

# DS-GA 1003: Machine Learning

Lecture 1: Intro & Supervised Learning Framework

Slides adapted from material from David Rosenberg's version of DS-GA 1003.

# Outline

## Course Overview and Logistics

Introduction to Machine Learning

Statistical Learning Setup

Statistical Learning: Bayes Risk

Statistical Learning: Empirical Risk and ERM

Statistical Learning: Hypothesis Class

Excess Risk Decomposition and Three Types of Error

# Course Website



<https://nyu-dsga-1003.github.io/sp26/>

A screenshot of the course website for NYU DS-GA 1003: Machine Learning (Spring 2026). The page features a navigation sidebar on the left with links for 'About', 'Calendar', 'Course Content', 'Homework', and 'Staff'. The main content area is titled 'About' and includes a 'TABLE OF CONTENTS' with links to 'What's this course?', 'Prerequisites' (with sub-links for Linear Algebra, Multivariable Calculus, and Probability Theory and Statistics), 'Logistics', 'Resources', 'Assessment', and 'Additional Information'. A note states the site is under construction until the first week of class on January 20th. The footer mentions the site uses 'Just the Docs' theme.

NYU DS-GA 1003: Machine Learning (Spring 2026)

Search NYU DS-GA 1003: Machine Learning (Spring 2026) Gradescope Ed Anonymous Feedback

## About

TABLE OF CONTENTS

- [What's this course?](#)
- [Prerequisites](#)
  - [Linear Algebra](#)
  - [Multivariable Calculus](#)
  - [Probability Theory and Statistics](#)
- [Logistics](#)
- [Resources](#)
- [Assessment](#)
- [Additional Information](#)

**NOTE:** This site is still under construction and will be undergoing changes until the first week of class, on January 20th! Please stay tuned for updates and monitor your email for communication about the course.

### What's this course?

This course is a graduate-level introduction to machine learning. We try to present machine learning as a story where many algorithmic techniques drop out of a common statistical learning framework.

The course covers a wide variety of topics in machine learning and statistical modeling. While

This site uses [Just the Docs](#), a documentation theme for Jekyll.

# Staff & Office Hours

8 course staff teaching assistants.

Check the **Staff** page: <https://nyu-dsga-1003.github.io/sp26/staff/>

Office hours available every day of the week!

Check the **Calendar** page: <https://nyu-dsga-1003.github.io/sp26/calendar/>

Strongly welcome and encouraged to come to office hours!

Sam's: Tues 5 - 6pm; Wed 1 - 2pm

Nick's: Wed 3 - 4pm

The screenshot displays the 'Staff' section of a course website. It is divided into 'Instructors' and 'Teaching Assistants'. Under 'Instructors', there are two entries: Nicholas Tomlin and Sam Deng. Each entry includes a profile picture, name, email, a short bio, and office hours. Under 'Teaching Assistants', there is one entry for Ansh Sharma. Below the staff list is a 'Course Calendar' section with a brief description. The calendar itself is a weekly view for January 2026, showing office hours for various staff members on specific days and times.

**Staff**

**Instructors**

**Nicholas Tomlin**  
n.tomlin@nyu.edu  
Hi! I'm a current faculty fellow at NYU, working on LLMs, reasoning, and interaction. In my spare time, I enjoy playing chess, eating bagels, and entertaining my cat Coco.  
**Office Hours:** Wednesdays 3:00pm - 4:00pm (CDS 617)

**Sam Deng**  
samuel.deng@nyu.edu  
Hi! I'm currently an adjunct instructor at NYU and final-year PhD student at Columbia, working on topics in machine learning theory. In my free time, I love going to the movies and running.  
**Office Hours:** Tuesdays 5:00pm - 6:00pm (after class in CDS 242); Wednesdays 1:00pm - 2:00pm (CDS 242)

**Teaching Assistants**

Ansh Sharma

**Course Calendar**

This page includes the course calendar, which will include all the important and recurring dates for the course, including office hours. If needed, we will reflect changes here.

Today	<	>	January 2026	Week			
	SUN	MON	TUE	WED	THU	FRI	SAT
	18	19	20	21	22	23	24
7 AM							
10 AM							
11 AM							
12 PM							
1 PM							
2 PM							
3 PM							
4 PM							
5 PM							

# EdStem

We'll use Ed for all course communications!

By default, please make your questions public; if you have a question, it's likely many other people do too!

If necessary (e.g., if your question reveals the answer to a homework question), post privately

Only email the instructors as a last resort. We are flooded with emails!

*We are not using Brightspace!*

ed NYU DS-GA 1003 – Ed Discussion

New Thread

Search

Filter

Welcome to DS-GA 1003! #1

Samuel Deng STAFF 18 hours ago in Announcements 119 VIEWS

UNPIN STAR WATCH

Hi everyone!

1 This is a start of semester announcement just to make sure that a couple things are on your radar before our lecture on Tuesday. Feel free to start posting questions on Ed if you have any!

- **Ed.** This is the main discussion board/communications channel of the course. Let this be your first stop to post questions/answer questions/stay updated with the course -- the course staff will be closely monitoring it. **We will not be using Brightspace/emails for communication/materials.** All announcements on Ed will automatically send an email and you should be automatically enrolled in the course if you are on the course Brightspace.
- **Course website.** We will be posting all the material for this course (lectures, labs, problem sets, syllabus, etc.) on the [course website](#). It's pending a couple more changes before the start of the semester Tuesday, but please take a look to get acquainted with the syllabus/structure of the course.
- **PS 0.** An introductory calibration problem set was released several days ago with the first and only Brightspace announcement. It's worth zero points and is just meant to get you acquainted with the resources/tools of the course. It shouldn't take long -- please complete it and
- **Problem Set 1.** The first problem set of the course, PS 1, will be released on Tuesday right before lecture (stay tuned for an

# Grading Rubric

Homeworks: 20%

Midterm Exam: 35%

Final Project: 35%

Lab Attendance: 10%

# Course Format

Lectures: Tuesdays, 2:45-4:45PM, 36 E 8th St (Cantor Film Ctr) Room 200

Labs (mandatory!): Thursday, 7:10-8PM, 238 Thompson St (GCASL) Room C95

# Course Format

Lectures: Tuesdays, 2:45-4:45PM, 36 E 8th St (Cantor Film Ctr) Room 200

Labs (mandatory!): Thursday, 7:10-8PM, 238 Thompson St (GCASL) Room C95

Lectures:

- Weeks 1-11: "classical ML" (gradient descent, regularization, SVMs, etc.)
- Weeks 12-16: "applied ML" (neural nets, generative models, LLMs, etc.)
- We'll do our best to post lecture slides and other relevant materials before class!
- Lectures will be recorded (on Brightspace), but we strongly recommend in-person attendance



# Course Format

Lectures: Tuesdays, 2:45-4:45PM, 36 E 8th St (Cantor Film Ctr) Room 200

Labs (mandatory!): Thursday, 7:10-8PM, 238 Thompson St (GCASL) Room C95

Lab grading policy:

- There are 12 labs in total
- You must attend 10+ labs to receive full credit for lab attendance
- You can receive 1 point of extra credit for each additional lab you attend

# Midterm

The midterm will be held in-class on Tuesday, March 10th, from 2:45-4:45PM.

*IMPORTANT:* Please make sure you are available at this time as we will not be able to offer makeup midterms! If you have a conflict, then you should consider not taking this course.

# Final Project

Form groups of 2-3 students and write an 8-page paper:

- Track 1: Applied ML - choose a real-world problem, identify how and why machine learning could be helpful, and find or collect a relevant dataset for the problem. Then, establish baselines and compare performance of many different ML techniques learned in class.
- Track 2: Research - identify a gap in the literature for an ML topic of interest, and then propose and execute experiments to address the gap. Then, write a NeurIPS/ICML/ICLR-style paper.

