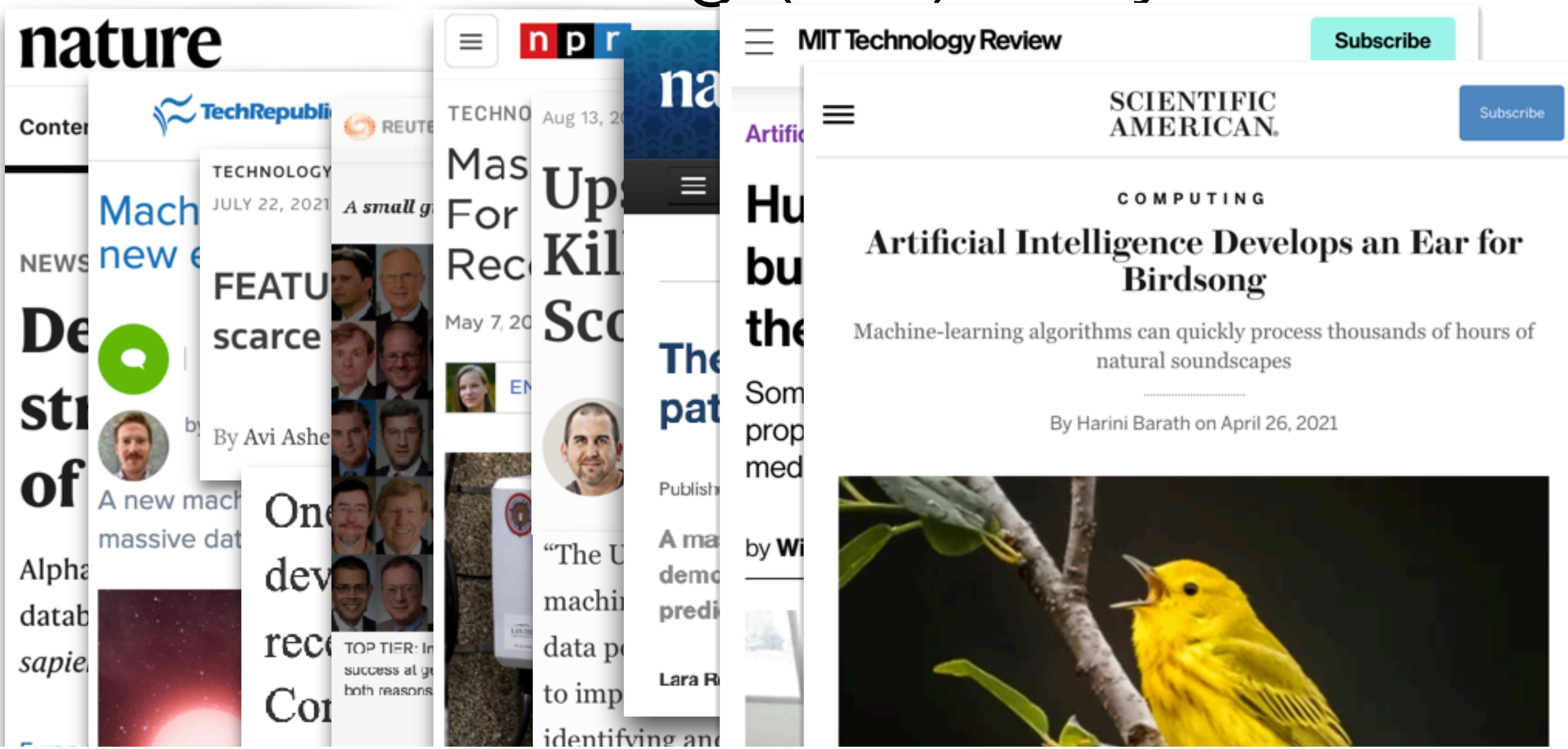
[https://www.technologyreview.com/2021/07/30/1030329/machine-learning-ai-failed-covid-hospital-diagnosis-pandemic/]

Machine learning (ML): why & what



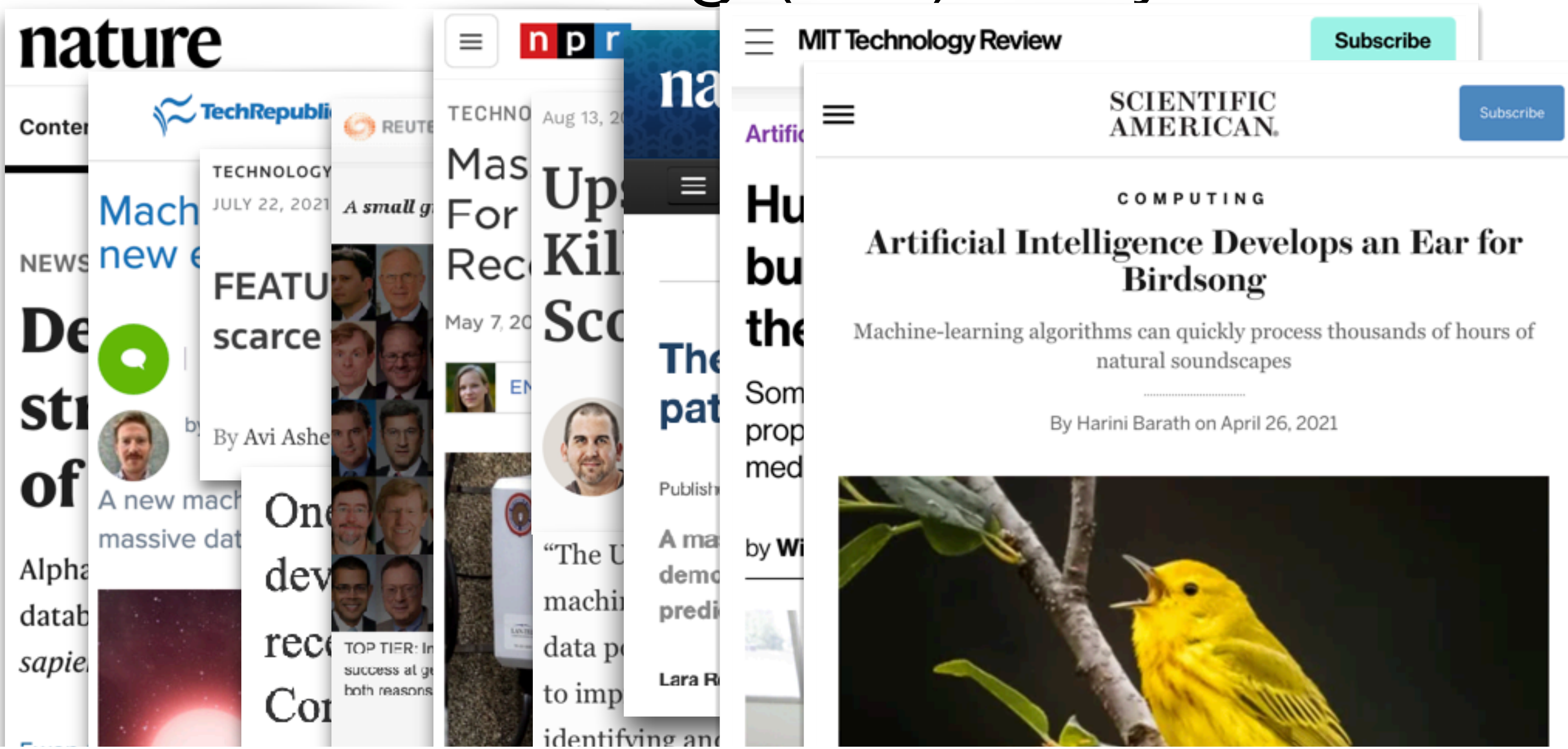
[https://www.scientificamerican.com/article/artificial-intelligence-develops-an-ear-for-birdsong1/#]

Machine learning (ML): why & what



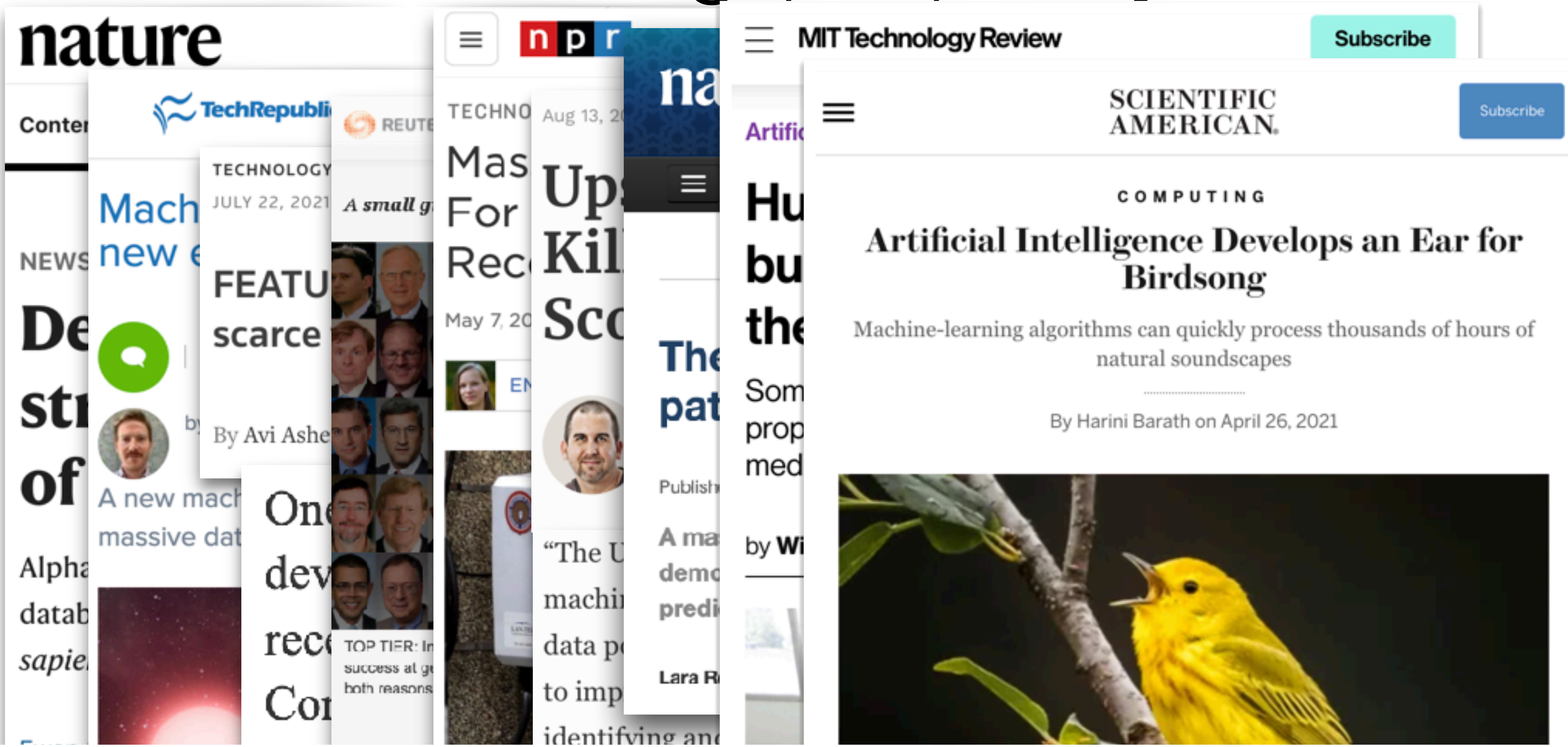
- **What is ML?**

Machine learning (ML): why & what



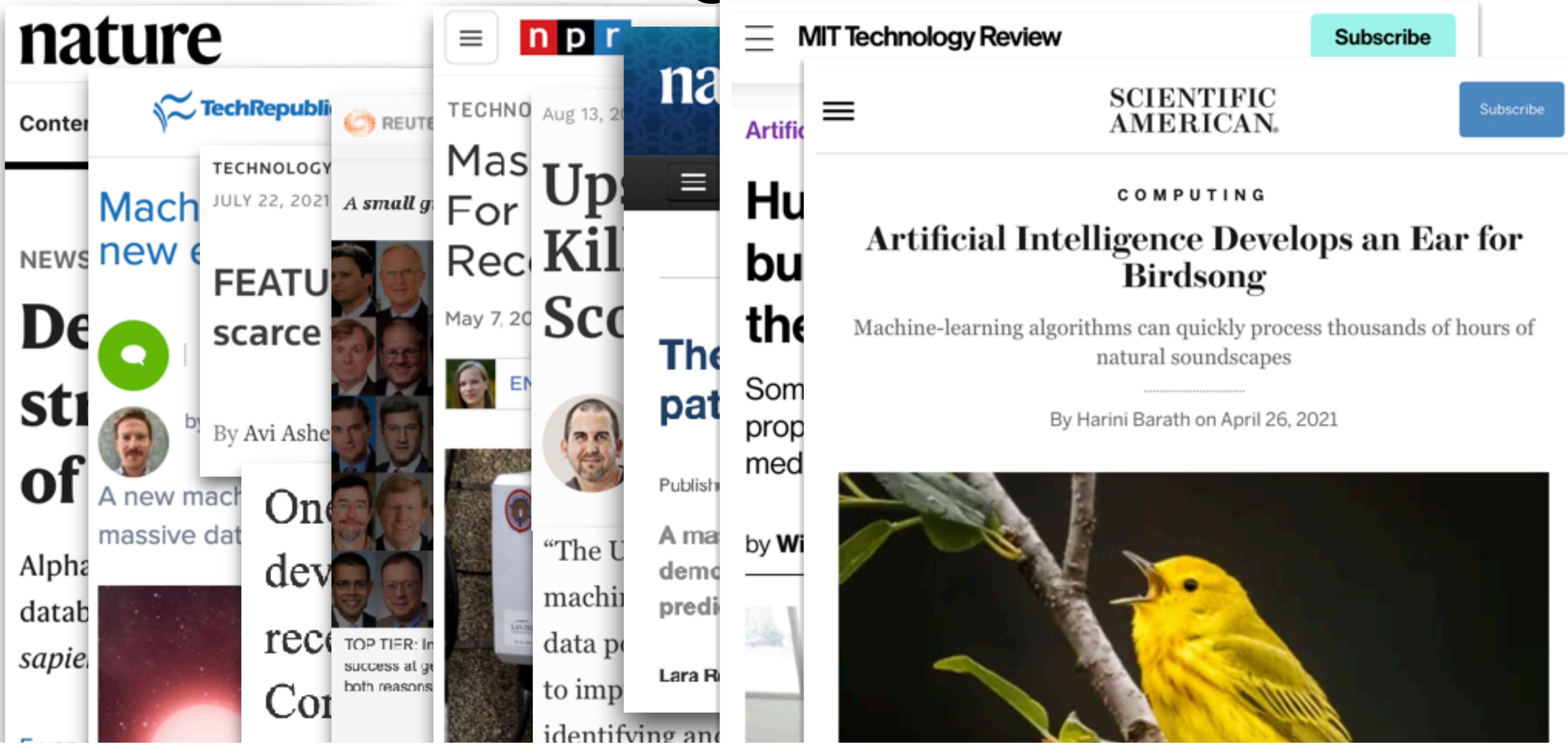
- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)

Machine learning (ML): why & what



- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)
- **Why study ML?** To apply; to understand; to evaluate

Machine learning (ML): why & what



- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)
- **Why study ML?** To apply; to understand; to evaluate
- **Notes:** ML is a tool with pros & cons. ML is built on math

Getting started: regression

Getting started: regression

Example: predict pollution level

Getting started: regression

Example: predict pollution level

What do we have?

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

Satellite reading

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$

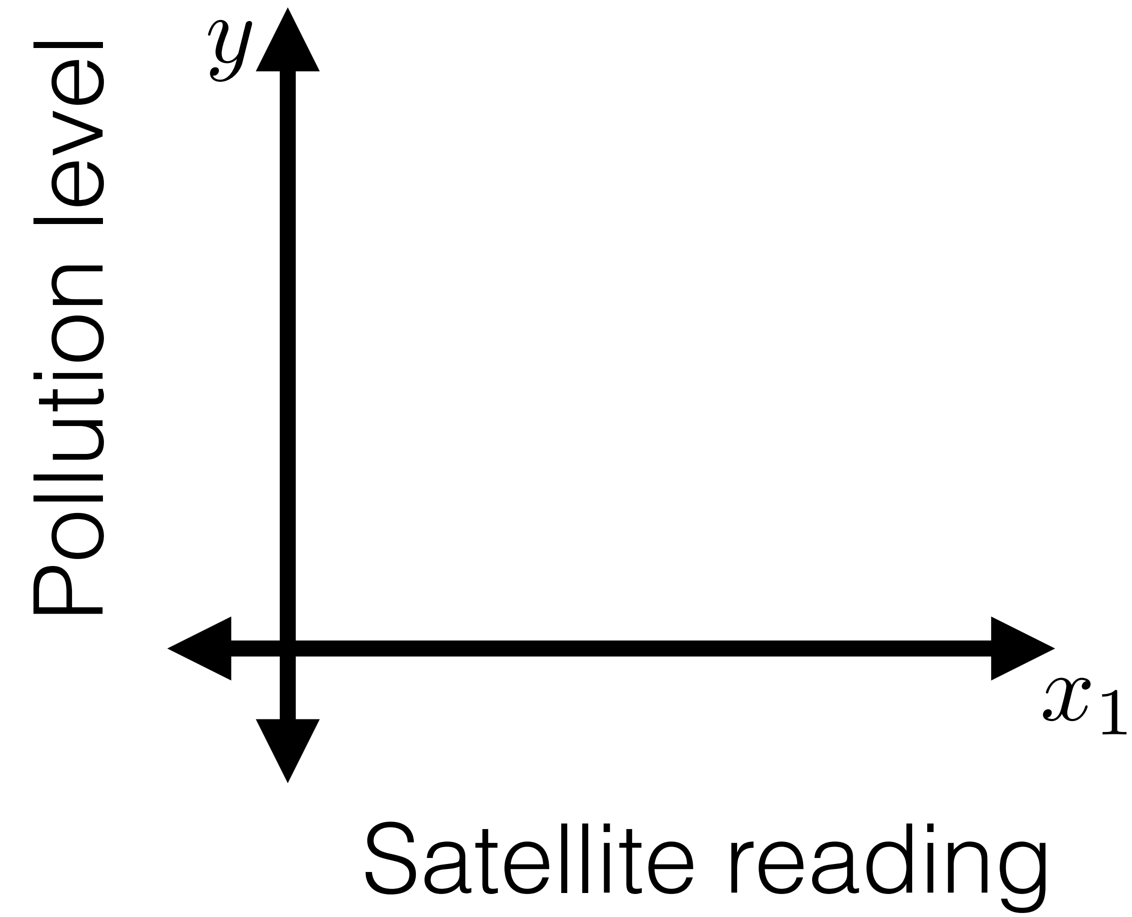
Satellite reading

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- n training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$

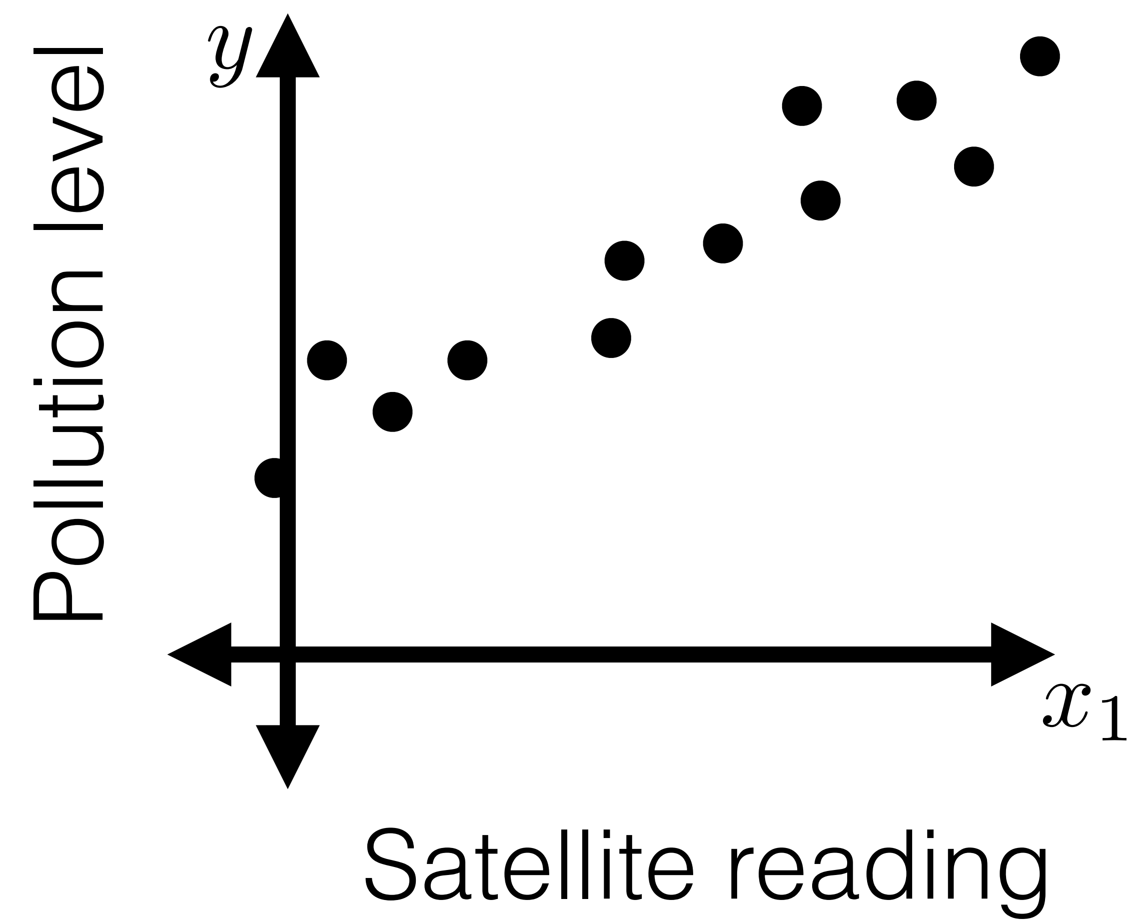


Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$

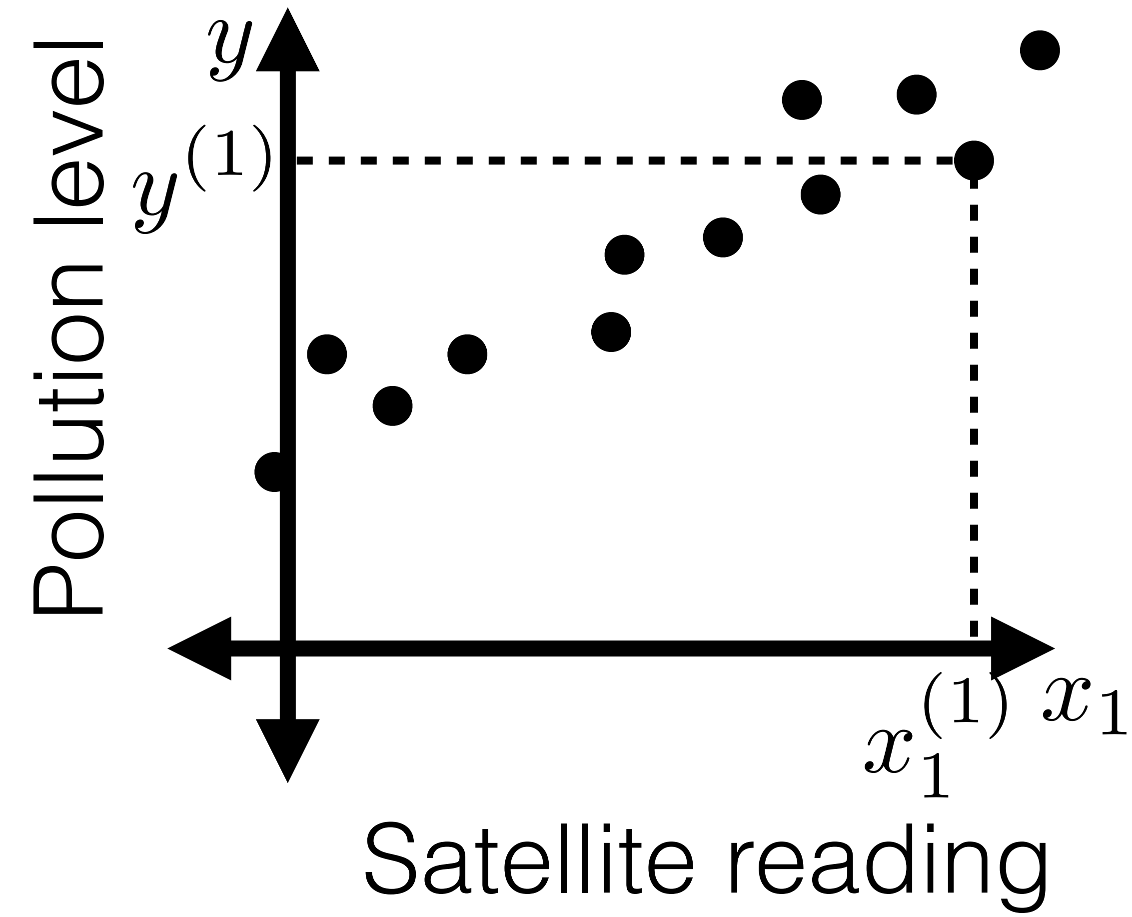


Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \dots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$

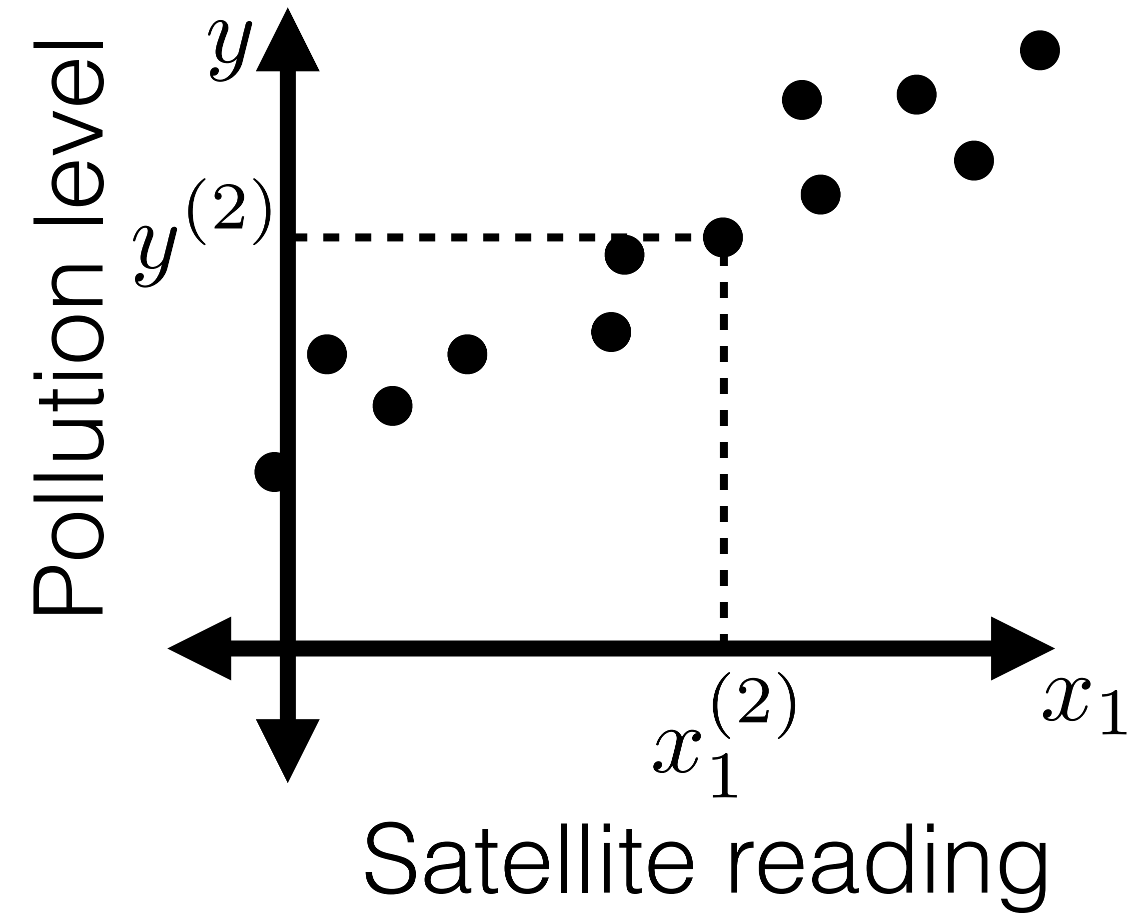


Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$

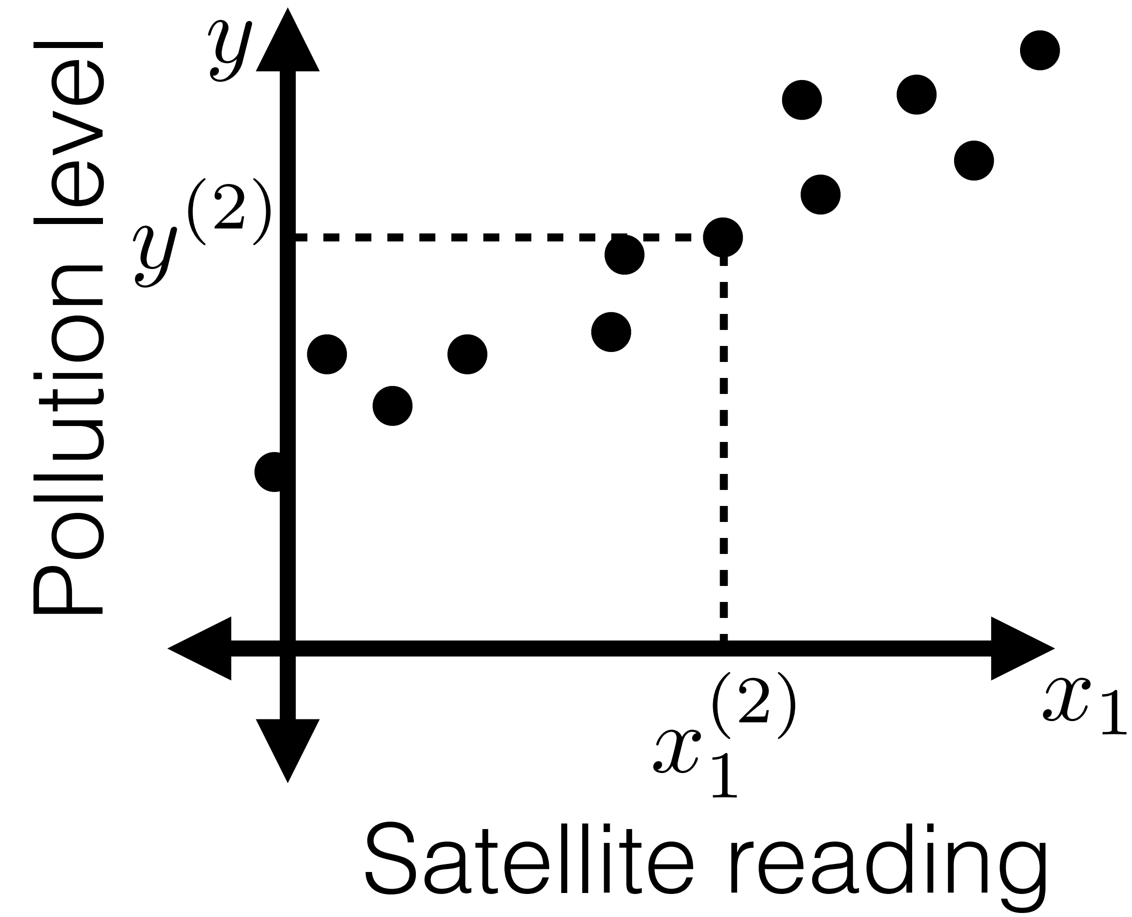


Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \dots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$

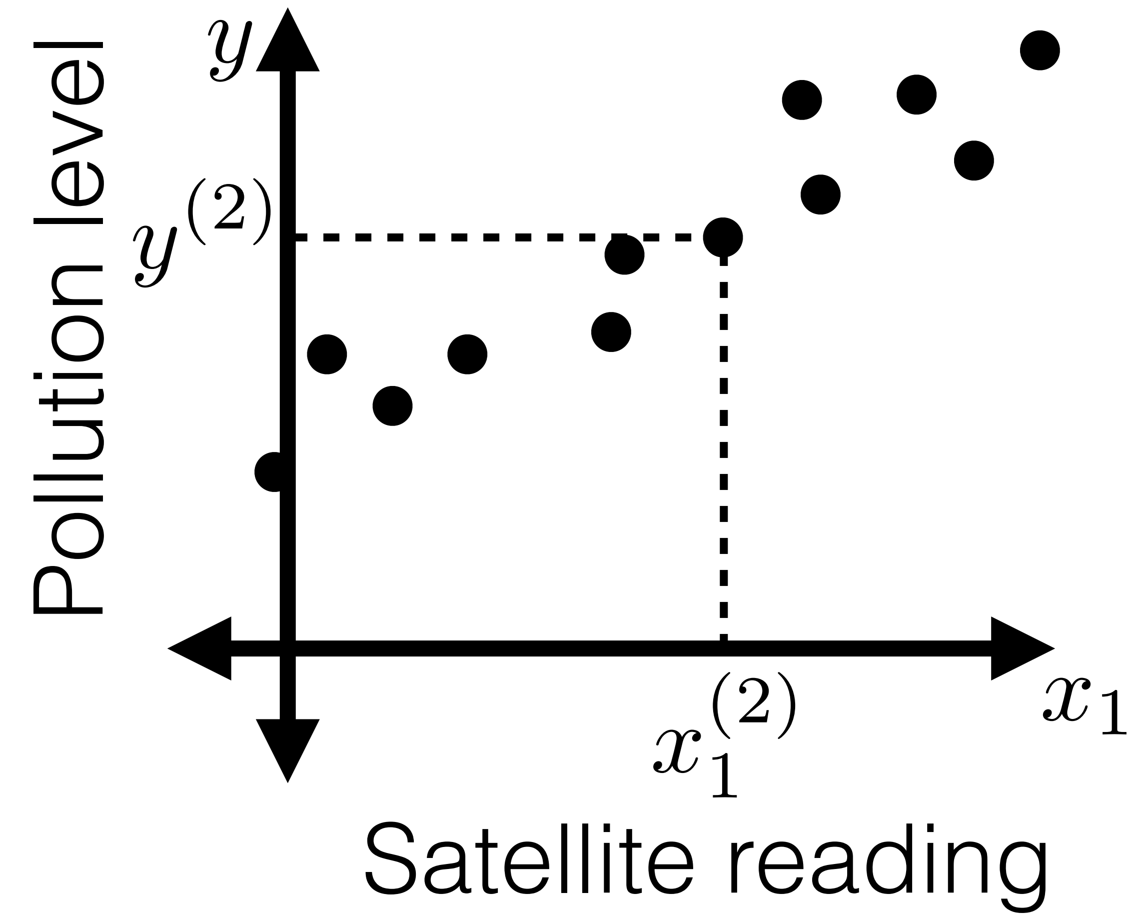


Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



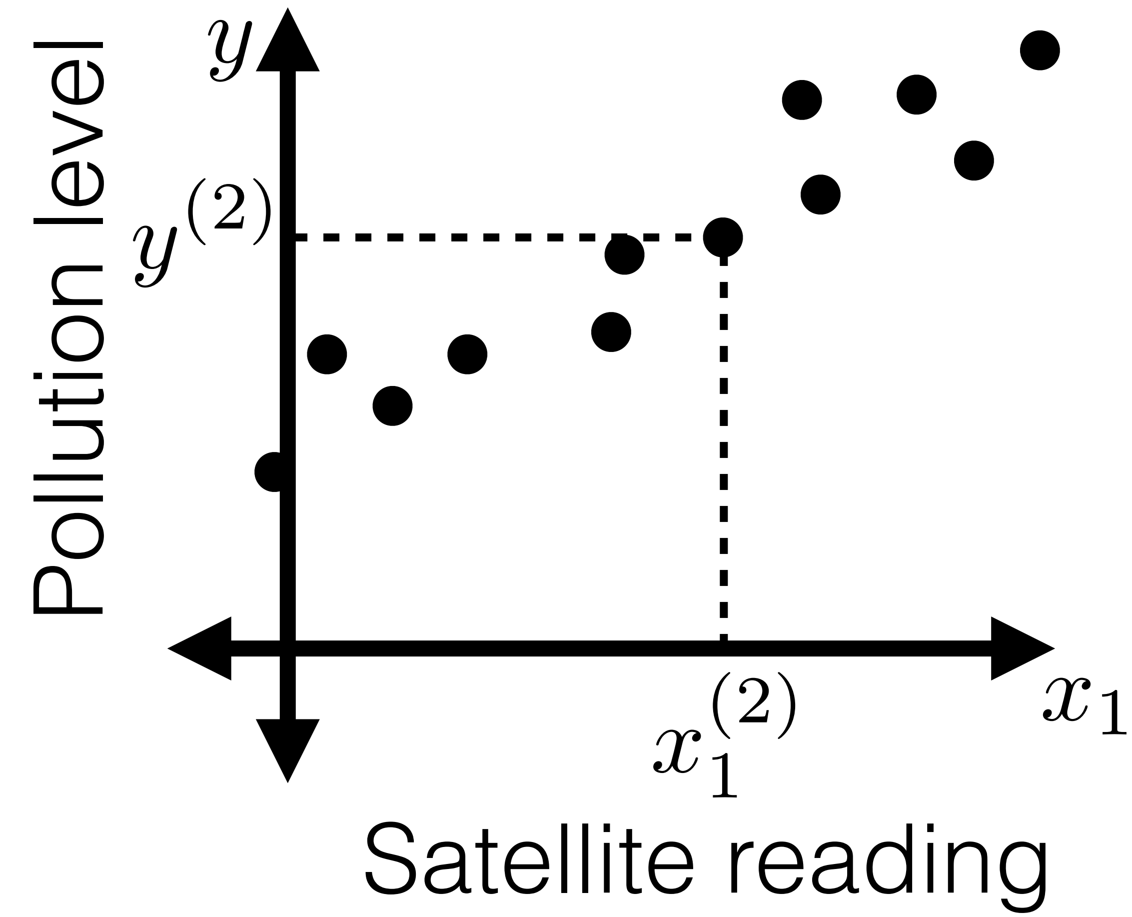
What do we want?

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



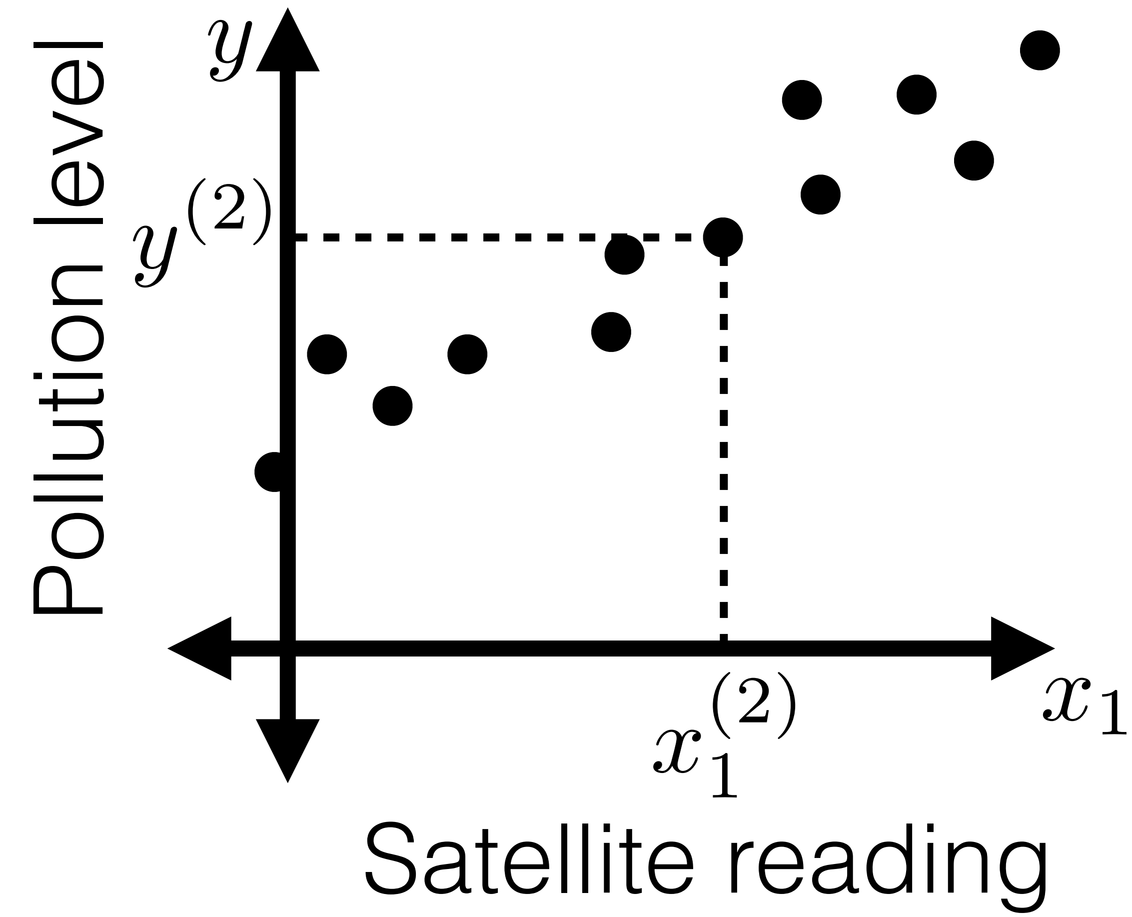
What do we want? A good way to label new points

Getting started: regression

Example: predict pollution level

What do we have? (Training) data

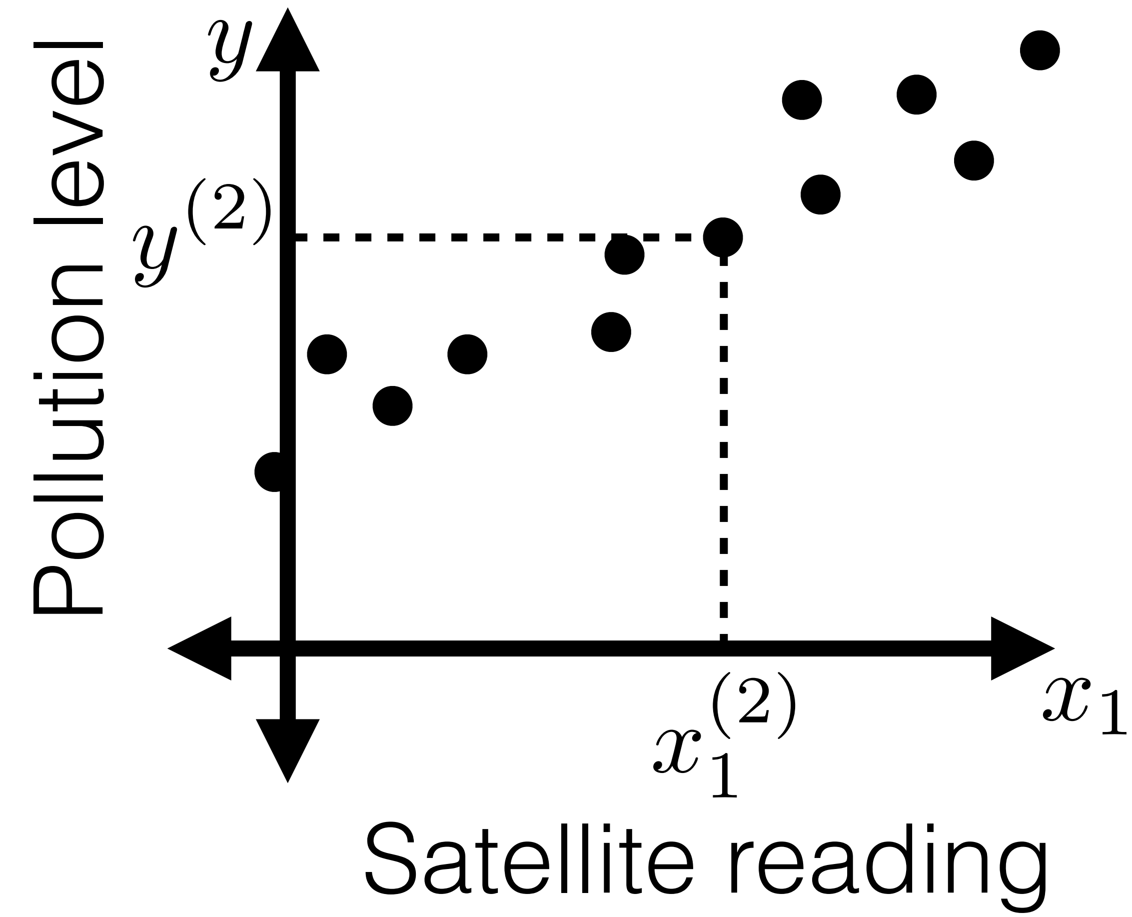
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



What do we want? A good way to label new points

- How to label?