Key dates:

- Groups formed for projects: Feb 28th
- Project proposal (~2 pages): March 31st
- Final project submitted: May 8th

# Homeworks

Seven homeworks, plus Homework 0 ("Submitting and typesetting your homework")

You will have roughly two weeks to complete each homework once it is assigned

Late policy:

- You have 6 late days in total across the semester; if you want to submit 1-2 days late but have already used your late days, you will incur a 20% grade penalty per day
- However, you can use a maximum of 2 late days per homework. Gradescope will close 48 hours after the assignment deadline
- You can drop your lowest homework grade

# Homeworks

1. Regression & Statistical Learning
2. Regularization & GD
3. Linear Classification & SVM
4. MLE & Conditional Probability Models
5. Decision/RFs & Boosting
6. NNs
7. Generative Models & RL

# Homeworks

1. Regression & Statistical Learning
2. Regularization & GD
3. Linear Classification & SVM
4. MLE & Conditional Probability Models
5. Decision/RFs & Boosting
6. NNs
7. Generative Models & RL

## Homeworks

In this section, we will post the biweekly homeworks for the semester. Check back here for the latest problem set!

### Homework 0

<b>Material:</b>	<a href="#">Submitting &amp; typesetting your homework</a>	<a href="#">ps0-submission.zip, ps0.pdf</a>
<b>Release:</b>	Monday, January 12th, 7:30 PM ET	
<b>Due:</b>	Friday, January 23rd, 11:59 PM ET	

### Homework 1

<b>Material:</b>	<a href="#">Error decomposition and regression</a>	<a href="#">ps1-statlearning.zip, ps1.pdf</a>
<b>Release:</b>	Tuesday, January 20th, 2:30 PM ET	
<b>Due:</b>	Tuesday, February 3rd, 11:59 PM ET	

# Homeworks

1. Regression & Statistical Learning
2. Regularization & GD
3. Linear Classification & SVM
4. MLE & Conditional Probability Models
5. Decision/RFs & Boosting
6. NNs
7. Generative Models & RL

## Homeworks

In this section, we will post the biweekly homeworks for the semester. Check back here for the latest problem set!

### Homework 0

Material:	<a href="#">Submitting &amp; typesetting your homework</a>	<a href="#">ps0-submission.zip</a> , <a href="#">ps0.pdf</a>
Release:	Monday, January 12th, 7:30 PM ET	
Due:	Friday, January 23rd, 11:59 PM ET	

### Homework 1

Material:	<a href="#">Error decomposition and regression</a>	<a href="#">ps1-statlearning.zip</a> , <a href="#">ps1.pdf</a>
Release:	Tuesday, January 20th, 2:30 PM ET	
Due:	Tuesday, February 3rd, 11:59 PM ET	

# LLM Policy

Don't use LLMs



# LLM Policy

You are not allowed to use LLMs like ChatGPT or Cursor for homework or final projects

LLMs are great! But there's a growing body of evidence that LLMs can harm learning

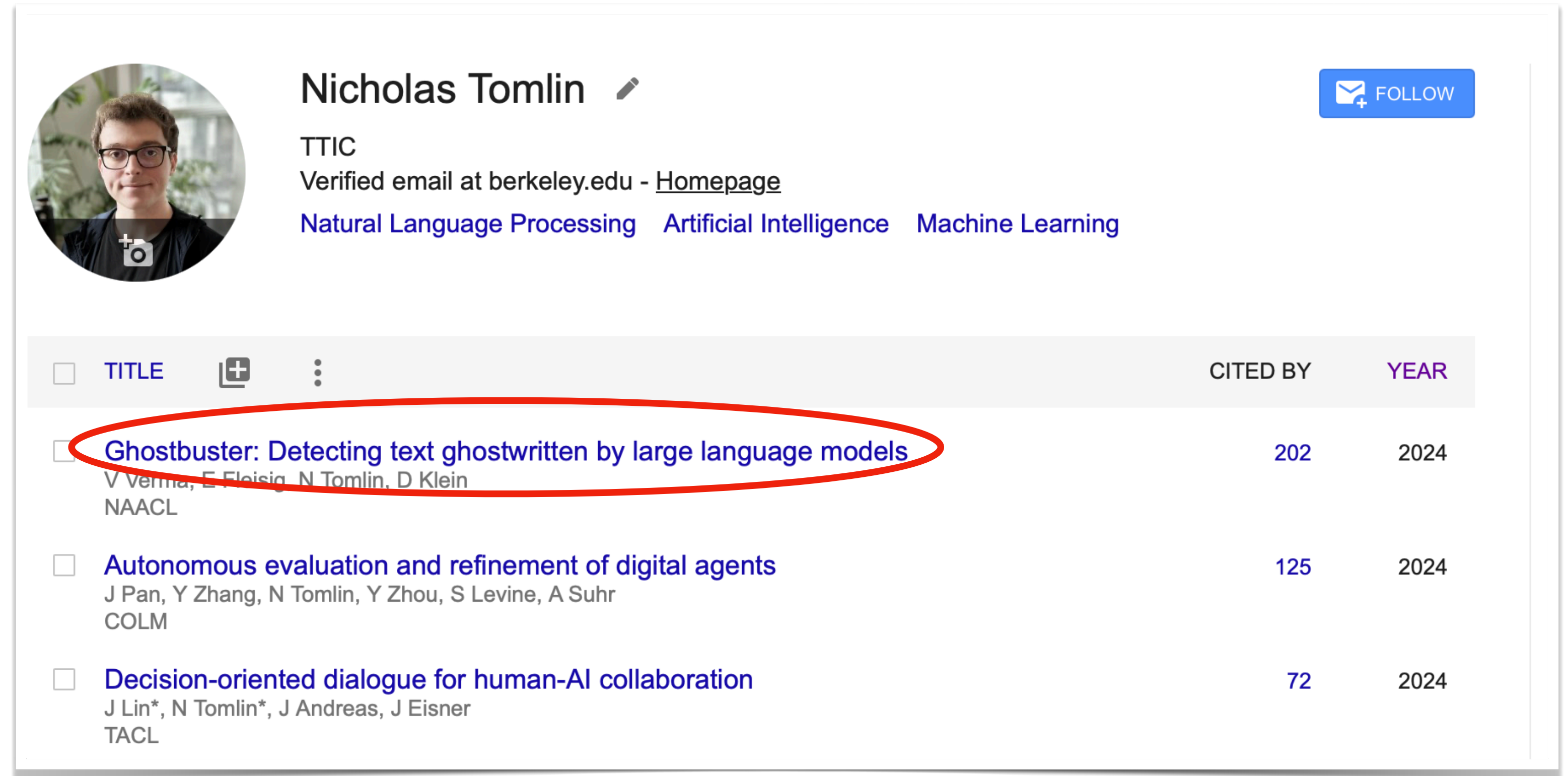
Using LLMs on homeworks may leave you unprepared for the midterm exam


# LLM Policy

You are not allowed to use LLMs like ChatGPT or Cursor for homework or final projects



LLMs are great! But there's a growing body of evidence that LLMs can harm learning

Using LLMs on homeworks may leave you unprepared for the midterm exam



**Nicholas Tomlin**  FOLLOW

TTIC  
Verified email at berkeley.edu - [Homepage](#)  
[Natural Language Processing](#) [Artificial Intelligence](#) [Machine Learning](#)

<input type="checkbox"/>	TITLE  	CITED BY	YEAR
<input type="checkbox"/>	<b>Ghostbuster: Detecting text ghostwritten by large language models</b> V Verma, E Fleisig, N Tomlin, D Klein NAACL	202	2024
<input type="checkbox"/>	<b>Autonomous evaluation and refinement of digital agents</b> J Pan, Y Zhang, N Tomlin, Y Zhou, S Levine, A Suhr COLM	125	2024
<input type="checkbox"/>	<b>Decision-oriented dialogue for human-AI collaboration</b> J Lin*, N Tomlin*, J Andreas, J Eisner TACL	72	2024

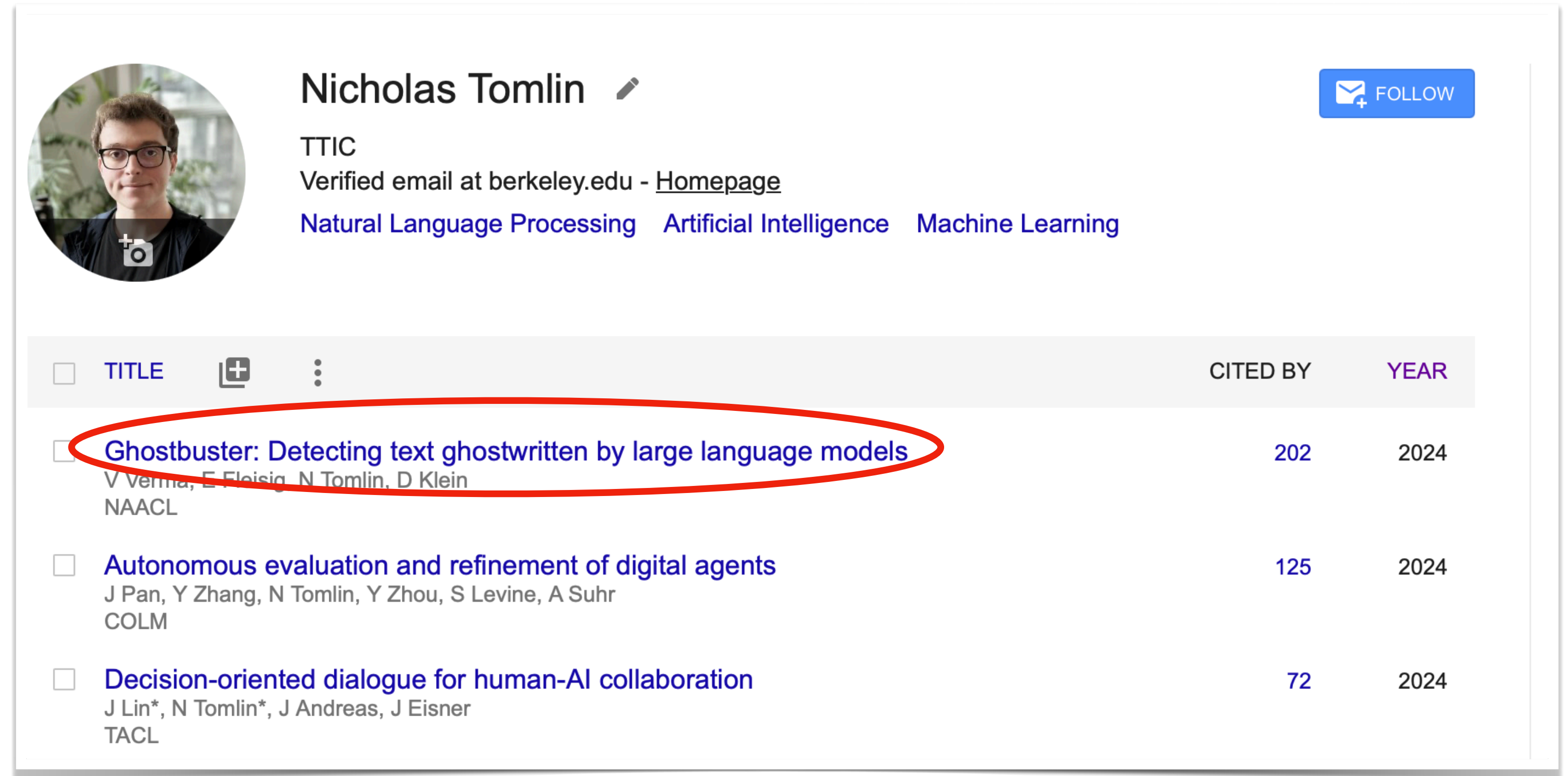
# LLM Policy


You are not allowed to use LLMs like ChatGPT or Cursor for homework or final projects

LLMs are great! But there's a growing body of evidence that LLMs can harm learning


Using LLMs on homeworks may leave you unprepared for the midterm exam



One exception: coding tools like Cursor and Claude Code are allowed if you are doing the "research track" for the final project. However, using LLMs for writing your final report is not allowed under either track.



**Nicholas Tomlin** 

TTIC  
Verified email at berkeley.edu - [Homepage](#)  
[Natural Language Processing](#) [Artificial Intelligence](#) [Machine Learning](#)



<input type="checkbox"/>	TITLE  	CITED BY	YEAR
<input type="checkbox"/>	<b>Ghostbuster: Detecting text ghostwritten by large language models</b> V Verma, E Fleisig, N Tomlin, D Klein NAACL	202	2024
<input type="checkbox"/>	<b>Autonomous evaluation and refinement of digital agents</b> J Pan, Y Zhang, N Tomlin, Y Zhou, S Levine, A Suhr COLM	125	2024
<input type="checkbox"/>	<b>Decision-oriented dialogue for human-AI collaboration</b> J Lin*, N Tomlin*, J Andreas, J Eisner TACL	72	2024

# Accommodations

If you need accommodations for the midterm or have accessibility concerns, please contact the Moses Center for Disabilities: [mosescsd@nyu.edu](mailto:mosescsd@nyu.edu)

If there are things we can do to help accommodate, let us know.

# Key dates and deadlines

Jan 23rd: Homework 0 due

Feb 2nd: last day to add/drop classes on Albert

Feb 3rd: Homework 1 due

Feb 28th: project groups formed

Mar 10th: midterm, in-class

# Should I take this class?

Yes, if:

- You are a CDS MS or PhD student
- You have familiarity with linear algebra, calculus, and basic programming
- You have taken DS-GA 1001 and DS-GA 1002
- You are available on Tuesday, March 10th from 2:45-4:45PM

# Should I take this class?

Yes, if:

- You are a CDS MS or PhD student
- You have familiarity with linear algebra, calculus, and basic programming
- You have taken DS-GA 1001 and DS-GA 1002
- You are available on Tuesday, March 10th from 2:45-4:45PM

If you think you have equivalent experience but haven't met the prerequisites: please email us your transcript and relevant course syllabi and we can review your waiver request

# Enrollment priority

Currently: 154 students (max of 200)

Priority order for registration:

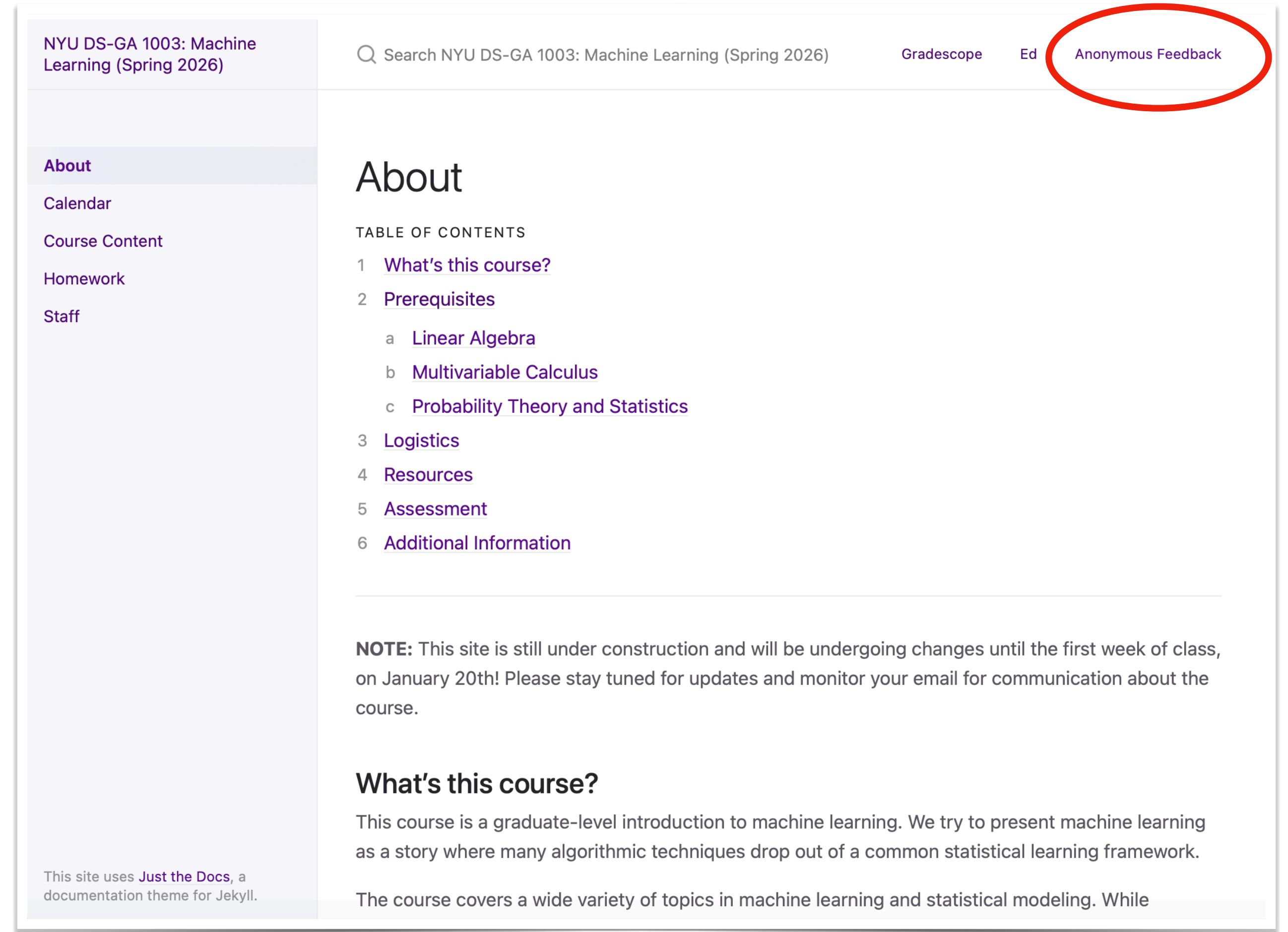
- Data science graduate students (MS and PhD)
- Non-data science PhD students: please ask your advisor to reach out to Tina Lam ([tina.lam@nyu.edu](mailto:tina.lam@nyu.edu)) to request your enrollment in this course
- MS students from other departments with appropriate prerequisites: registration should now be open. If you have issues, please contact [cds-masters@nyu.edu](mailto:cds-masters@nyu.edu)