Getting started: regression



Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

What do we want? A good way to label new points

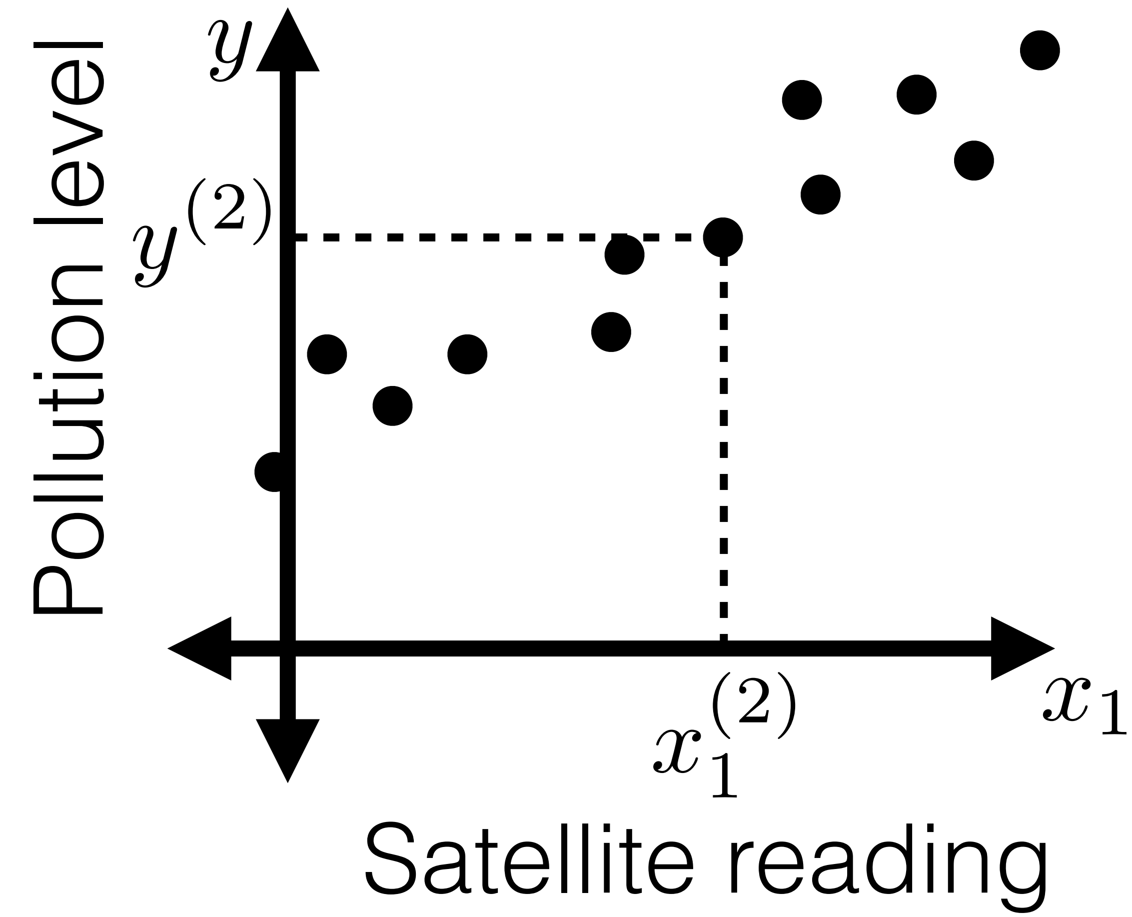
- How to label? Hypothesis $h : \mathbb{R}^d \to \mathbb{R}$

Getting started: regression

Example: predict pollution level

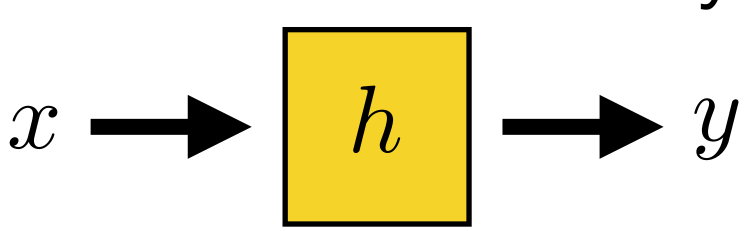
What do we have? (Training) data

- n training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \to \mathbb{R}$

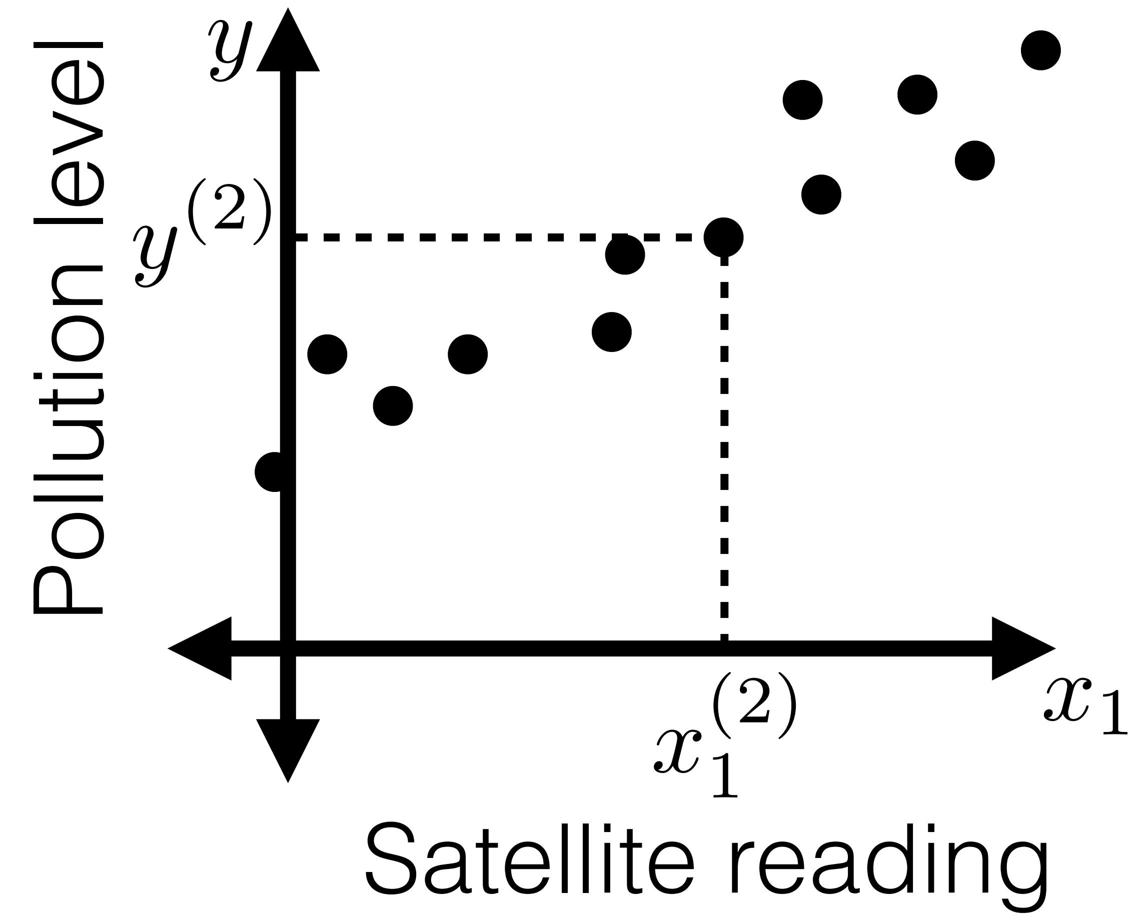


Getting started: regression

Example: predict pollution level

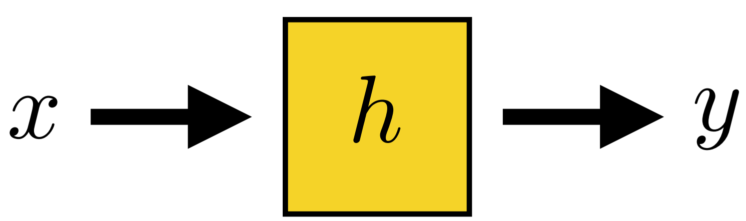
What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \rightarrow \mathbb{R}$



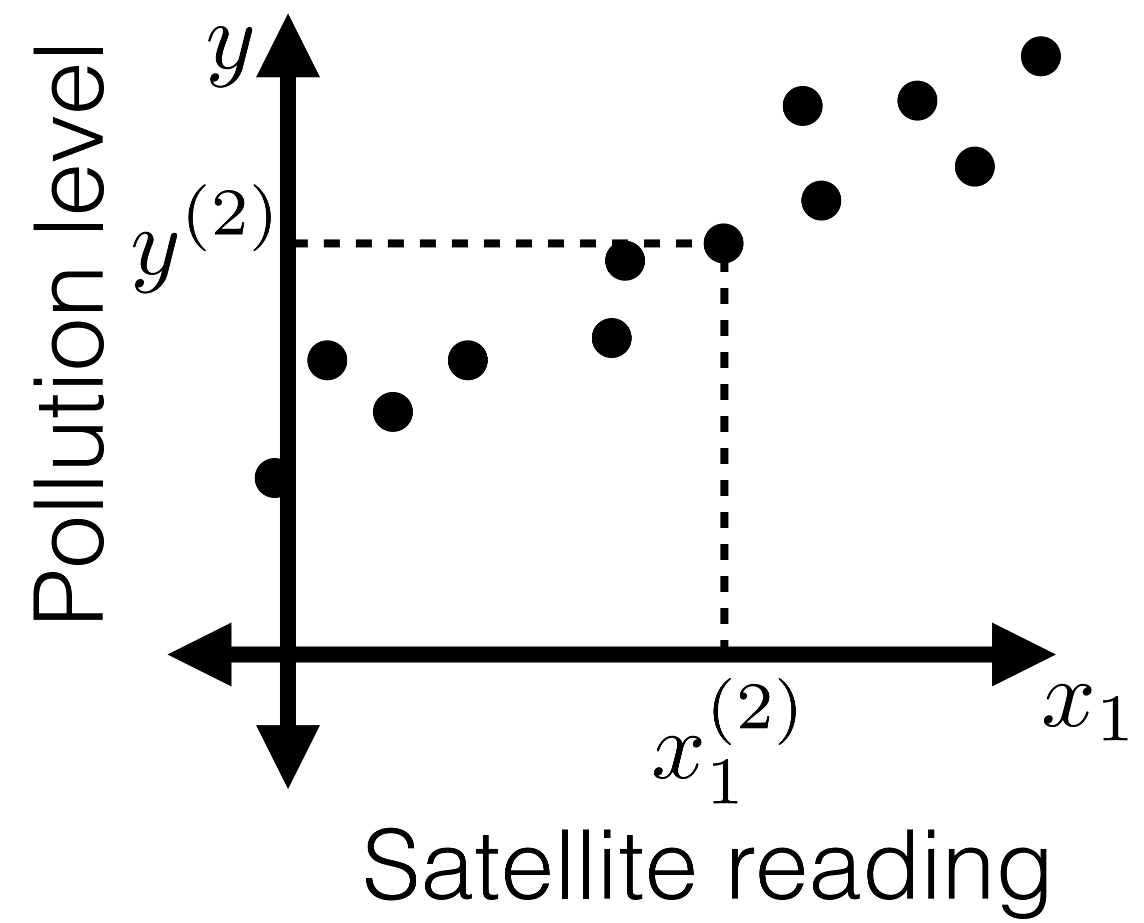
  - Example *h*: For any *x*, *h*(*x*) = 1,000,000

Getting started: regression

Example: predict pollution level

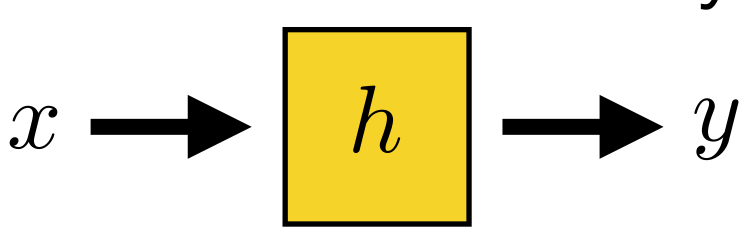
What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \rightarrow \mathbb{R}$



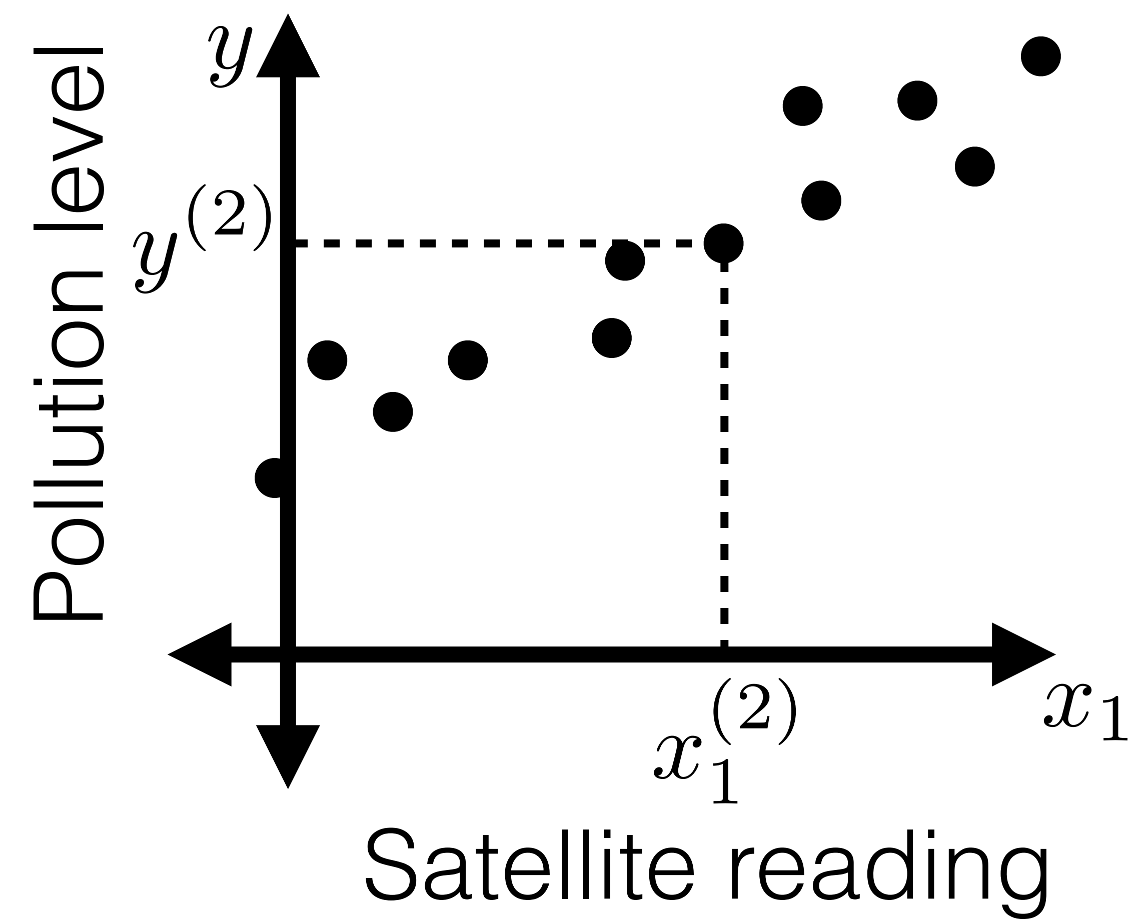
Is this a hypothesis?

- Example *h*: For any *x*, *h*(*x*) = 1,000,000

Getting started: regression

Example: predict pollution level

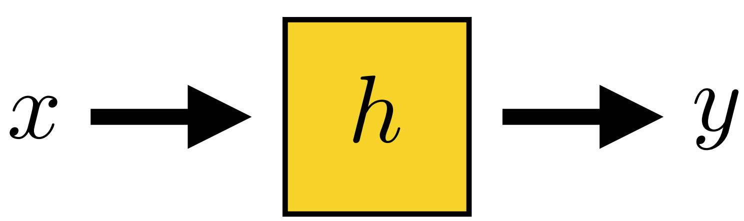
What do we have? (Training) data



- n training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

What do we want? A good way to label new points

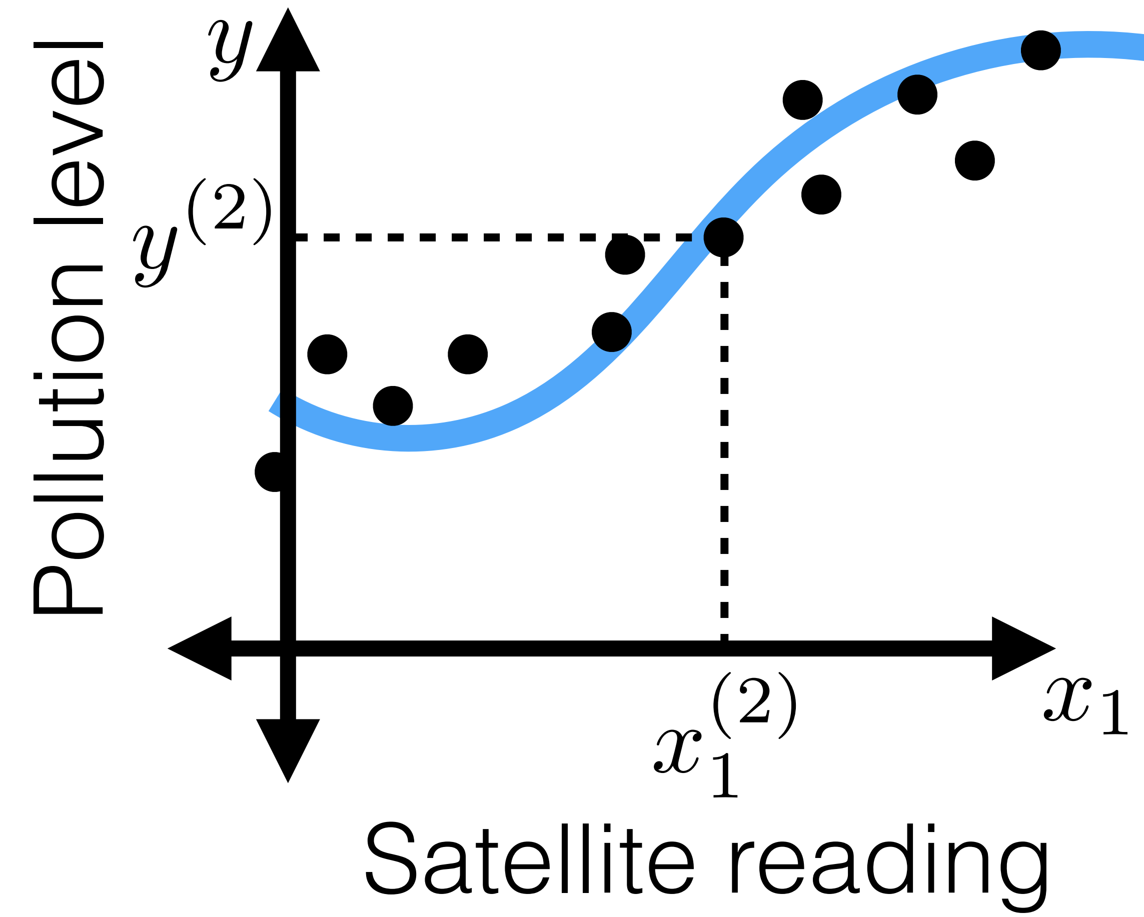
- How to label? Hypothesis $h : \mathbb{R}^d \to \mathbb{R}$



Is this a ***good*** hypothesis?

  - Example *h*: For any *x*, *h*(*x*) = 1,000,000

Getting started: regression



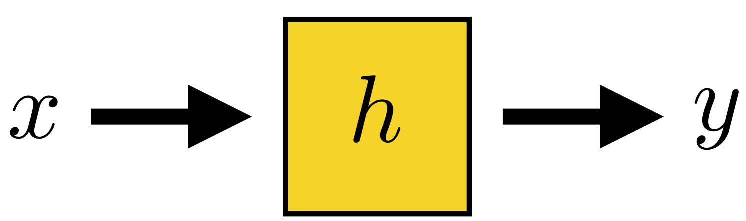
Example: predict pollution level

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \rightarrow \mathbb{R}$



Is this a ***good*** hypothesis?

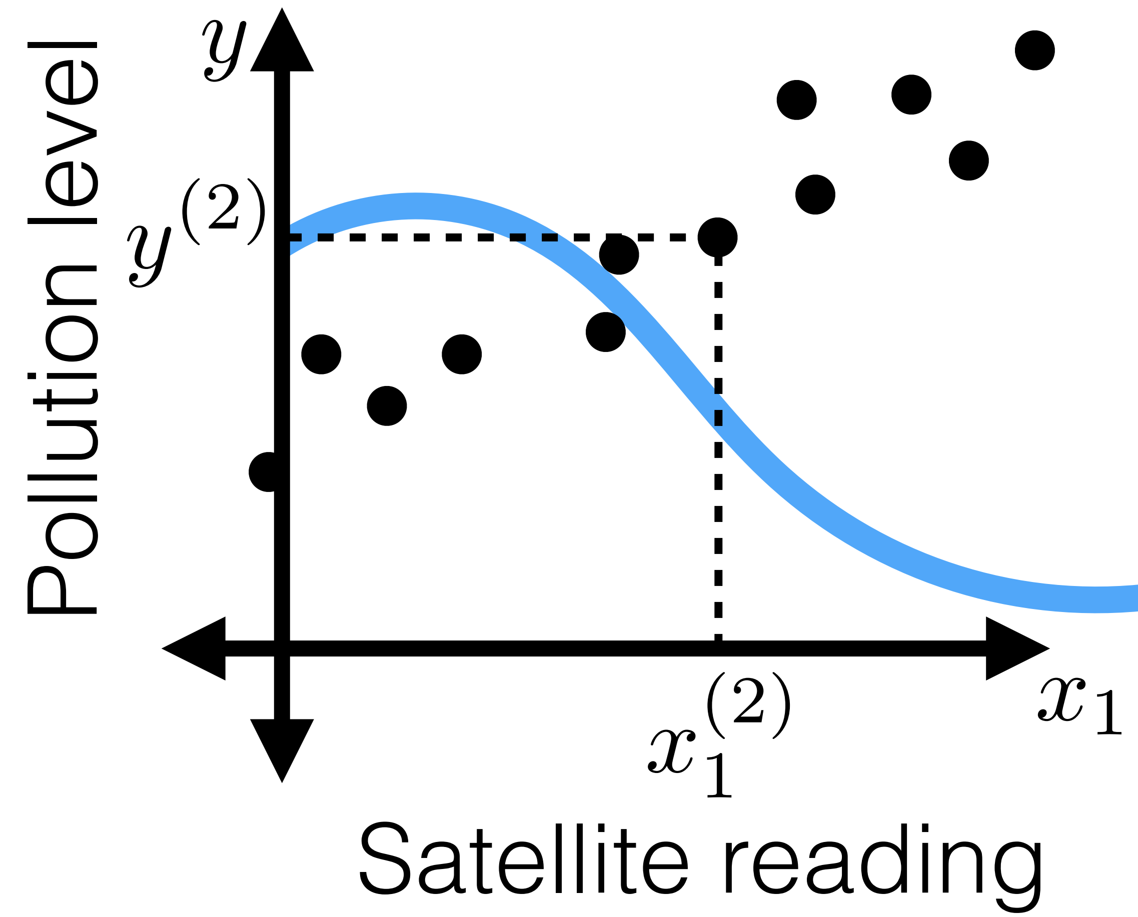
- Example *h*: For any *x*, *h*(*x*) = 1,000,000

Getting started: regression

Example: predict pollution level

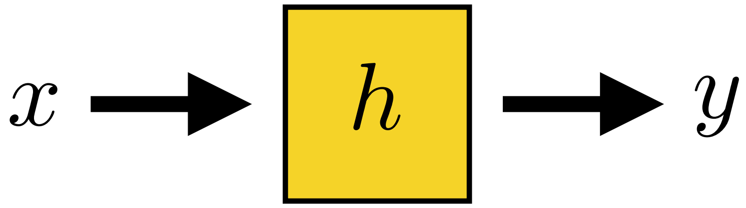
What do we have? (Training) data

- n training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



What do we want? A good way to label new points

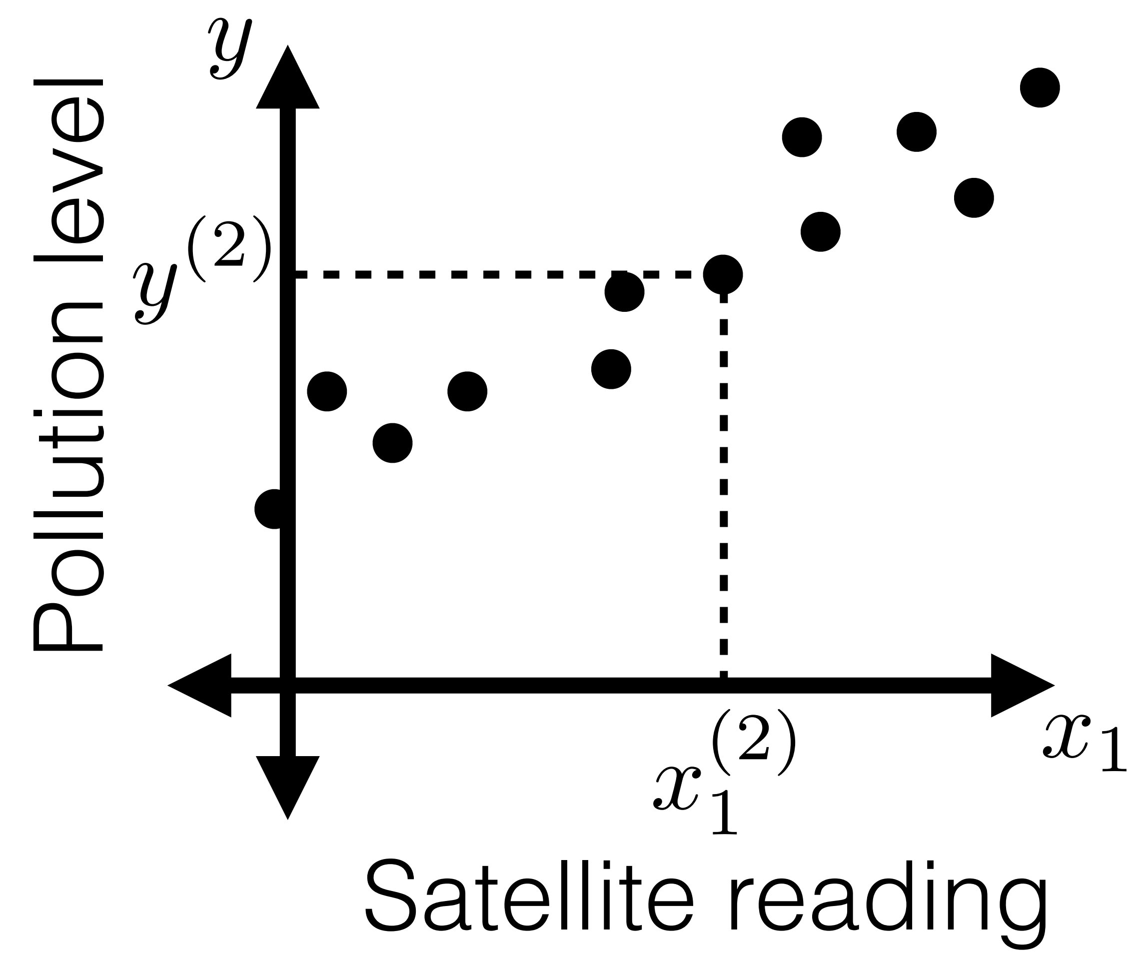
- How to label? Hypothesis $h : \mathbb{R}^d \to \mathbb{R}$



Is this a ***good*** hypothesis?

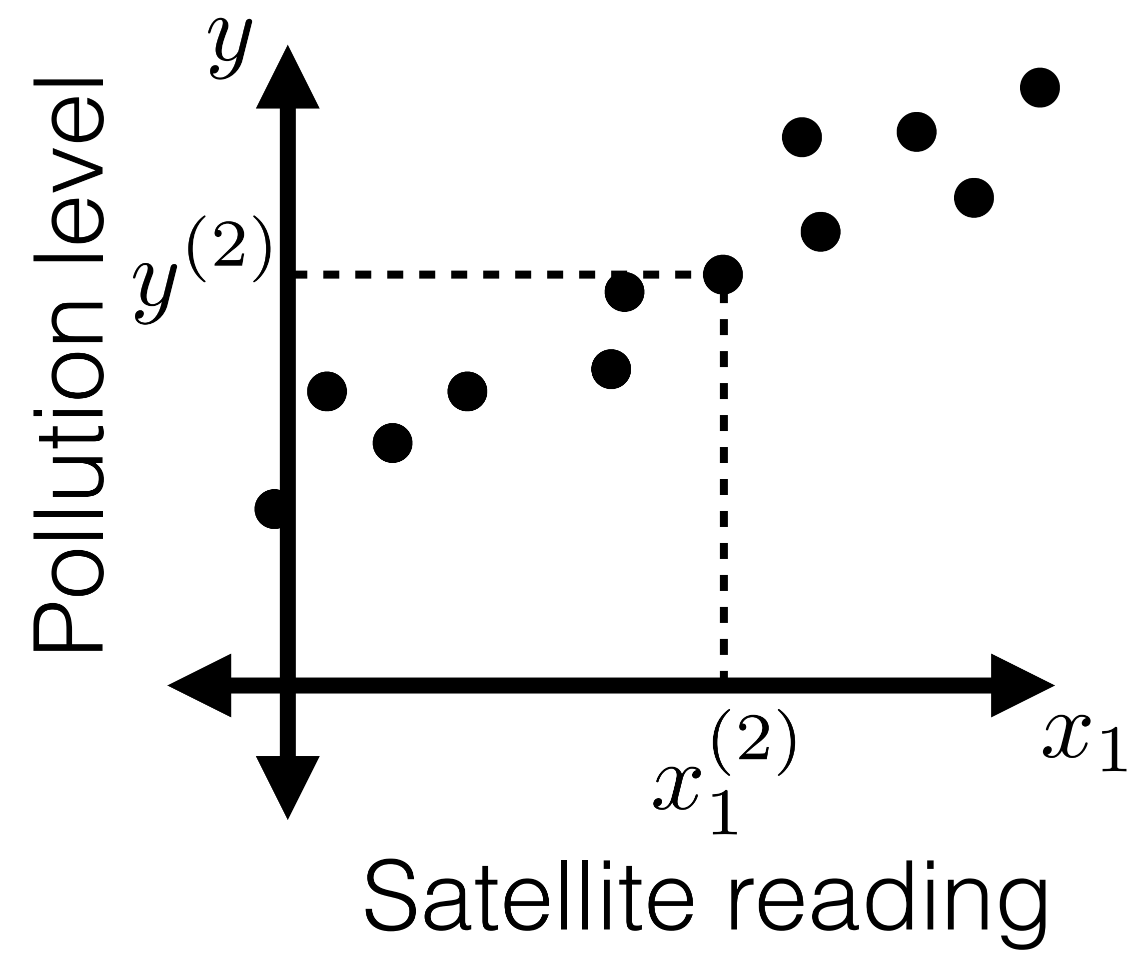
- Example h: For any x, $h(x)$ = 1,000,000

Linear regressors



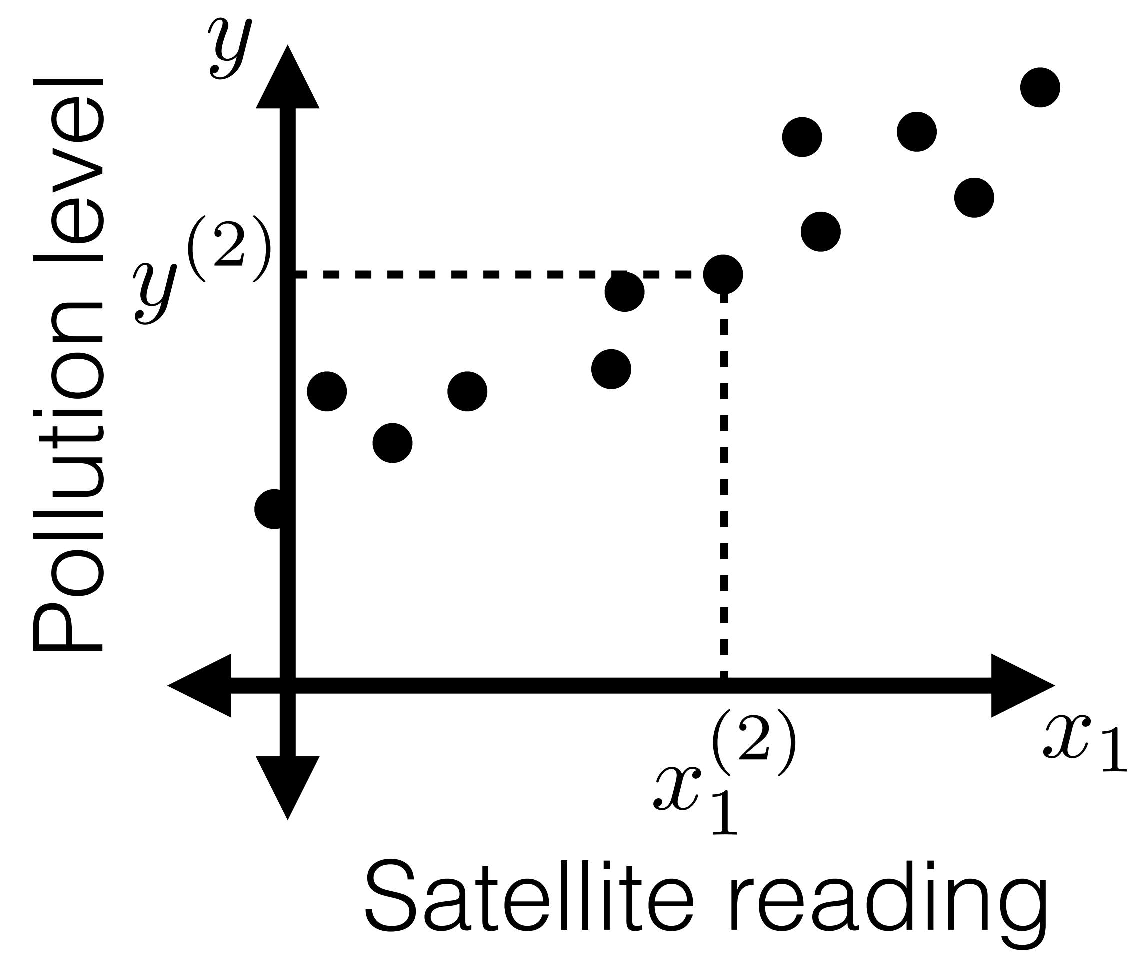
Linear regressors

- Hypothesis class $\mathcal{H}$: set of h



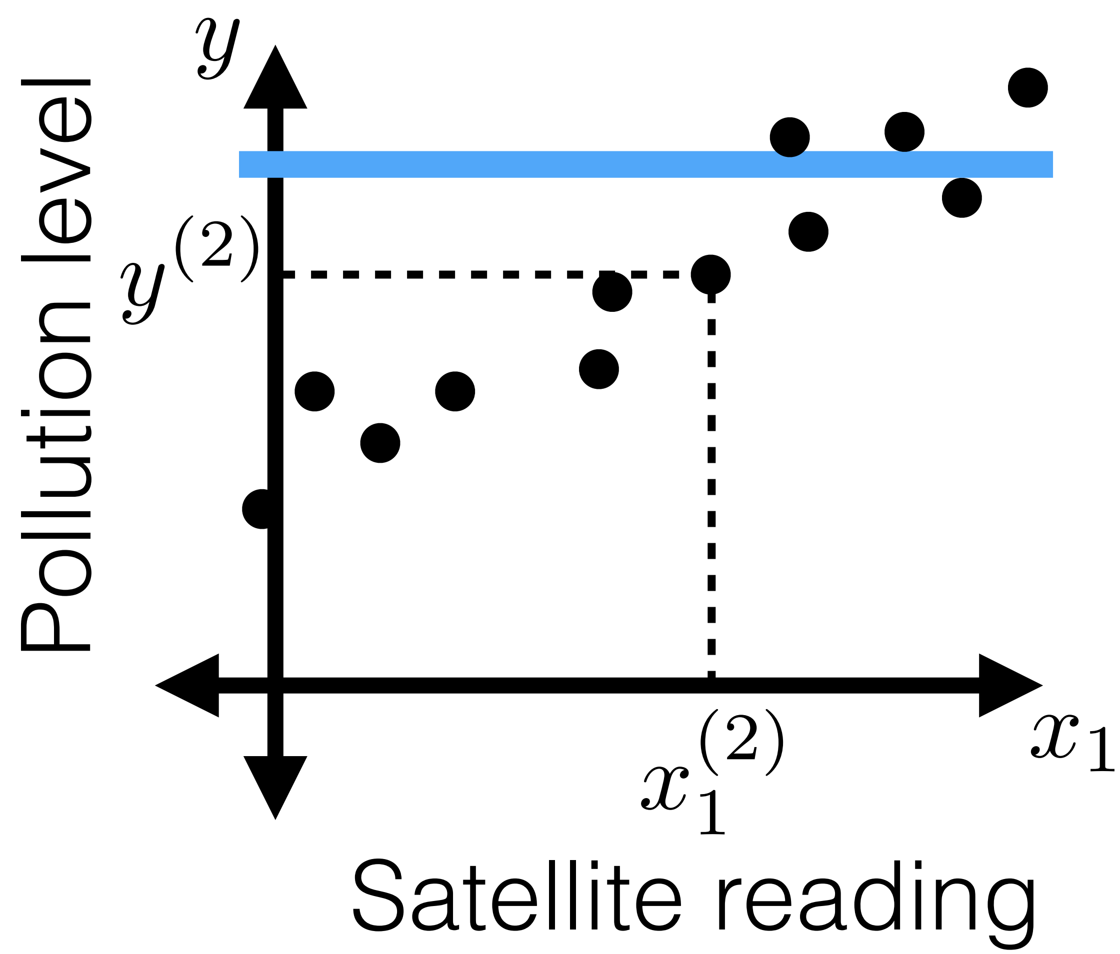
Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions



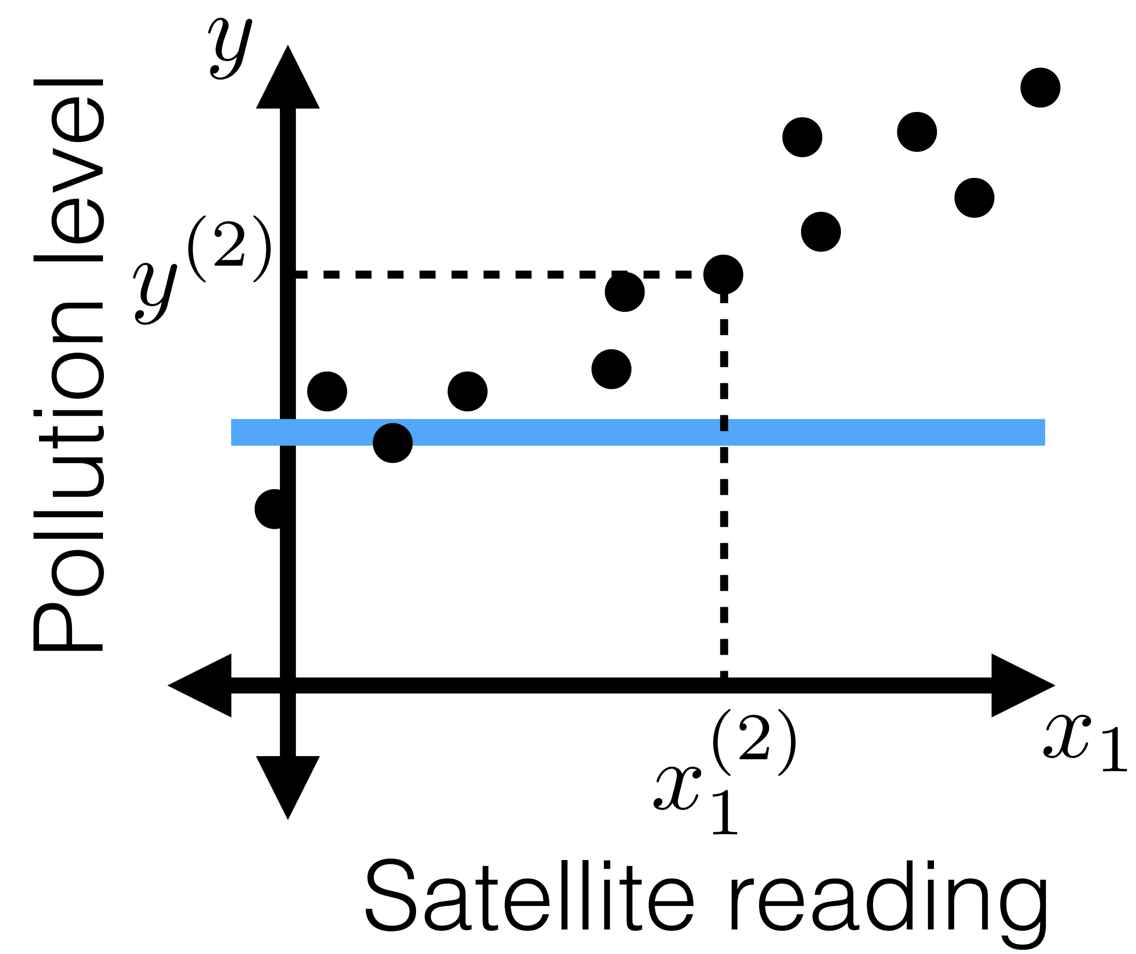
Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions



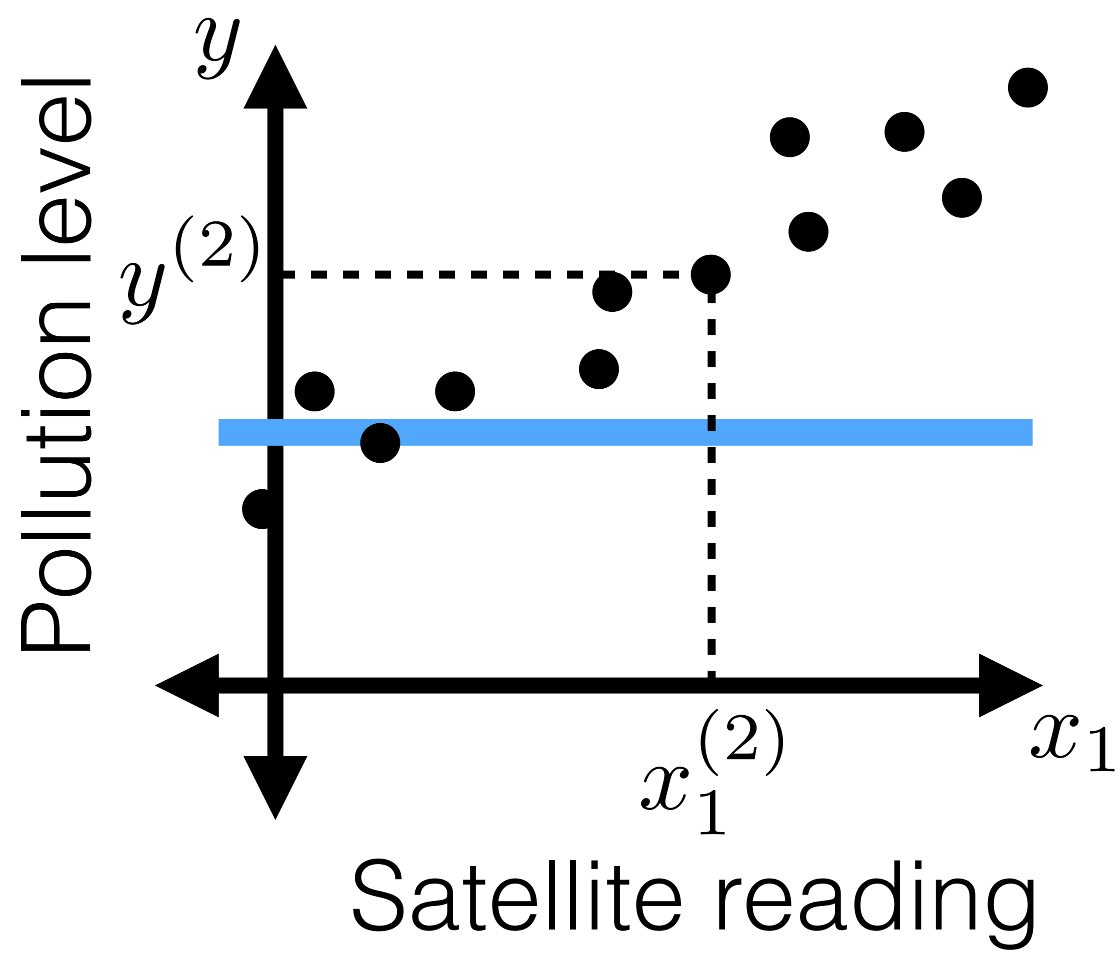
Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions



Linear regressors

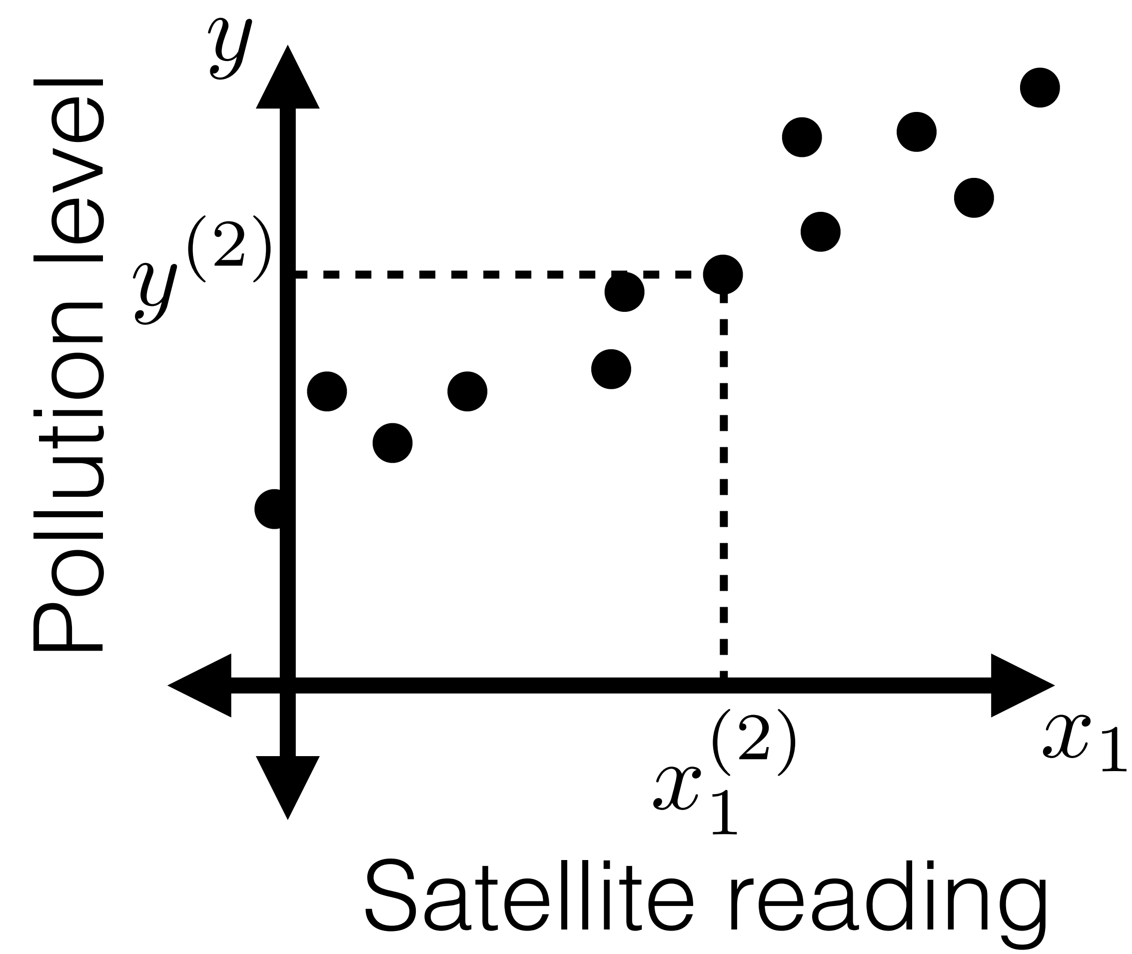
- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions



Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

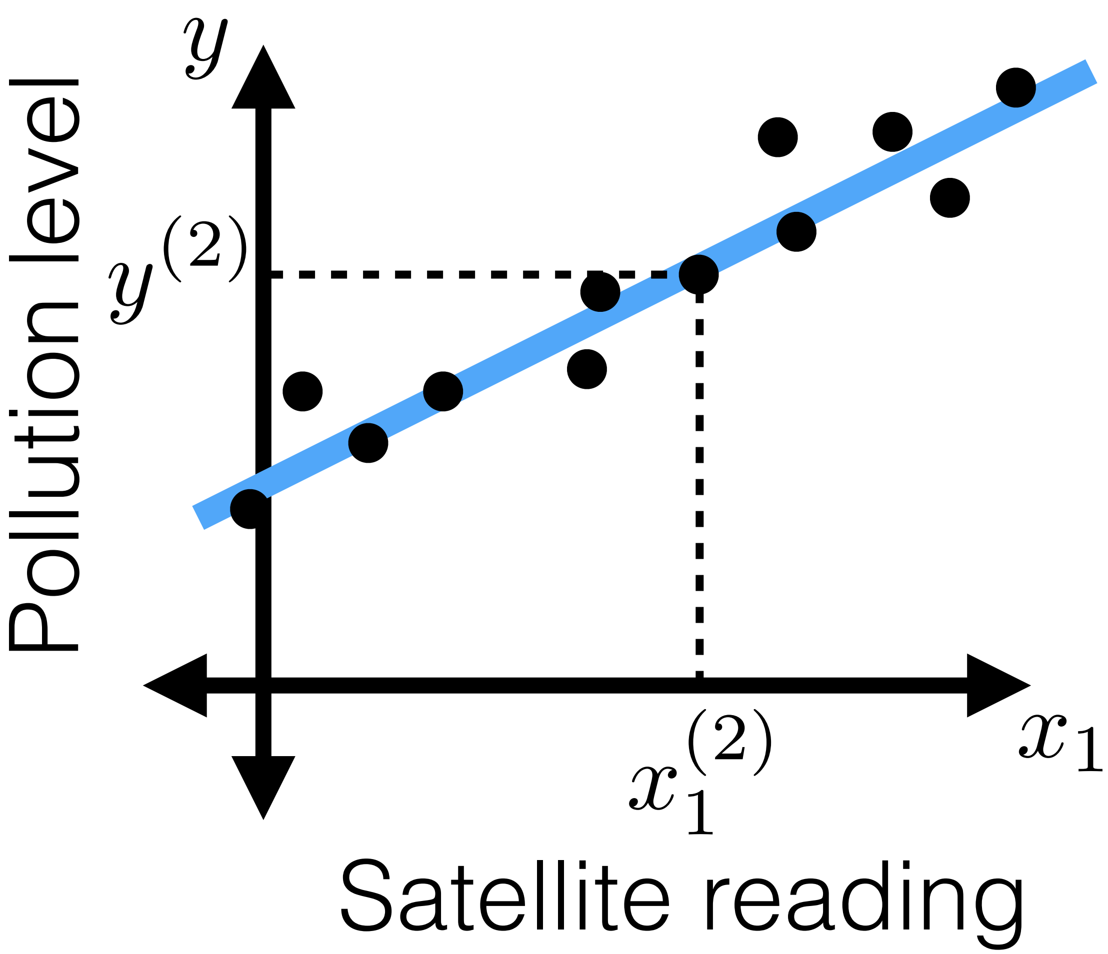
$$h(x) = \theta x + \theta_0$$



Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

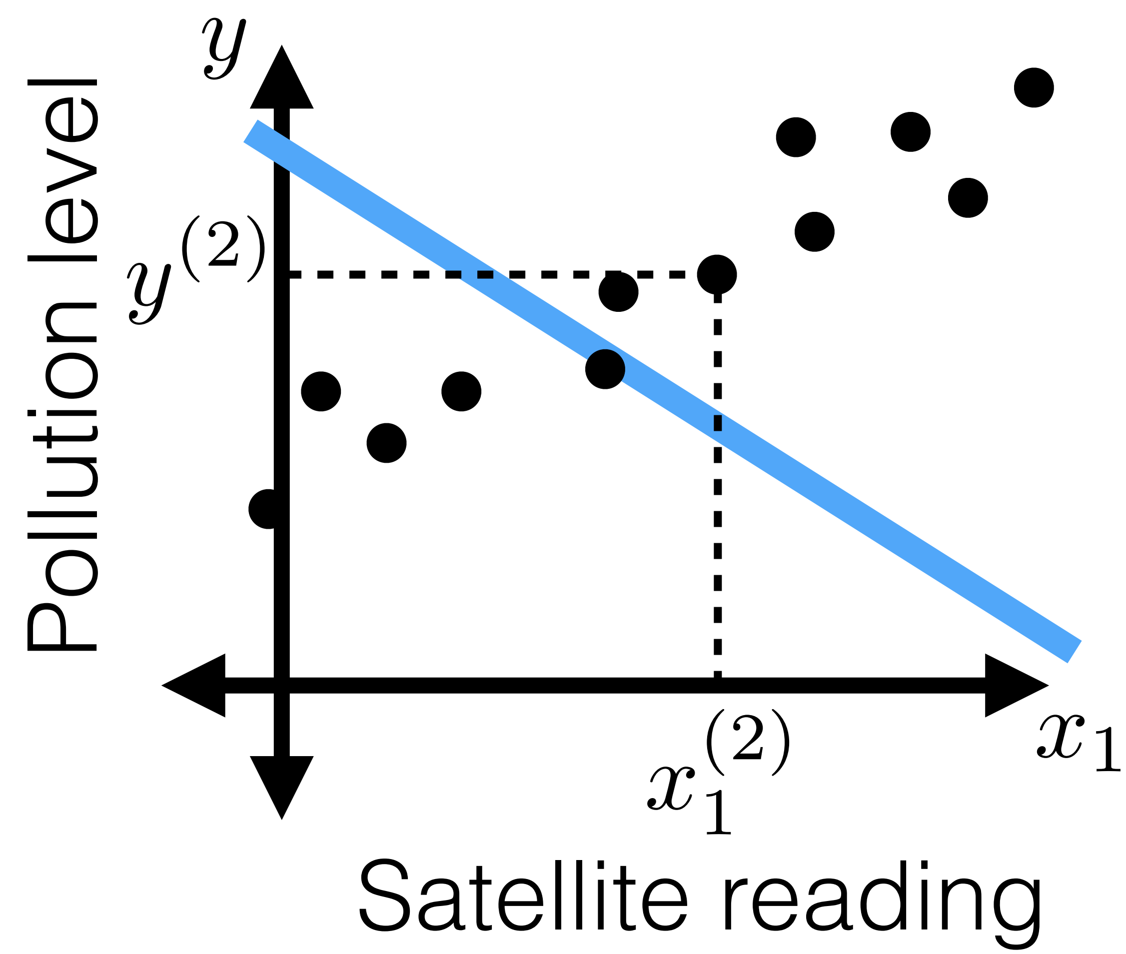
$$h(x) = \theta x + \theta_0$$



Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

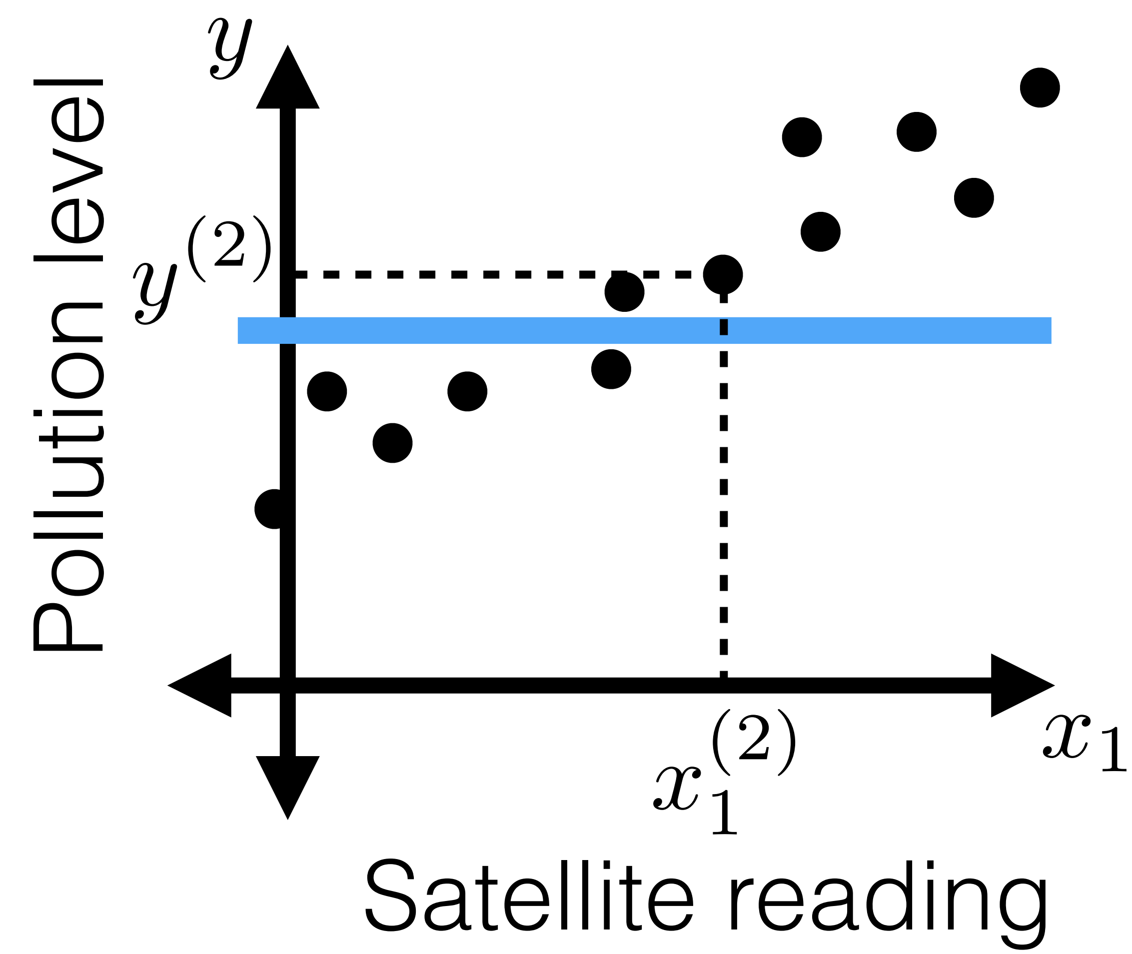
$$h(x) = \theta x + \theta_0$$



Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

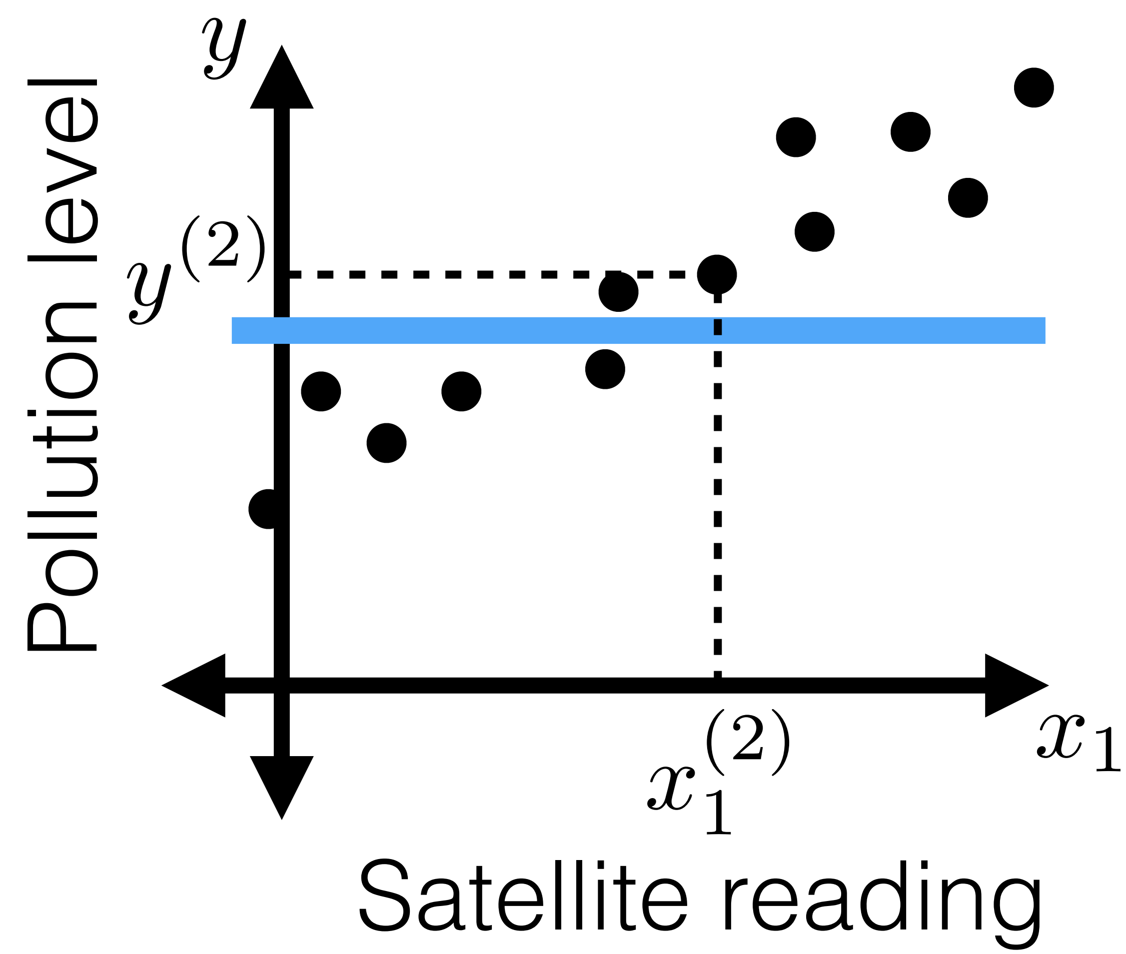
$$h(x) = \theta x + \theta_0$$



Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

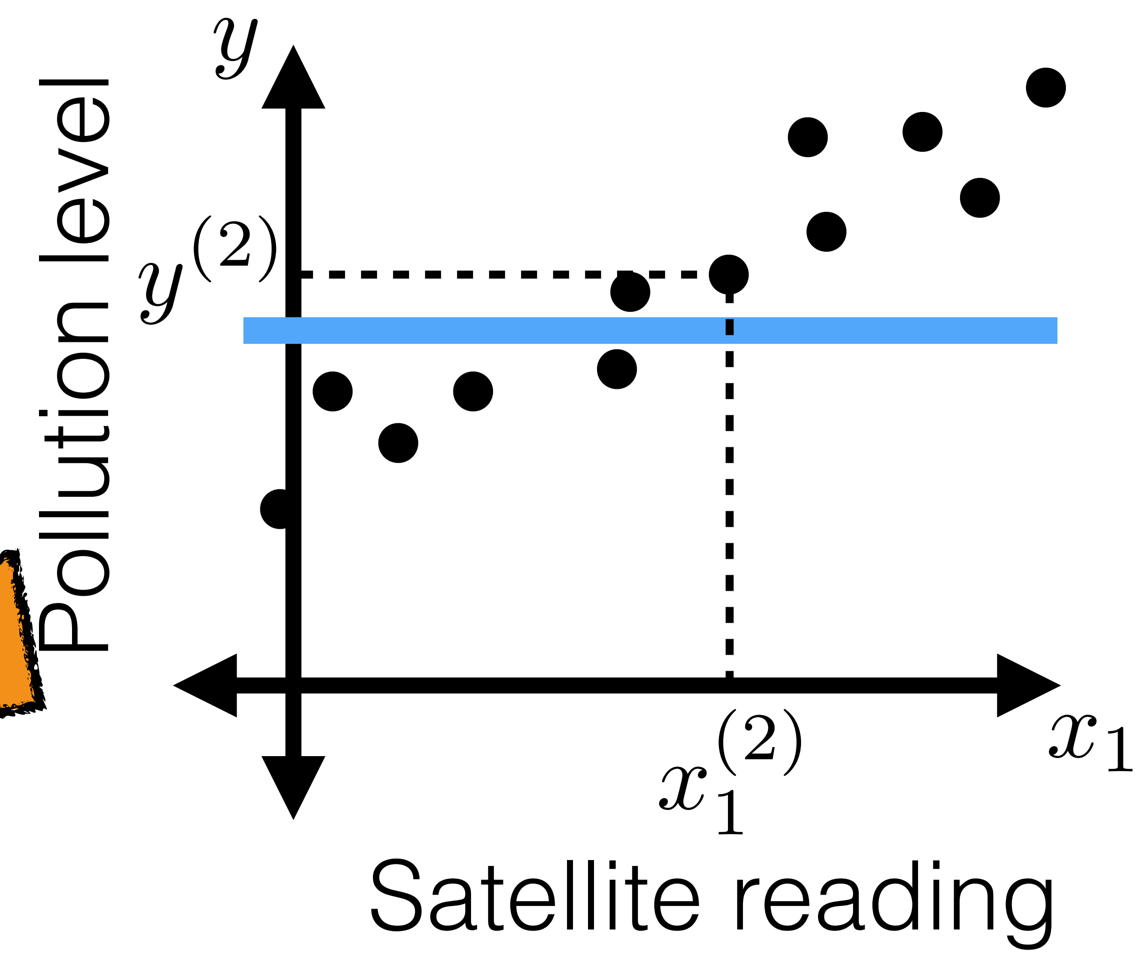


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters



Linear regressors

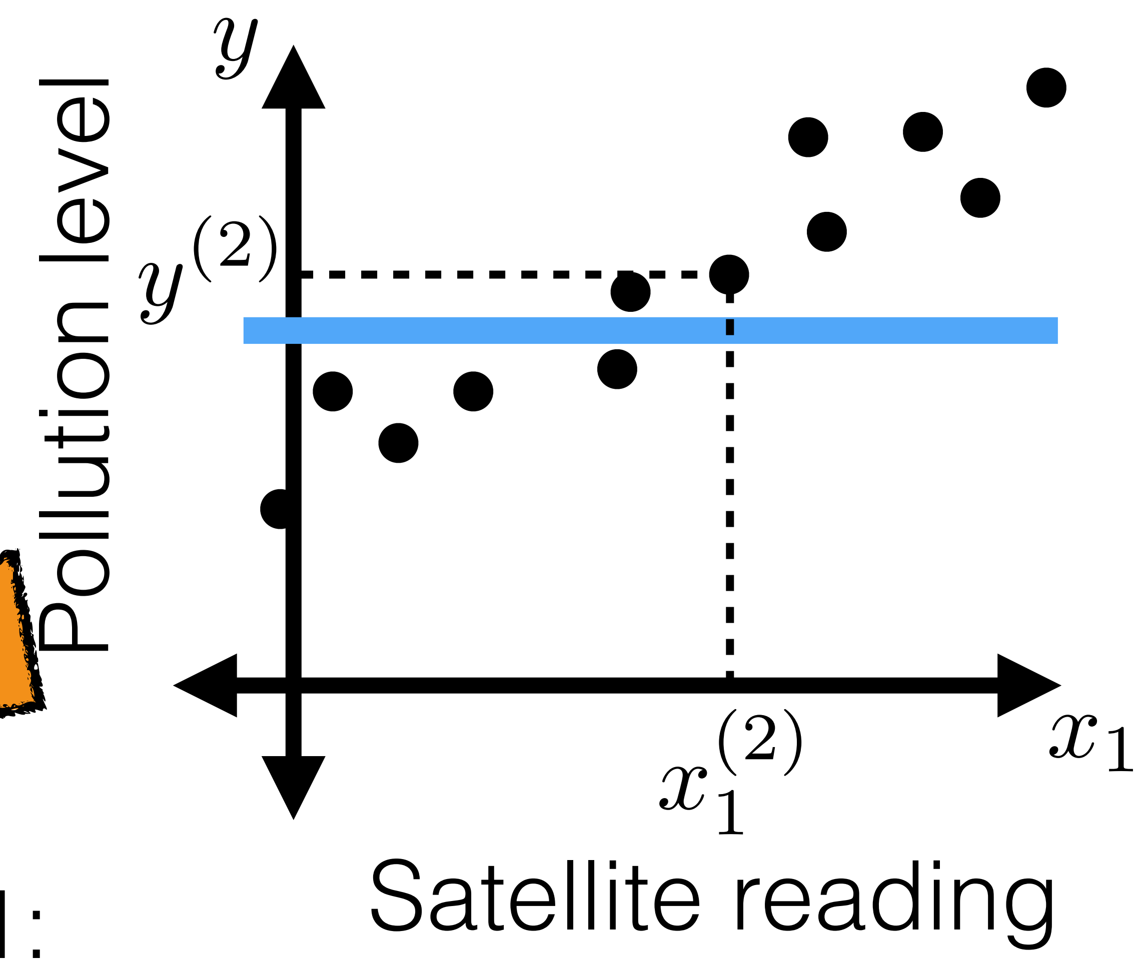
- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters

- A linear reg. hypothesis when *d*≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$



Linear regressors

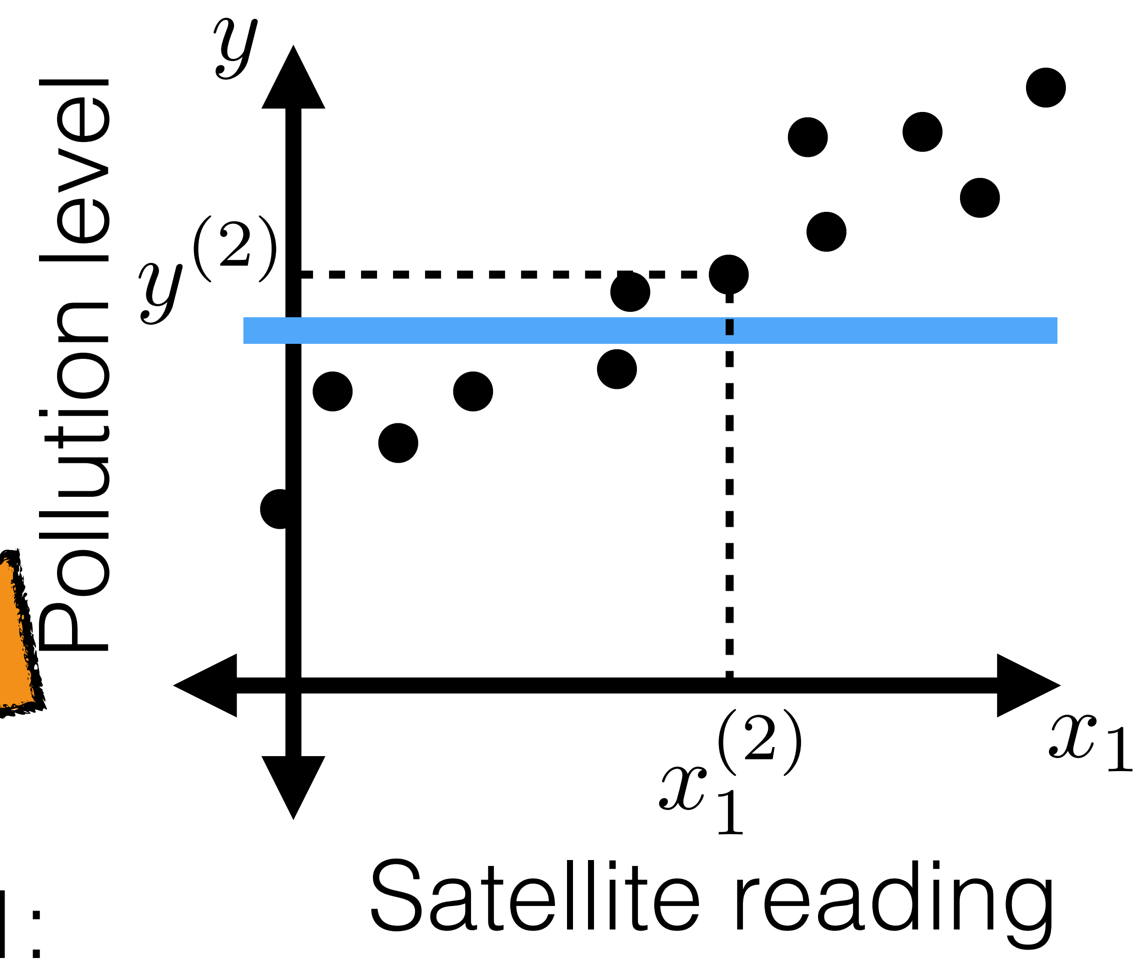
- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters

- A linear reg. hypothesis when *d*≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0 = \theta^\top x + \theta_0$$

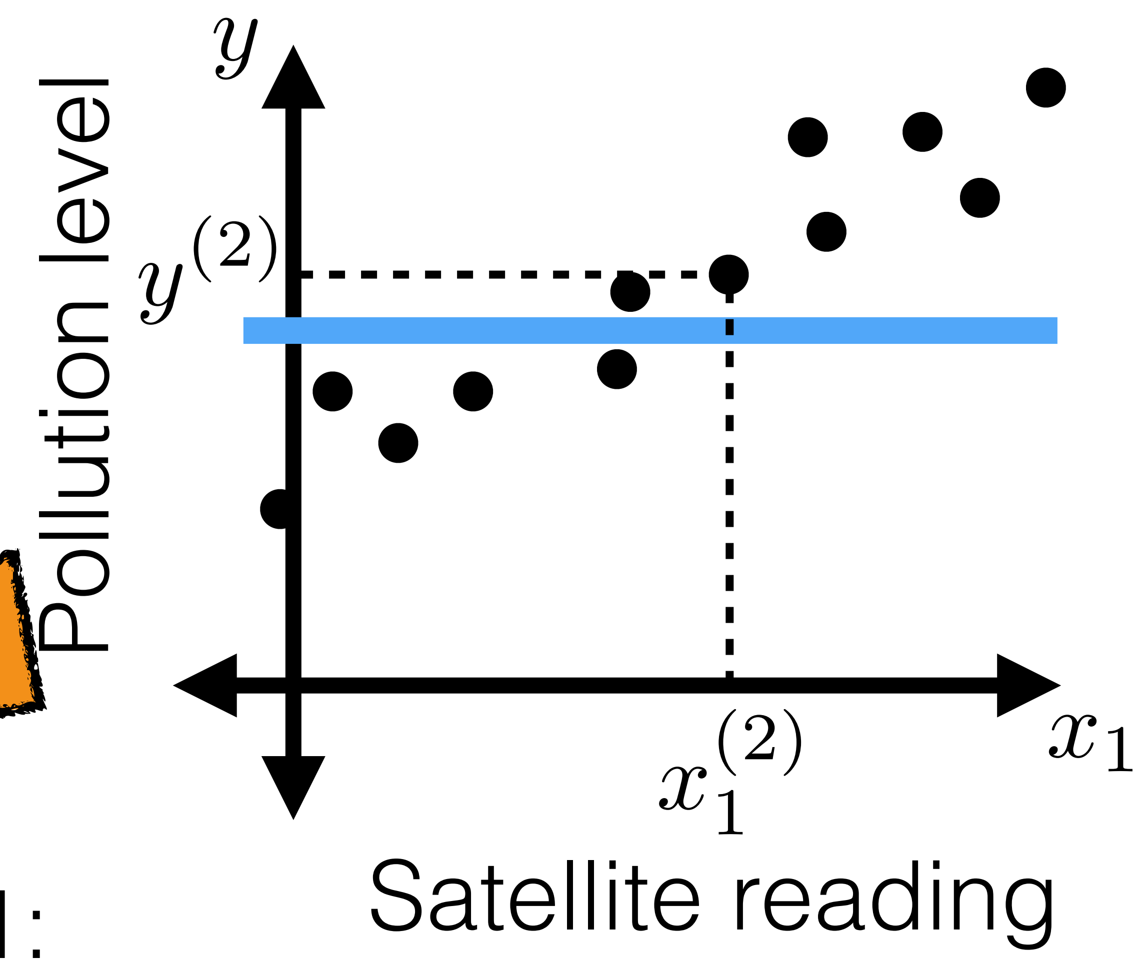


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x;\theta,\theta_0) = \theta x + \theta_0$$

parameters



- A linear reg. hypothesis when *d*≥1:

$$h(x;\theta,\theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

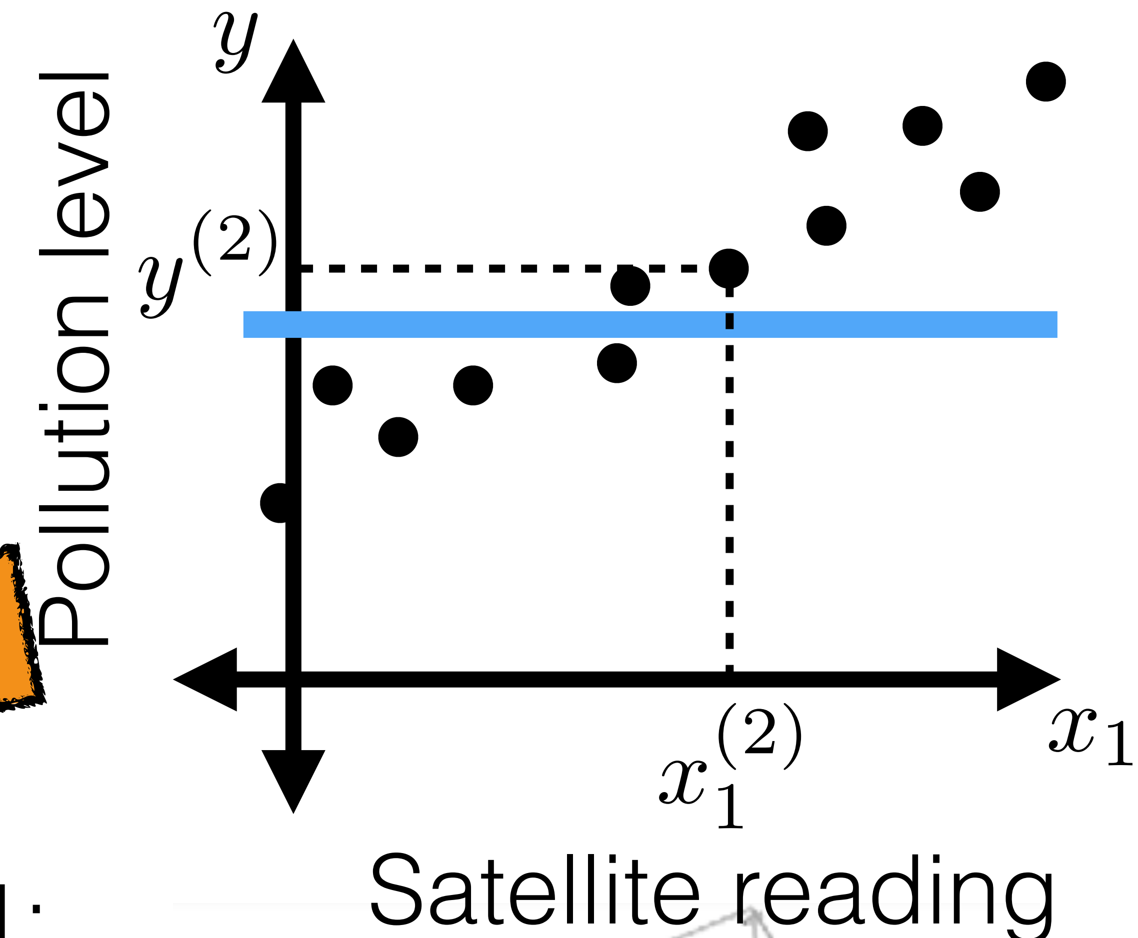
1xd,dx1

Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

parameters

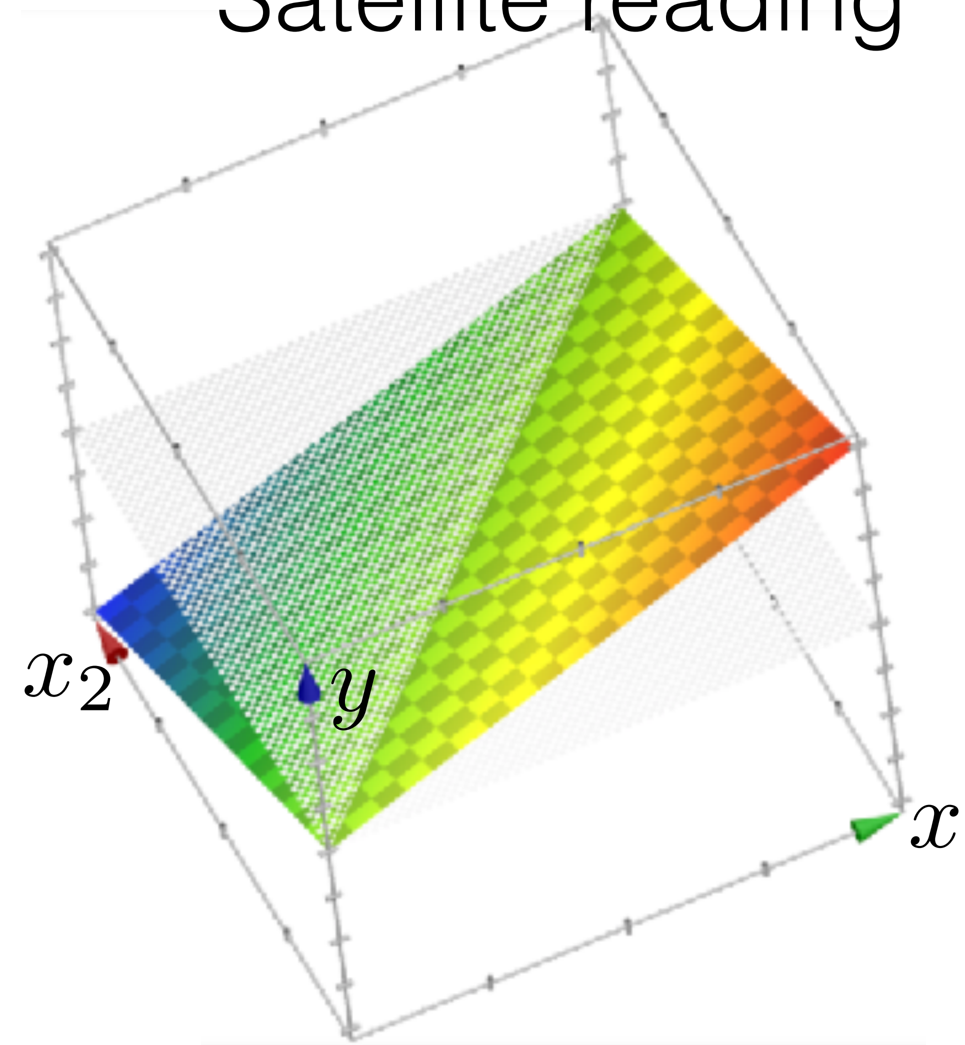
$$h(x; \theta, \theta_0) = \theta x + \theta_0$$



- A linear reg. hypothesis when *d*≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1xd,dx1

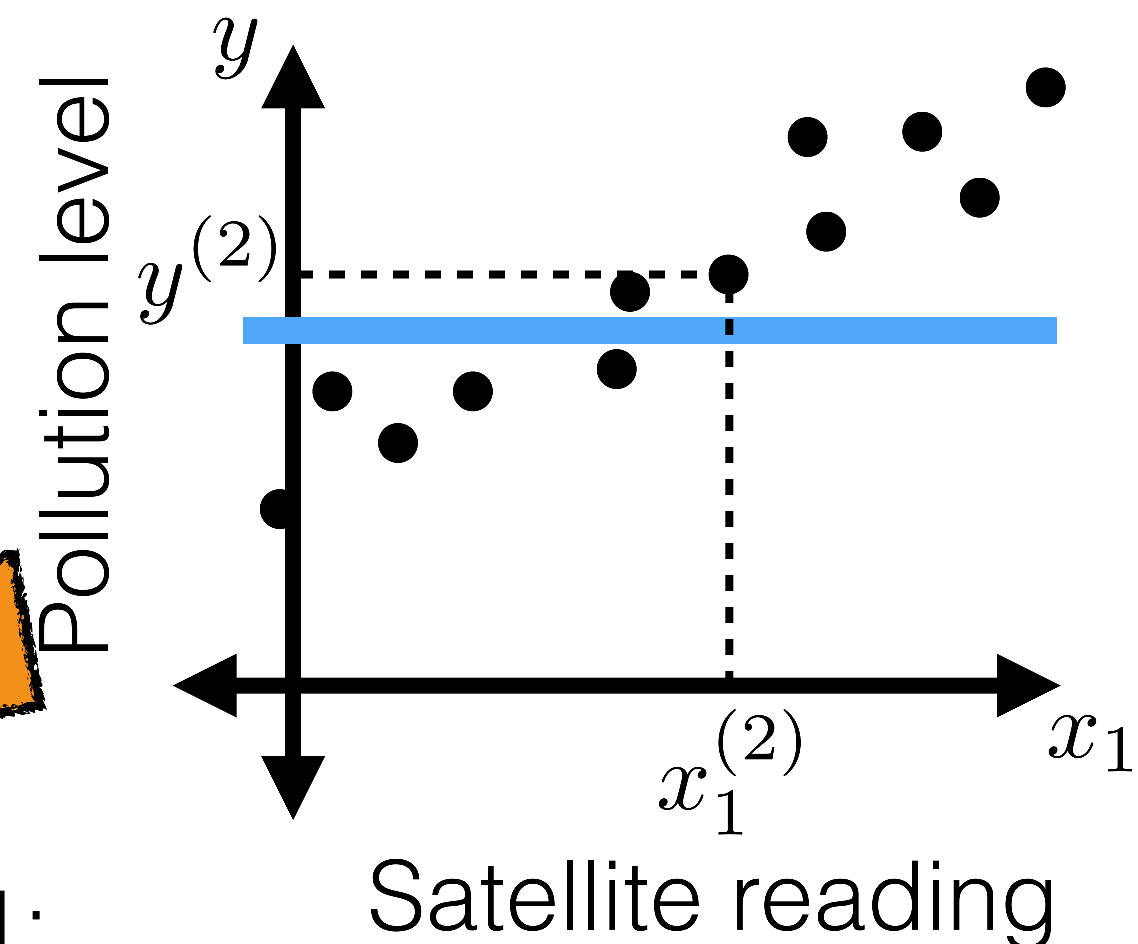


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when d=1:

$$h(x;\theta,\theta_0) = \theta x + \theta_0$$

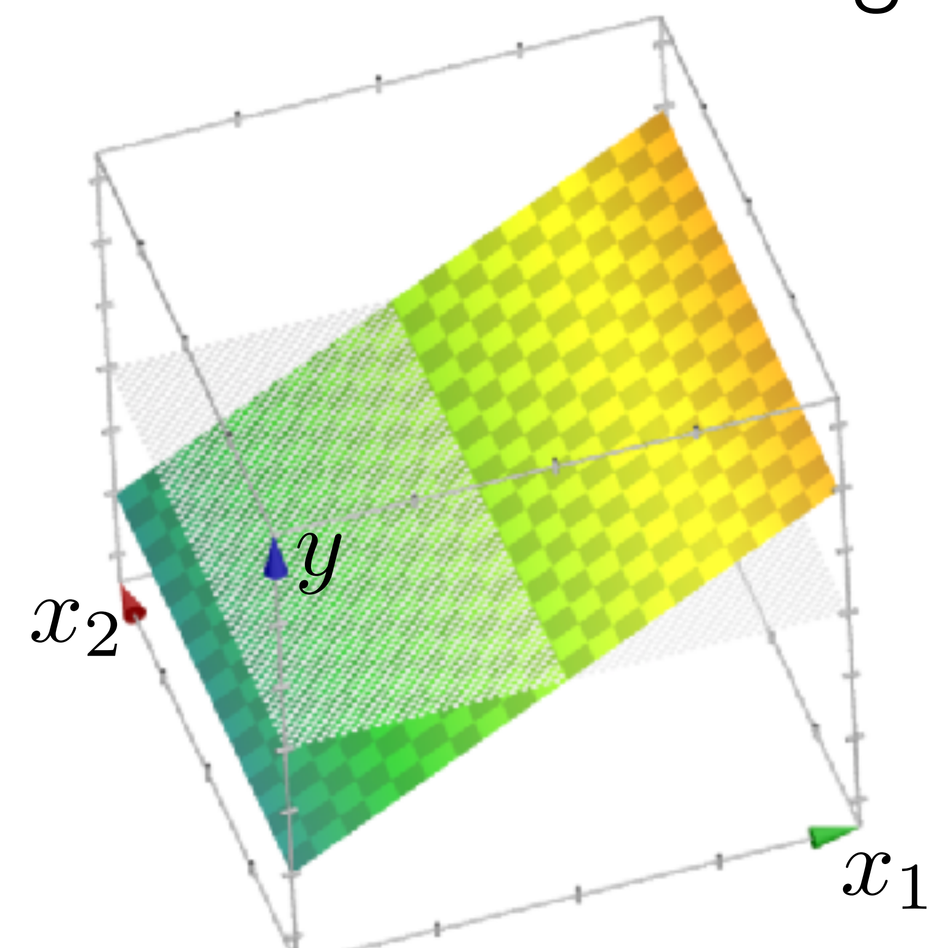
parameters



- A linear reg. hypothesis when d≥1:

$$h(x;\theta,\theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1xd,dx1

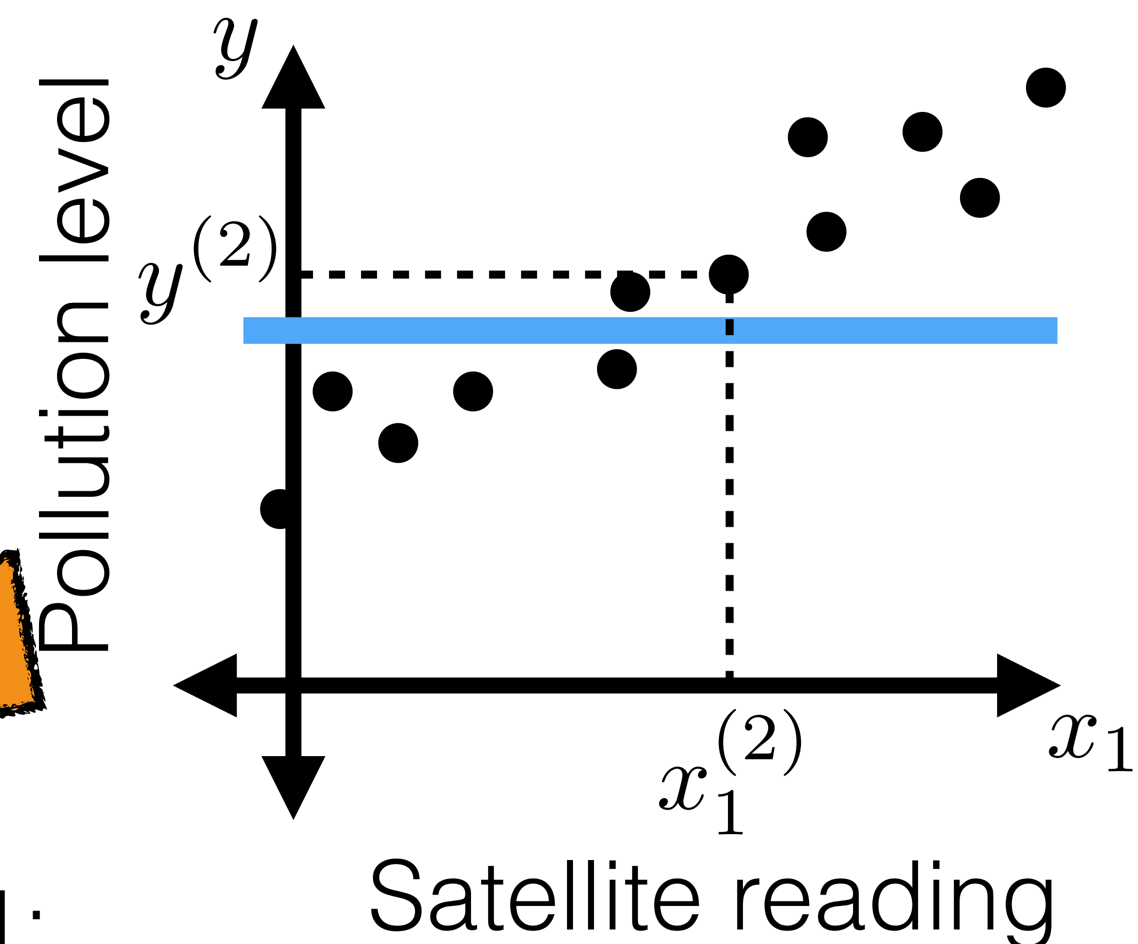


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters

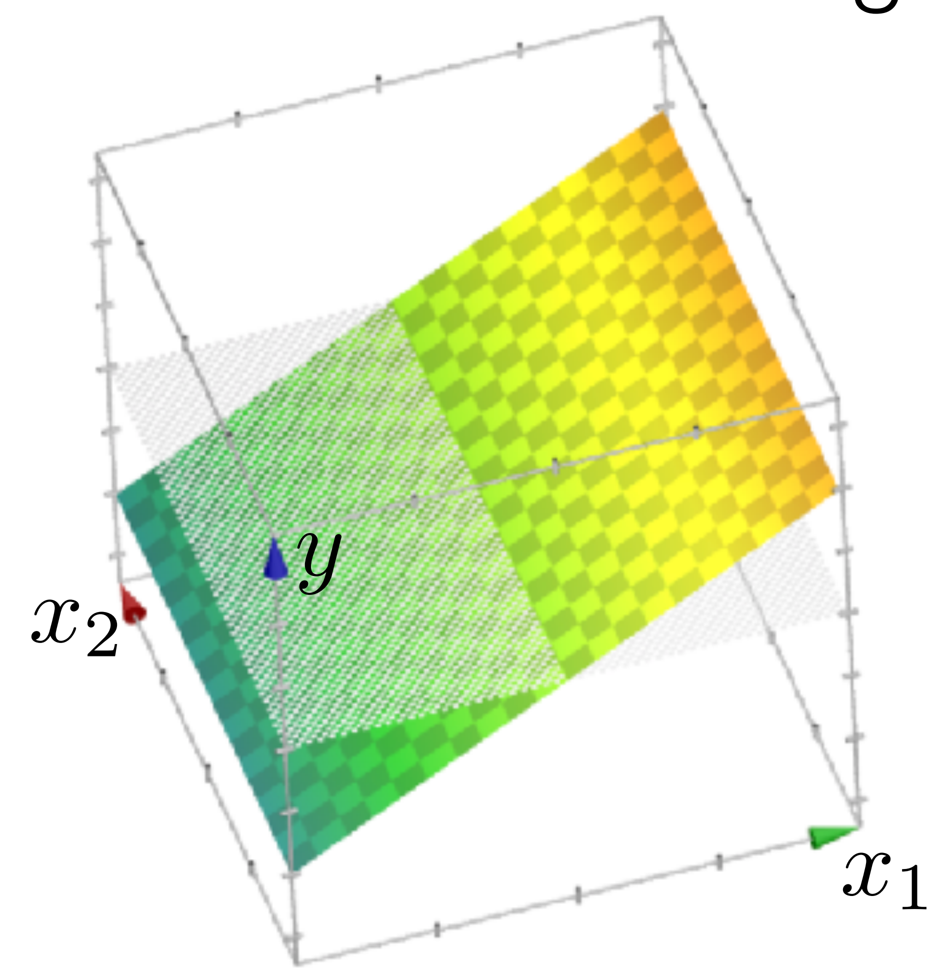


- A linear reg. hypothesis when *d*≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1xd,dx1

OR

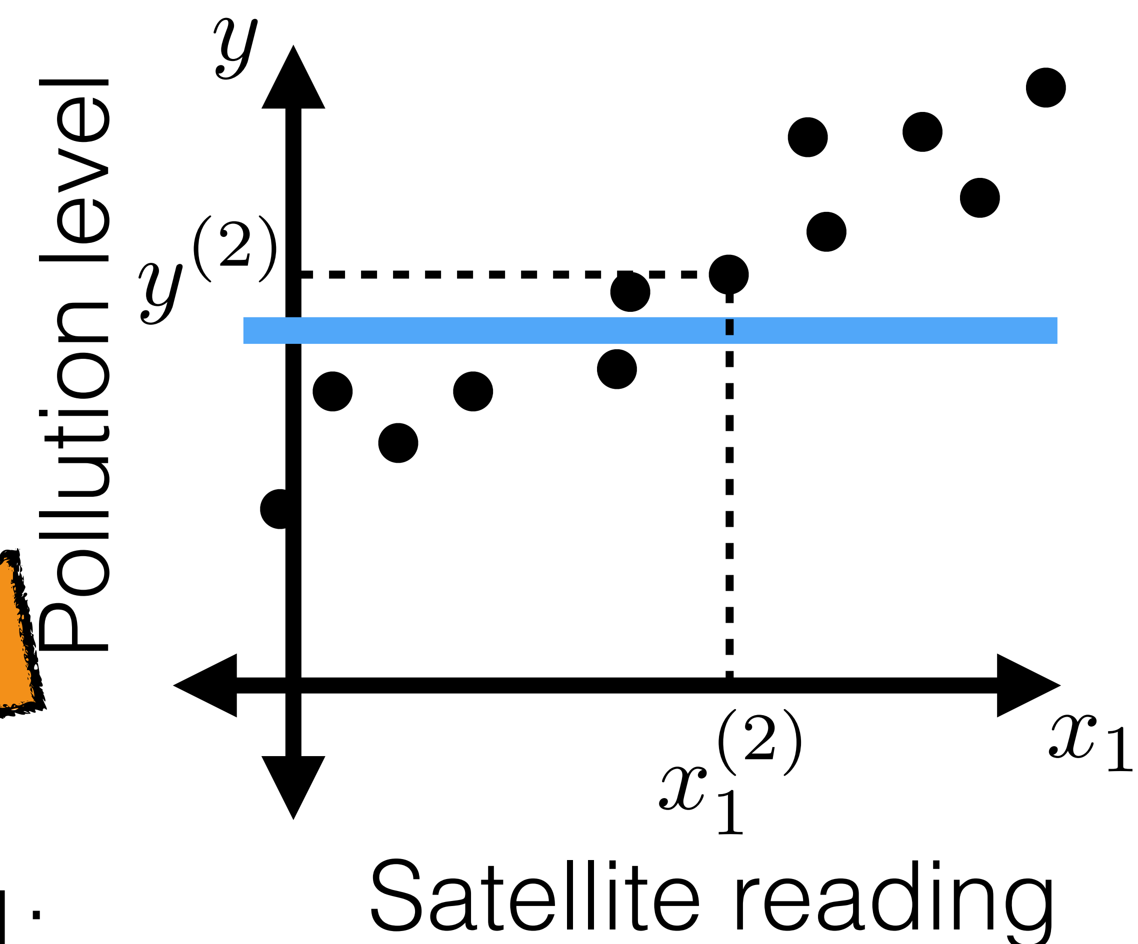


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters



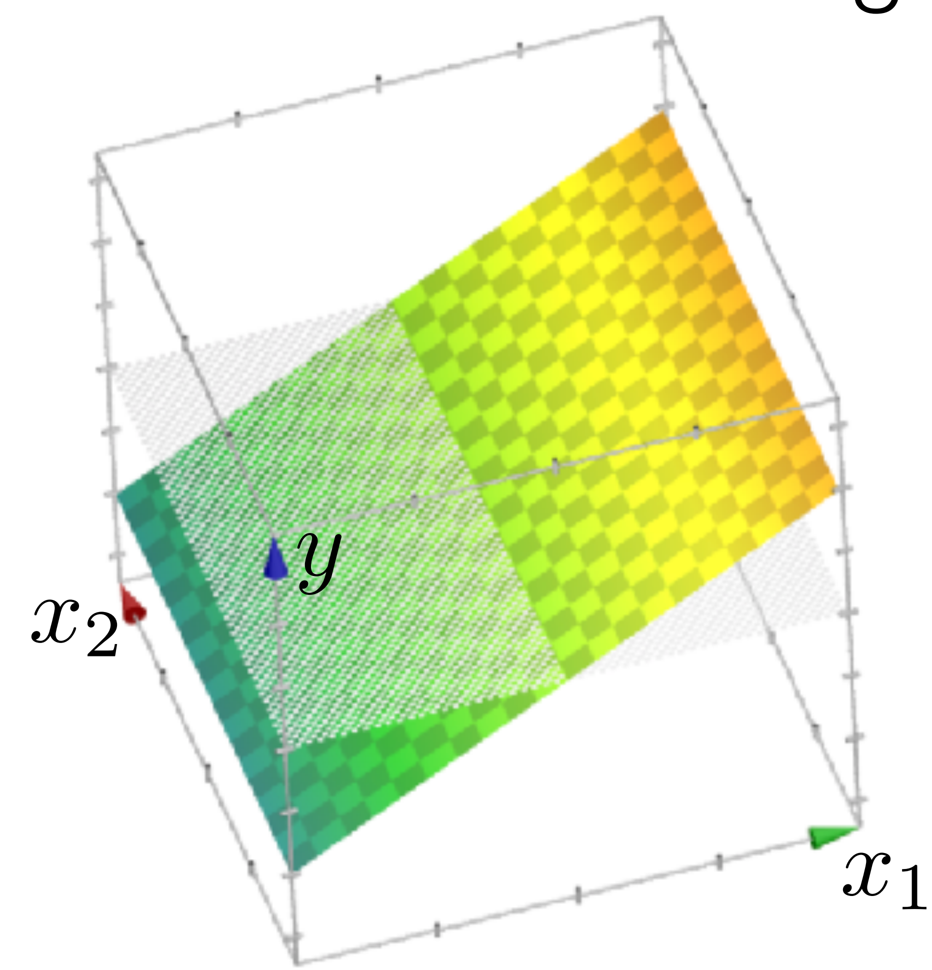
- A linear reg. hypothesis when *d*≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1xd,dx1