# Anonymous Feedback

We want to provide a good learning experience and improve this course for future semesters!

Anonymous feedback form available at all times.



The screenshot shows the course website for NYU DS-GA 1003: Machine Learning (Spring 2026). The page has a navigation bar at the top with a search bar, 'Gradescope', 'Ed', and a red circle around the 'Anonymous Feedback' link. A left sidebar contains links for 'About', 'Calendar', 'Course Content', 'Homework', and 'Staff'. The main content area is titled 'About' and includes a 'TABLE OF CONTENTS' with links to 'What's this course?', 'Prerequisites', 'Logistics', 'Resources', 'Assessment', and 'Additional Information'. A note at the bottom states: 'NOTE: This site is still under construction and will be undergoing changes until the first week of class, on January 20th! Please stay tuned for updates and monitor your email for communication about the course.' Below the note is the heading 'What's this course?' followed by a paragraph: 'This course is a graduate-level introduction to machine learning. We try to present machine learning as a story where many algorithmic techniques drop out of a common statistical learning framework.' The text continues: 'The course covers a wide variety of topics in machine learning and statistical modeling. While'.

# Outline

Course Overview and Logistics

**Introduction to Machine Learning**

Statistical Learning Setup

Statistical Learning: Bayes Risk

Statistical Learning: Empirical Risk and ERM

Statistical Learning: Hypothesis Class

Excess Risk Decomposition and Three Types of Error



Given a dataset of photos of cats, predict the breed of a cat.



→ "Siamese"

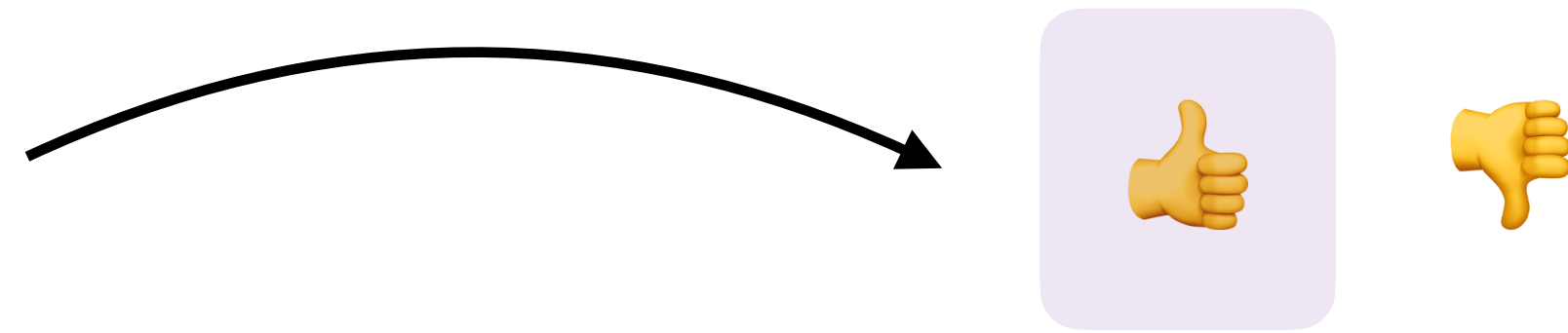
By Karin Langner-Bahmann, upload von Martin Bahmann - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=3020045>



Given a dataset of music listeners and songs, predict whether a user likes a song.



By <https://open.spotify.com/album/26ZV7BuCkdY3INkETgEJ0e?si=-5kn-WvIQsesSQGof-BD3w>, Fair use, <https://en.wikipedia.org/w/index.php?curid=4897516>



Given a written Chinese sentence, return the English translation.

Text Images Documents Websites

Detect language English **Chinese (Simplified)** Filipino

你好世界

Nǐ hǎo shìjiè

4 / 5,000 拼

“Hello world”

Given a dataset of meteorological measurements, forecast the temperature.

humidity	wind (mph)	cloud cover	month	pressure (in)
33%	7	2	march	29



81

Given a written English text passage, predict the (“most probable”) next word.

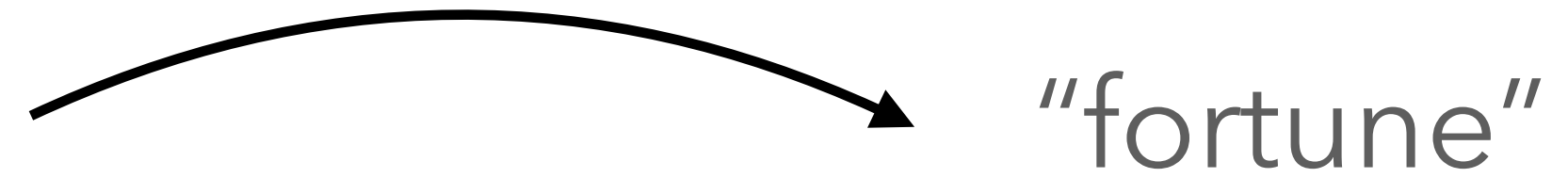
It is a truth universally acknowledged, that a single man in possession of a good

...fortune, must be in want of a wife.

— *Jane Austen, Pride and Prejudice* (1813)

That’s the famous opening line — would you like me to continue the paragraph, or do a short literary analysis of why this sentence is so iconic?

📄 👍 💬 ↗️ ↻️ ⋮



# “Traditional Programs” vs. Machine Learning

Many problems are difficult to “program by hand.”

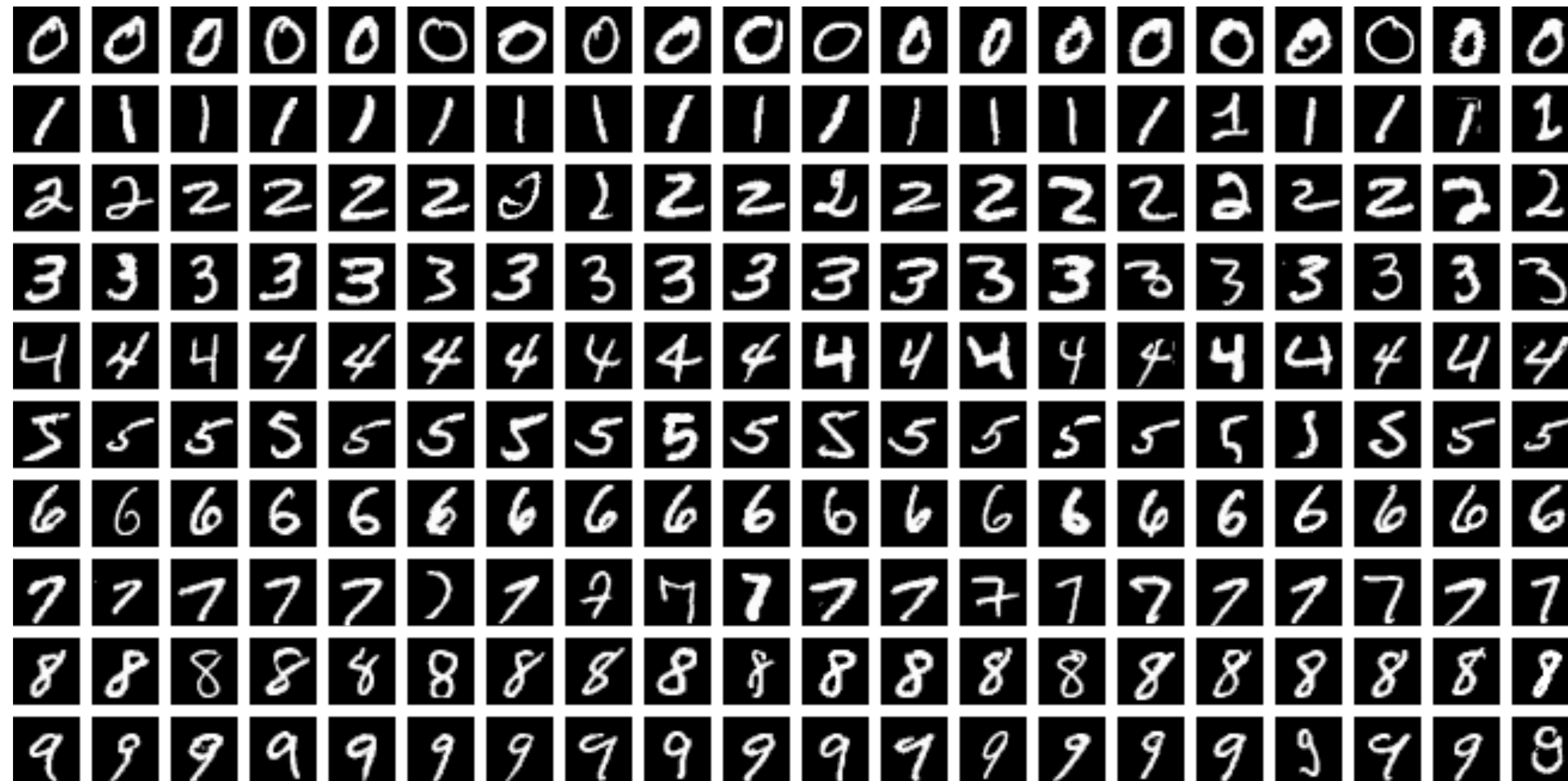
*Image recognition, language processing, product recommendation, etc.*

Machine learning approach: construct an algorithm that learns automatically from data or experience, and output a program, typically to solve a prediction problem:

Given an input  $x$ , predict the output  $y$ .



# “Traditional Programs”



Suppose we want to classify handwritten digits (example: MNIST dataset).

*How would you handwrite code to distinguish between digits?*

# Example: Image Classification

## Binary Classification

Given an input  $x$ , predict the output  $y$ .

Input  $x$ : 1000x1000 pixel image of a cat or dog.

Output  $y$ : "CAT" or "DOG"



This is a binary classification problem, where  $y$  is one of two possible outputs.

# Example: Medical Diagnosis

## Multiclass Classification

Given an input  $x$ , predict the output  $y$ .

Input  $x$ : Symptoms of an individual patient (*fever, cough, nausea...*)

Output  $y$ : Diagnosis (*pneumonia, flu, cold, bronchitis, ...*)

This is a multiclass classification problem, where  $y$  is from a *discrete* set of possible outputs.

$$\Pr(\text{pneumonia}) = 0.7$$

$$\Pr(\text{flu}) = 0.1$$

⋮

# Example: Stock Price Prediction

## Regression

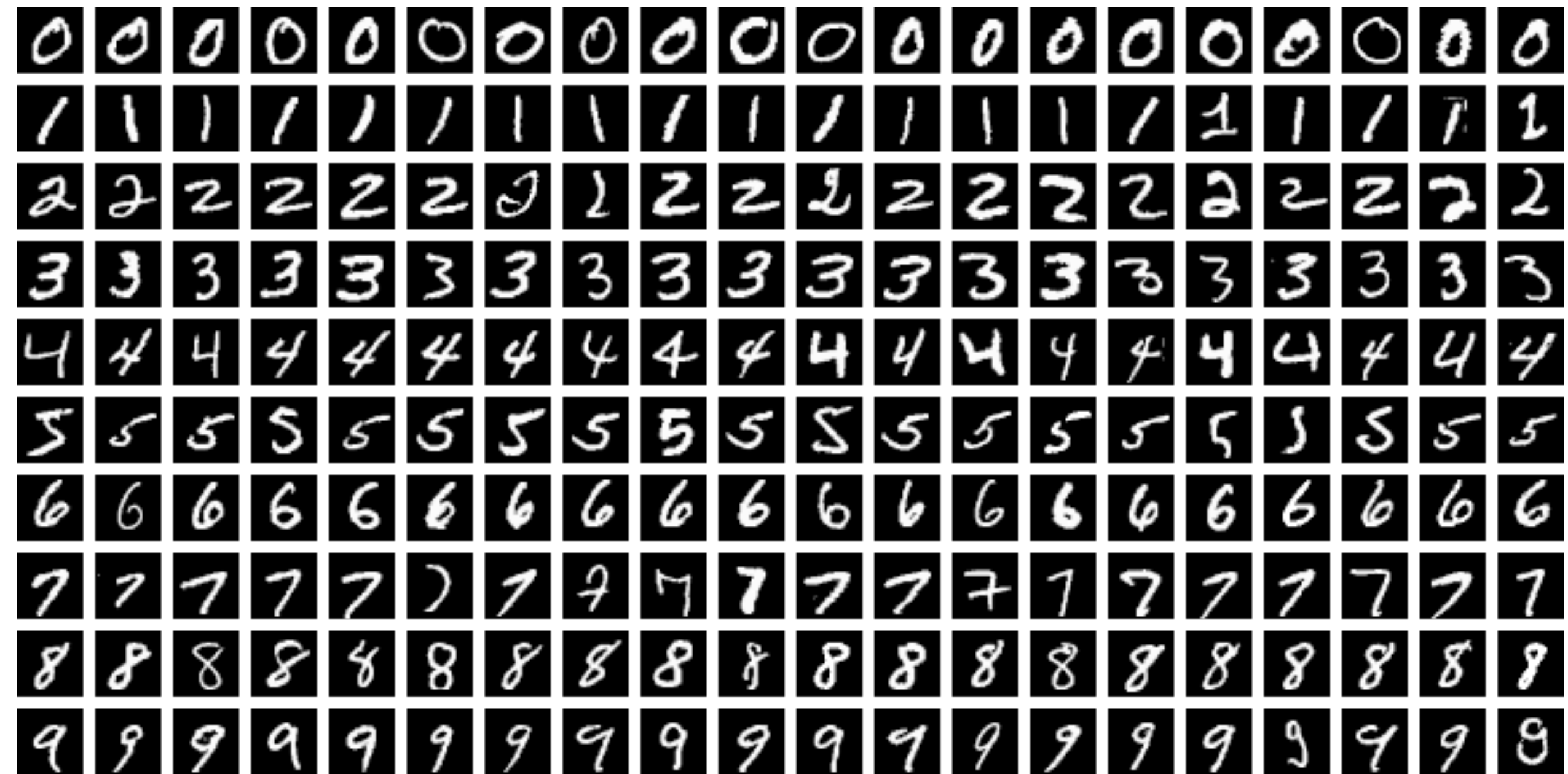
Given an input  $x$ , predict the output  $y$ .

Input  $x$ : History of stock prices, volume of stock.

Output  $y$ : Price of a stock at the close of the next day.

This is a regression problem, where  $y$  is a *continuous* output.

# Machine Learning Approach



Suppose we want to classify handwritten digits (example: MNIST dataset).