OR

$$h(x) = \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1)$$

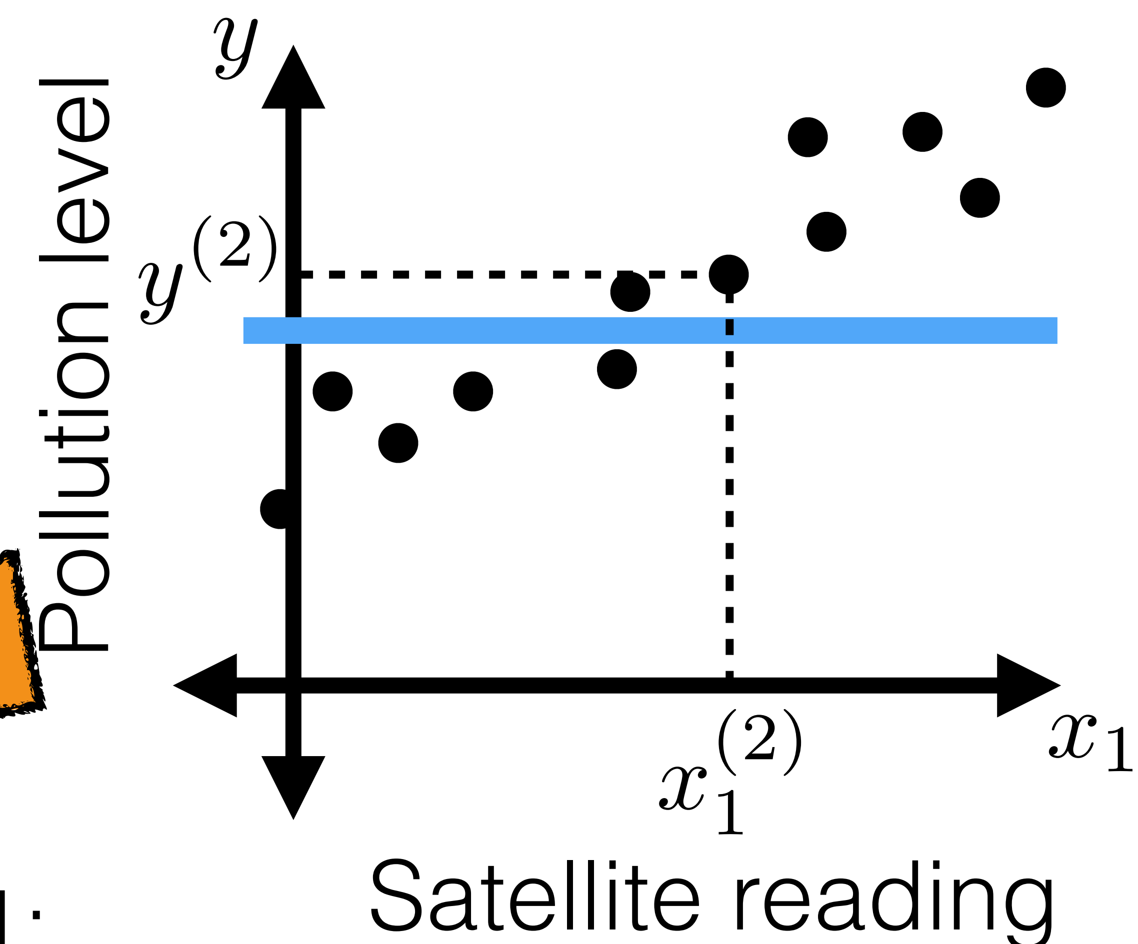


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters



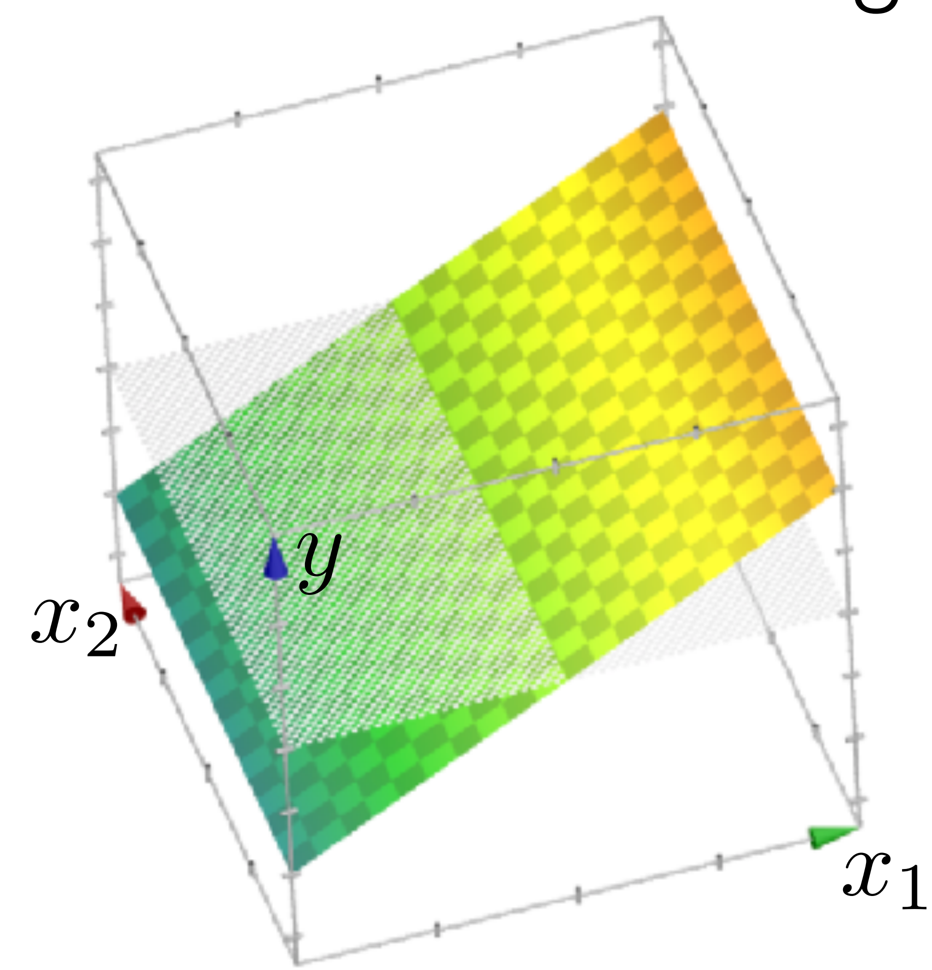
- A linear reg. hypothesis when *d*≥1:

$$\begin{aligned} h(x; \theta, \theta_0) &= \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0 \\ &= \theta^\top x + \theta_0 \end{aligned}$$

1xd,dx1

OR

$$\begin{aligned} h(x) &= \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1) \\ &= \theta^\top x \end{aligned}$$

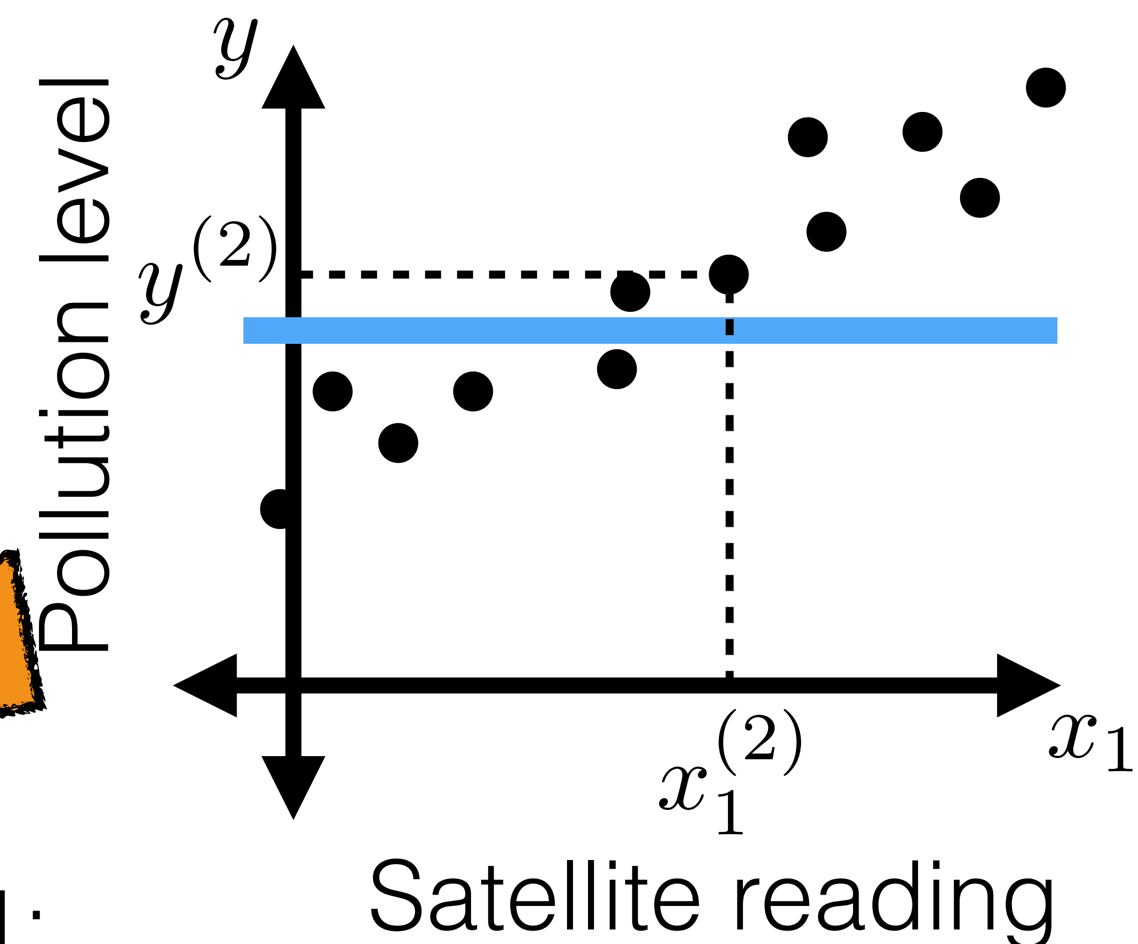


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

parameters

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$



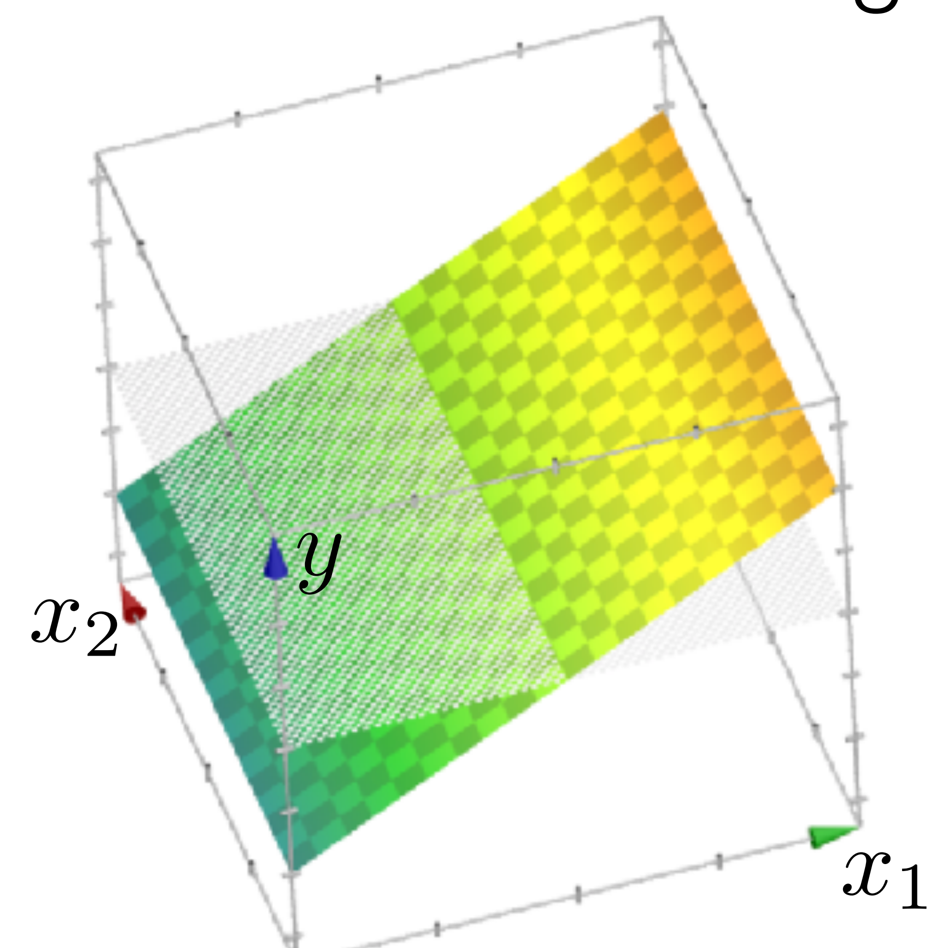
- A linear reg. hypothesis when *d*≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1xd,dx1

OR

$$h(x; \theta) = \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1)$$
$$= \theta^\top x$$

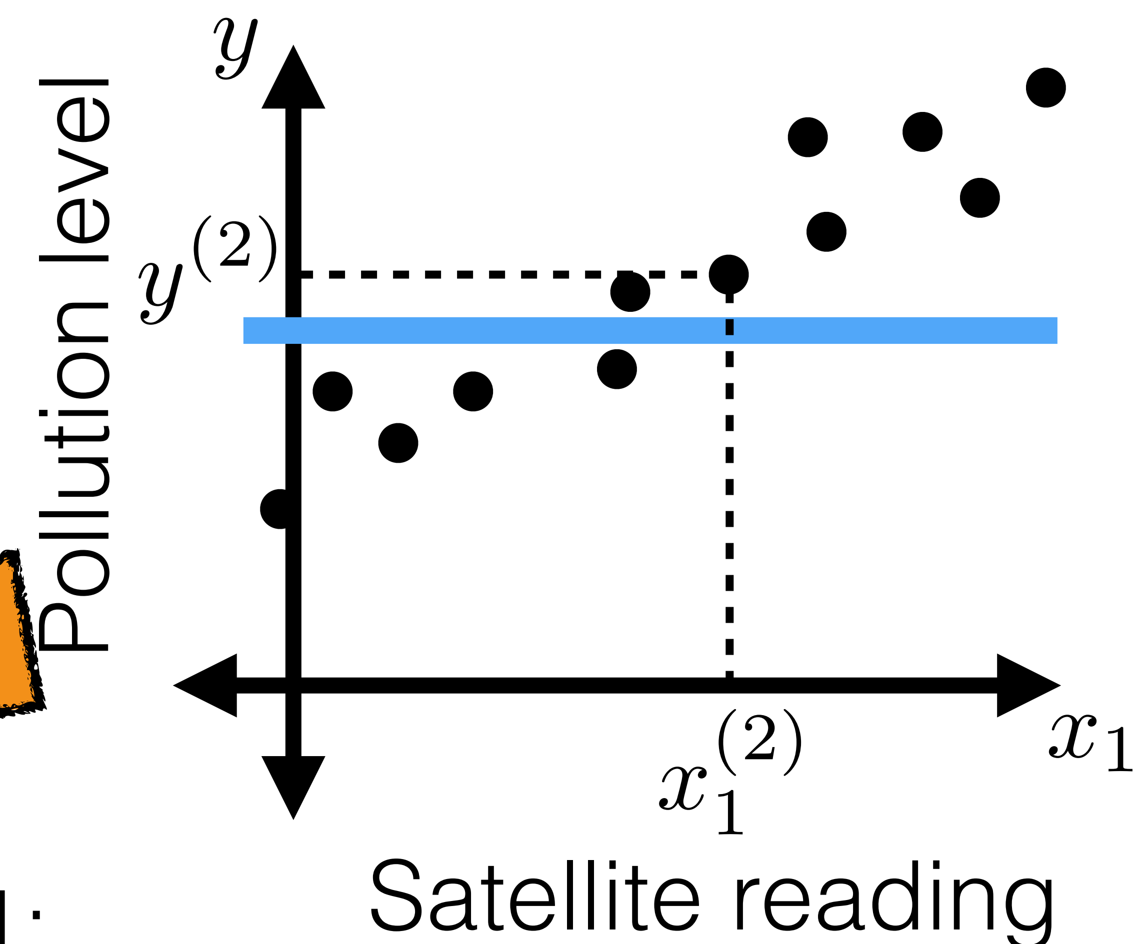


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when d=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters



- A linear reg. hypothesis when d≥1:

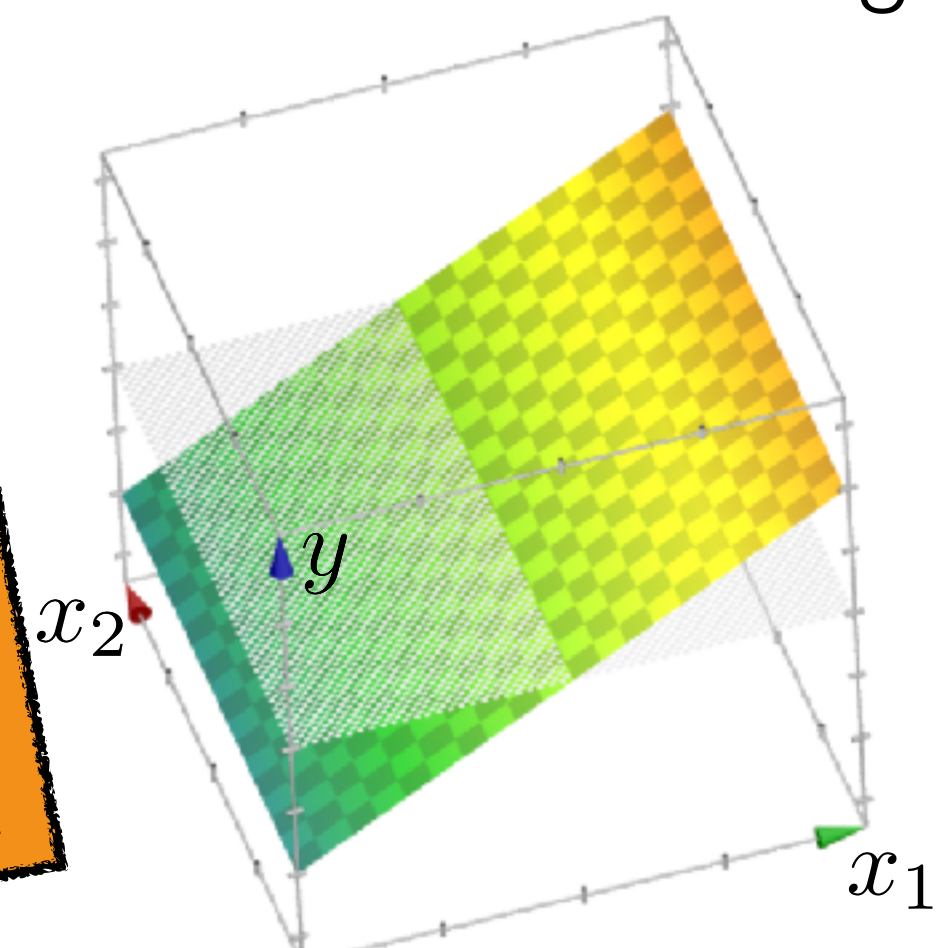
$$\begin{aligned} h(x; \theta, \theta_0) &= \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0 \\ &= \theta^\top x + \theta_0 \end{aligned}$$

1xd,dx1

OR

$$\begin{aligned} h(x; \theta) &= \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1) \\ &= \theta^\top x \end{aligned}$$

Notational trick: *not* the same θ & x!

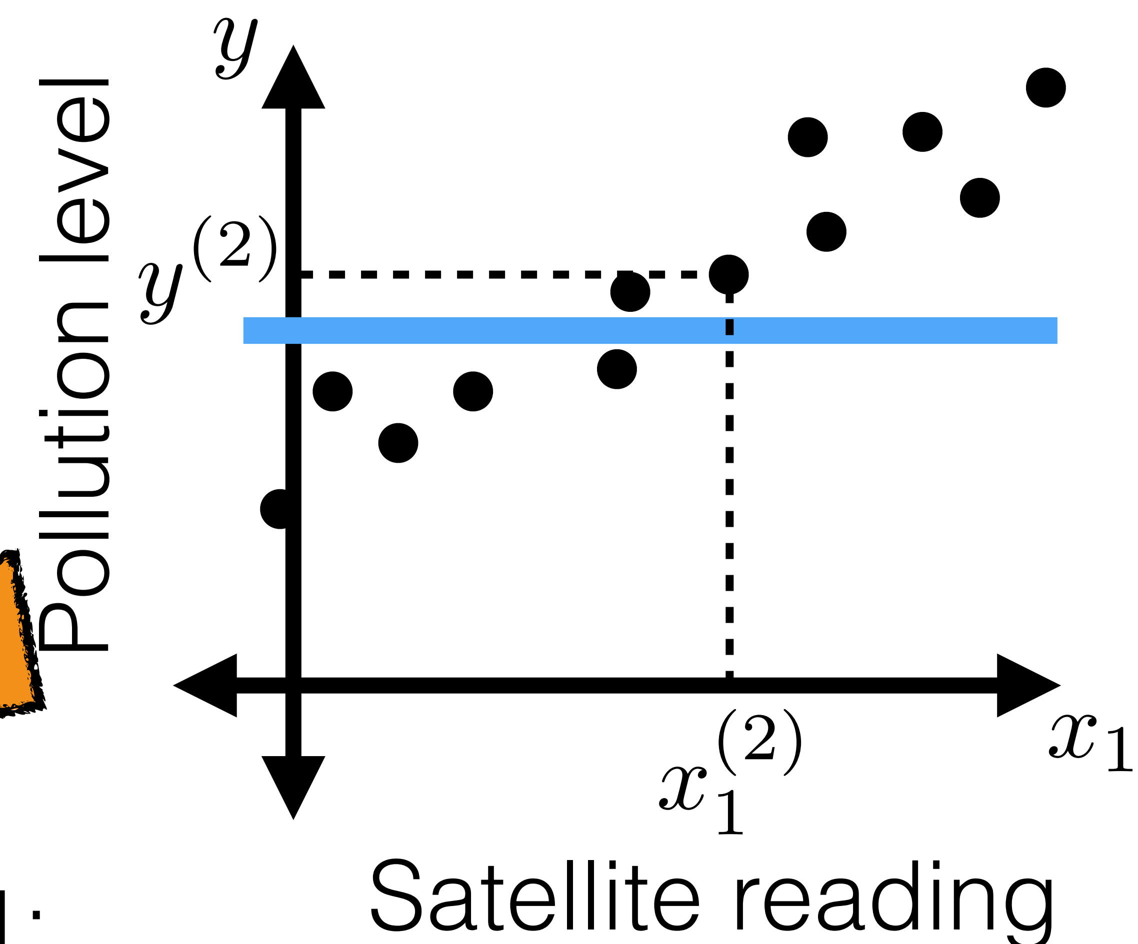


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when *d*=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters



- A linear reg. hypothesis when *d*≥1:

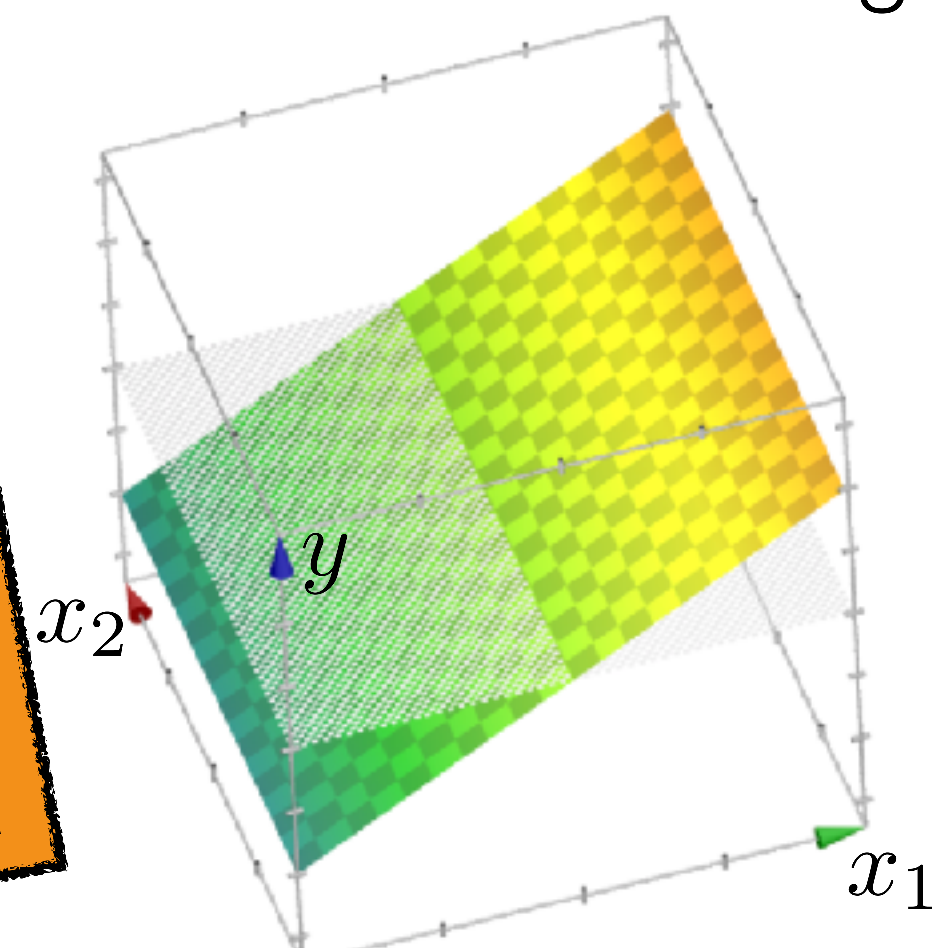
$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1x2,2x1

OR

$$h(x; \theta) = \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1)$$
$$= \theta^\top x$$

Notational trick: *not* the same θ & x!

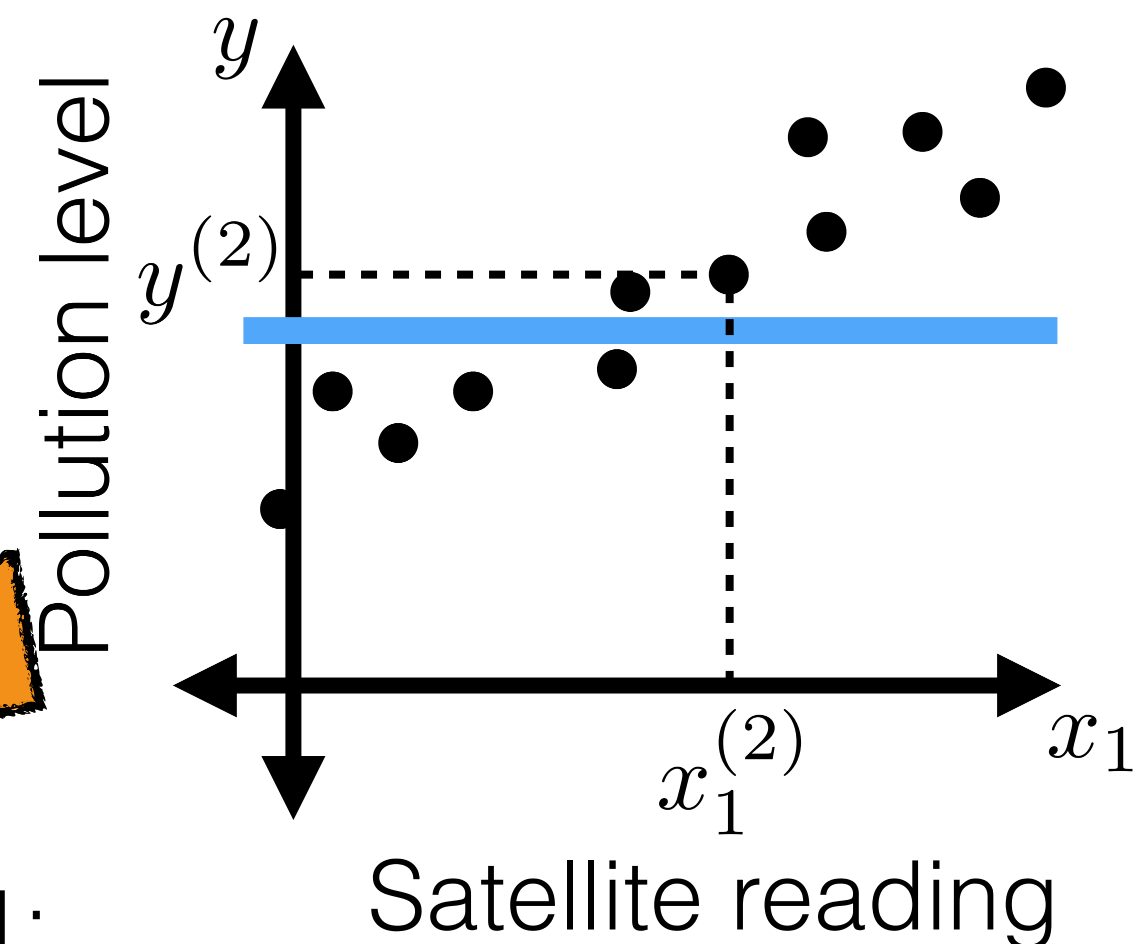


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when d=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters



- A linear reg. hypothesis when d≥1:

$$\begin{aligned} h(x; \theta, \theta_0) &= \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0 \\ &= \theta^\top x + \theta_0 \end{aligned}$$

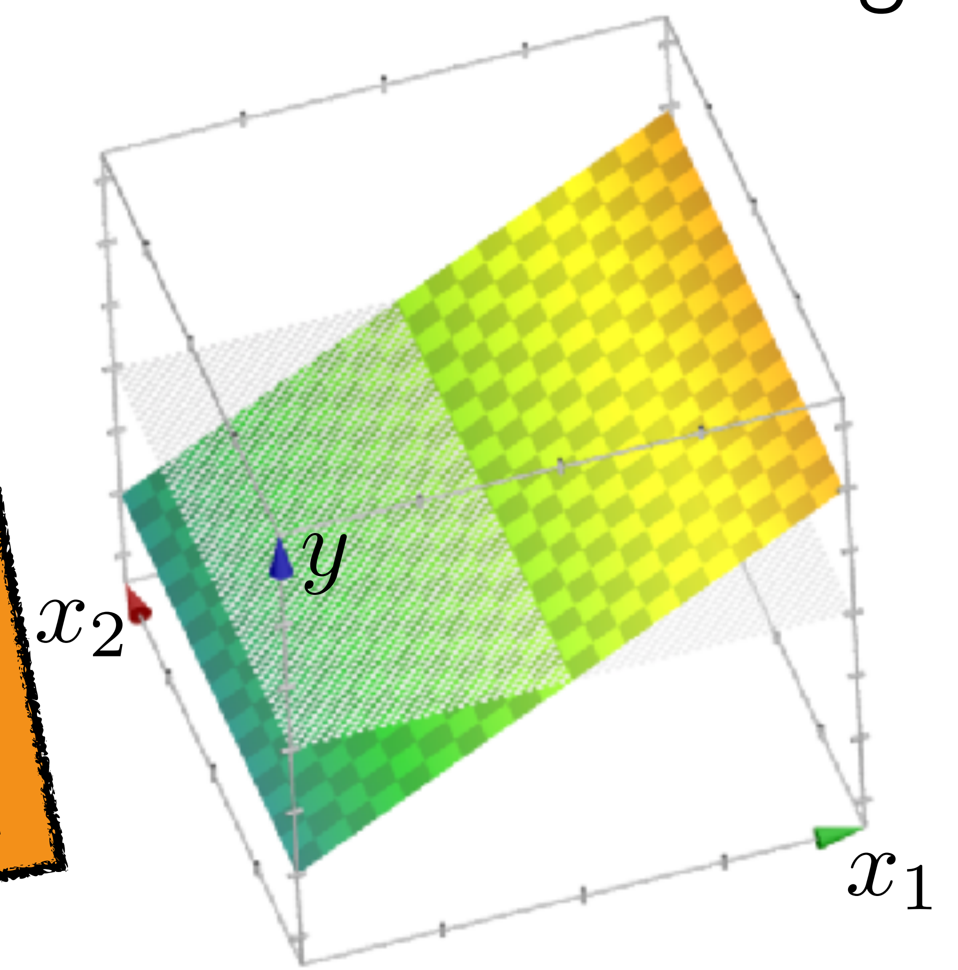
1x2,2x1

OR

$$\begin{aligned} h(x; \theta) &= \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1) \\ &= \theta^\top x \end{aligned}$$

1x3,3x1

Notational trick: *not* the same θ & x!

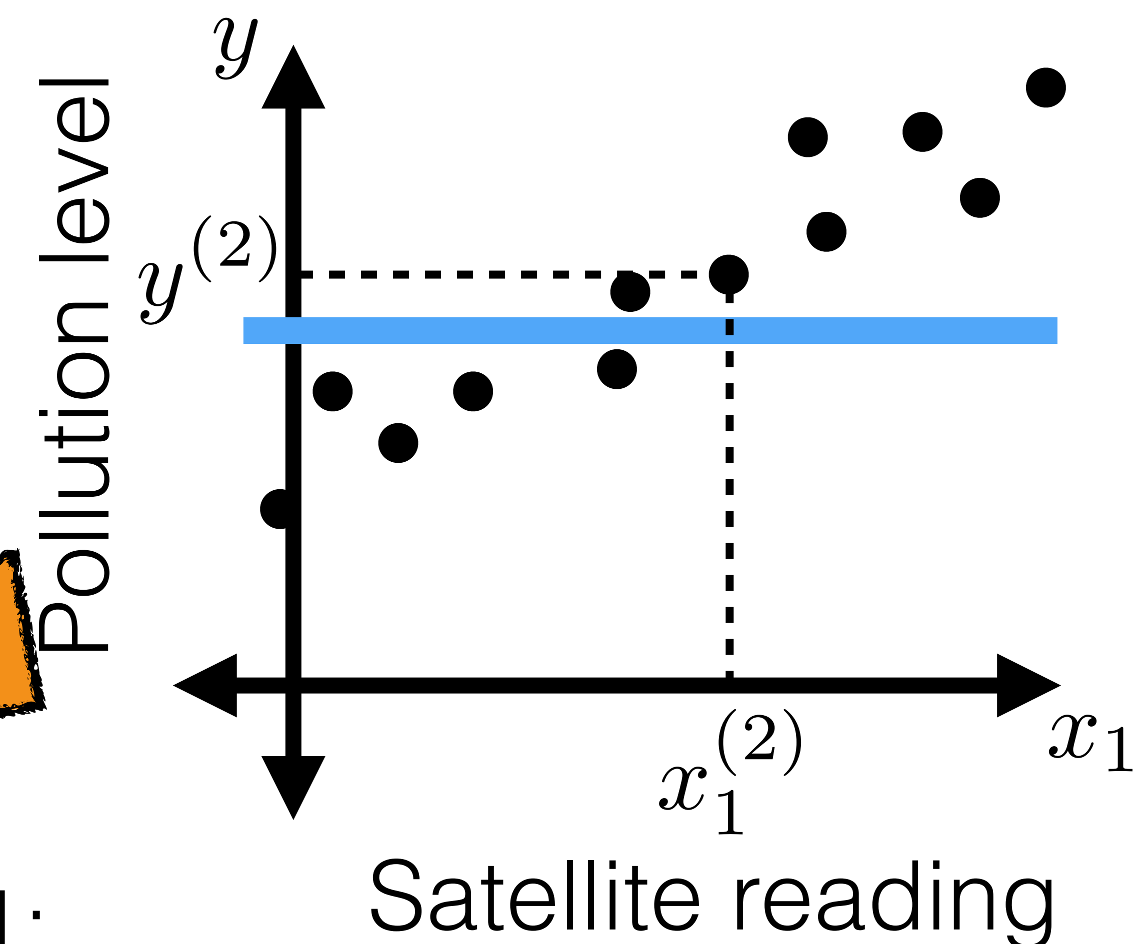


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when d=1:

$$h(x;\theta,\theta_0) = \theta x + \theta_0$$

parameters



- A linear reg. hypothesis when d≥1:

$$h(x;\theta,\theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

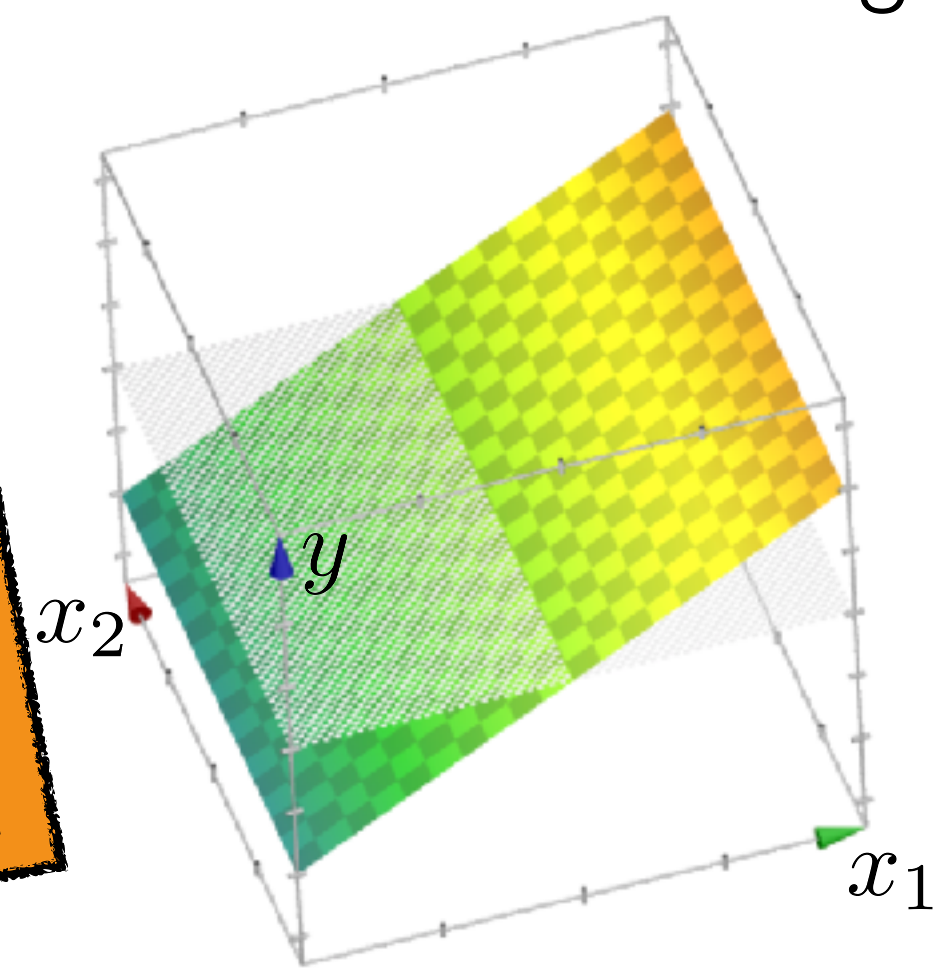
1x2,2x1

OR

$$h(x;\theta) = \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1)$$
$$= \theta^\top x$$

1x3,3x1

Notational trick: *not* the same θ & x!

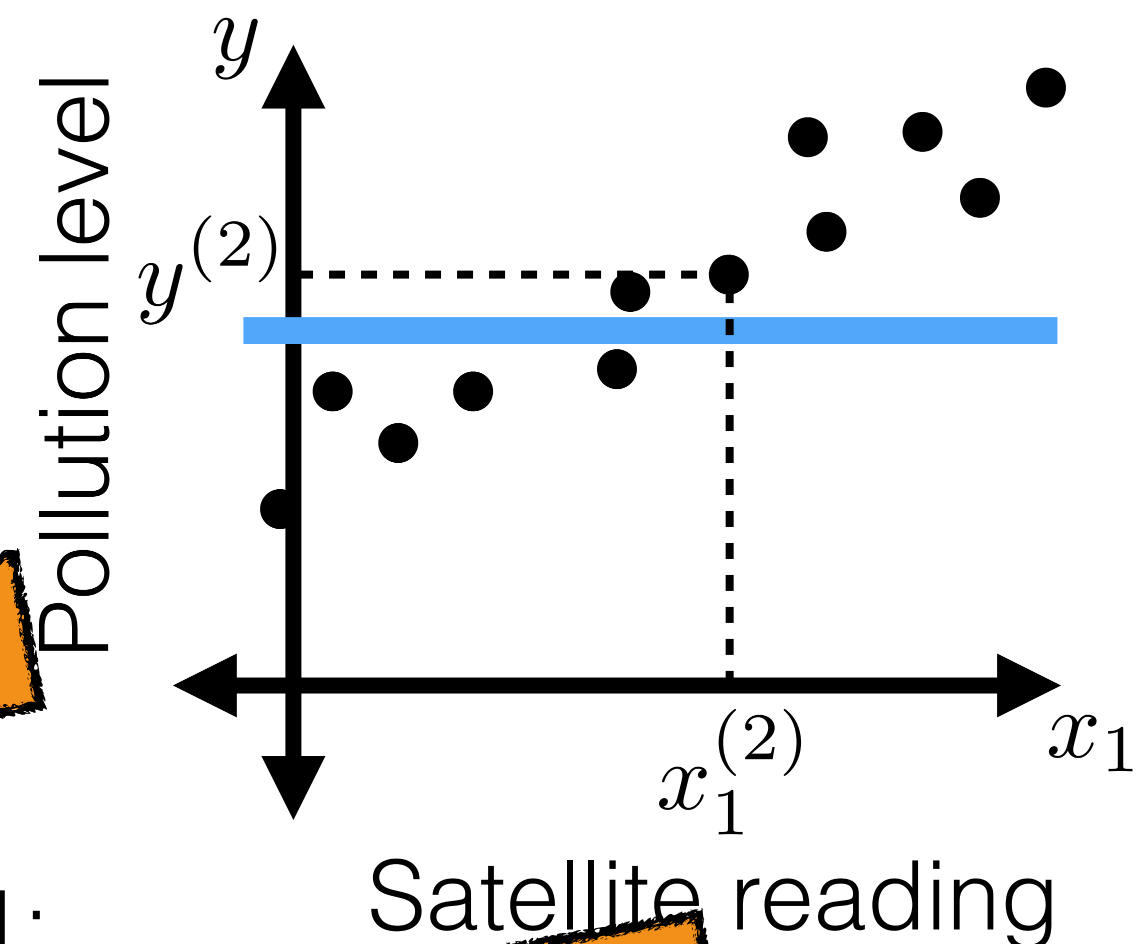


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when d=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters



- A linear reg. hypothesis when d≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1x2,2x1

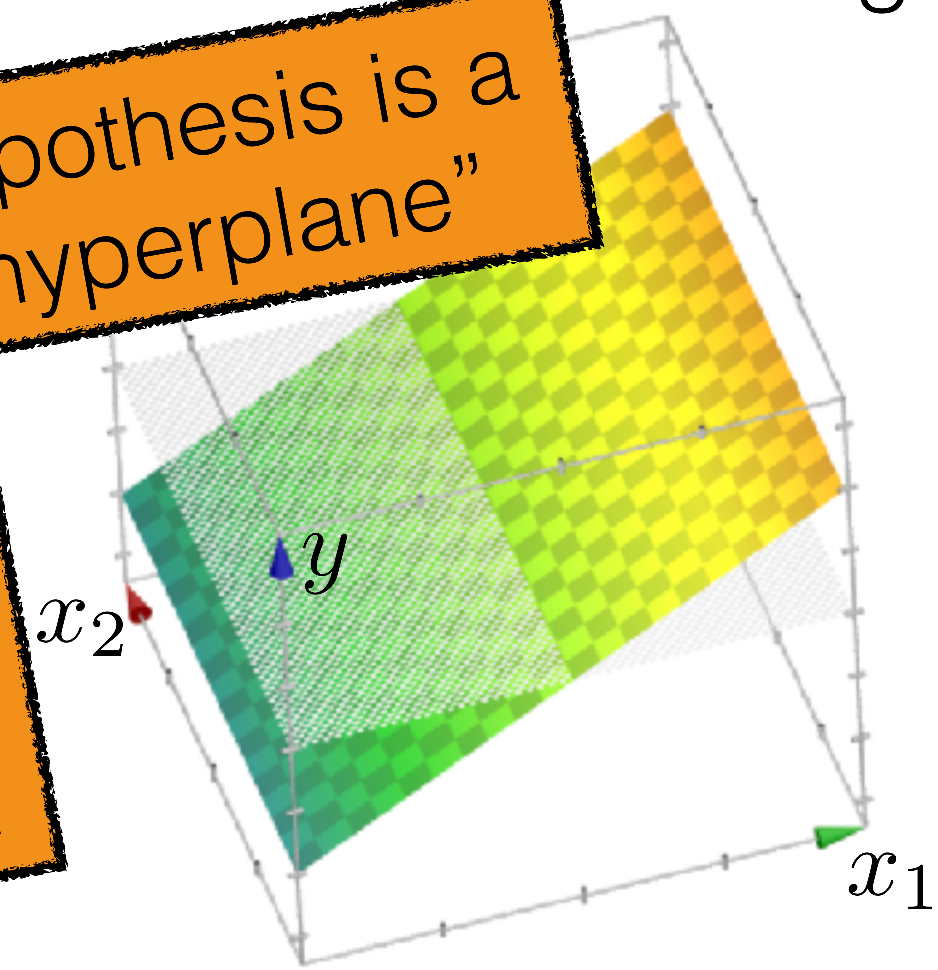
OR

$$h(x; \theta) = \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1)$$
$$= \theta^\top x$$

1x3,3x1

Hypothesis is a "hyperplane"

Notational trick: *not* the same θ & x!