*Gather a labeled dataset of inputs and outputs.*

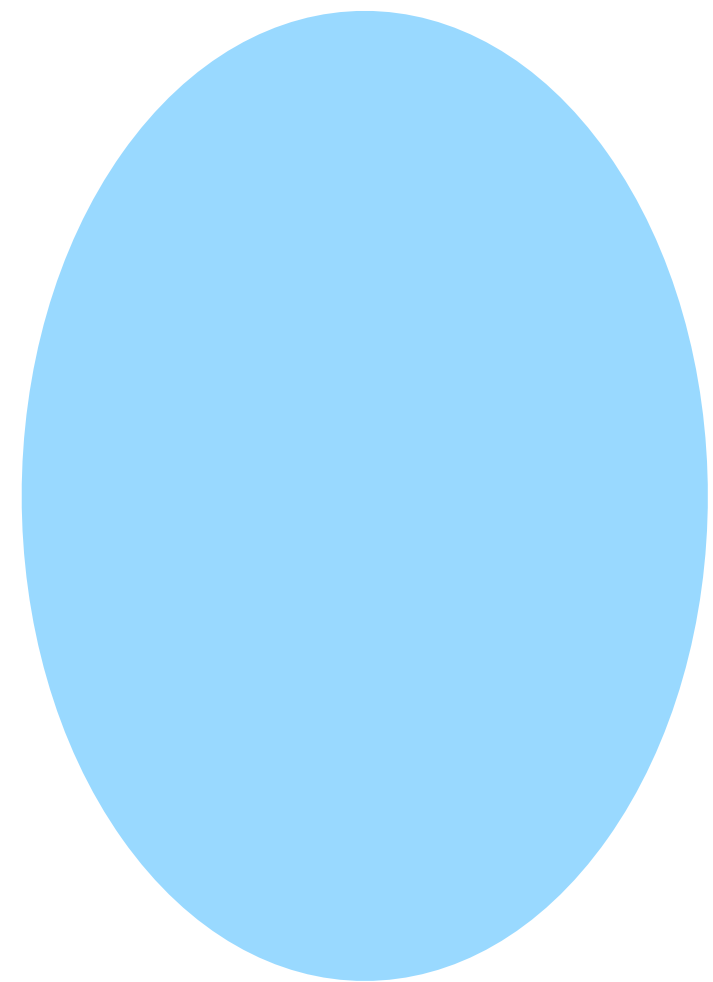
*Use this data to “automatically” find the best rule for classifying digits.*

# Supervised Machine Learning

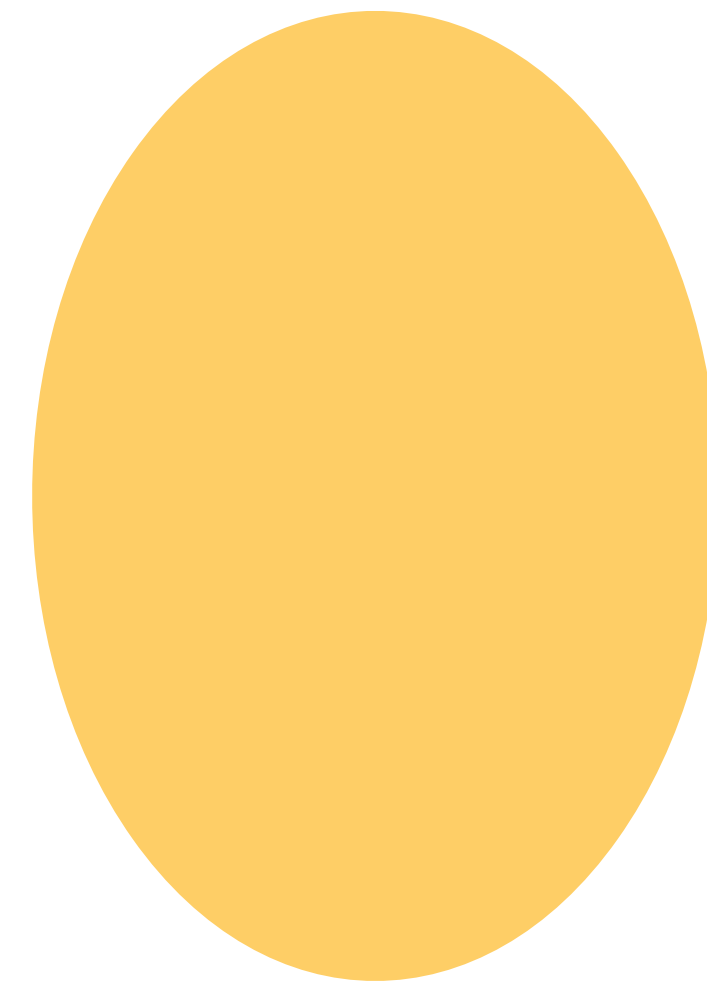
## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

The study of *making predictions* from *data*.



$\mathcal{X}$



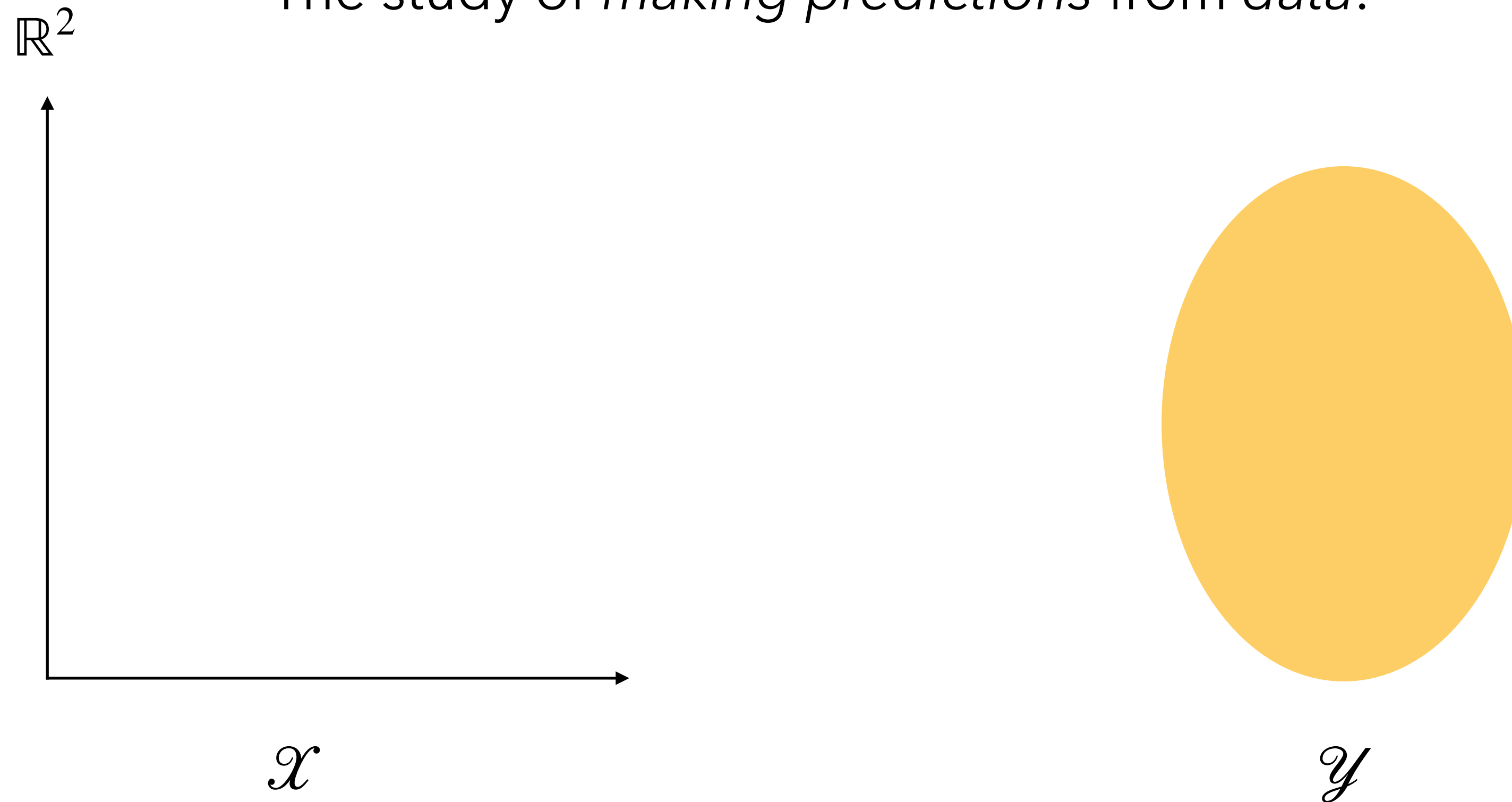
$\mathcal{Y}$

# Supervised Machine Learning

## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

The study of *making predictions from data*.