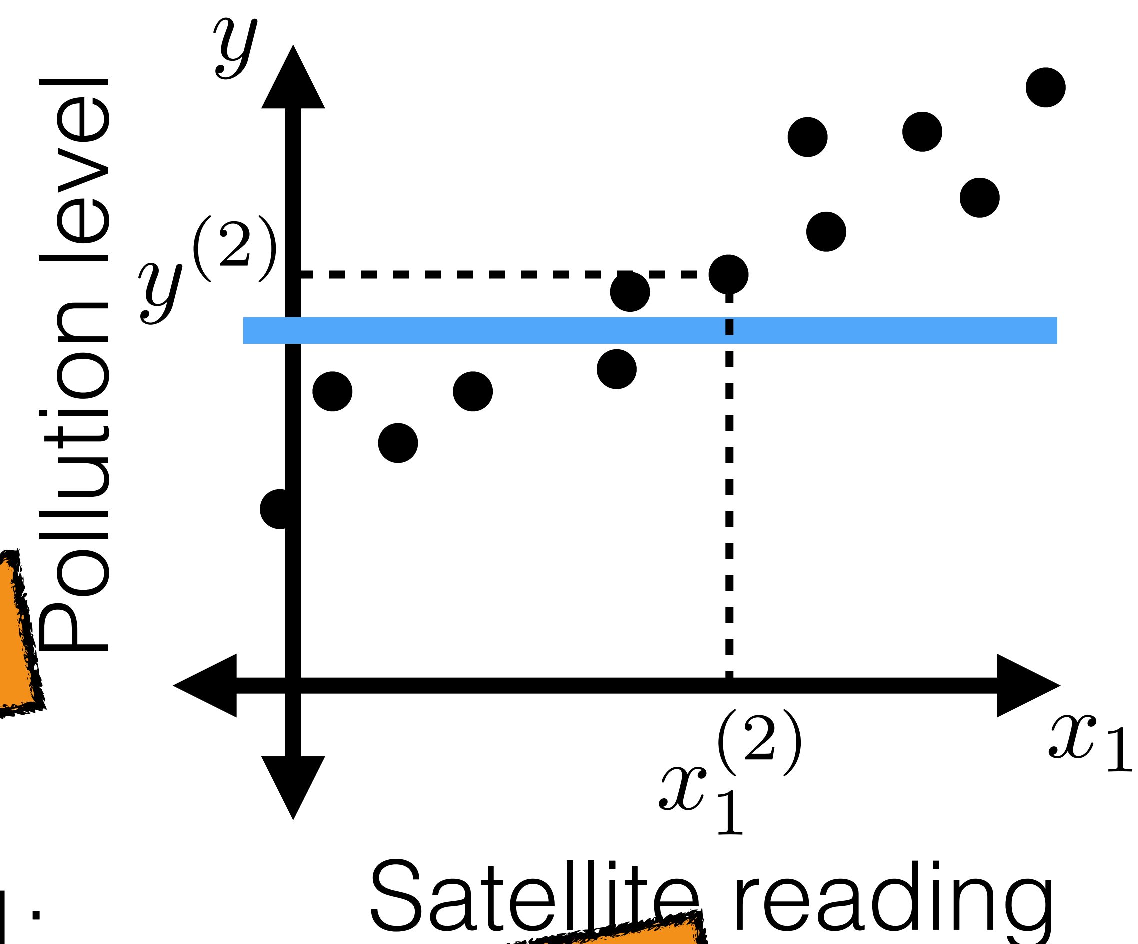


Linear regressors

- Hypothesis class $\mathcal{H}$: set of h
  - Example: all constant functions
- A linear regression hypothesis when d=1:

$$h(x; \theta, \theta_0) = \theta x + \theta_0$$

parameters

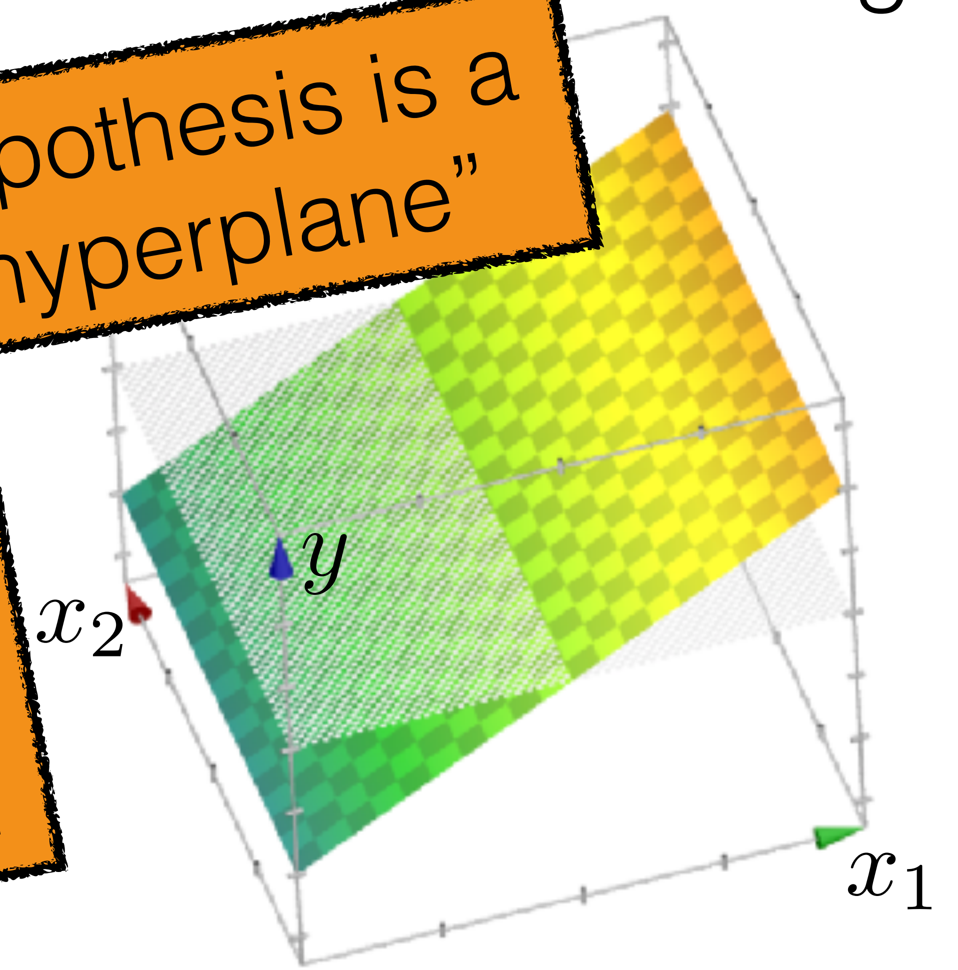


- A linear reg. hypothesis when d≥1:

$$h(x; \theta, \theta_0) = \theta_1 x_1 + \cdots + \theta_d x_d + \theta_0$$
$$= \theta^\top x + \theta_0$$

1x2,2x1

Hypothesis is a "hyperplane"



OR

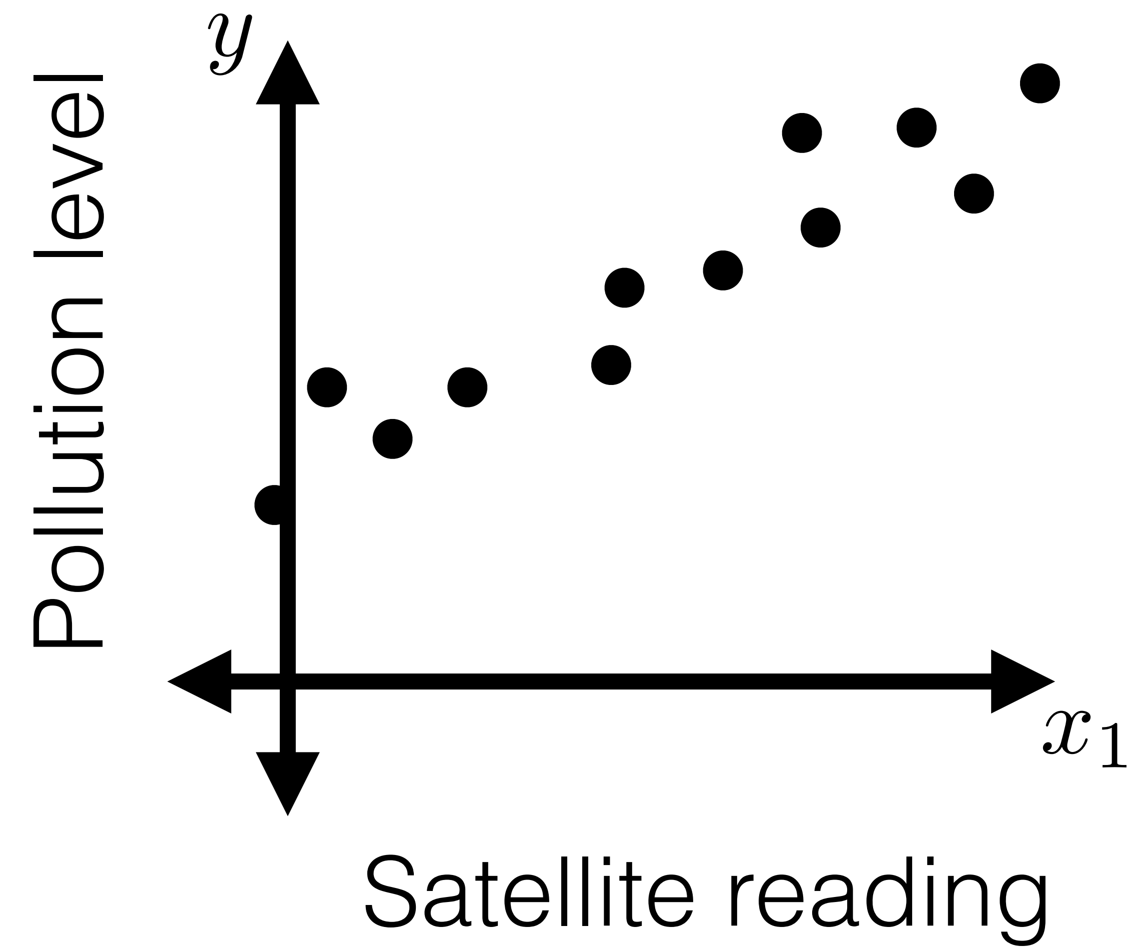
$$h(x; \theta) = \theta_1 x_1 + \cdots + \theta_d x_d + (\theta_0)(1)$$
$$= \theta^\top x$$

1x3,3x1

Notational trick: *not* the same θ & x!

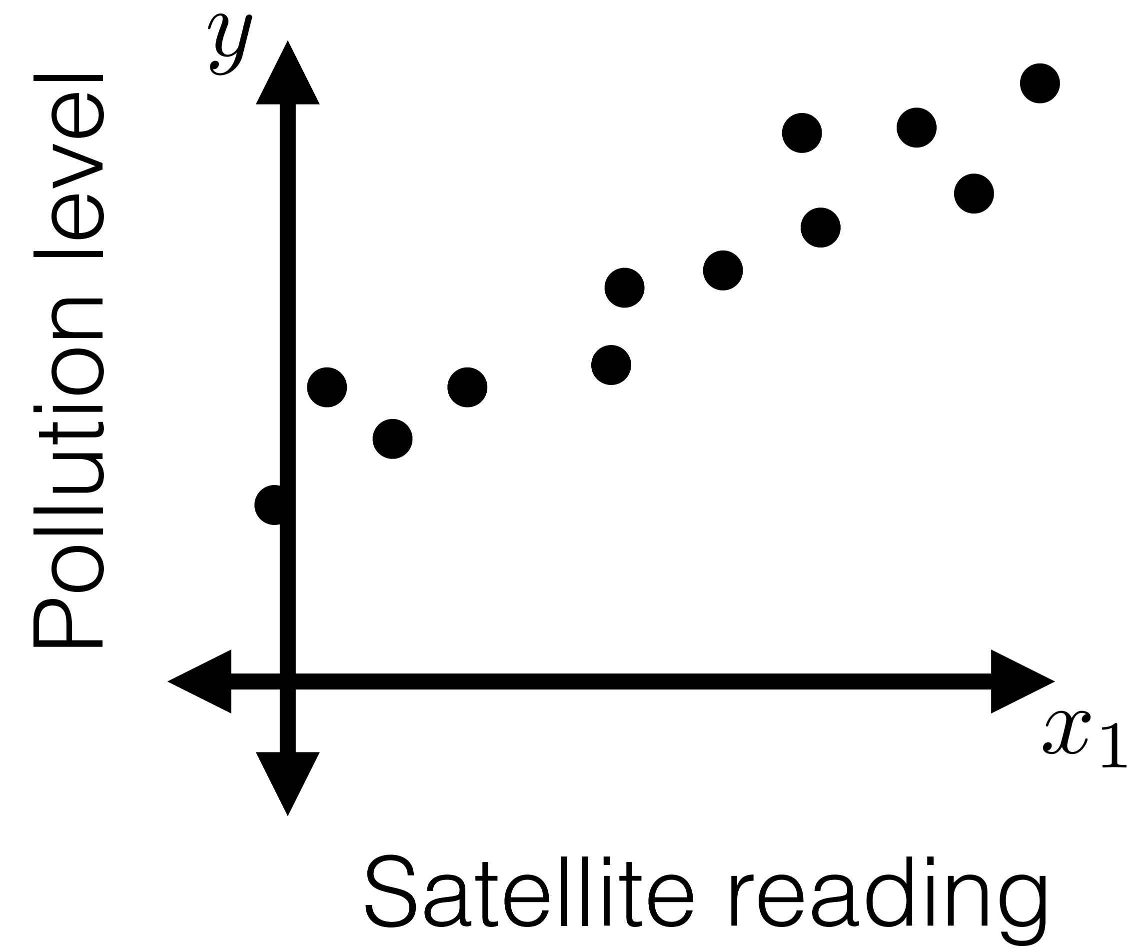
- Our hypothesis class in linear regression will be the set of all such h

How good is a regression hypothesis?



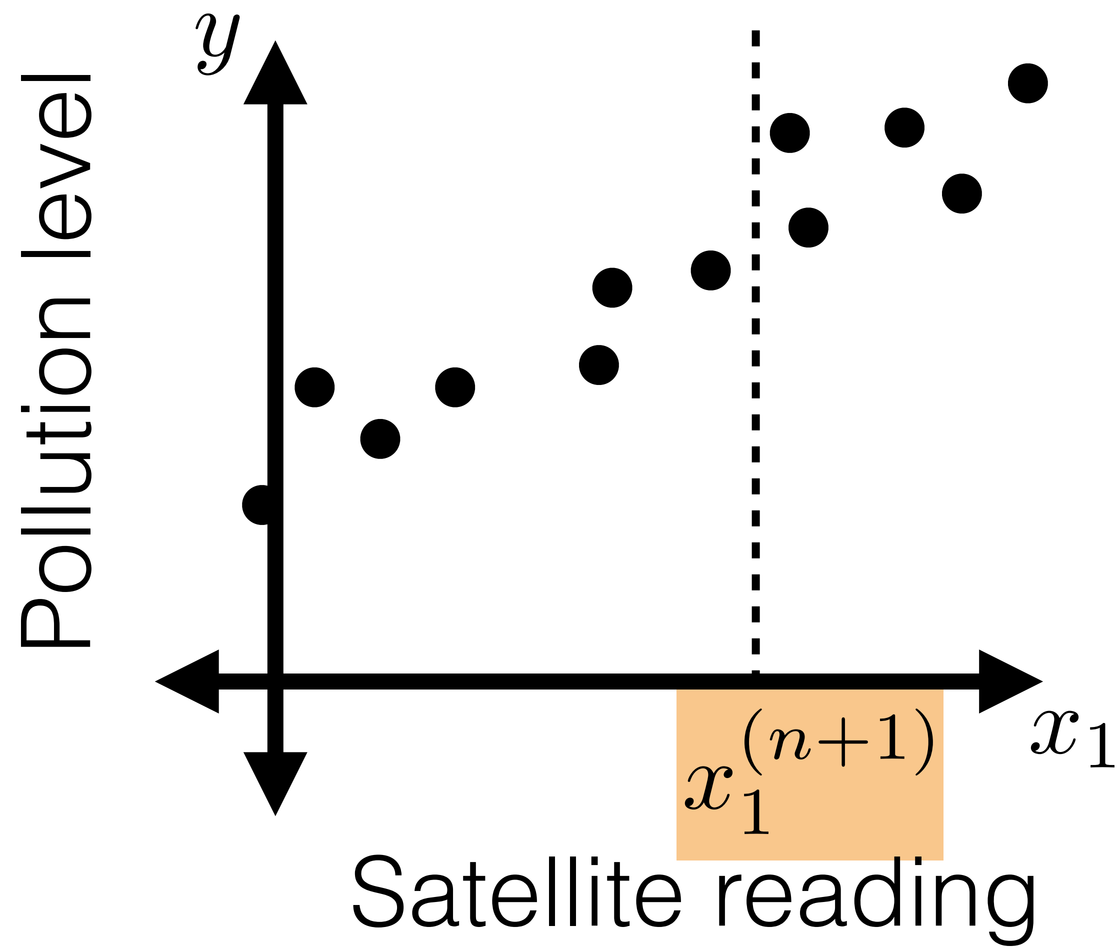
How good is a regression hypothesis?

- Should predict well on future data



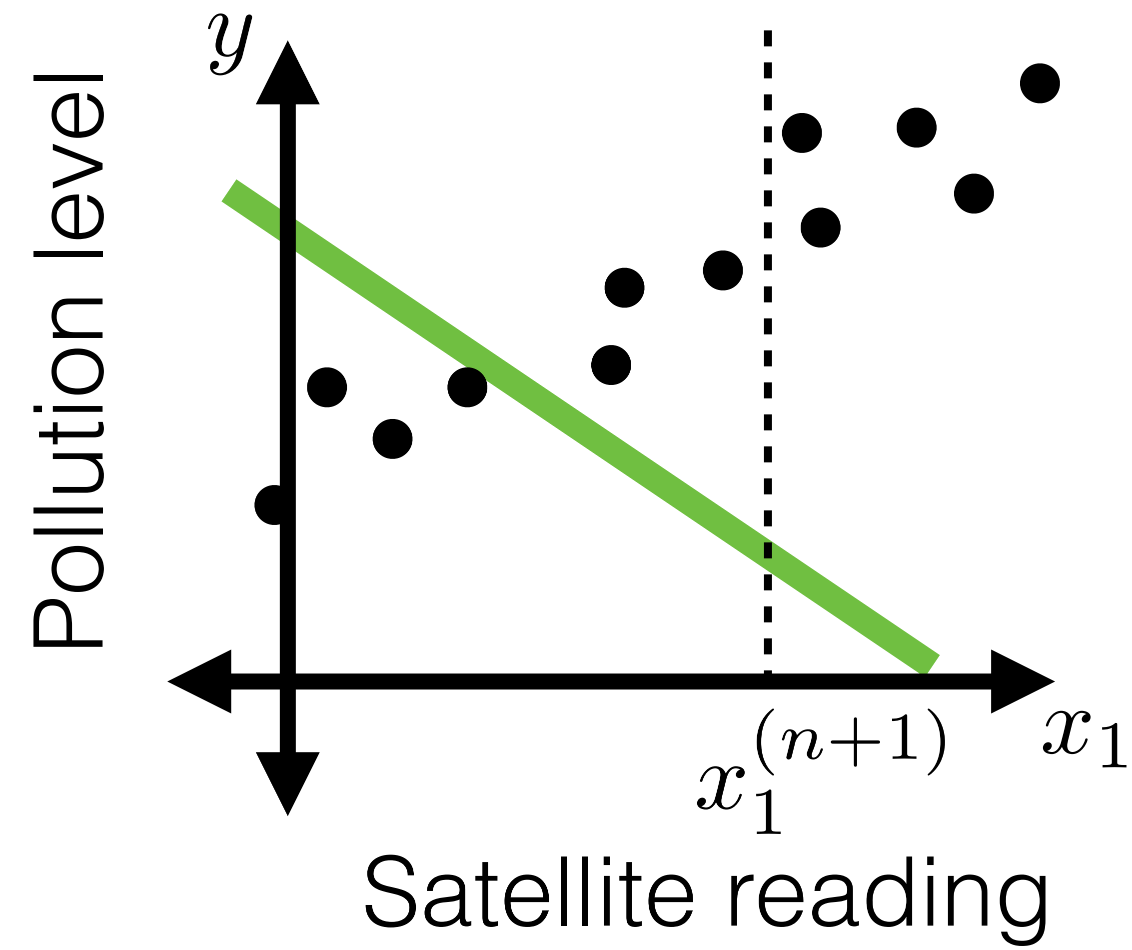
How good is a regression hypothesis?

- Should predict well on future data



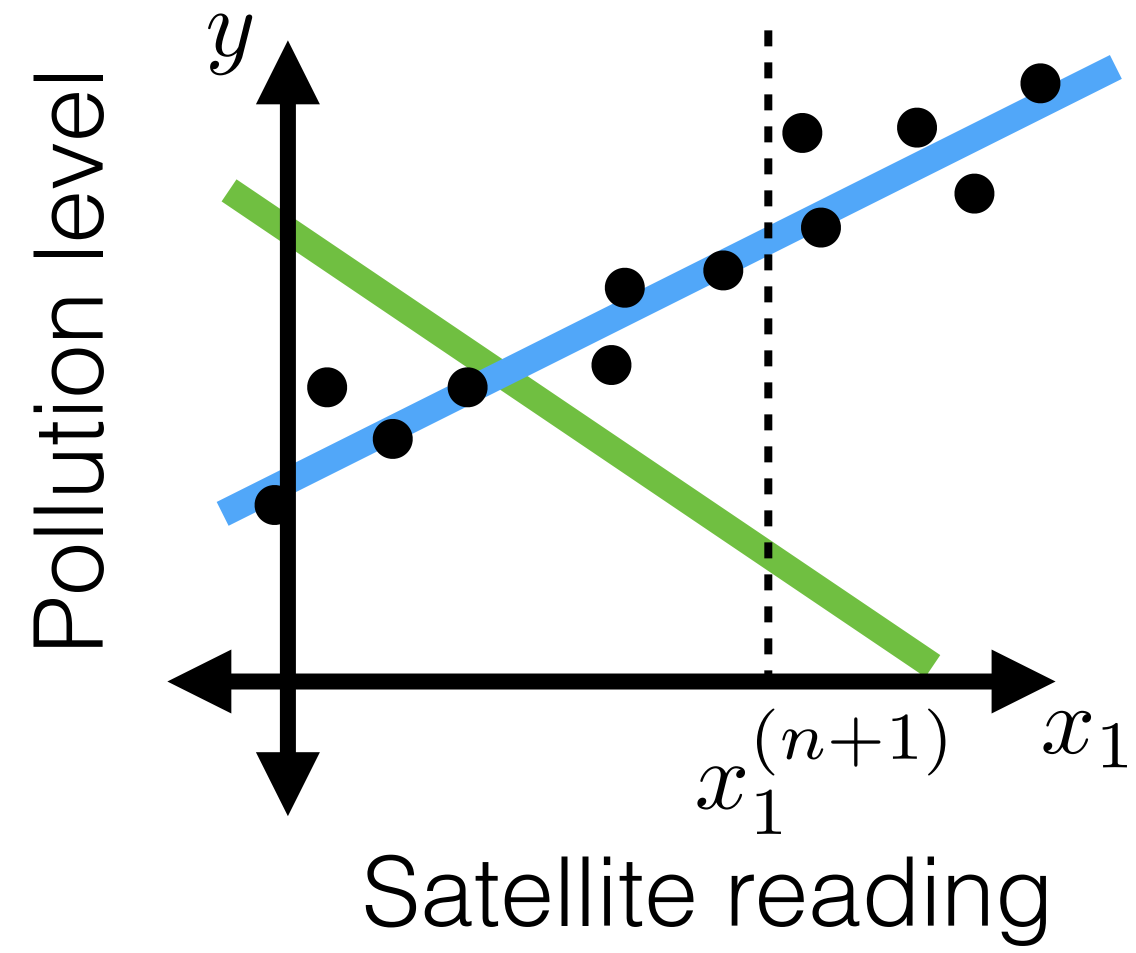
How good is a regression hypothesis?

- Should predict well on future data



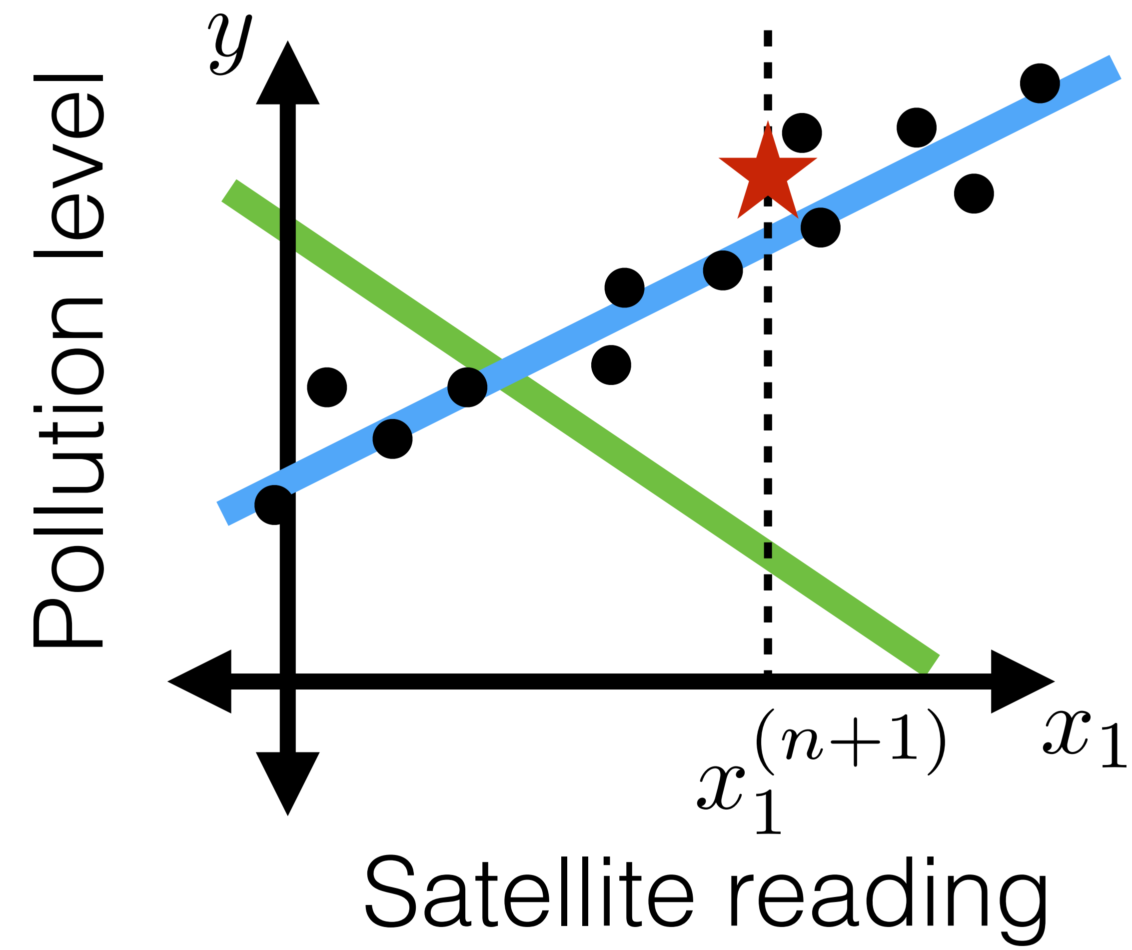
How good is a regression hypothesis?

- Should predict well on future data



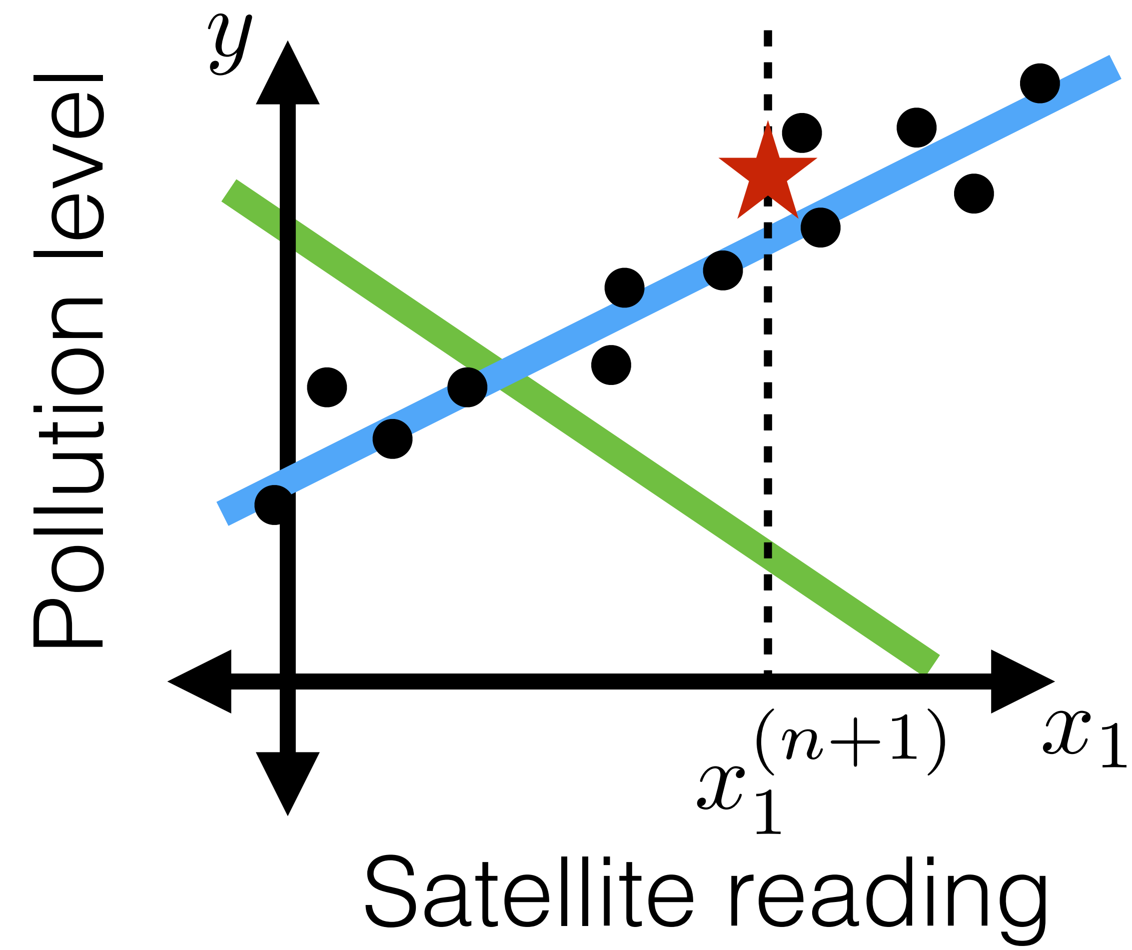
How good is a regression hypothesis?

- Should predict well on future data



How good is a regression hypothesis?

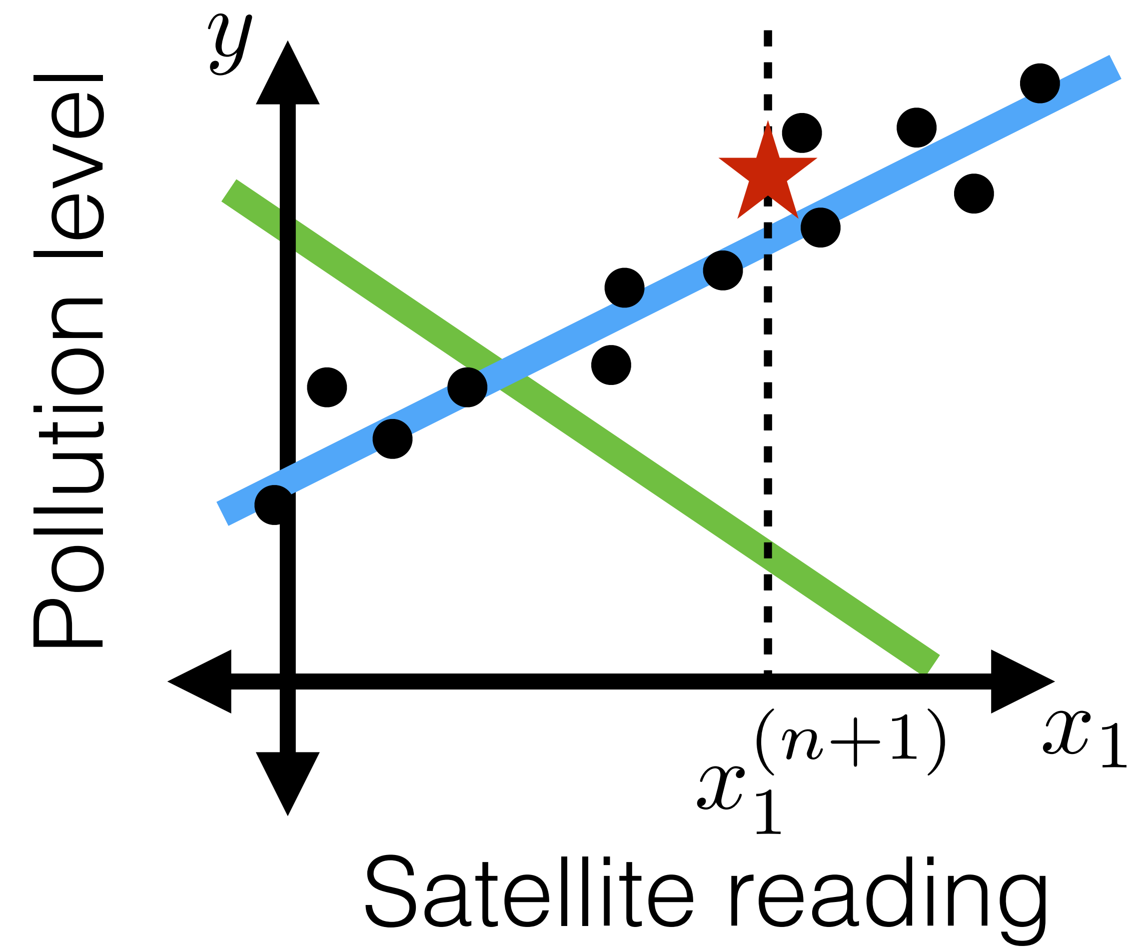
- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$



How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

g: guess,
a: actual



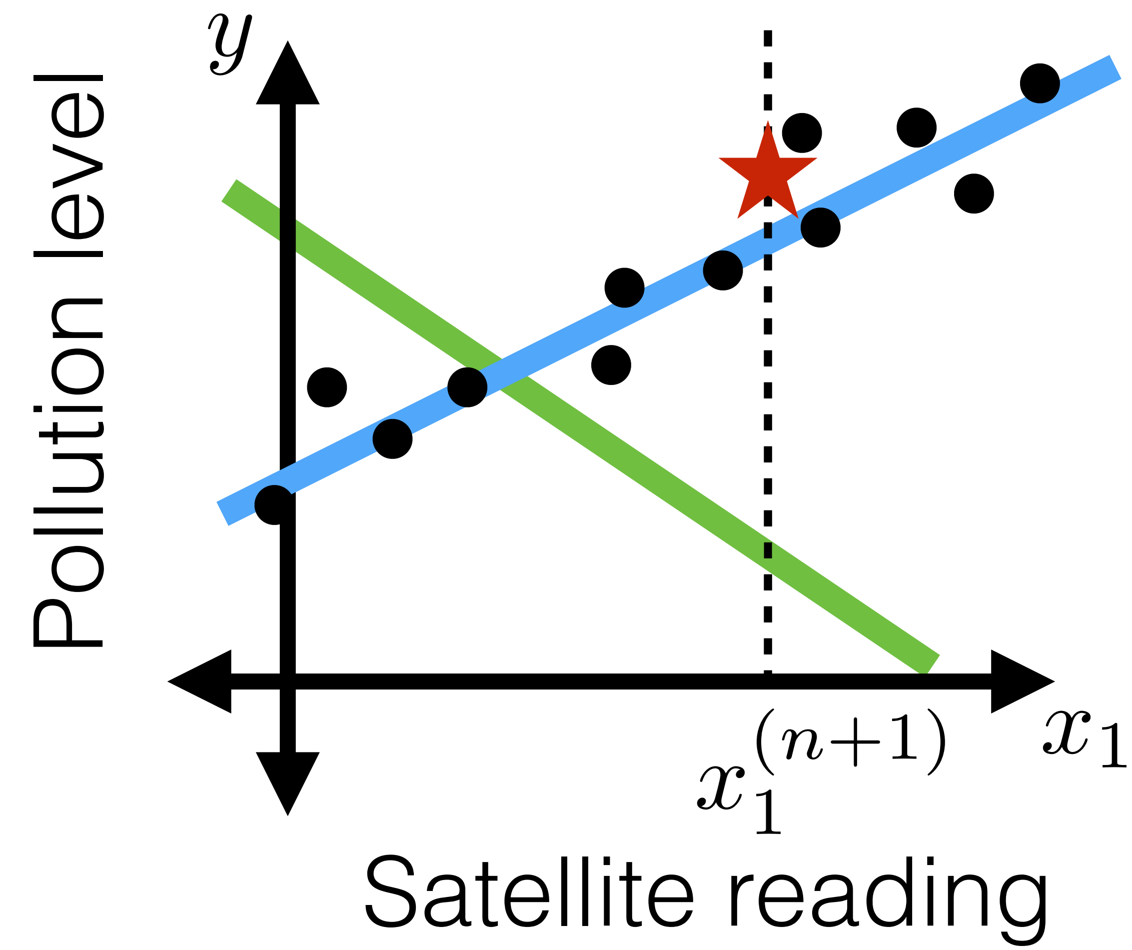
How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

  g: guess,
  a: actual

  - Ex: squared loss

$$L(g, a) = (g - a)^2$$



How good is a regression hypothesis?

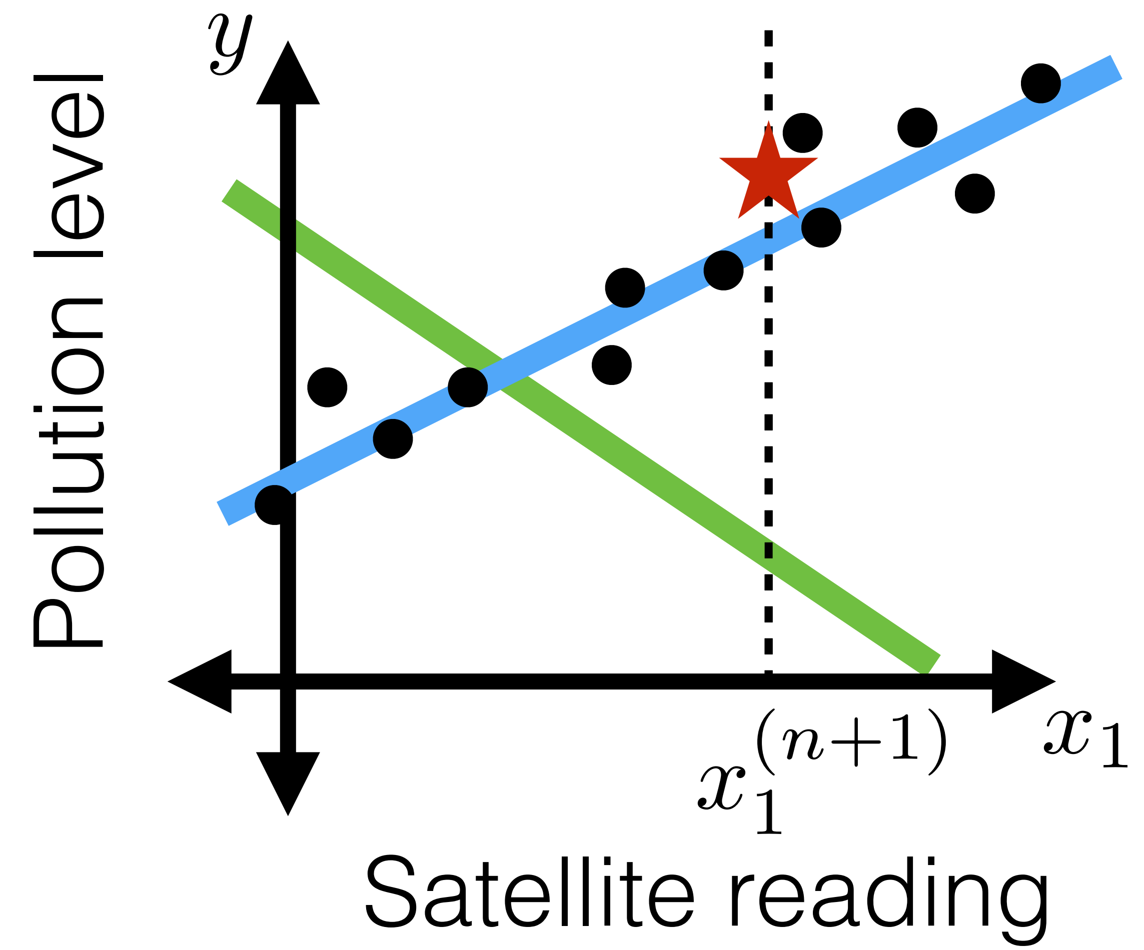
- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

g: guess,
a: actual

  - Ex: squared loss

$$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss



How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

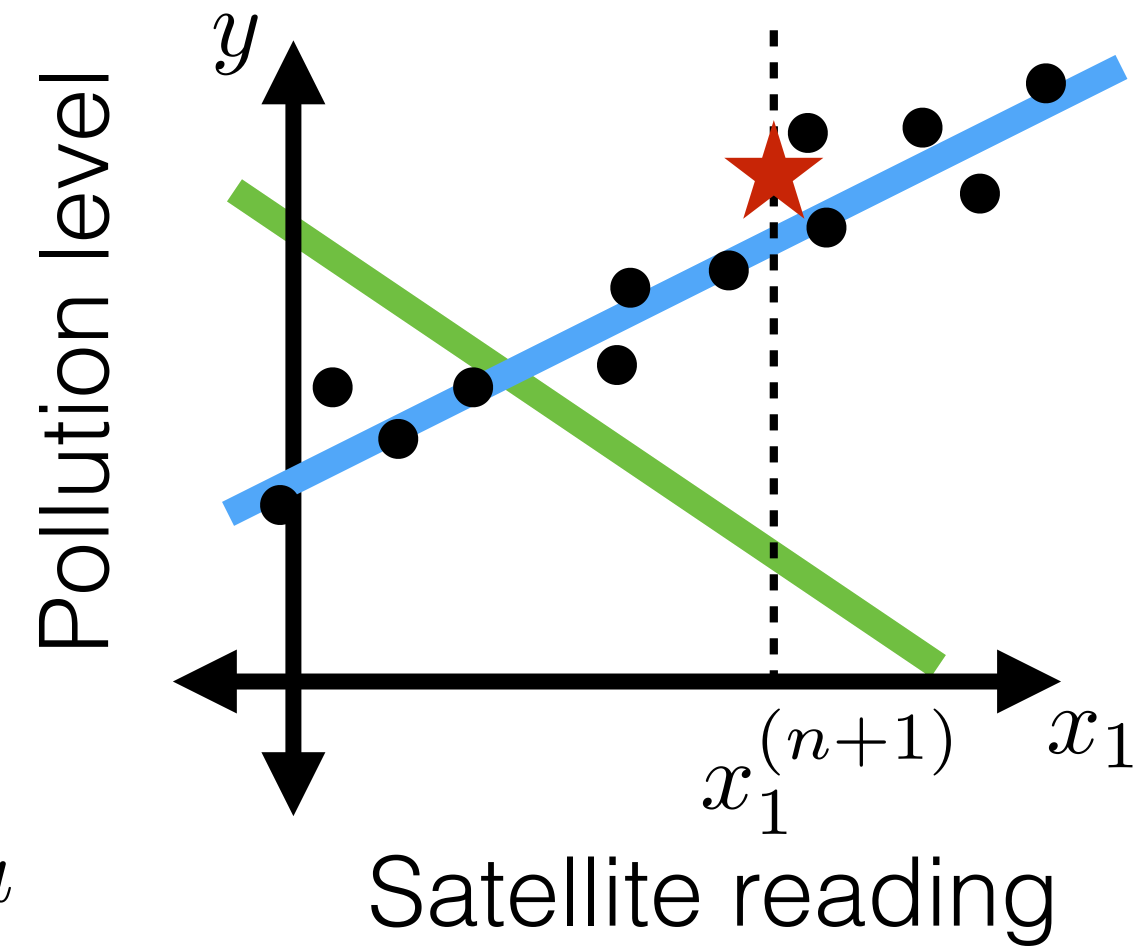
g: guess,
a: actual

  - Ex: squared loss

$$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

$$L(g, a) = \begin{cases} (g-a)^2 \text{ if } g > a \\ 2(g-a)^2 \text{ if } g \leq a \end{cases}$$



How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

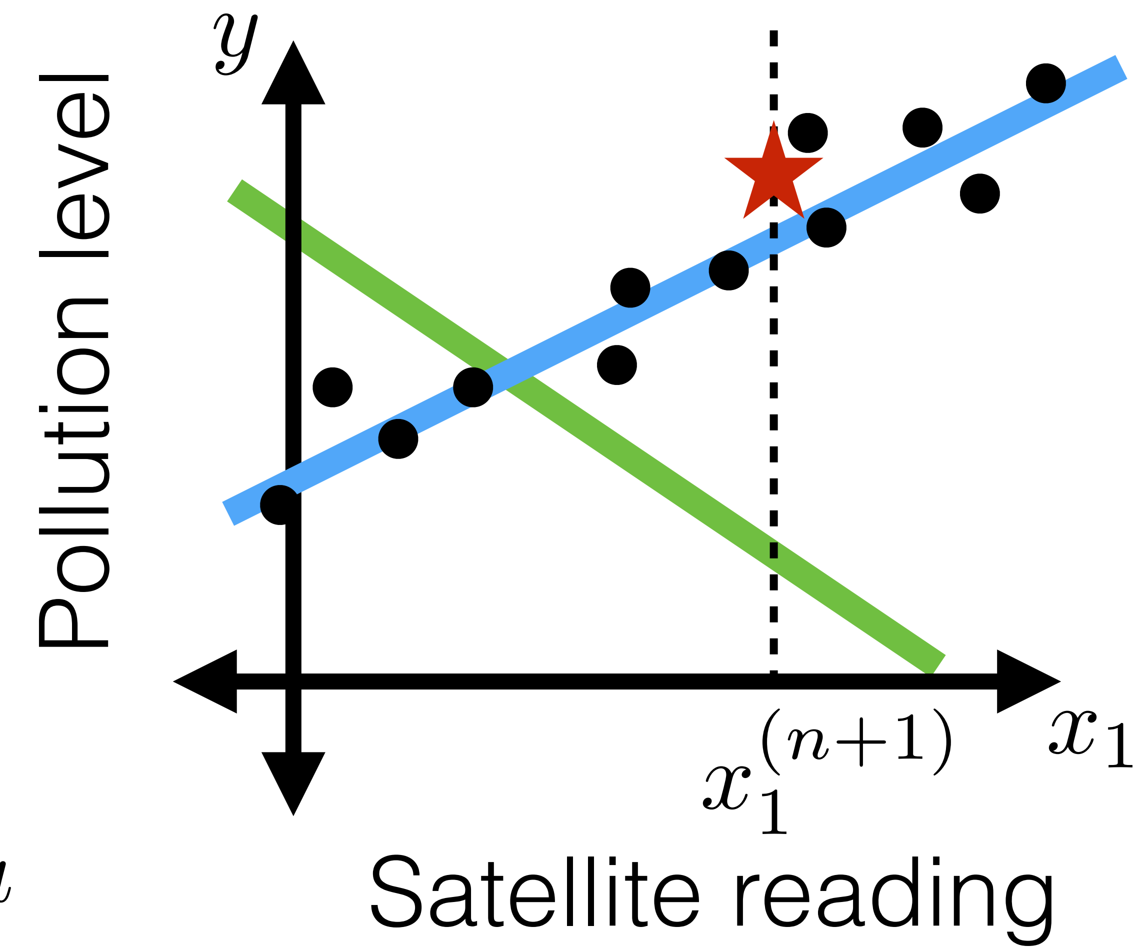
g: guess,
a: actual

  - Ex: squared loss

    $$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

    $$L(g, a) = \begin{cases} (g - a)^2 \text{ if } g > a \\ 2(g - a)^2 \text{ if } g \le a \end{cases}$$



- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$

How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

  g: guess,
  a: actual

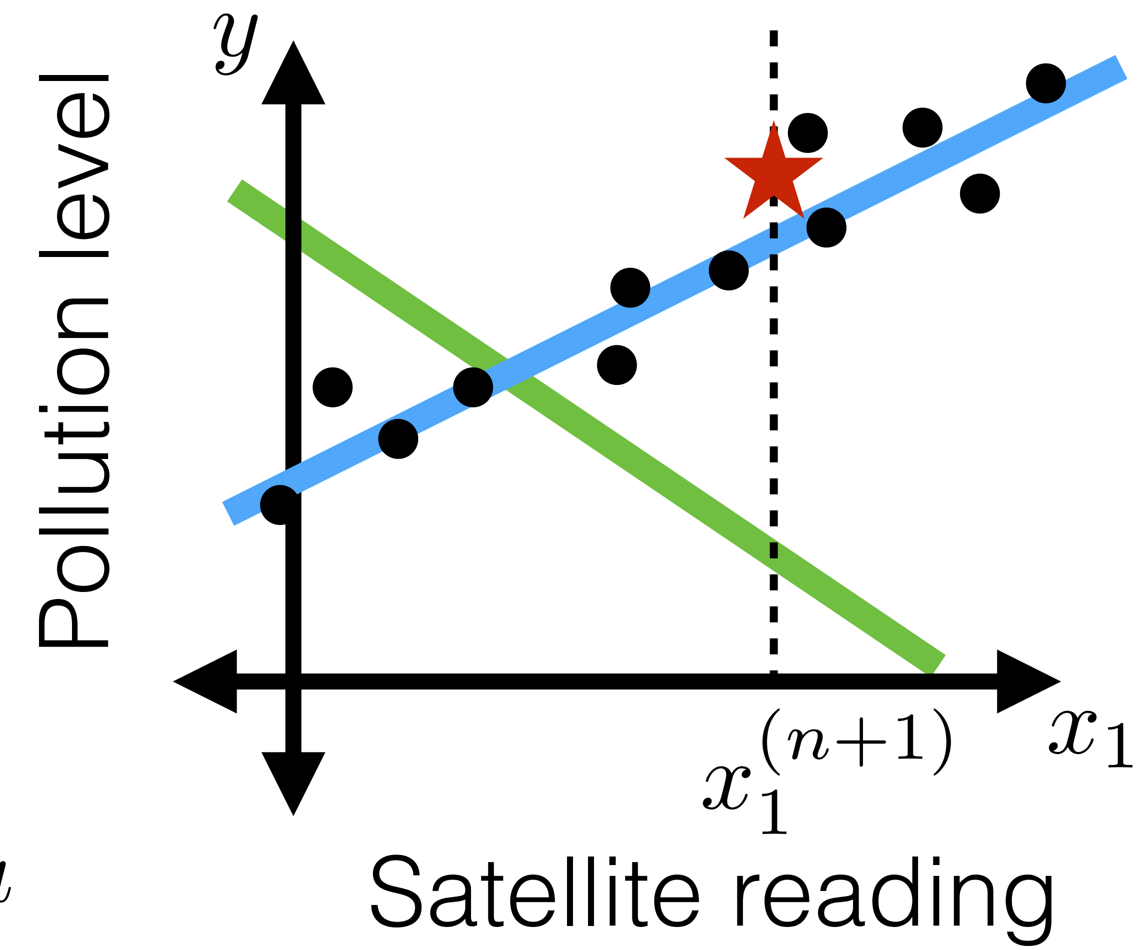
  - Ex: squared loss

    $$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

    $$L(g, a) = \begin{cases} (g-a)^2 \text{ if } g > a \\ 2(g-a)^2 \text{ if } g \leq a \end{cases}$$

- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$



How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

  g: guess,
  a: actual

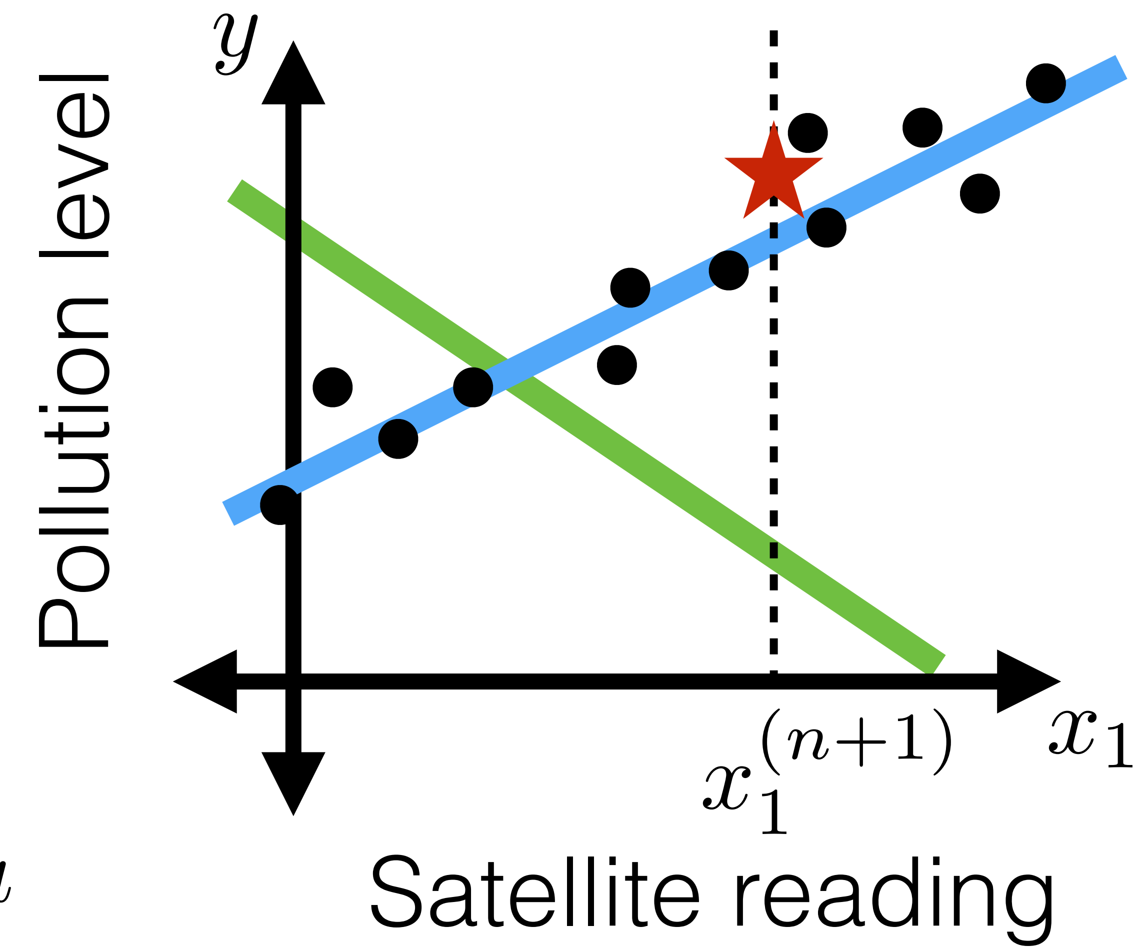
  - Ex: squared loss

    $$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

    $$L(g, a) = \begin{cases} (g - a)^2 \text{ if } g > a \\ 2(g - a)^2 \text{ if } g \leq a \end{cases}$$

- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$



How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

  g: guess,
  a: actual

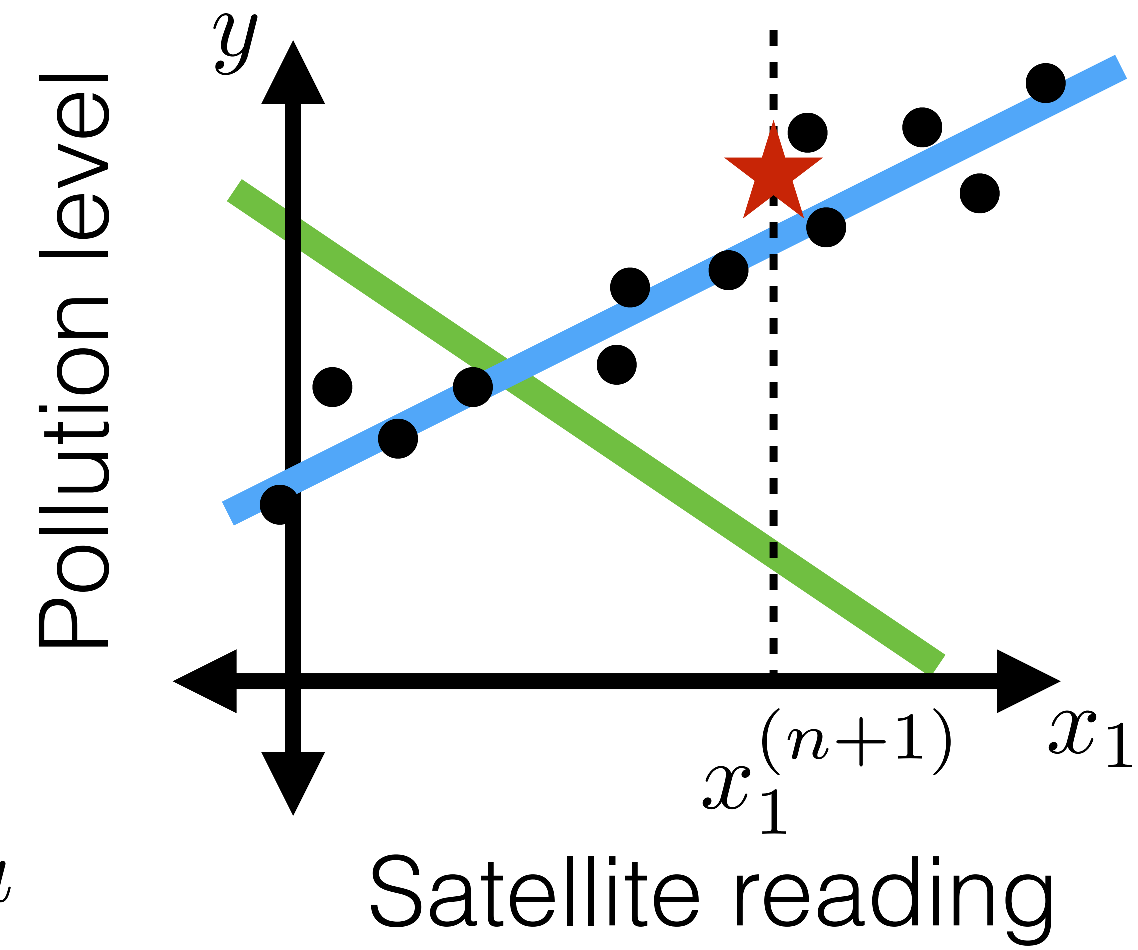
  - Ex: squared loss

    $$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

    $$L(g, a) = \begin{cases} (g-a)^2 \text{ if } g > a \\ 2(g-a)^2 \text{ if } g \le a \end{cases}$$

- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$



How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

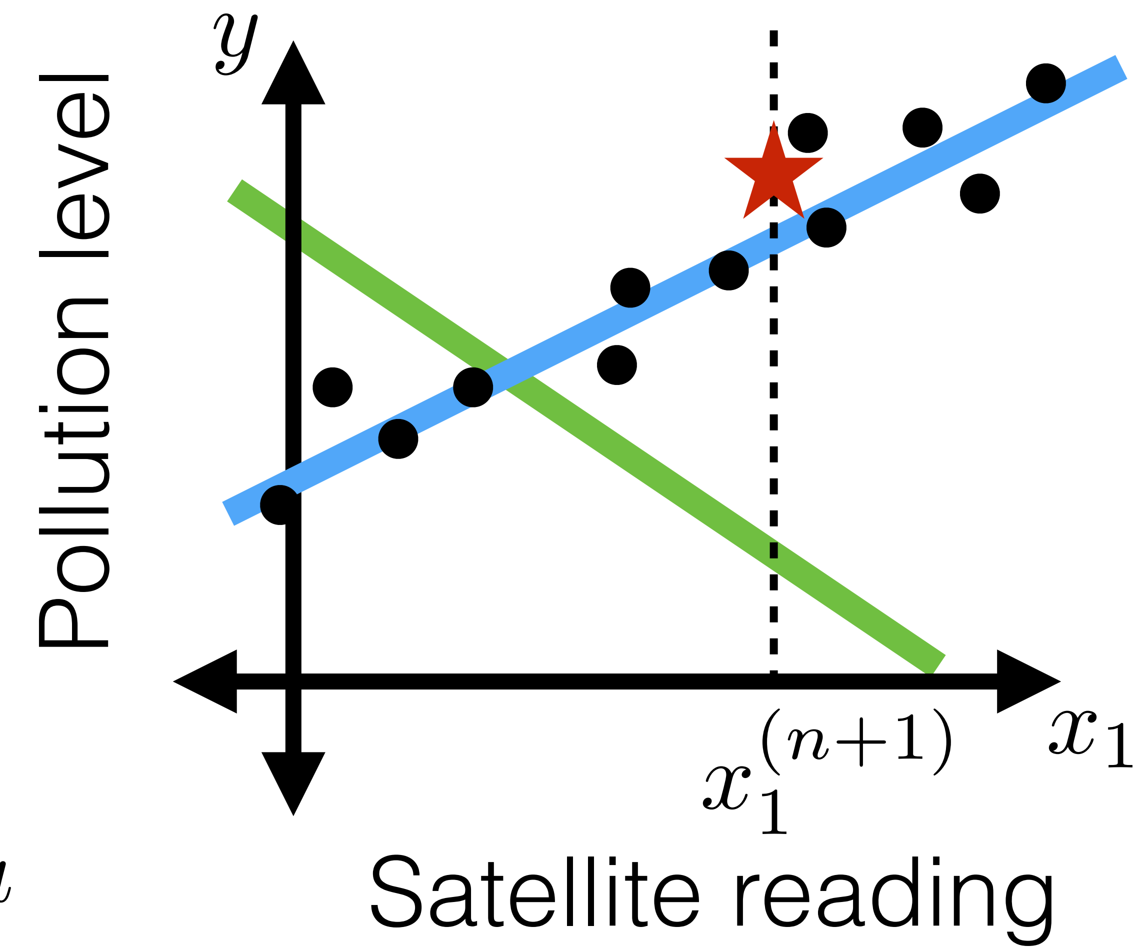
  g: guess,
  a: actual

  - Ex: squared loss

    $$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

    $$L(g, a) = \begin{cases} (g - a)^2 \text{ if } g > a \\ 2(g - a)^2 \text{ if } g \leq a \end{cases}$$



- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$

How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

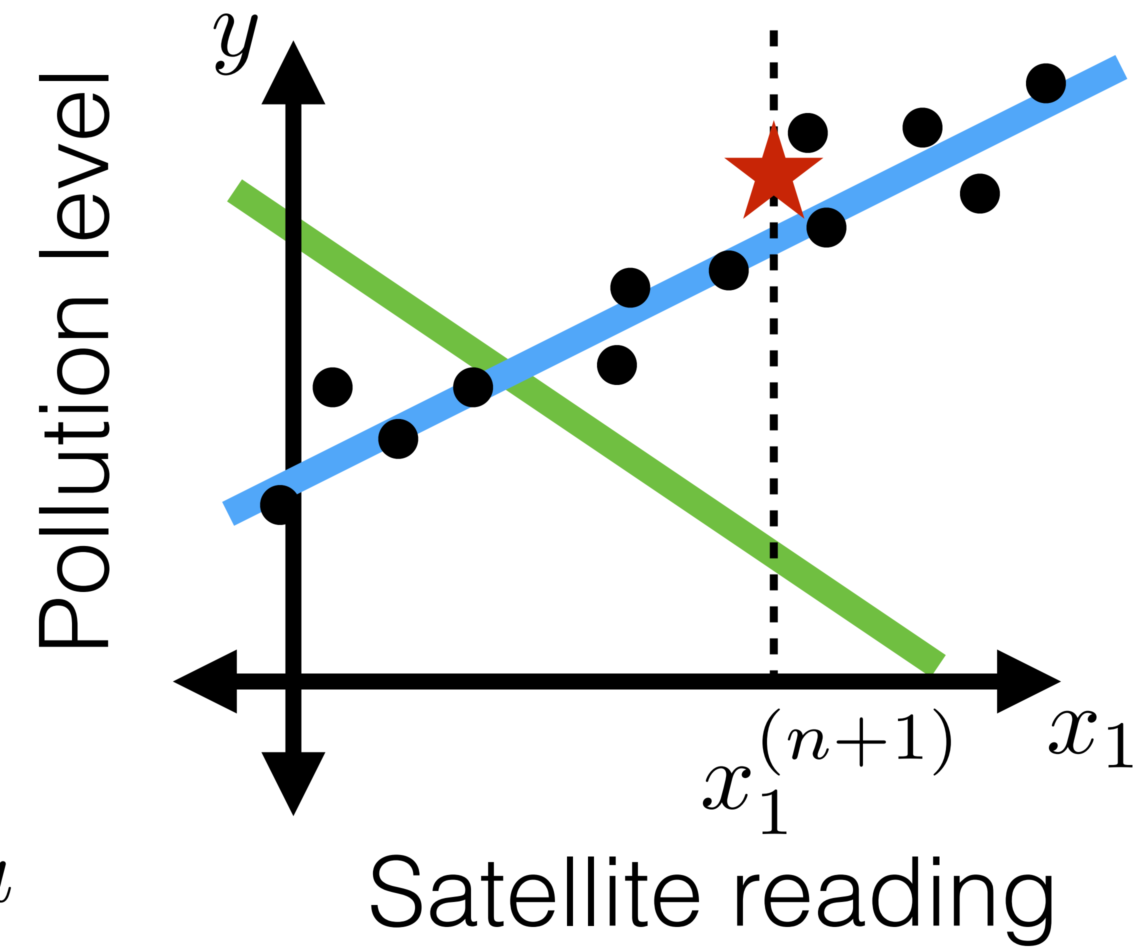
  g: guess,
  a: actual

  - Ex: squared loss

    $$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

    $$L(g, a) = \begin{cases} (g-a)^2 \text{ if } g > a \\ 2(g-a)^2 \text{ if } g \leq a \end{cases}$$



- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$

How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

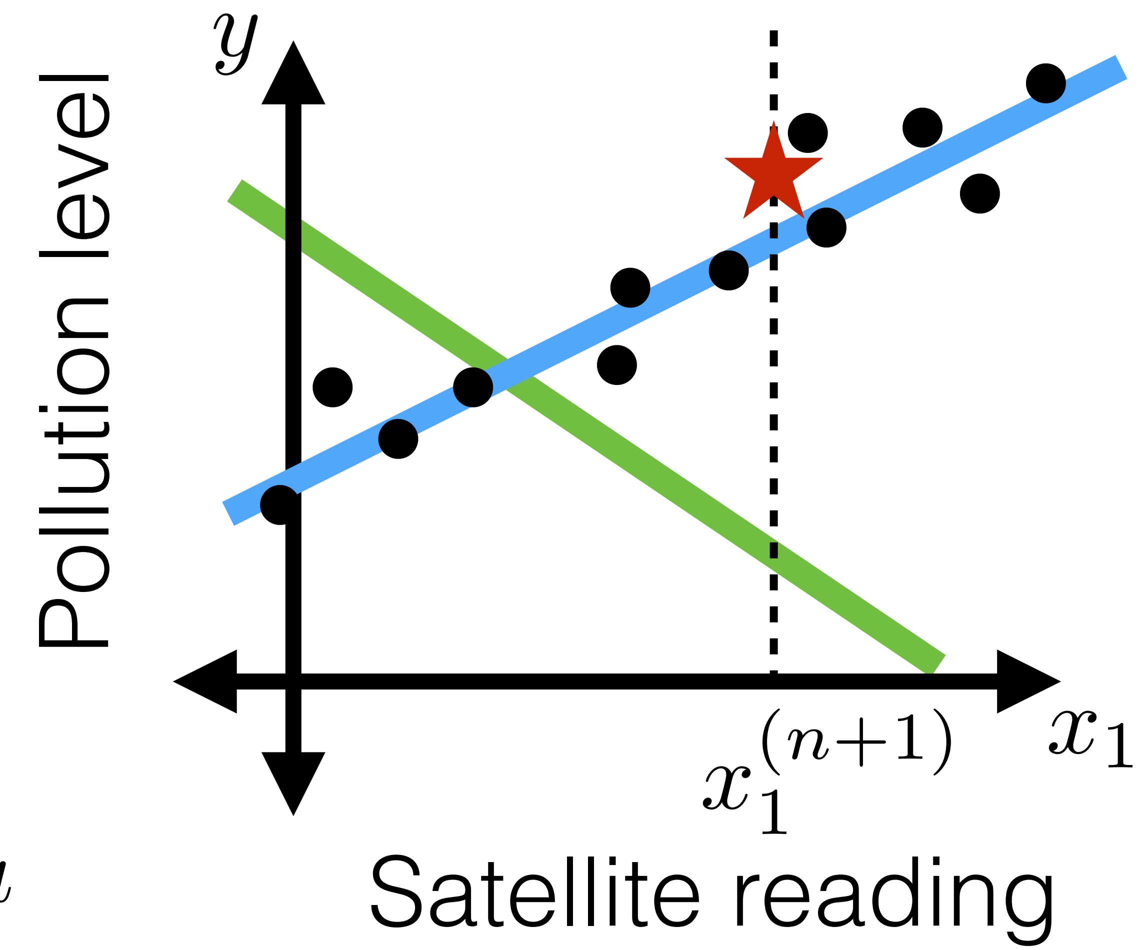
g: guess,
a: actual

  - Ex: squared loss

$$L(g, a) = (g - a)^2$$

  - Example: asymmetric loss

$$L(g, a) = \begin{cases} (g-a)^2 \text{ if } g > a \\ 2(g-a)^2 \text{ if } g \leq a \end{cases}$$



- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$
- Training error: $\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$

How good is a regression hypothesis?

- Should predict well on future data
- How good is a regressor at one point? Loss $L(g, a)$

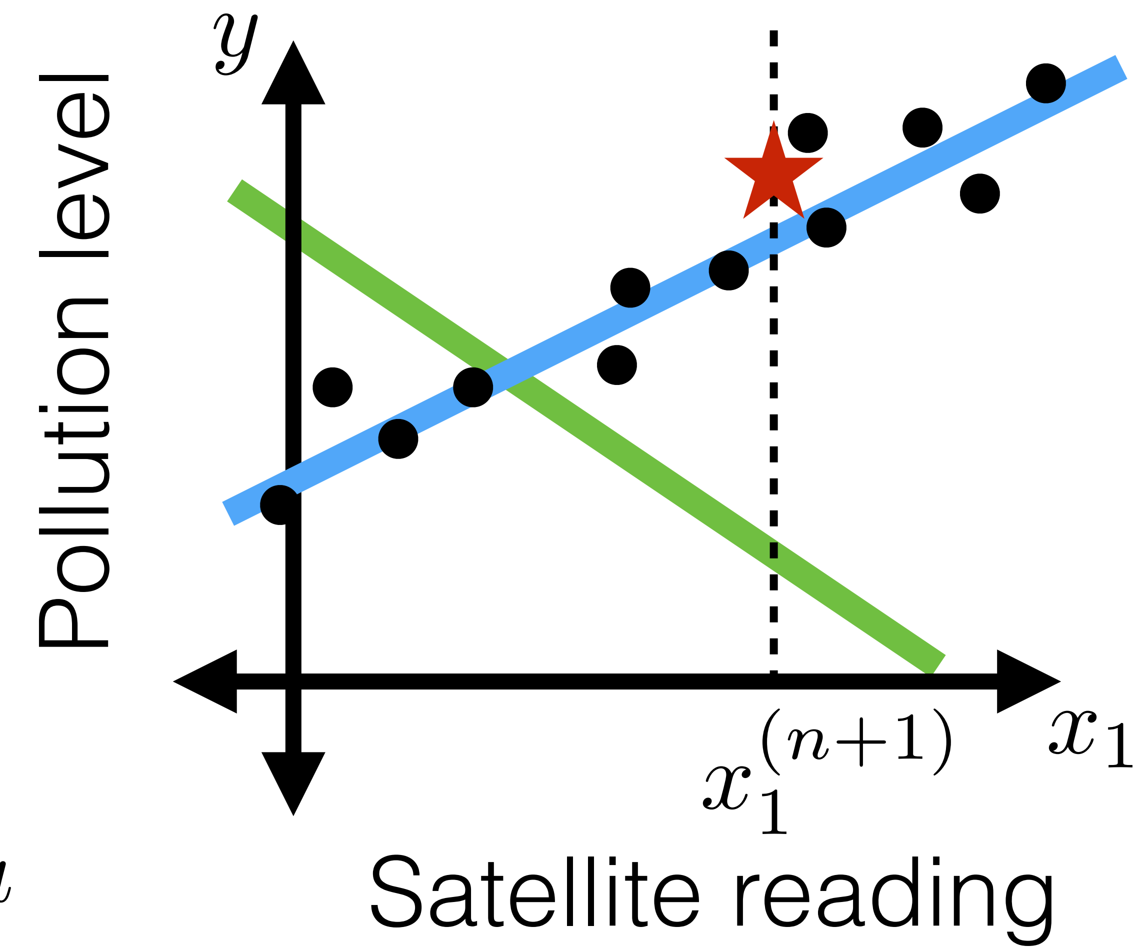
g: guess,
a: actual

  - Ex: squared loss

$$L(g, a) = (g - a)^2$$

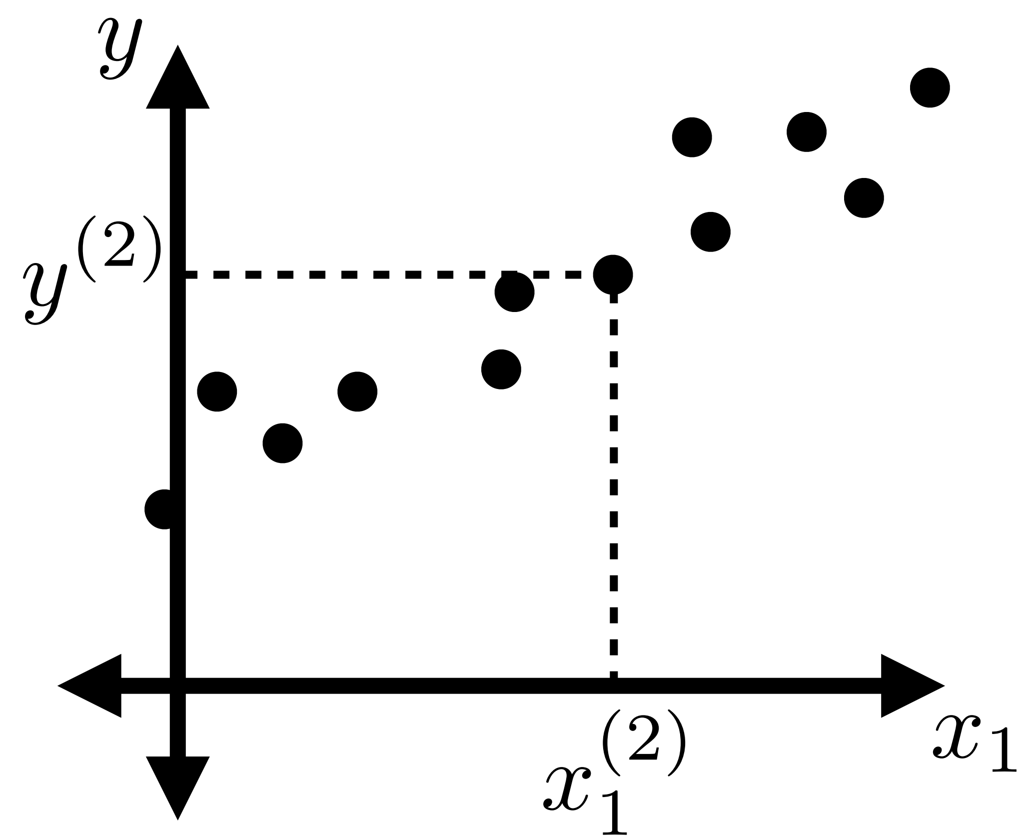
  - Example: asymmetric loss

$$L(g, a) = \begin{cases} (g-a)^2 \text{ if } g > a \\ 2(g-a)^2 \text{ if } g \leq a \end{cases}$$



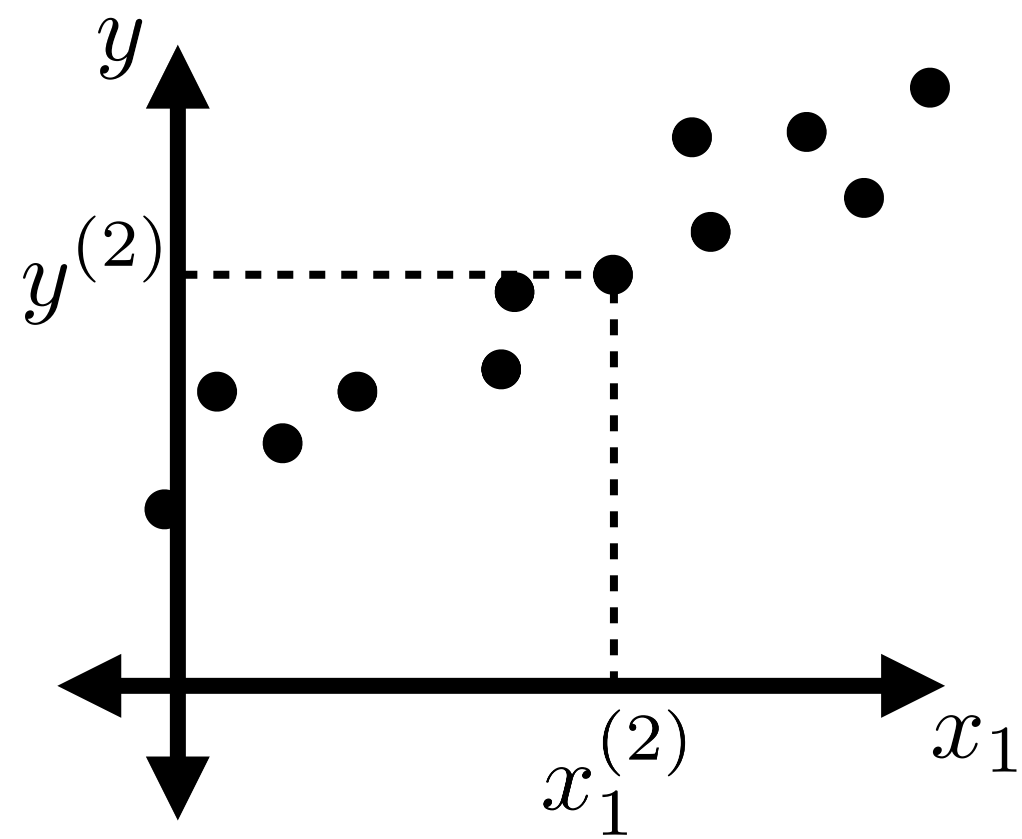
- Test error (n' new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$
- Training error: $\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- One idea: prefer h to $\tilde{h}$ if $\mathcal{E}_n(h) < \mathcal{E}_n(\tilde{h})$

Learning a regressor



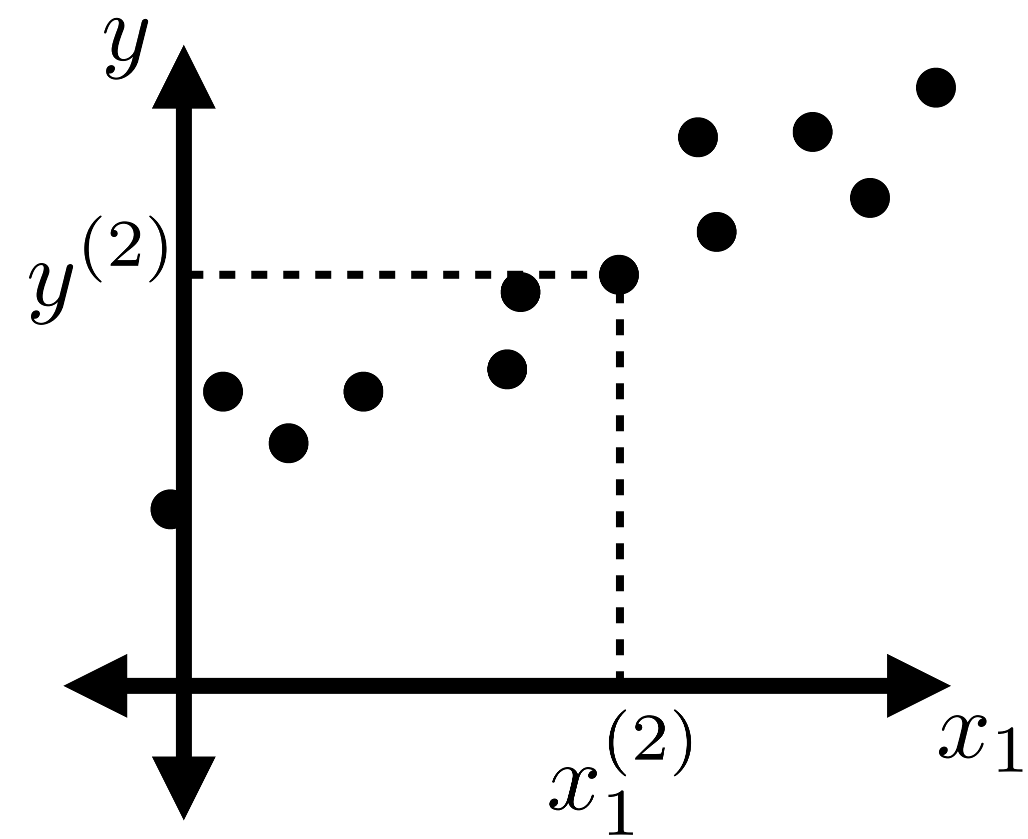
Learning a regressor

- Have data; have hypothesis class

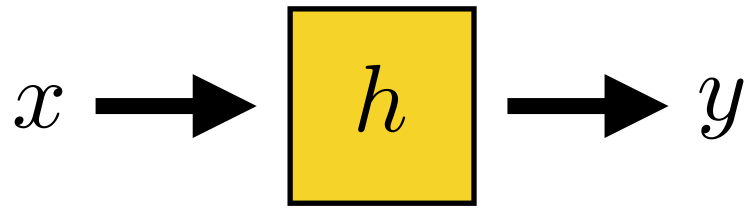


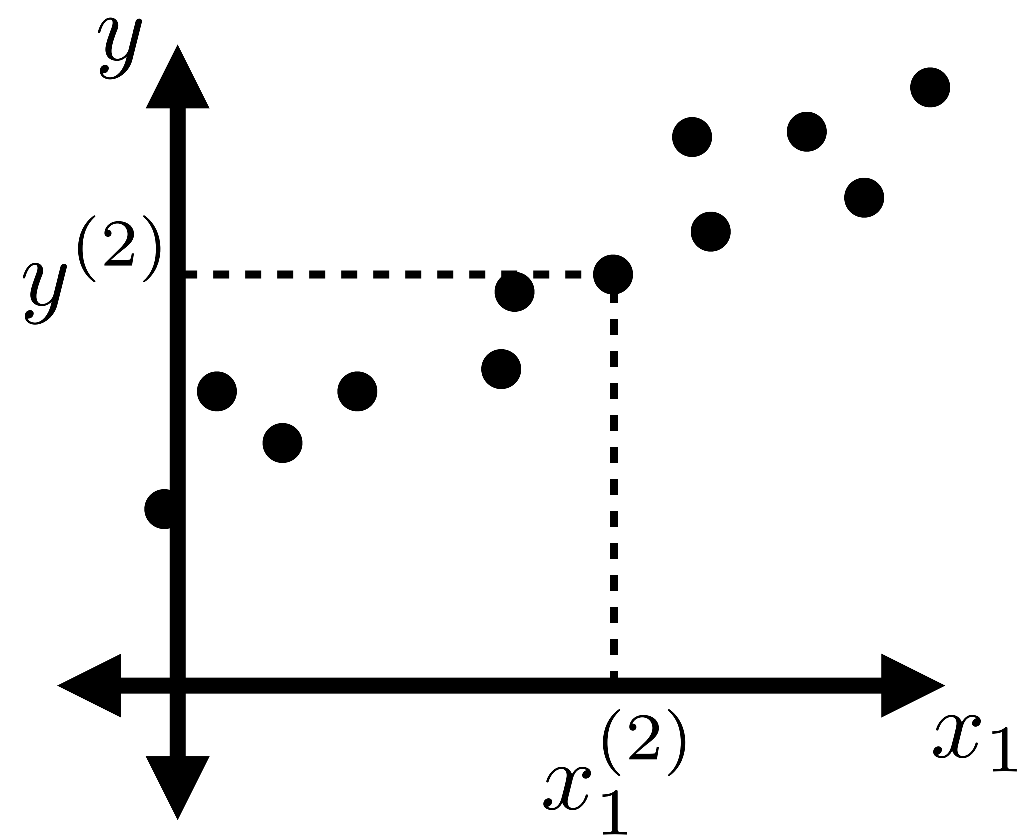
Learning a regressor

- Have data; have hypothesis class
- Want to choose a good regressor



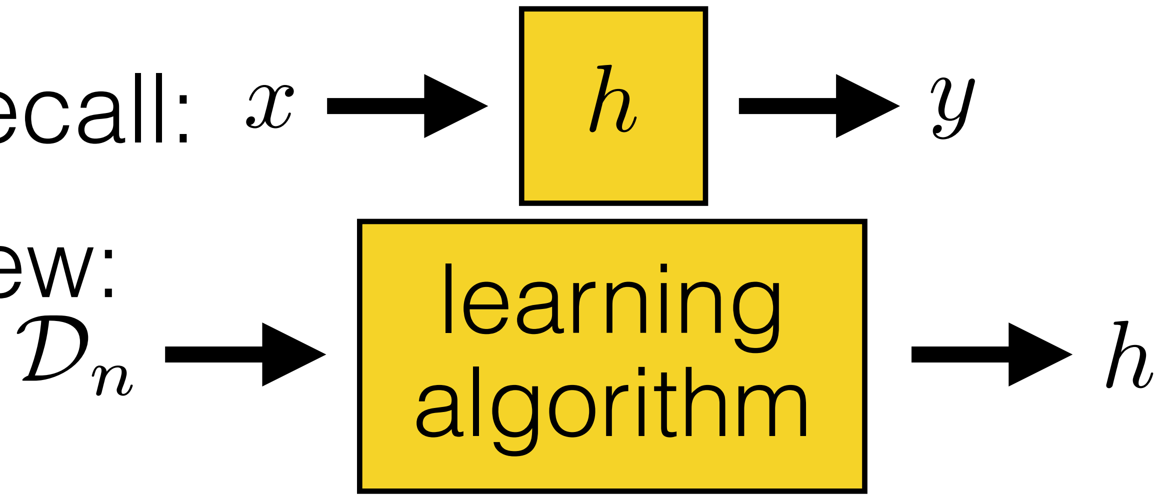
Learning a regressor

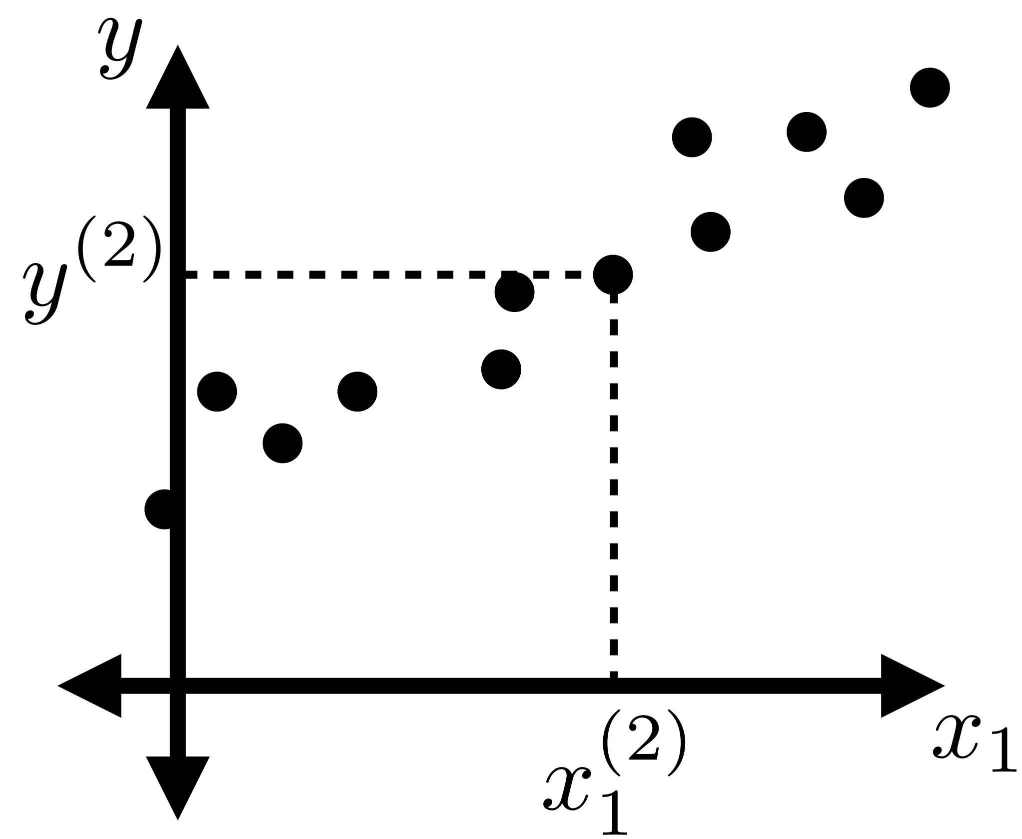
- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall: 



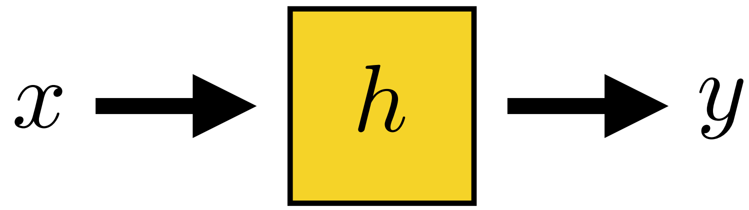
Learning a regressor

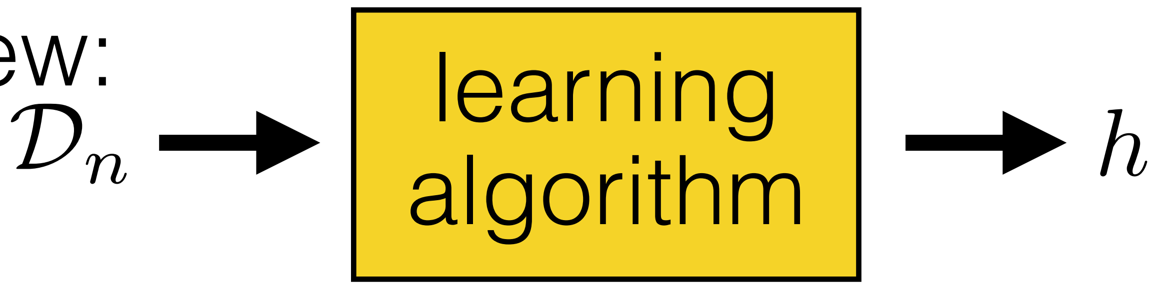
- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall:
  - New:



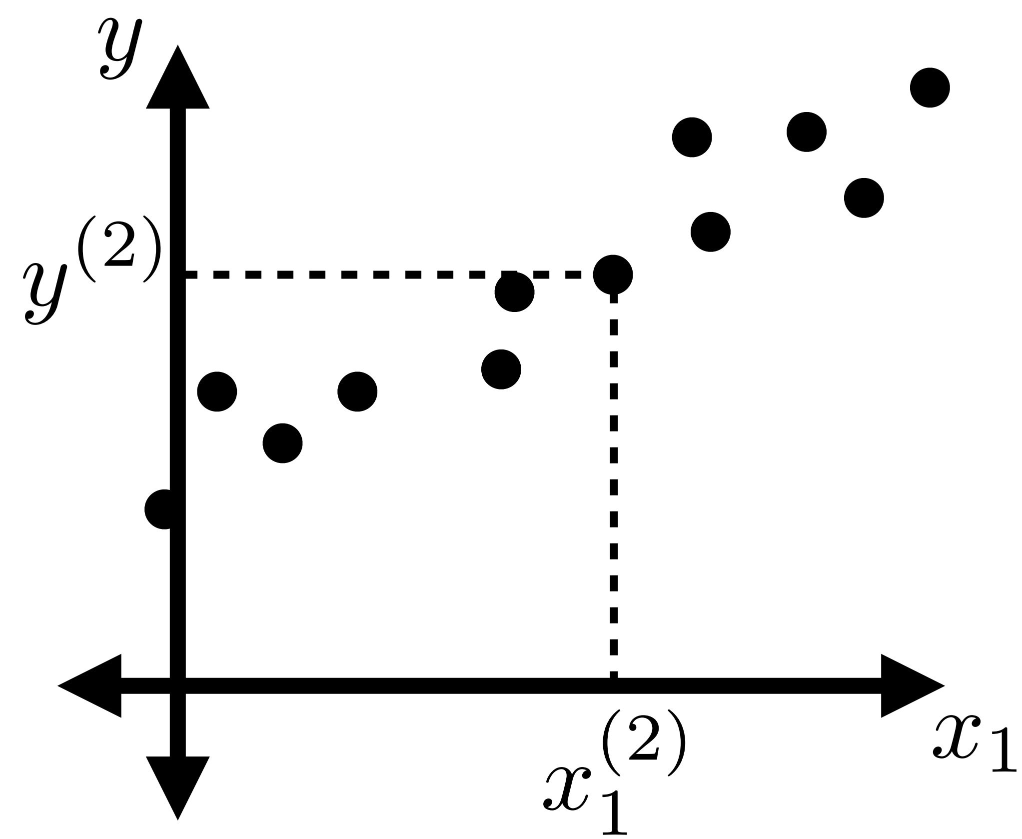


Learning a regressor

- Have data; have hypothesis class
- Want to choose a good regressor
    - Recall: 

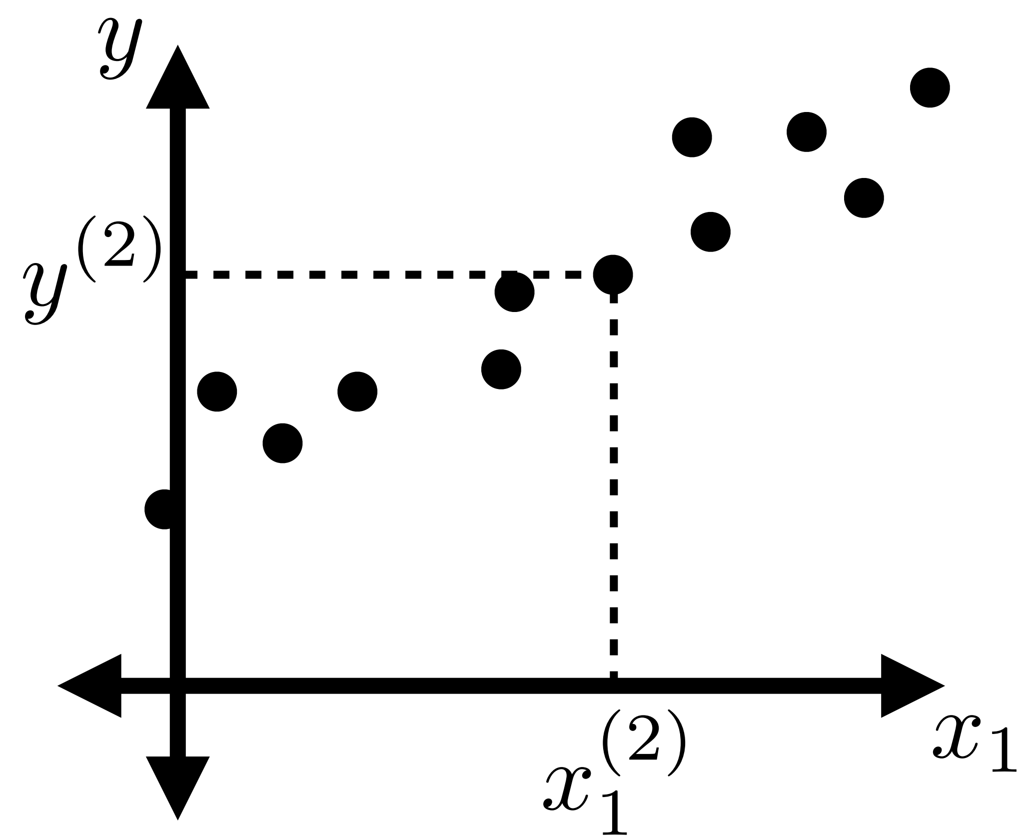
    - New: 

- Example:



Learning a regressor

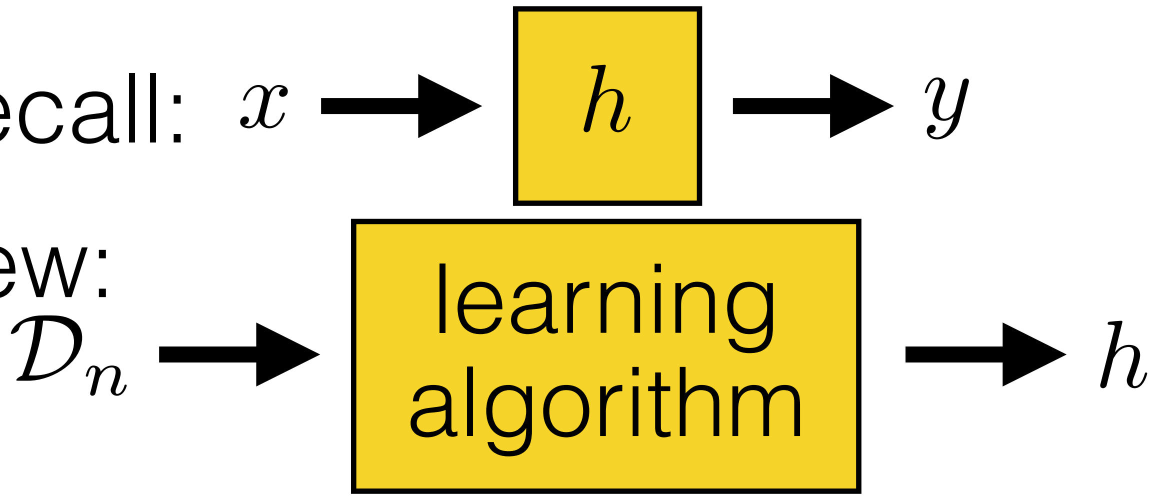
- Have data; have hypothesis class
- Want to choose a good regressor
    - Recall: $x \rightarrow h \rightarrow y$
    - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$

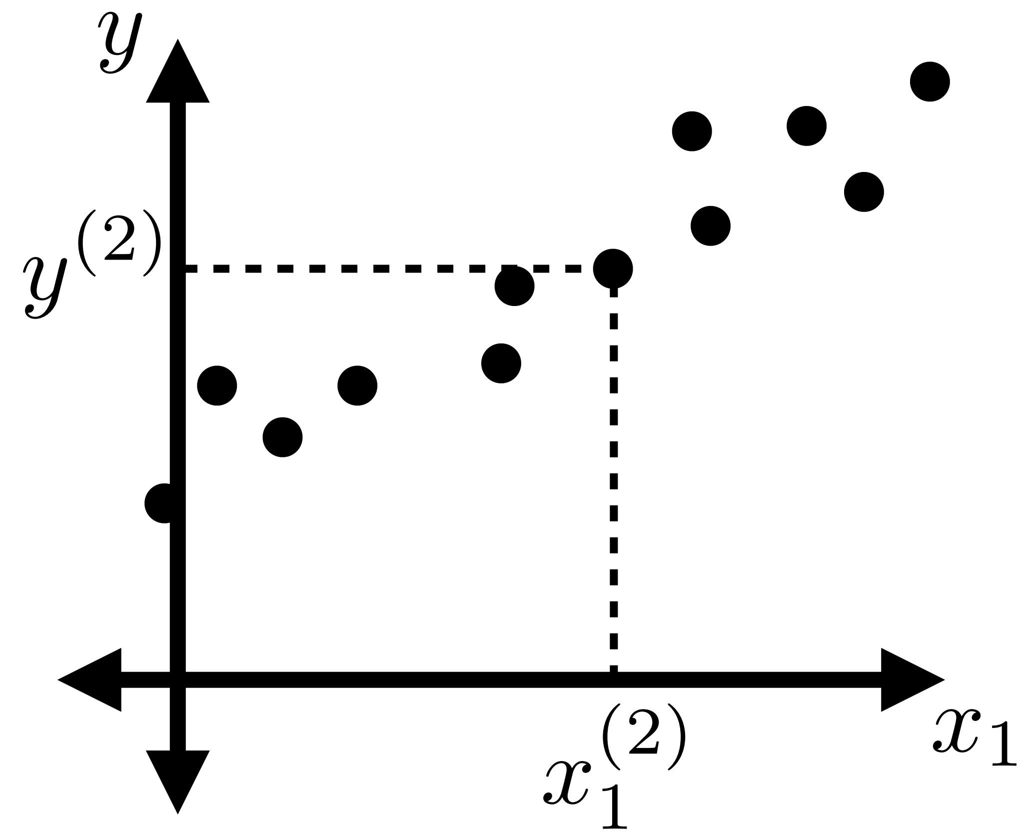


- Example:
    - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

Learning a regressor

- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall:
  - New:



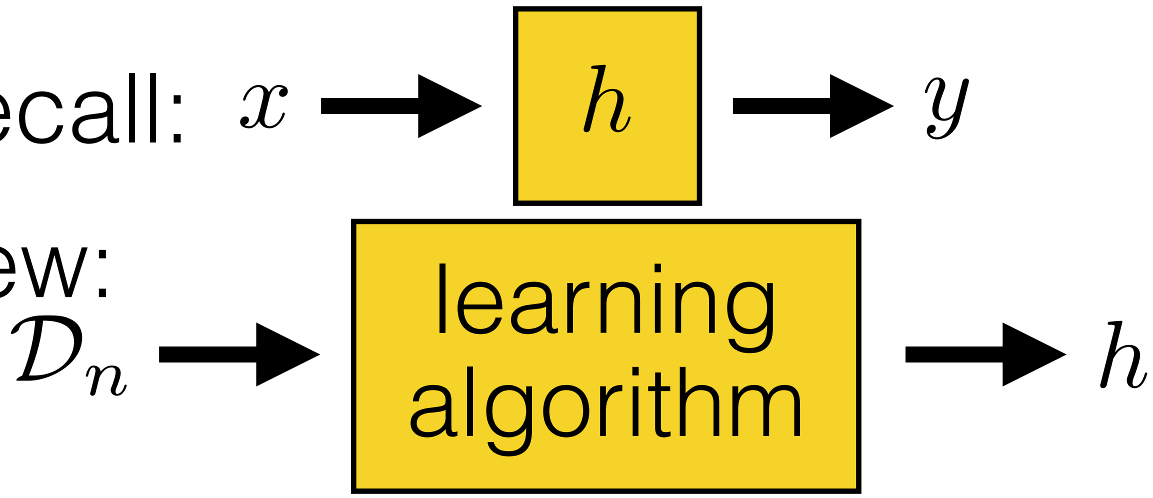


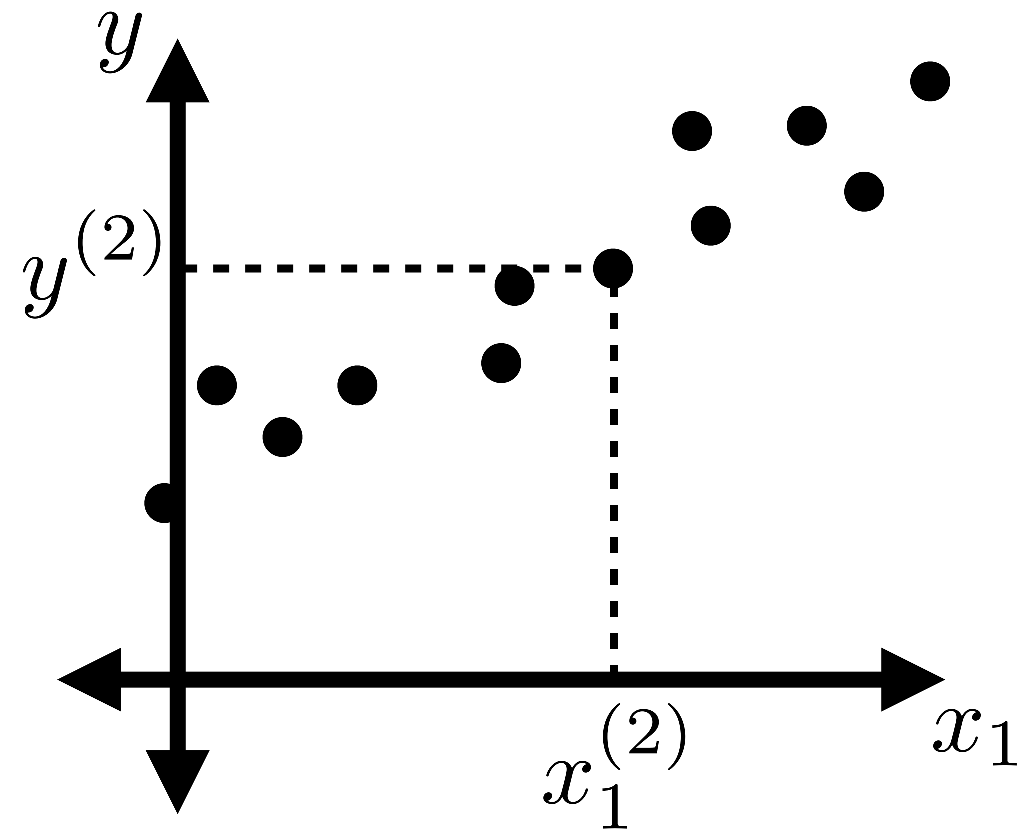
- Example:
  - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

$$h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$$

Learning a regressor

- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall:
  - New:



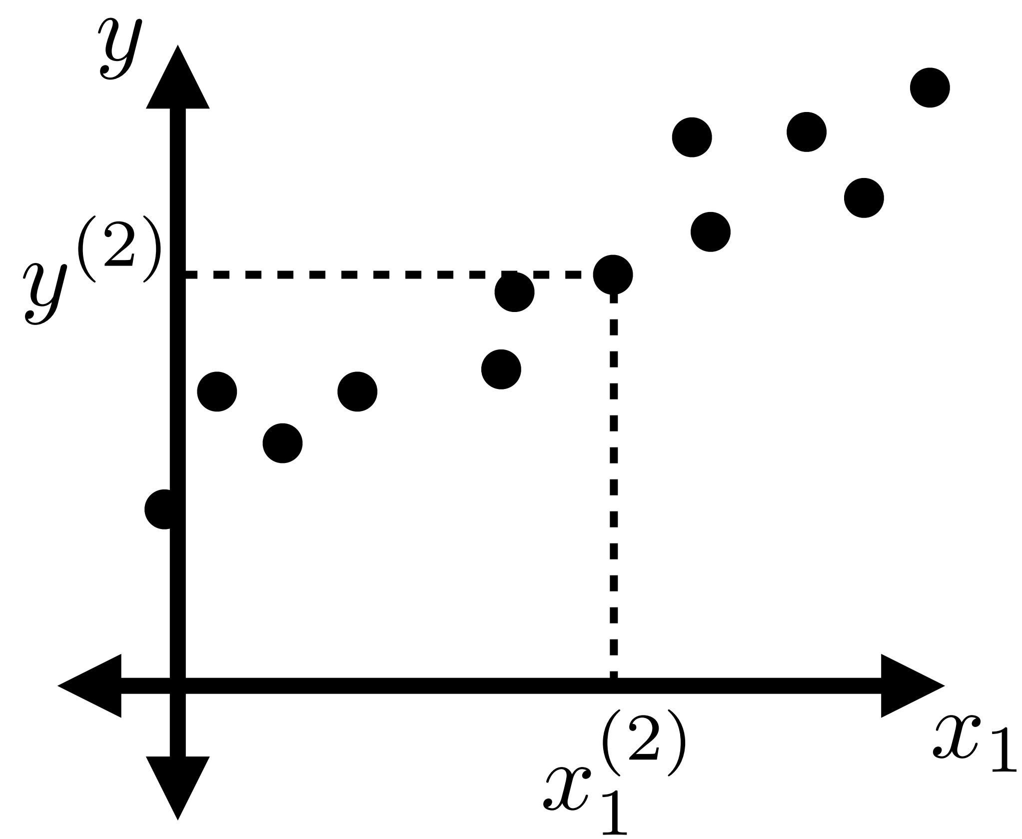


- Example:
  - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

$$h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$$

Learning a regressor

- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall: $x \rightarrow$ **h** $\rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ **learning algorithm** $\rightarrow h$

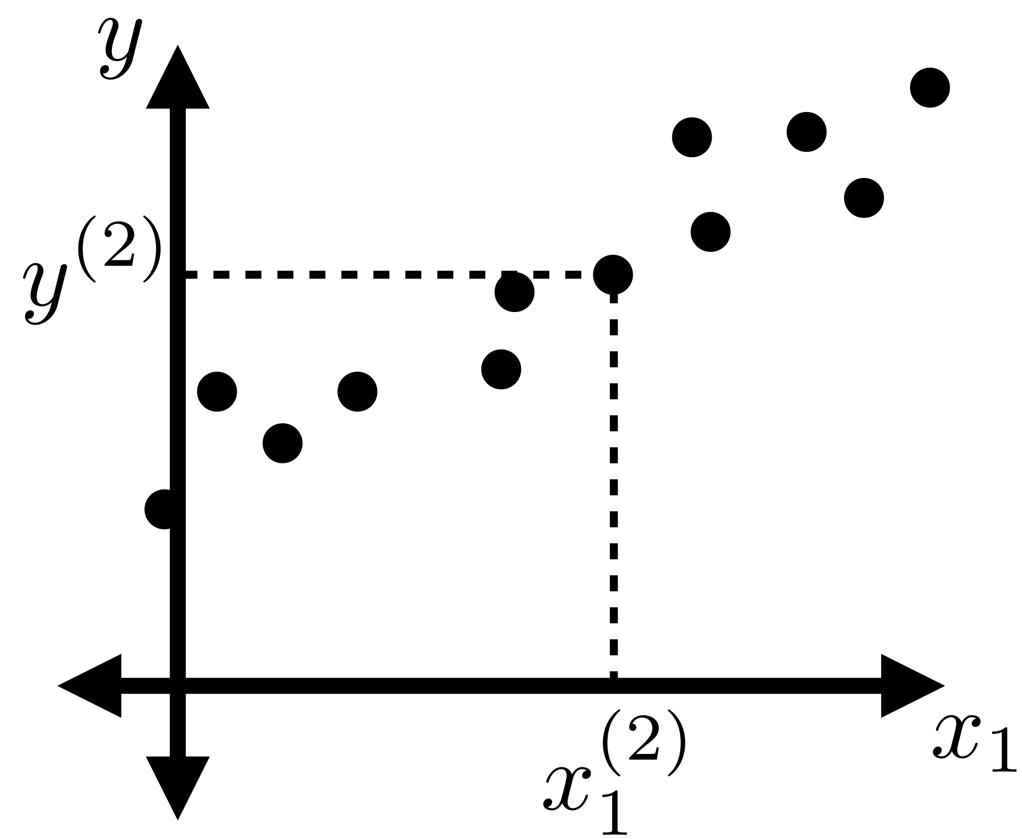


- Example:
  - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

$$h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$$

`Ex_learning_alg(` $\mathcal{D}_n$ `;` k `)`

Learning a regressor



- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall: $x \rightarrow$ [h] $\rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ [learning algorithm] $\rightarrow h$
- Example:
  - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

$$h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$$

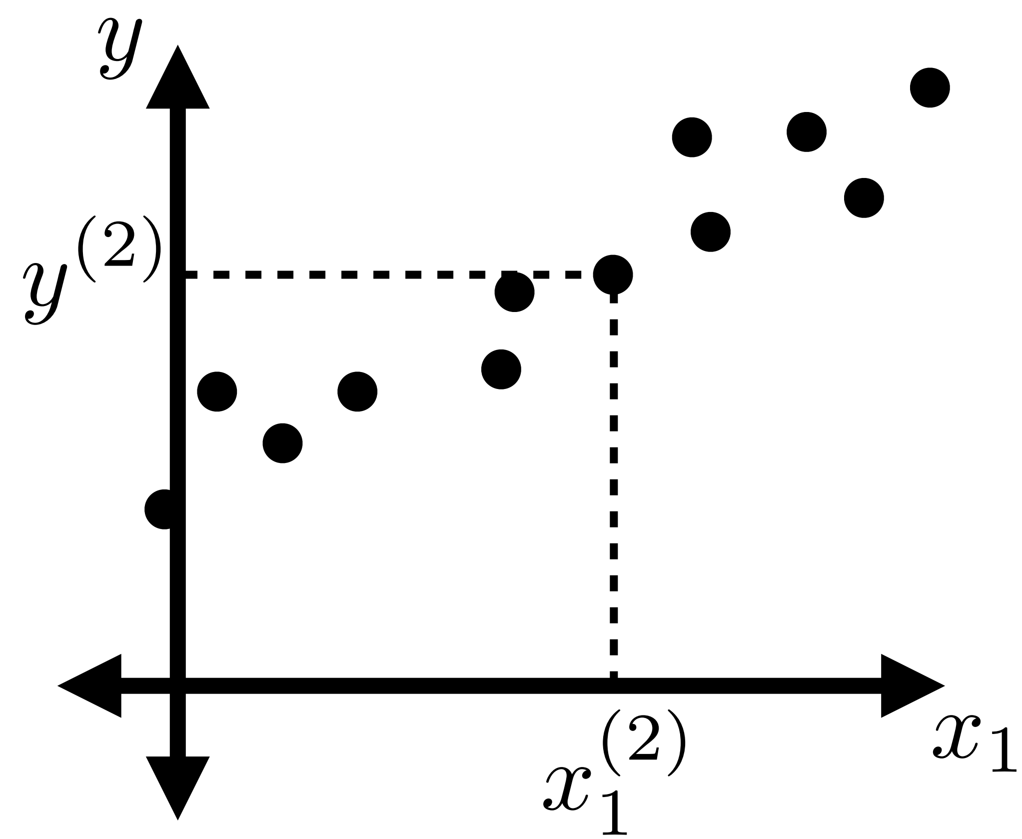
```
Ex_learning_alg( 𝒟ₙ ; k )
   Set j* = the j ∈ {1, ..., k} with lowest 𝓔ₙ(h^(j))
```

Ex_learning_alg($\mathcal{D}_n$; k)

Set $j^* =$ the $j \in \{1, \ldots, k\}$ with lowest $\mathcal{E}_n(h^{(j)})$

Learning a regressor

- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall: $x \rightarrow$ h $\rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$

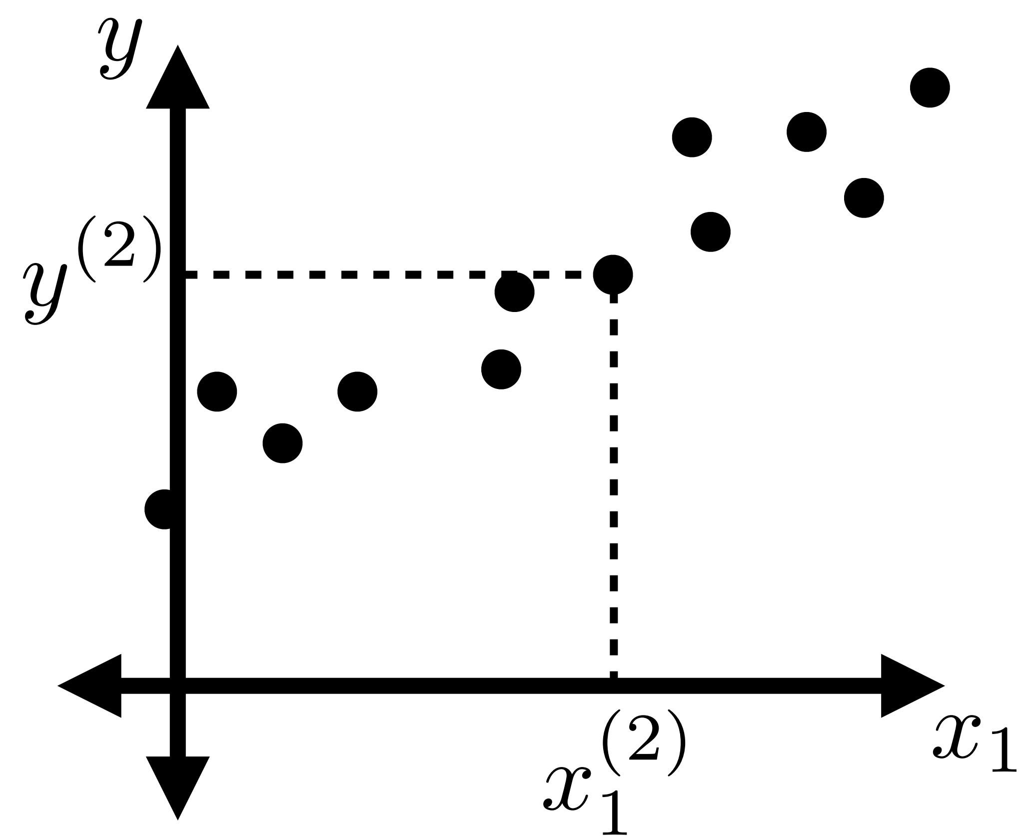


- Example:
  - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

$$h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$$

```
Ex_learning_alg( 𝒟_n ; k )
  Set j* = the j ∈ {1,...,k} with lowest ℰ_n(h^(j))
  Return h^(j*)
```

Learning a regressor



- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ $\boxed{\text{learning algorithm}}$ $\rightarrow h$
- Example:
  - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

$$h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$$

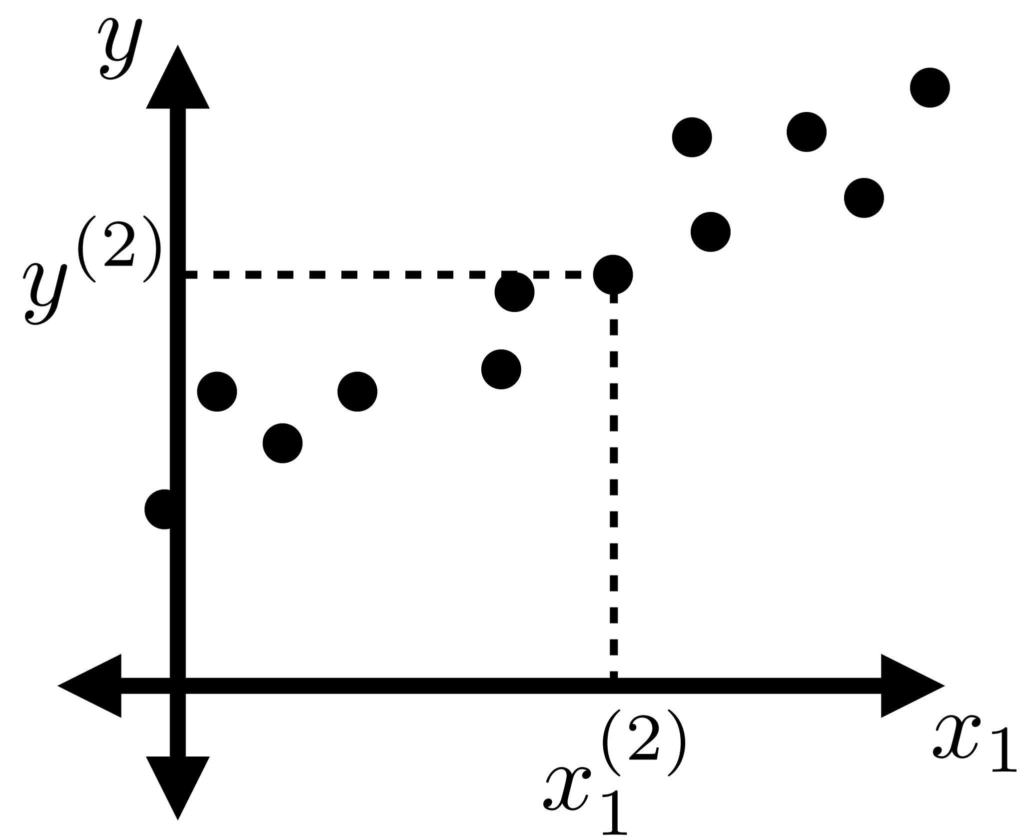
hyperparameter

`Ex_learning_alg(` $\mathcal{D}_n$ `;` k `)`

`Set` $j^* =$ the $j \in \{1, \ldots, k\}$ with lowest $\mathcal{E}_n(h^{(j)})$

`Return` $h^{(j^*)}$

Learning a regressor



- Have data; have hypothesis class
- Want to choose a good regressor
  - Recall: $x \rightarrow$ h $\rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$
- Example:
  - Suppose someone already generated 1 trillion hypotheses, e.g. at random, indexed by *j*:

$$h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$$

```
Ex_learning_alg( 𝒟ₙ ; k )
   Set j* = the j ∈ {1,...,k} with lowest ℰₙ(h^(j))
   Return h^(j*)
```

hyperparameter

- How does training error of `Ex_learning_alg`($\mathcal{D}_n$;1) compare to the training error of `Ex_learning_alg`($\mathcal{D}_n$;2)?

Linear regression: Another way

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
    - We’ll see: not typically straightforward

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss

$$\frac{1}{n} \sum_{i=1}^{n} (h(x^{(i)}) - y^{(i)})^2$$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$\frac{1}{n}\sum_{i=1}^{n}(h(x^{(i)}) - y^{(i)})^2$$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$\frac{1}{n}\sum_{i=1}^{n}(h(x^{(i)}) - y^{(i)})^2$$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$\frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2$$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2$$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2$$

Define $\tilde{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\theta^\top x^{(i)} - y^{(i)})^2$$

Define $\underset{\text{nxd}}{\tilde{X}} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2$$

Define $\underset{\text{nxd}}{\tilde{X}} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2$$

Define $\underset{\text{nxd}}{\tilde{X}} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$ $\qquad \tilde{Y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2$$

Define $\underset{\text{nxd}}{\tilde{X}} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$ $\quad \underset{\text{nx1}}{\tilde{Y}} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2$$

Define $\underset{\text{nxd}}{\tilde{X}} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$ $\qquad$ $\underset{\text{nx1}}{\tilde{Y}} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2 = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$$

Define $\underset{\text{nxd}}{\tilde{X}} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$ $\quad \underset{\text{nx1}}{\tilde{Y}} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

Linear regression: Another way

- How about we just consider all hypotheses in our class and choose the one with lowest training error?
  - We'll see: not typically straightforward
  - But for linear regression with square loss: can do it!
- Recall: training error: $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Training error: square loss, linear regr., extra "1" feature

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} - y^{(i)})^2 = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$$

Define $\underset{\text{nxd}}{\tilde{X}} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$ $\quad \underset{\text{nx1}}{\tilde{Y}} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

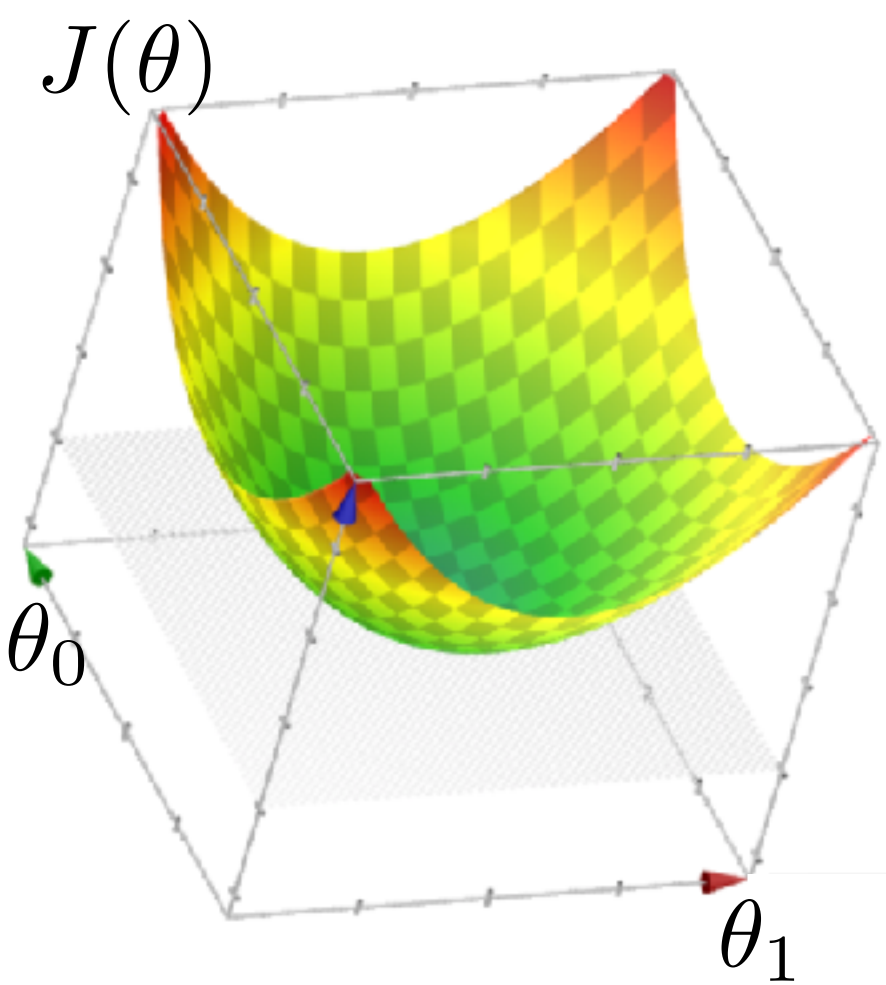
- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$

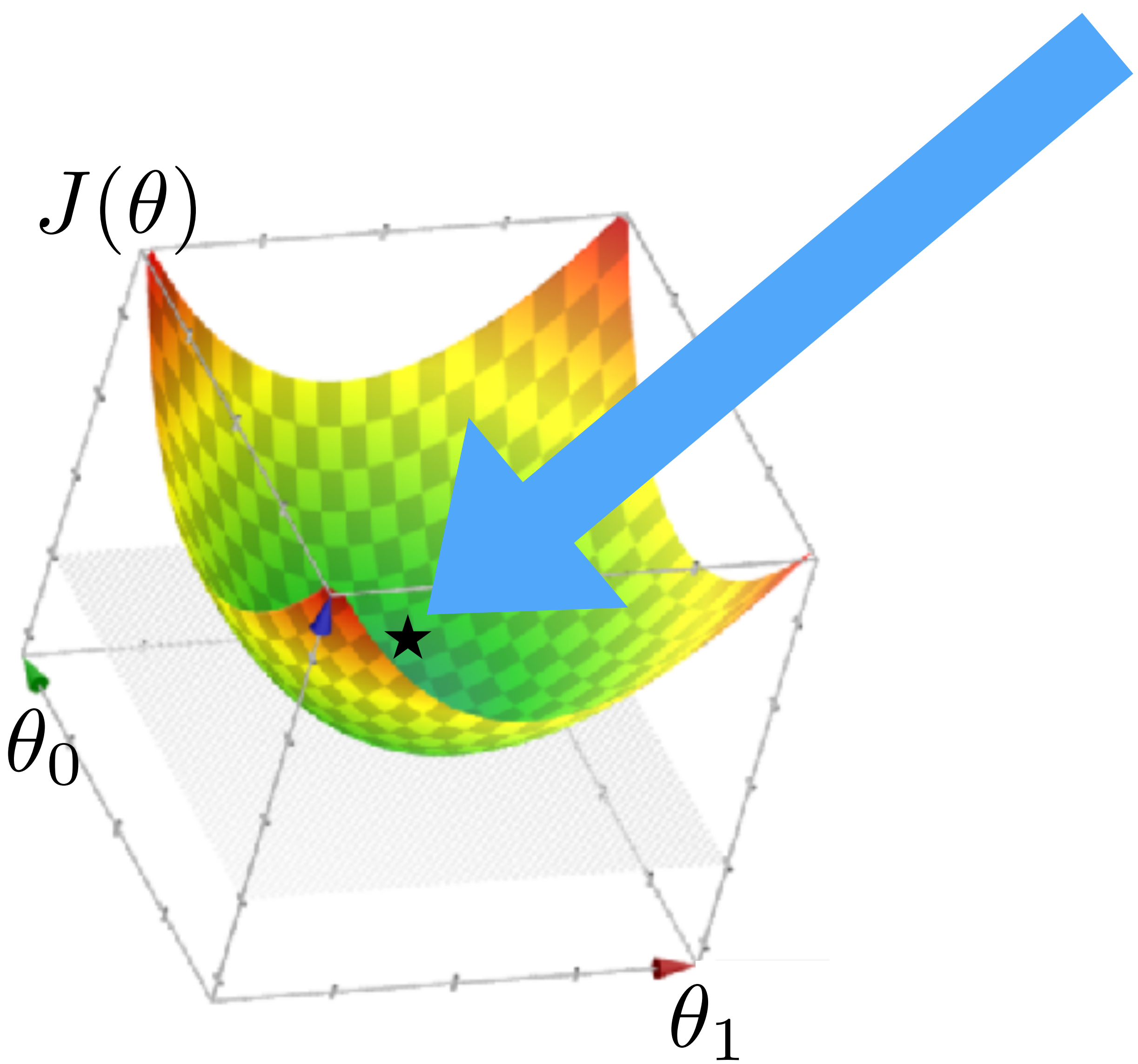
Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$



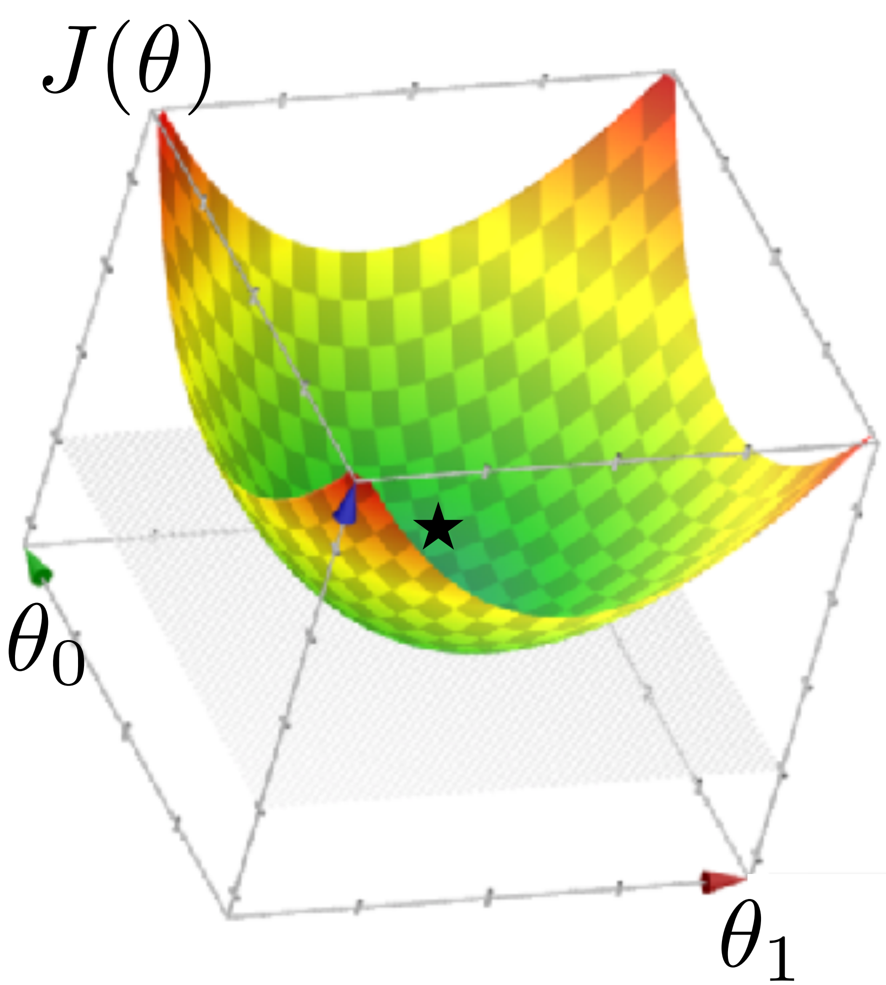
Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$



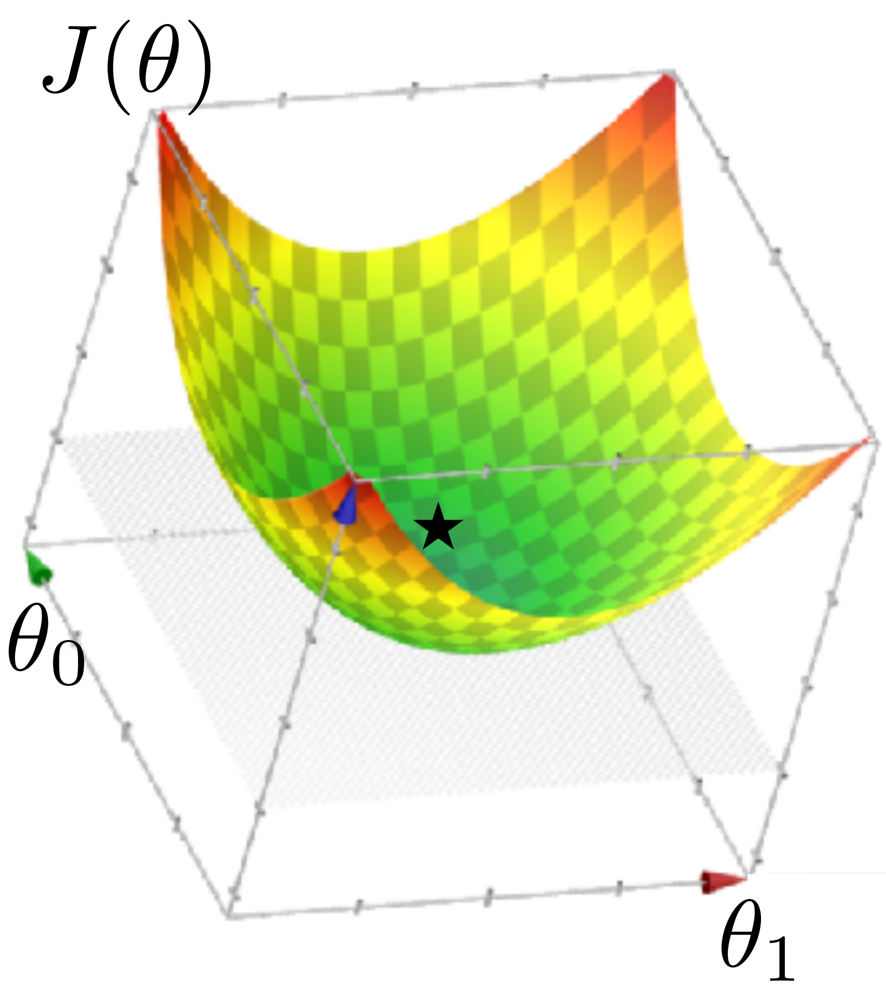
Linear regression: A Direct Solution

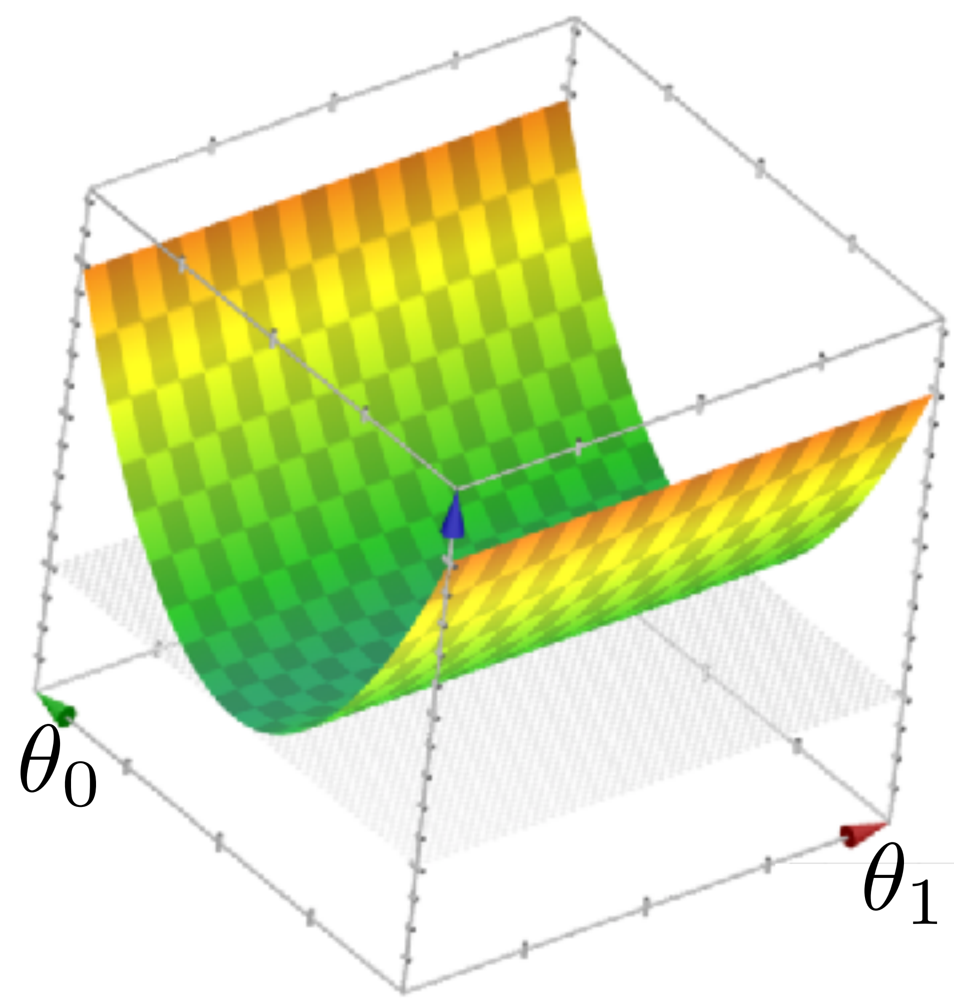
- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$



Linear regression: A Direct Solution

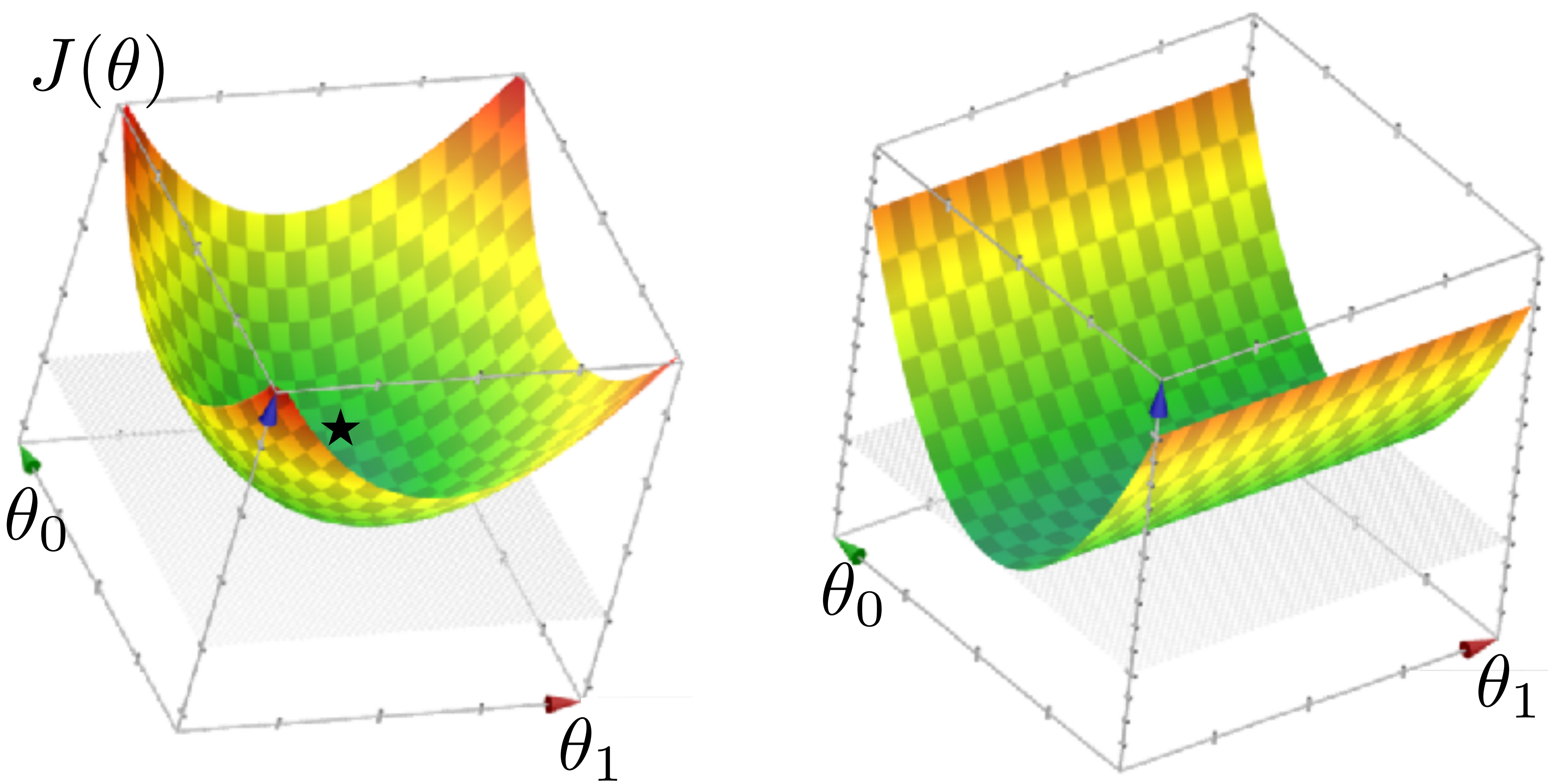
- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$





Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]



Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function “curves up” [see linear algebra]

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\nabla_\theta J(\theta)$

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\overset{\text{dx1}}{\nabla_\theta J(\theta)}$

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\underbrace{\nabla_\theta J(\theta)}_{}$ $\overset{\text{set}}{=} 0$ (dx1)

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\nabla_\theta J(\theta) = 0$ (annotations: "dx1" above $\nabla_\theta J(\theta)$, "set" above "=")

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\underbrace{\nabla_\theta J(\theta)}_{\text{dx1}} \overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{Y}$

Linear regression: A Direct Solution

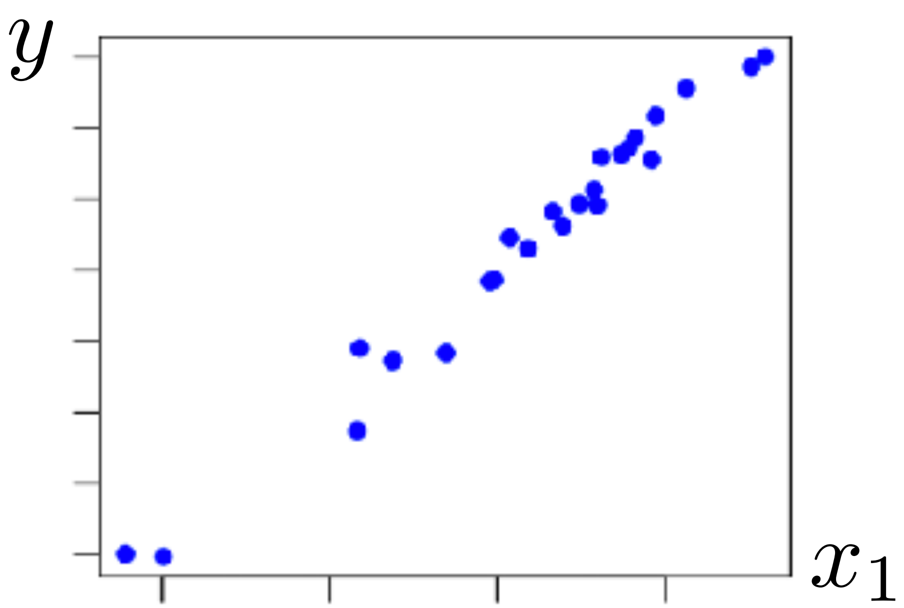
- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\underbrace{\nabla_\theta J(\theta)}_{\text{dx1}} \overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top \tilde{Y}$

Exercise: check n,d=1

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\underbrace{\nabla_\theta J(\theta)}_{\text{dx1}} \overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top \tilde{Y}$

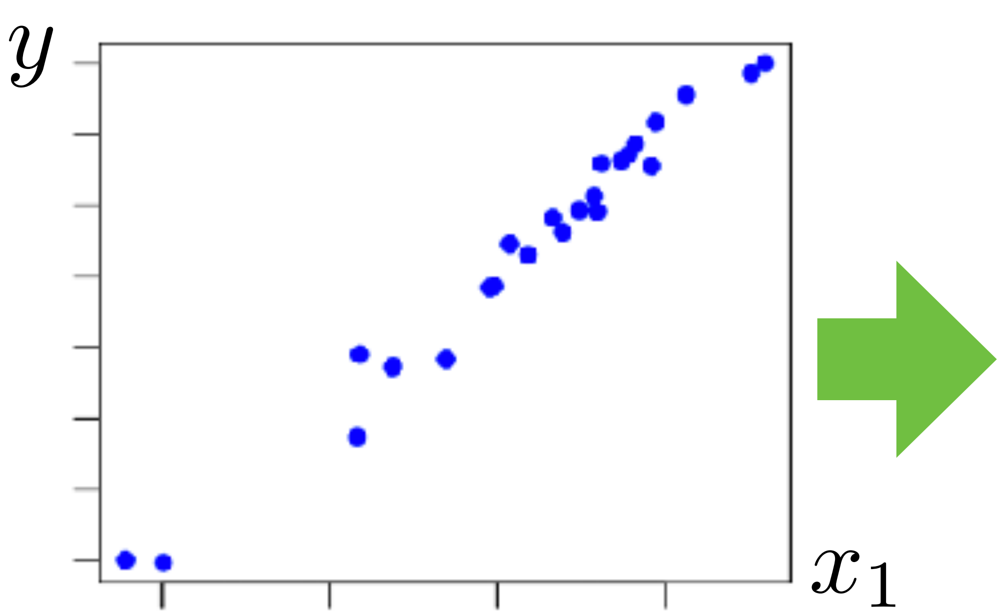
Exercise: check n,d=1



Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function “curves up” [see linear algebra]
- Gradient $\underbrace{\nabla_\theta J(\theta)}_{}$ (dx1) $\overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top\tilde{X})^{-1}\tilde{X}^\top\tilde{Y}$

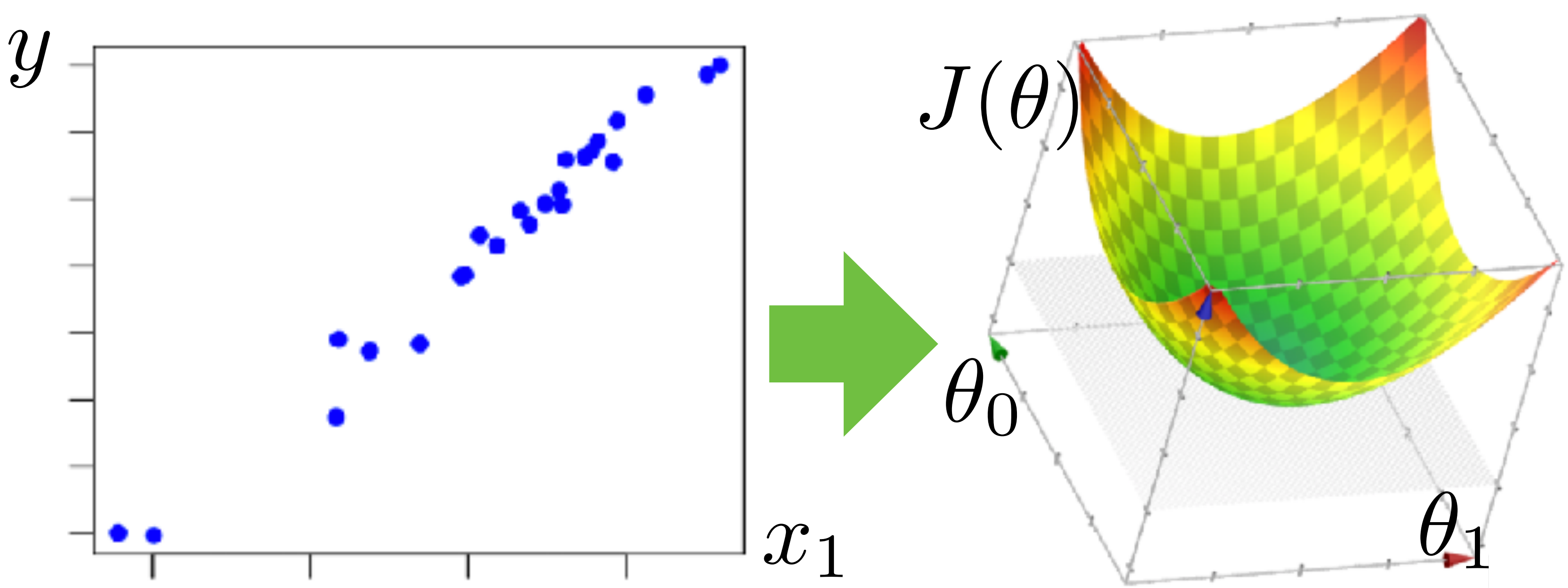
Exercise: check n,d=1



Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\underbrace{\nabla_\theta J(\theta)}_{\text{dx1}} \overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top \tilde{Y}$

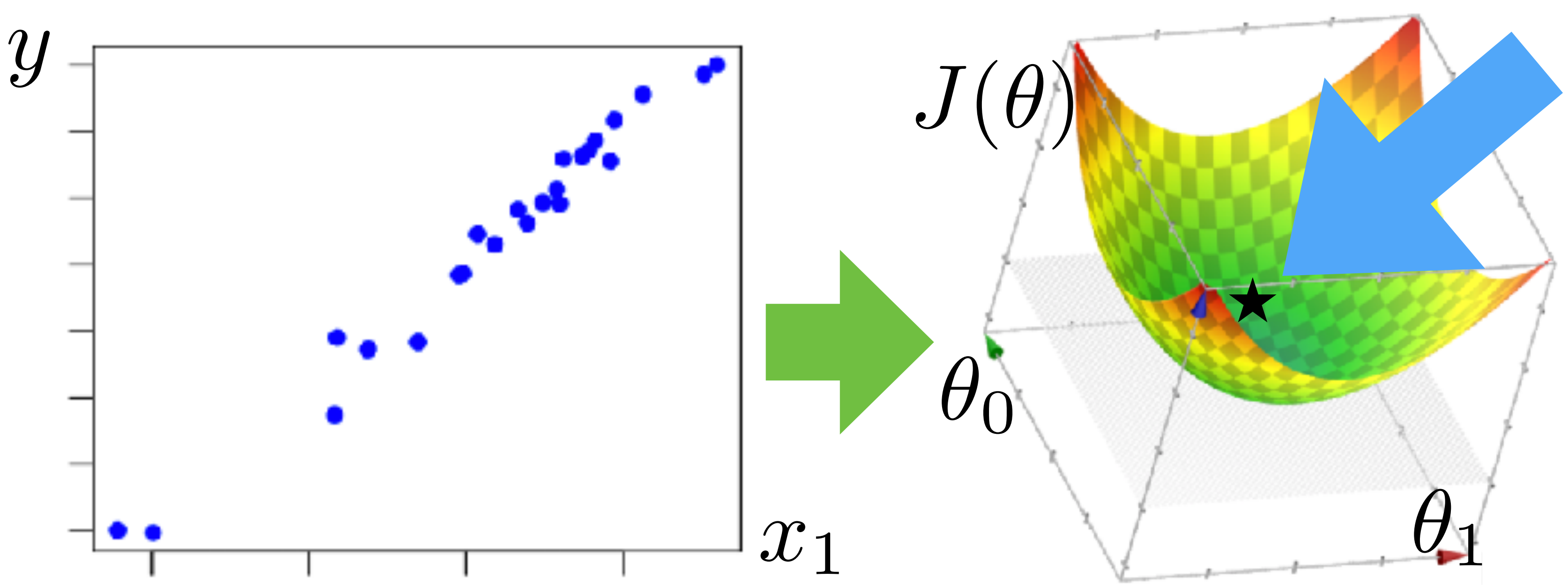
Exercise: check n,d=1



Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function “curves up” [see linear algebra]
- Gradient $\overset{\text{dx1}}{\nabla_\theta J(\theta)} \overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top \tilde{Y}$