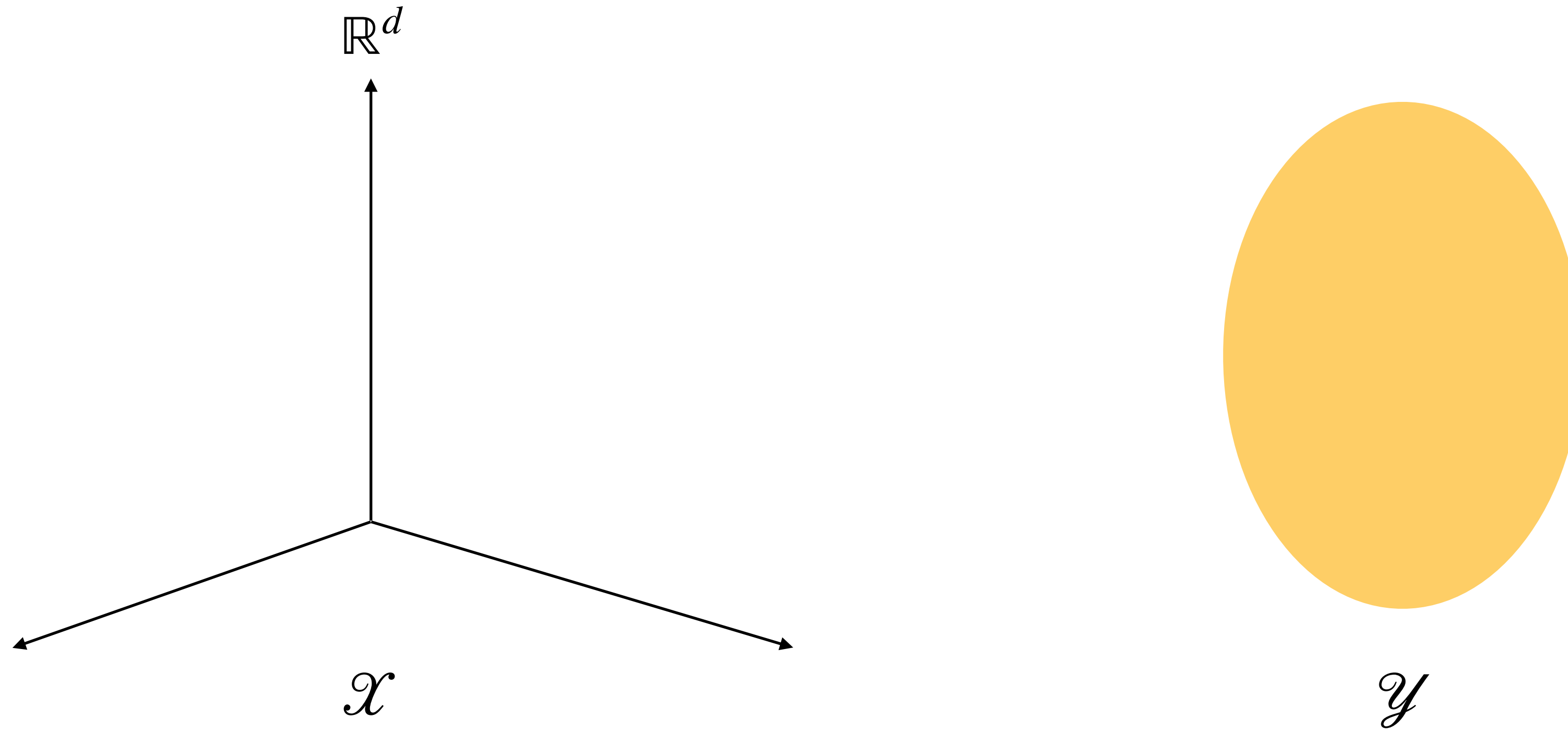


# Supervised Machine Learning

## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

The study of *making predictions* from *data*.



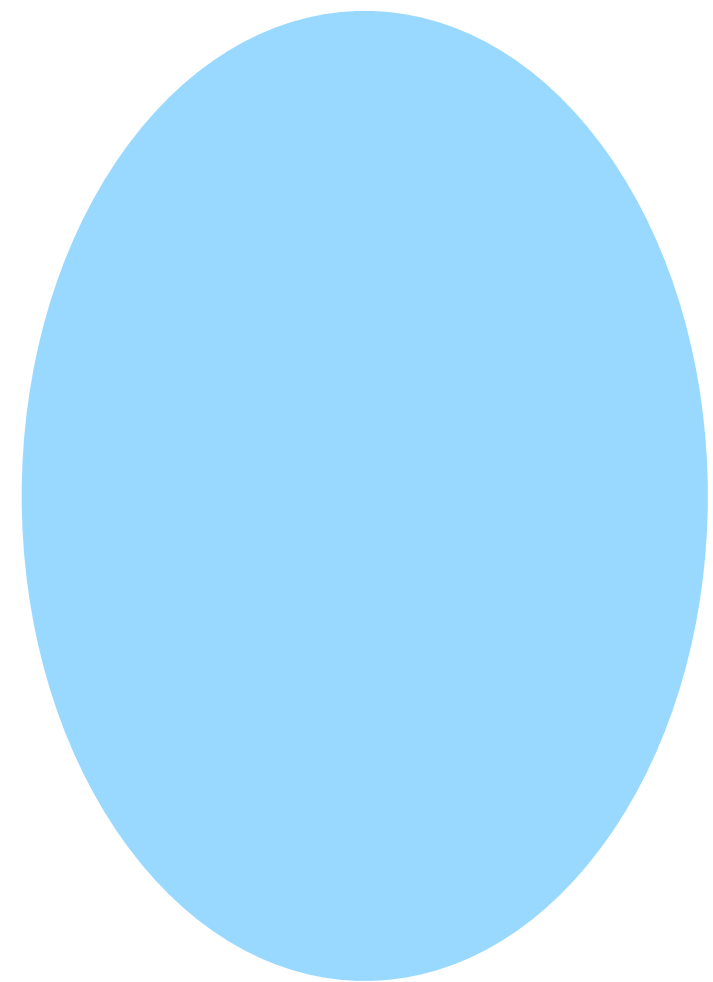


# Supervised Machine Learning

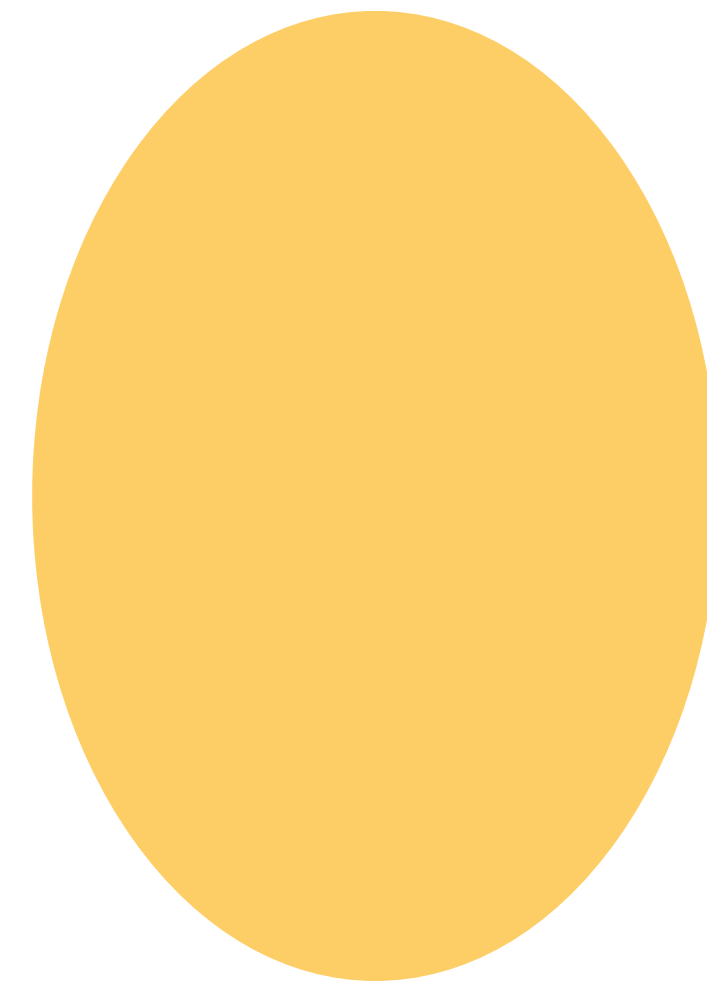
## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

The study of *making predictions* from *data*.