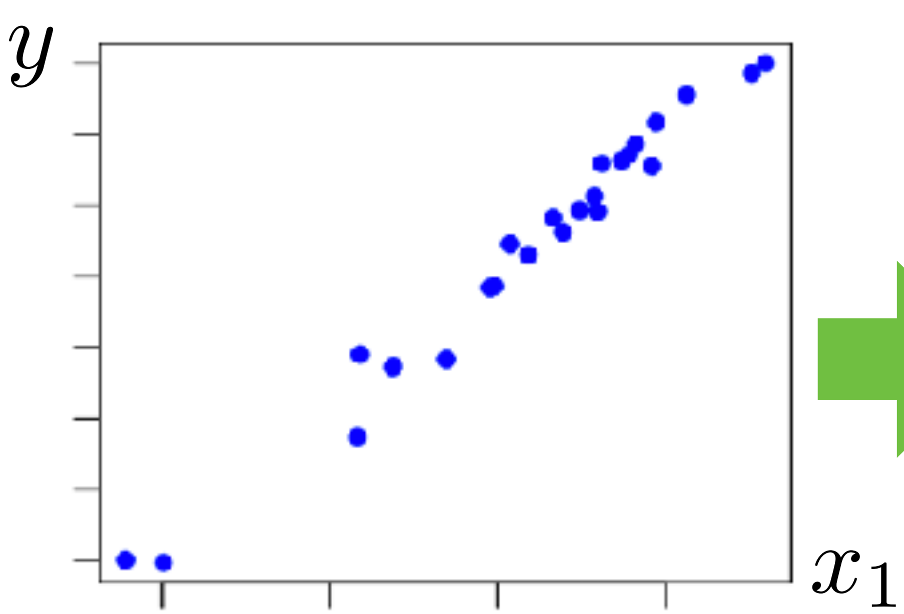
Exercise: check n,d=1

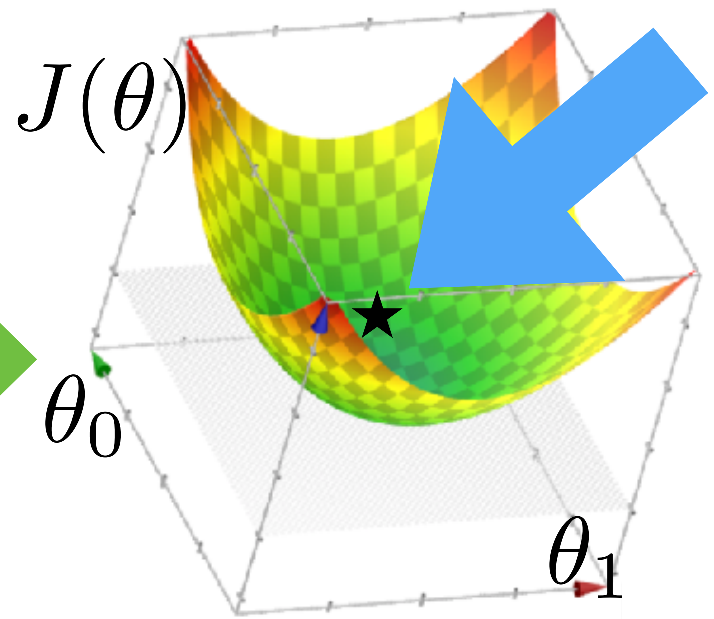


Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\nabla_\theta J(\theta) = 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top \tilde{Y}$ (dx1; set)

Exercise: check n,d=1

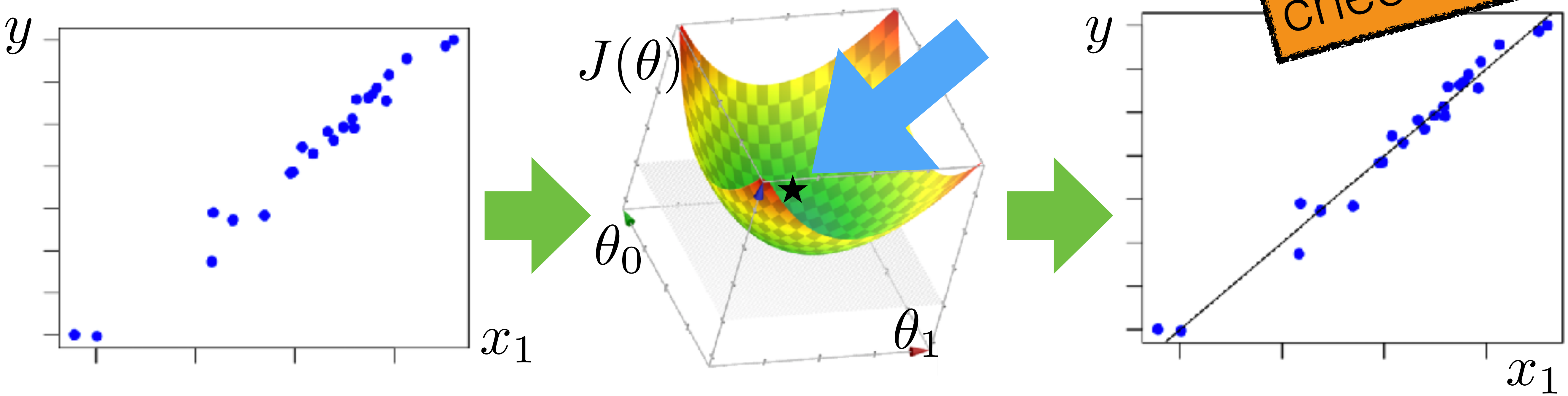




Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\nabla_\theta J(\theta) \overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top\tilde{X})^{-1}\tilde{X}^\top\tilde{Y}$ (dx1)

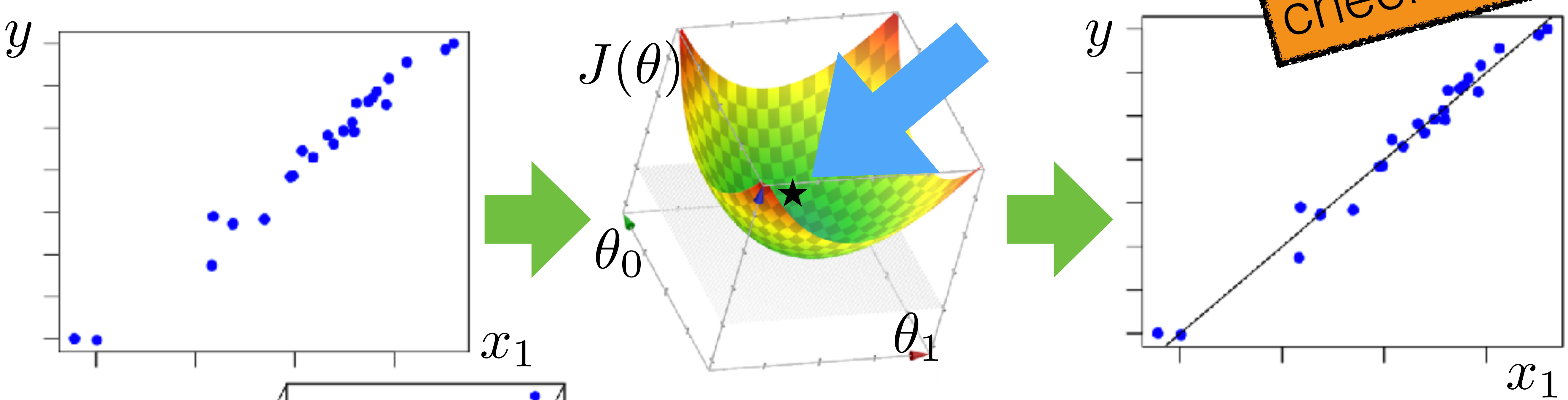
Exercise: check n,d=1

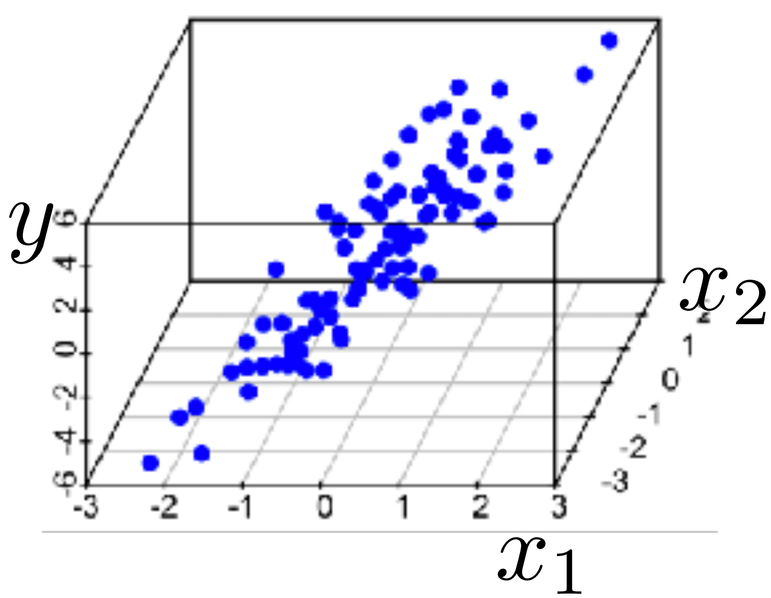


Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\underbrace{\nabla_\theta J(\theta)}_{\text{dx1}} \overset{\text{set}}{=} 0 \Rightarrow \theta = (\tilde{X}^\top\tilde{X})^{-1}\tilde{X}^\top\tilde{Y}$

Exercise: check n,d=1

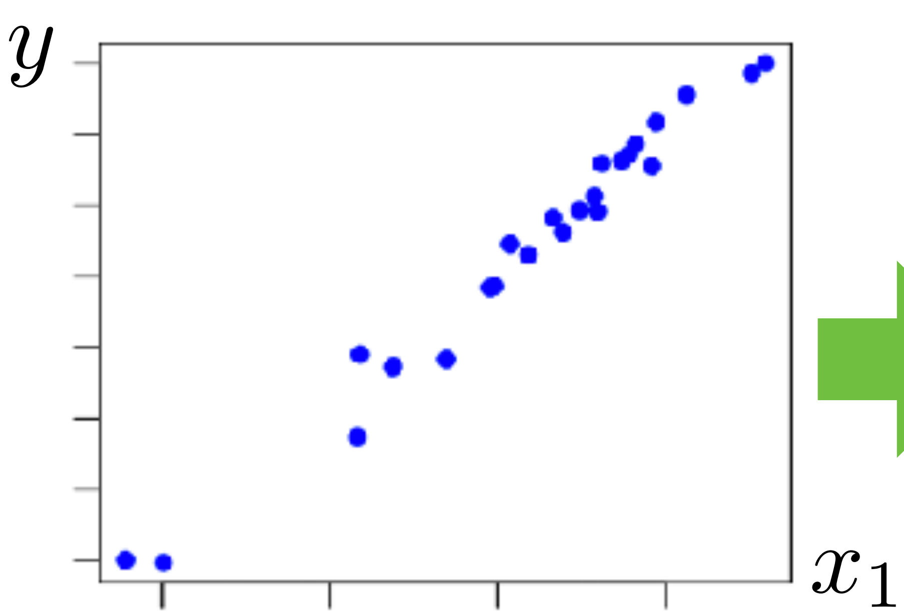


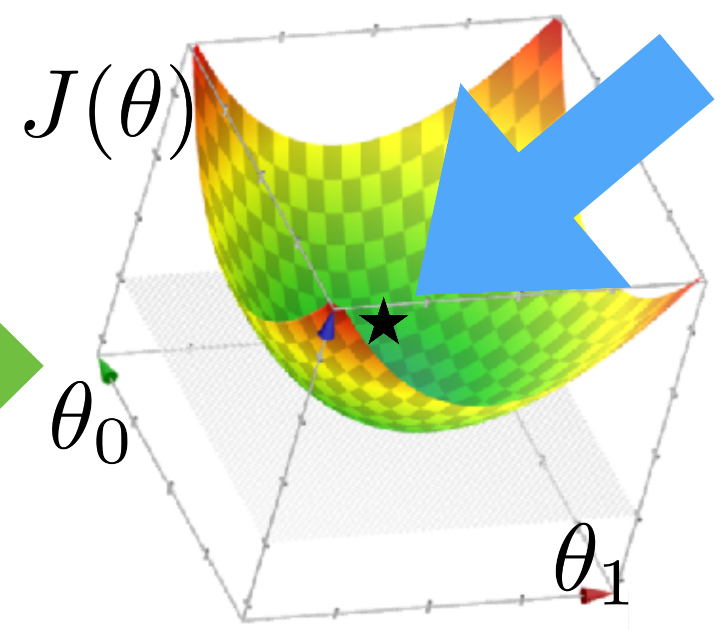


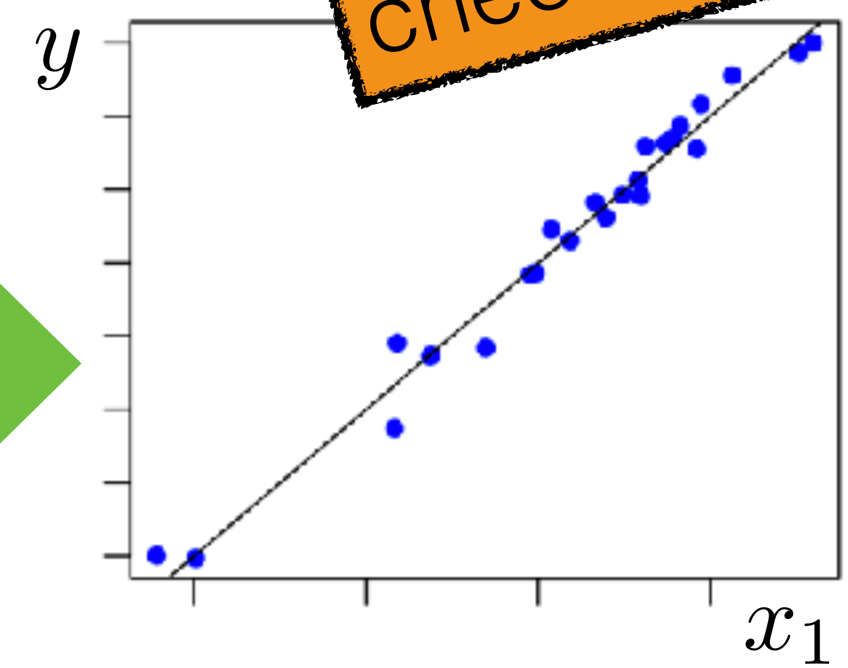
Linear regression: A Direct Solution

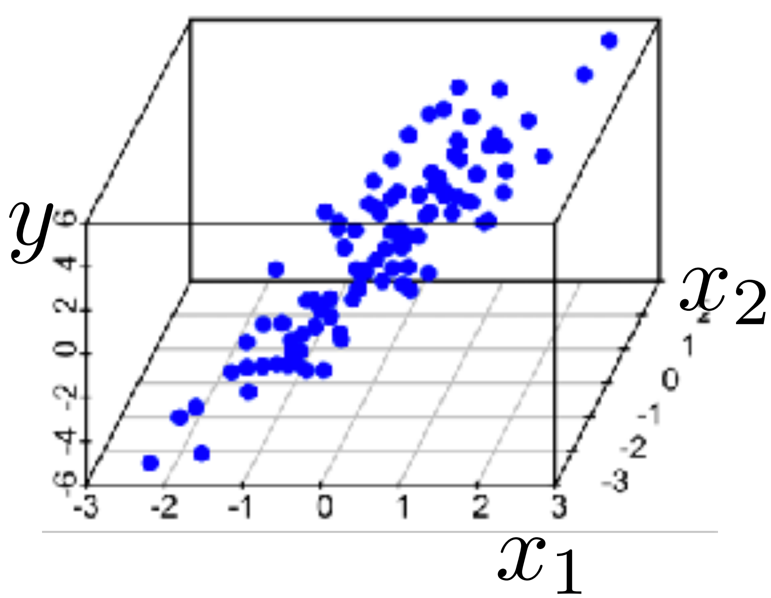
- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\nabla_\theta J(\theta) = 0 \Rightarrow \theta = (\tilde{X}^\top\tilde{X})^{-1}\tilde{X}^\top\tilde{Y}$ (annotations: "dx1" above θ; "set" above "= 0")

Exercise: check n,d=1







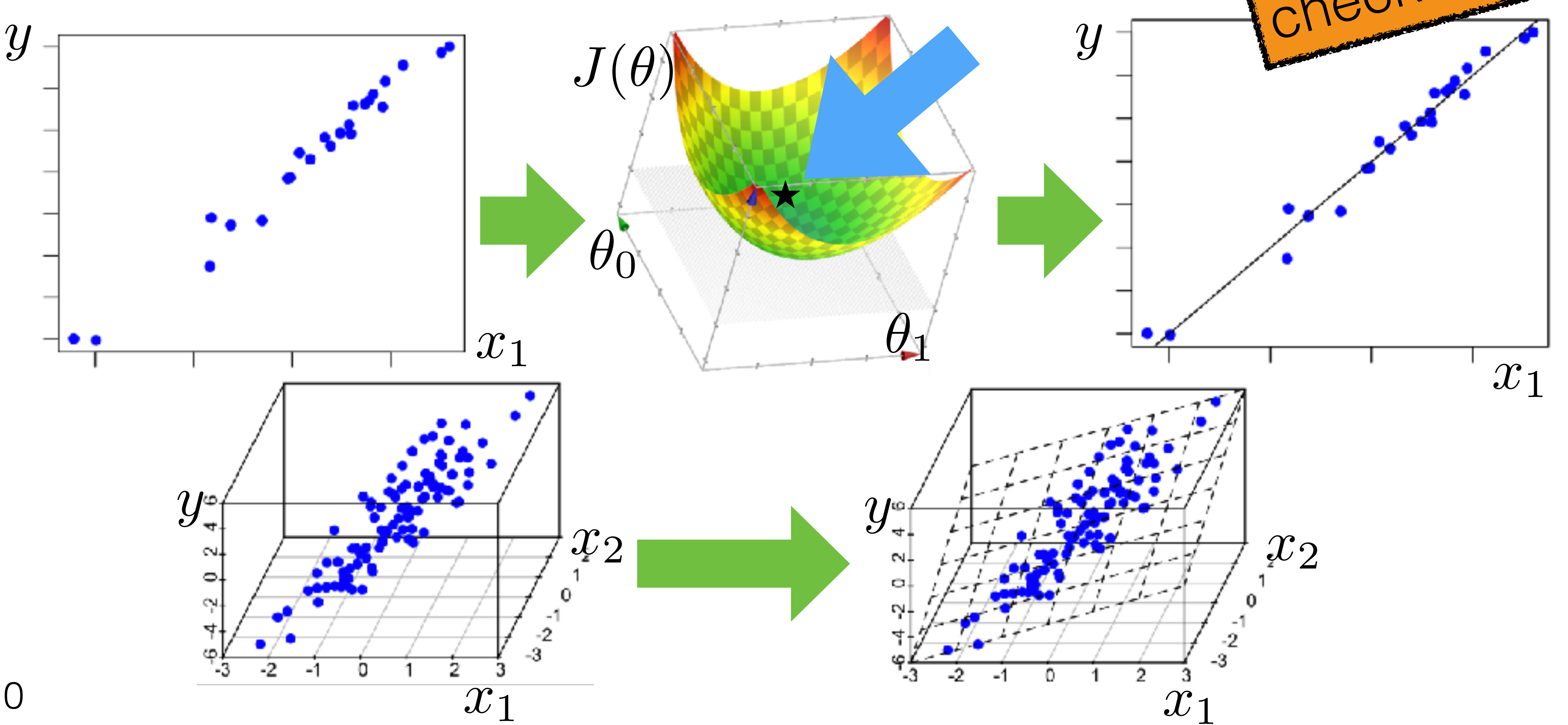


Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function "curves up" [see linear algebra]
- Gradient $\nabla_\theta J(\theta) = 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top \tilde{Y}$

dx1 set

Exercise: check n,d=1



What can go wrong in practice?

What can go wrong in practice?

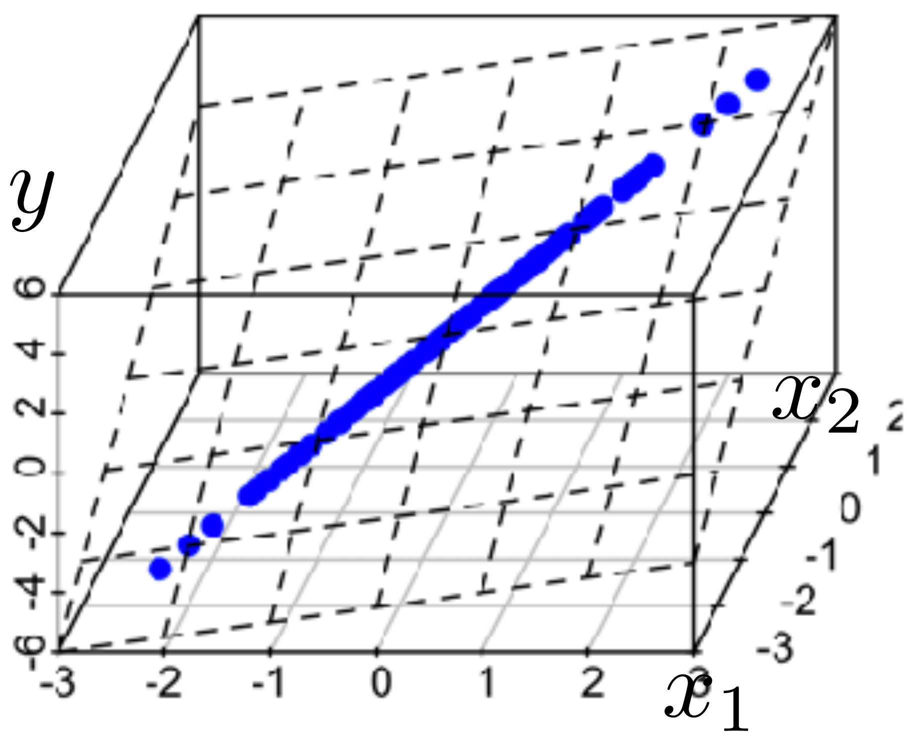
- Does the linear regr. objective always “curve up”? No!

What can go wrong in practice?

- Does the linear regr. objective always “curve up”? No!
- Sometimes there isn’t a unique best hyperplane

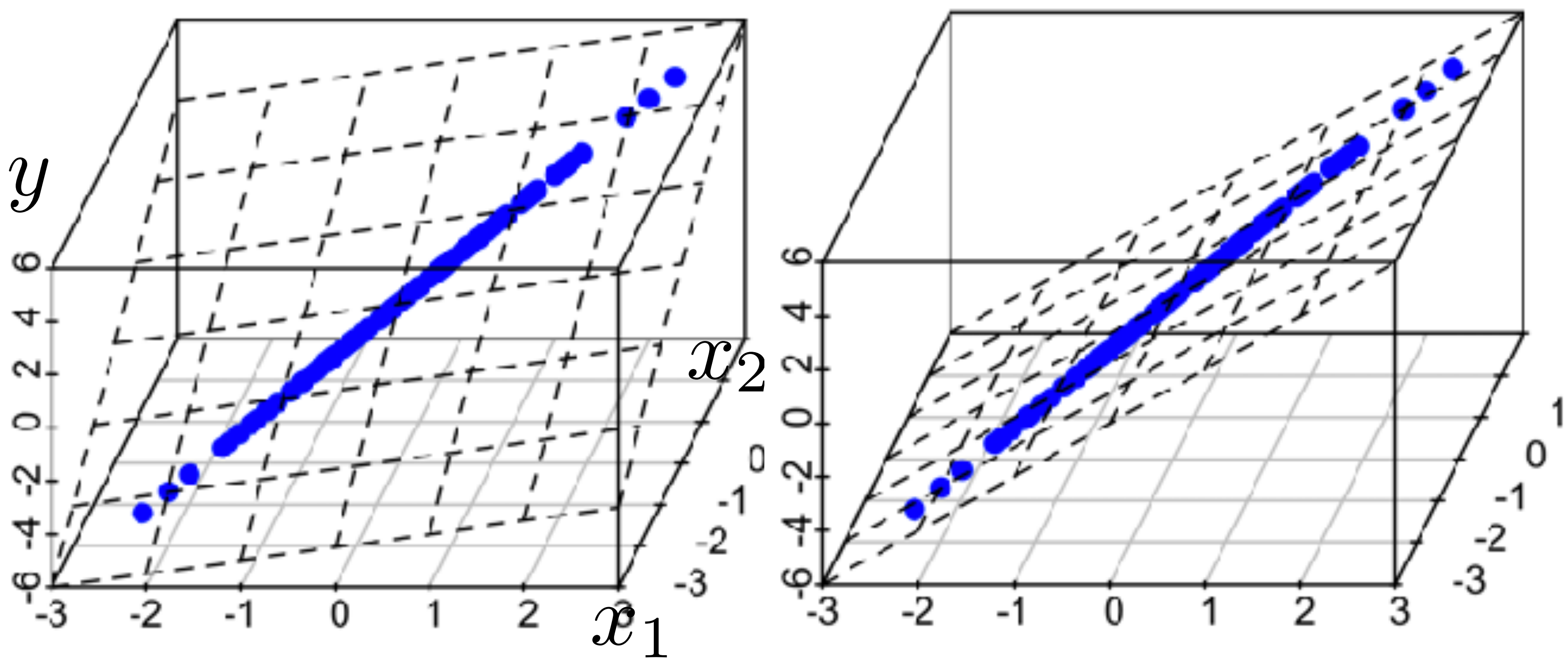
What can go wrong in practice?

- Does the linear regr. objective always "curve up"? No!
- Sometimes there isn't a unique best hyperplane



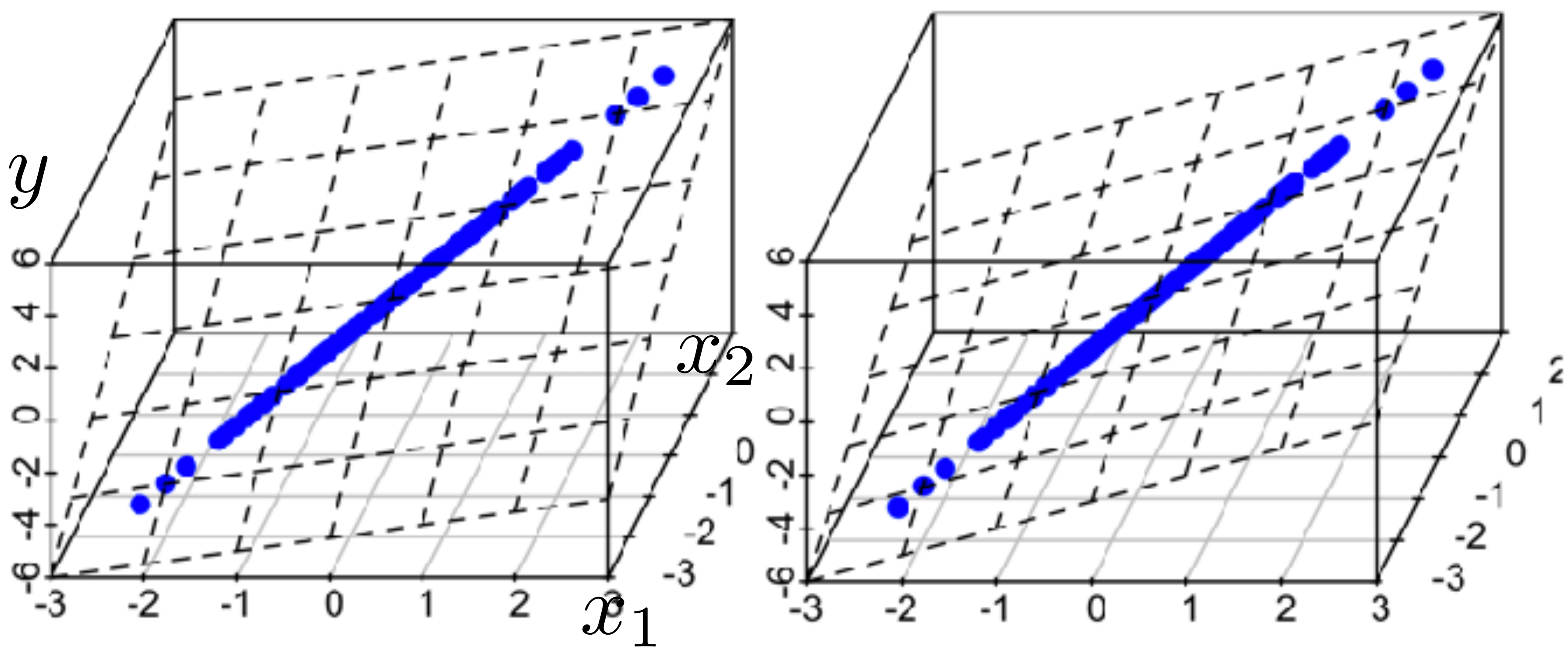
What can go wrong in practice?

- Does the linear regr. objective always “curve up”? No!
- Sometimes there isn’t a unique best hyperplane



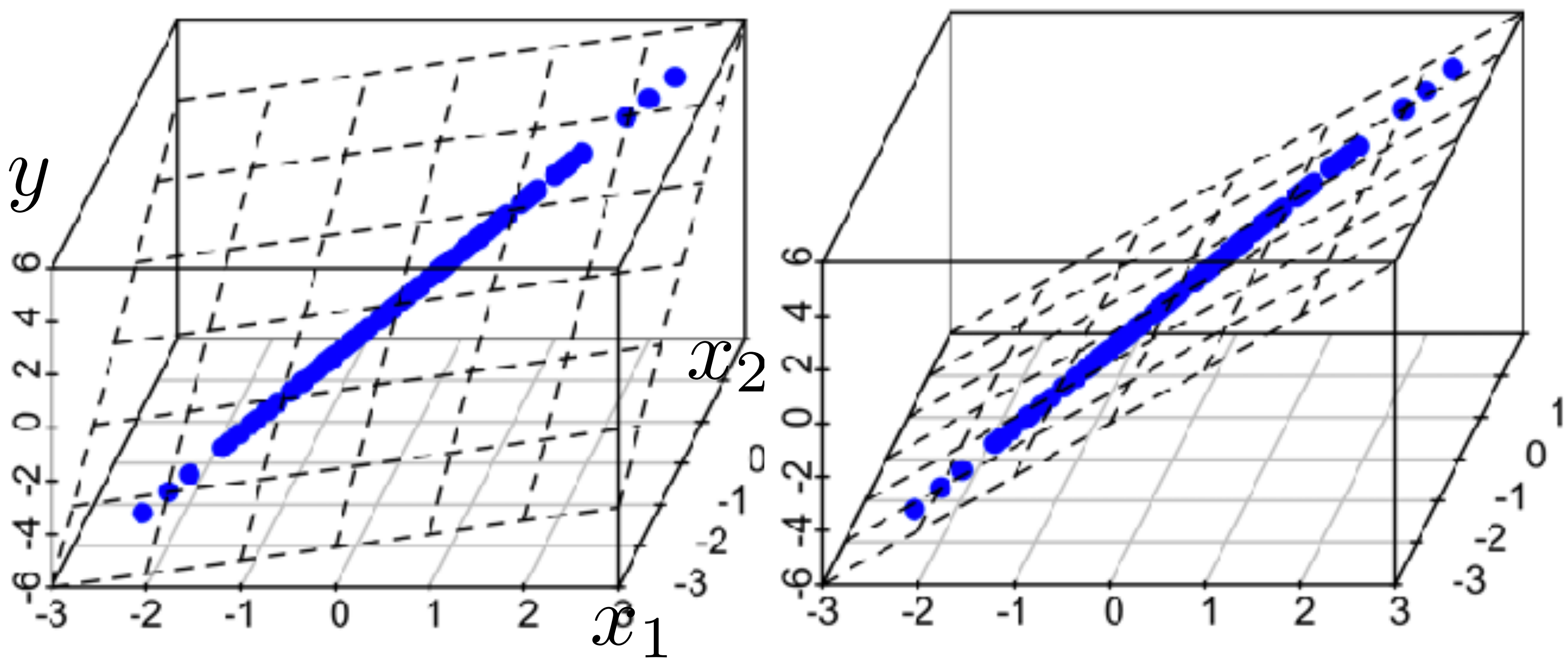
What can go wrong in practice?

- Does the linear regr. objective always “curve up”? No!
- Sometimes there isn’t a unique best hyperplane



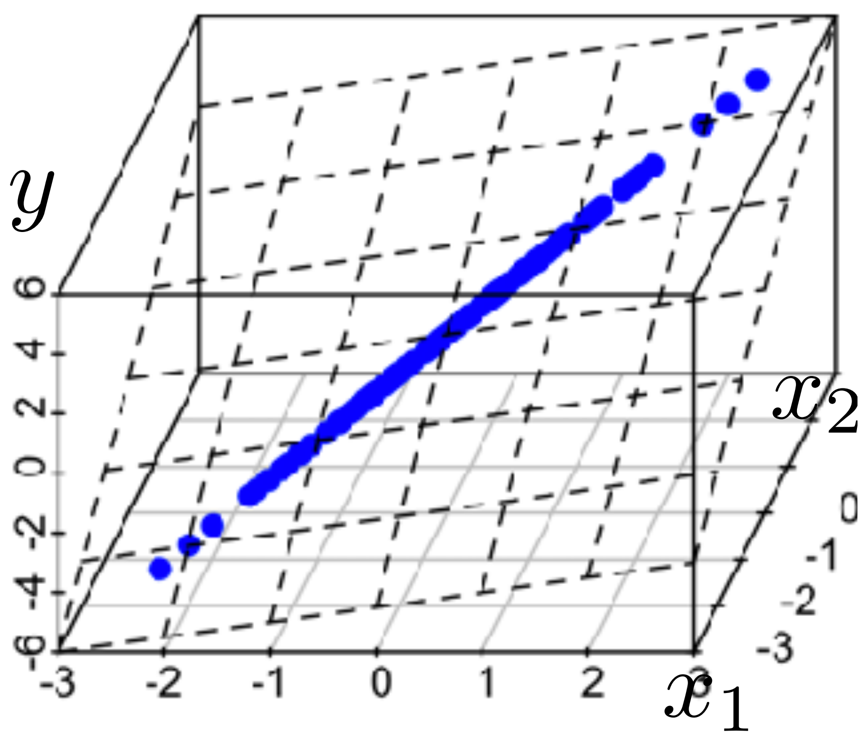
What can go wrong in practice?

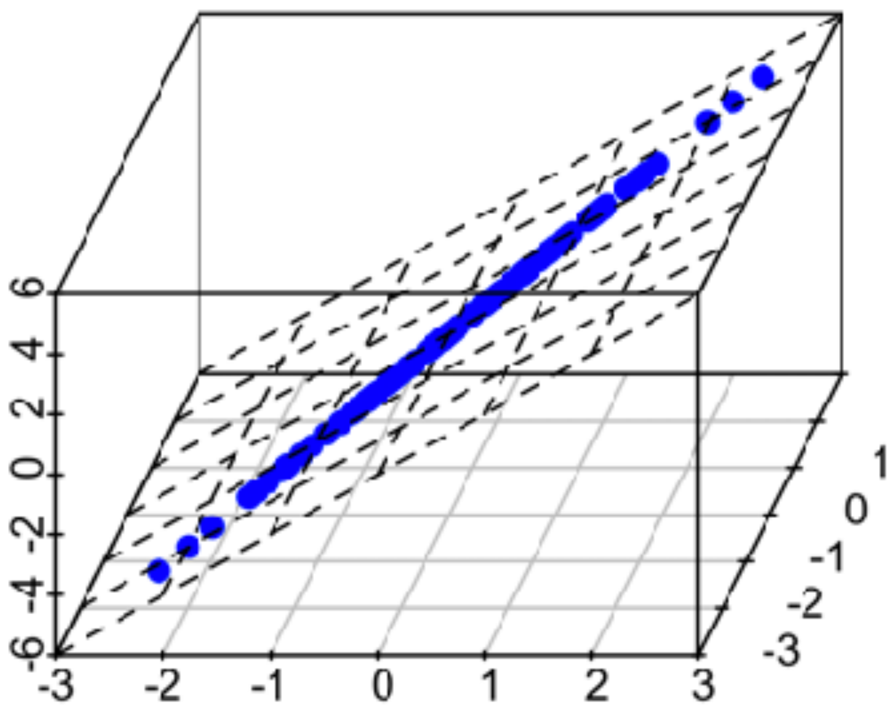
- Does the linear regr. objective always "curve up"? No!
- Sometimes there isn't a unique best hyperplane



What can go wrong in practice?

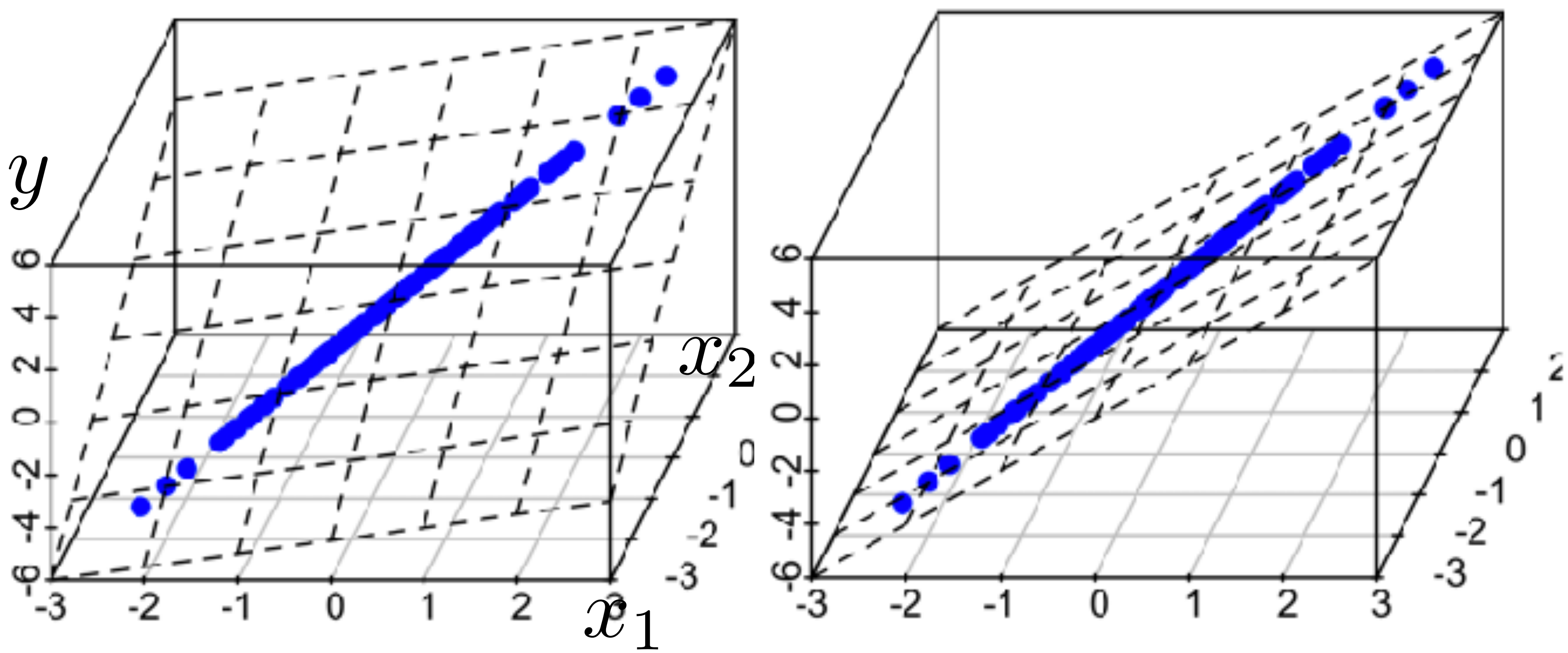
- Does the linear regr. objective always “curve up”? No!
- Sometimes there isn’t a unique best hyperplane
  - Then $X^\top X$ not invertible





What can go wrong in practice?

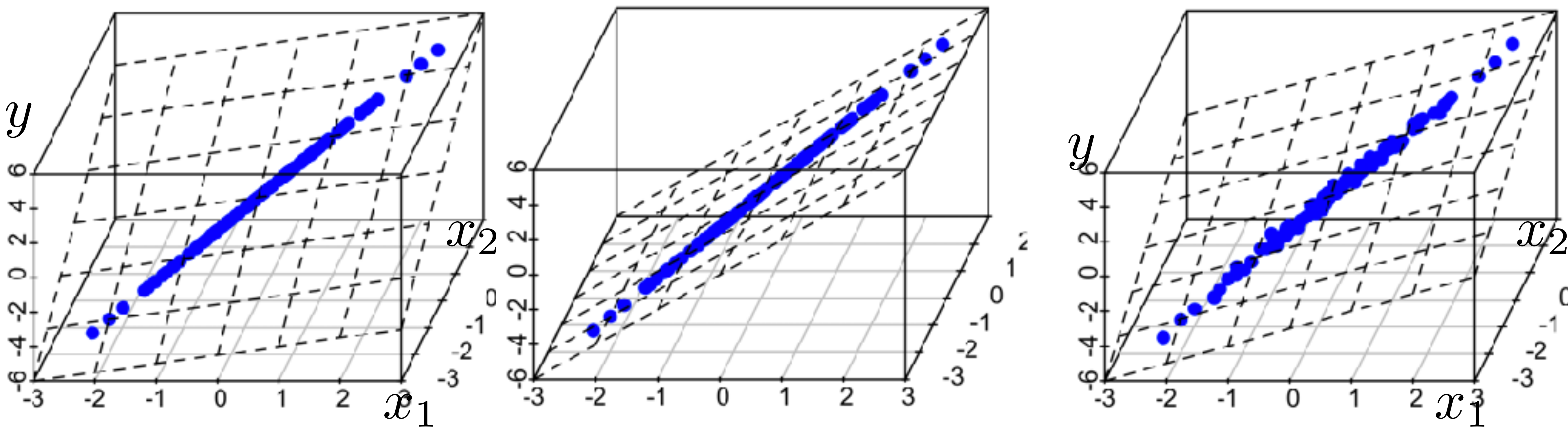
- Does the linear regr. objective always “curve up”? No!
- Sometimes there isn’t a unique best hyperplane
  - Then $X^\top X$ not invertible



- Sometimes there’s technically a unique best hyperplane, but just because of noise

What can go wrong in practice?

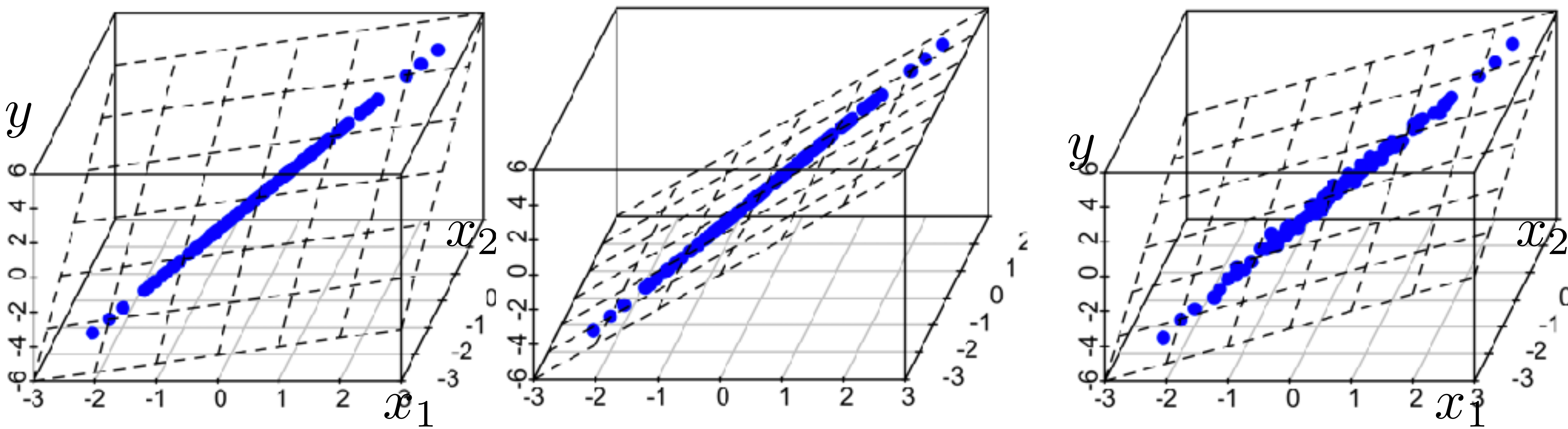
- Does the linear regr. objective always "curve up"? No!
- Sometimes there isn't a unique best hyperplane
  - Then $X^\top X$ not invertible



- Sometimes there's technically a unique best hyperplane, but just because of noise

What can go wrong in practice?

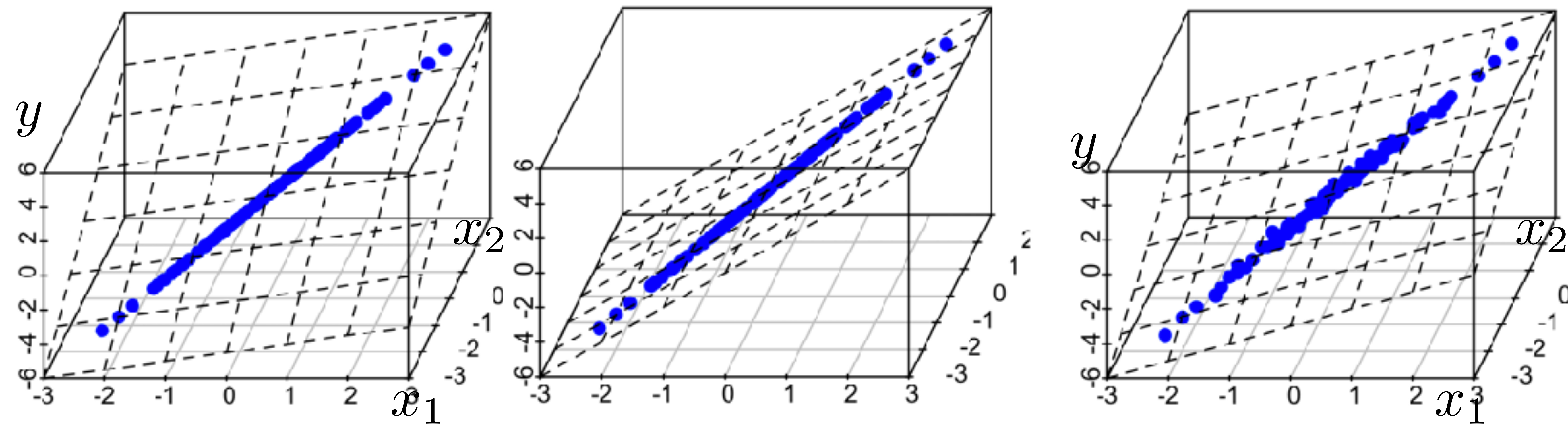
- Does the linear regr. objective always "curve up"? No!
- Sometimes there isn't a unique best hyperplane
  - Then $X^\top X$ not invertible



- Sometimes there's technically a unique best hyperplane, but just because of noise
- Practical: real-life features often have this issue

What can go wrong in practice?

- Does the linear regr. objective always “curve up”? No!
- Sometimes there isn’t a unique best hyperplane
  - Then $X^\top X$ not invertible



- Sometimes there’s technically a unique best hyperplane, but just because of noise
- Practical: real-life features often have this issue
- How to choose among hyperplanes? Preference for θ components being near zero

Regularizing linear regression

Regularizing linear regression

- Linear regression with square penalty: ridge regression

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2$$

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2$$

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2$$

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

What happens if $\lambda < 0$?

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

What happens if $\lambda < 0$?

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

  - Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

- Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

$$\Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1}\tilde{X}^\top \tilde{Y}$$

dxn,nxd dxd

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

- Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

$$\Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1}\tilde{X}^\top \tilde{Y}$$

dxn,nxd dxd

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

- Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

$$\Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1}\tilde{X}^\top \tilde{Y}$$

dxn,nxd dxd

- When $\lambda > 0$, always “curves up” & can invert

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

  - Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

$$\Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1}\tilde{X}^\top\tilde{Y}$$

dxn,nxd dxd

  - When $\lambda > 0$, always "curves up" & can invert
  - Can also solve with an offset

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda \|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top (\tilde{X}\theta - \tilde{Y}) + \lambda \|\theta\|^2$$

What happens if $\lambda < 0$?

  - Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

$$\Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1} \tilde{X}^\top \tilde{Y}$$

dxn,nxd dxd

  - When $\lambda > 0$, always "curves up" & can invert
  - Can also solve with an offset

- Can think of λ as hyperparameter of a learning algorithm

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

  - Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

$$\Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1}\tilde{X}^\top \tilde{Y}$$

dxn,nxd dxd

Exercise: write out the learning algorithm

  - When $\lambda > 0$, always "curves up" & can invert
  - Can also solve with an offset

- Can think of λ as hyperparameter of a learning algorithm

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda\|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$J_{\text{ridge}}(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y}) + \lambda\|\theta\|^2$$

What happens if $\lambda < 0$?

- Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0$

$$\Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1}\tilde{X}^\top \tilde{Y}$$

dxn,nxd dxd

Exercise: write out the learning algorithm

- When $\lambda > 0$, always “curves up” & can invert
- Can also solve with an offset

- Can think of λ as hyperparameter of a learning algorithm
- How to choose λ? One option: cross validation (see HW!)