$\mathcal{X}$



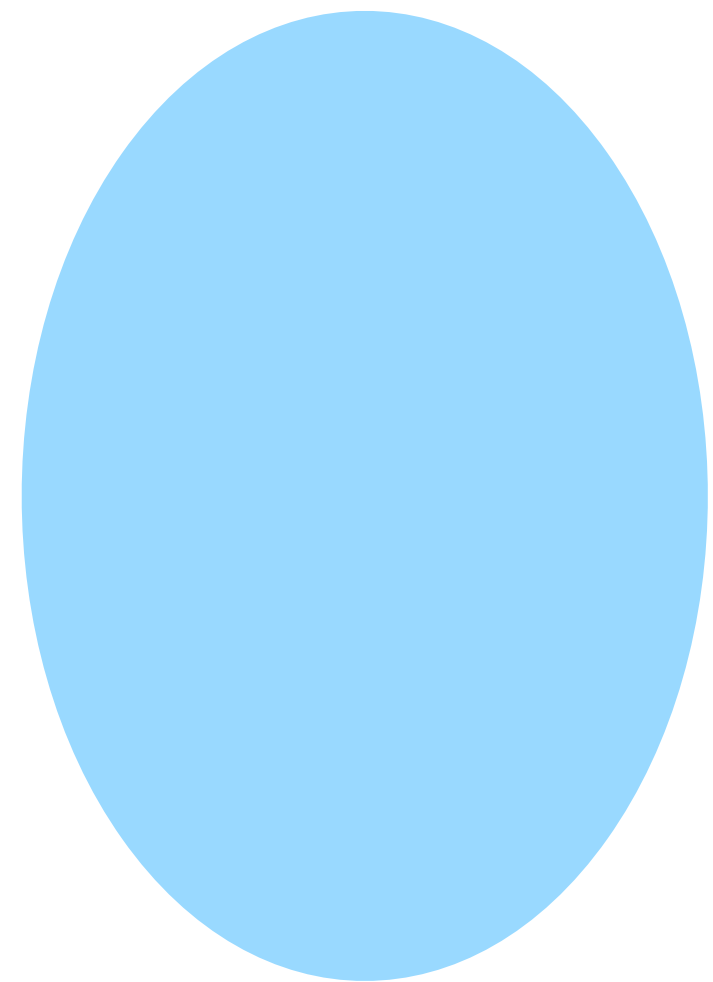
$\mathcal{Y}$

# Supervised Machine Learning

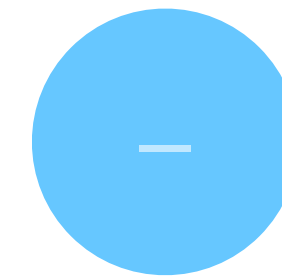
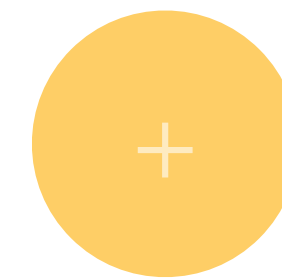
## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

The study of *making predictions* from *data*.



$\mathcal{X}$



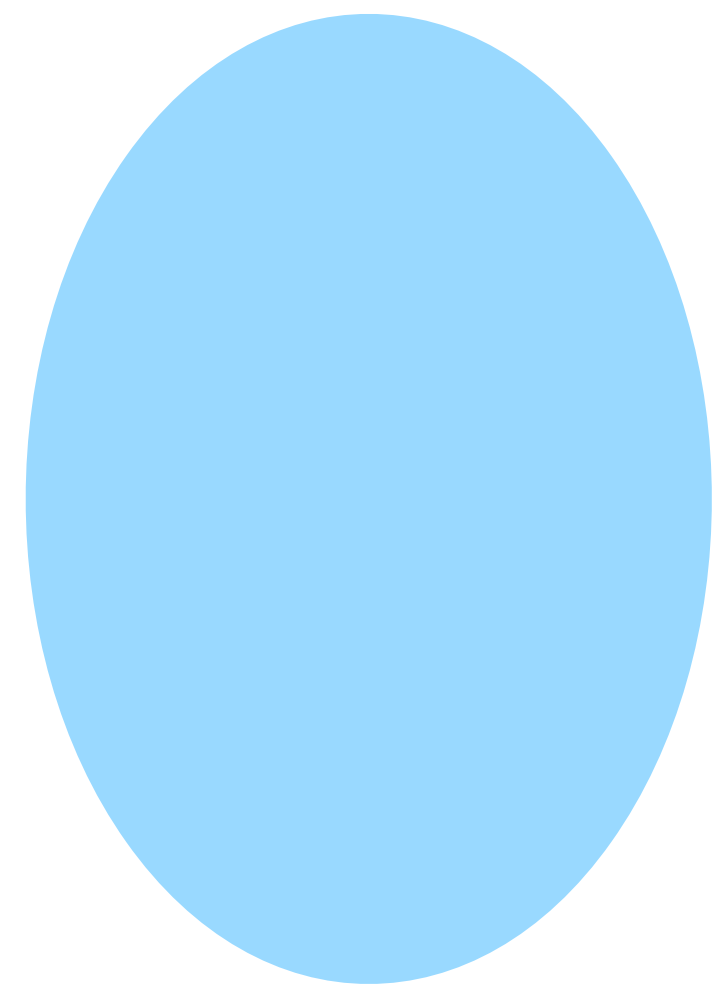
$\mathcal{Y}$

# Supervised Machine Learning

## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

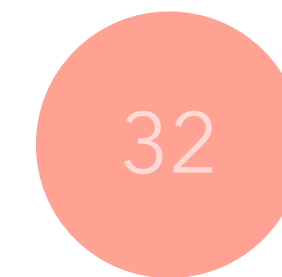
The study of *making predictions* from *data*.



$\mathcal{X}$



⋮



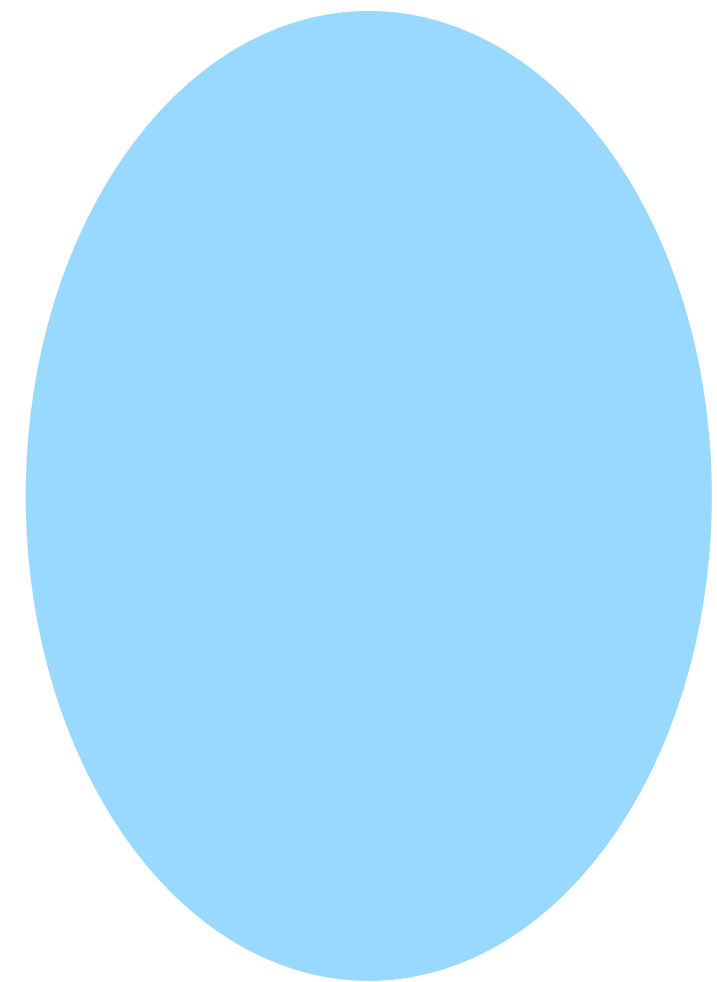
$\mathcal{Y}$

# Supervised Machine Learning

## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

The study of *making predictions* from *data*.



$\mathcal{X}$

$\mathbb{R}$

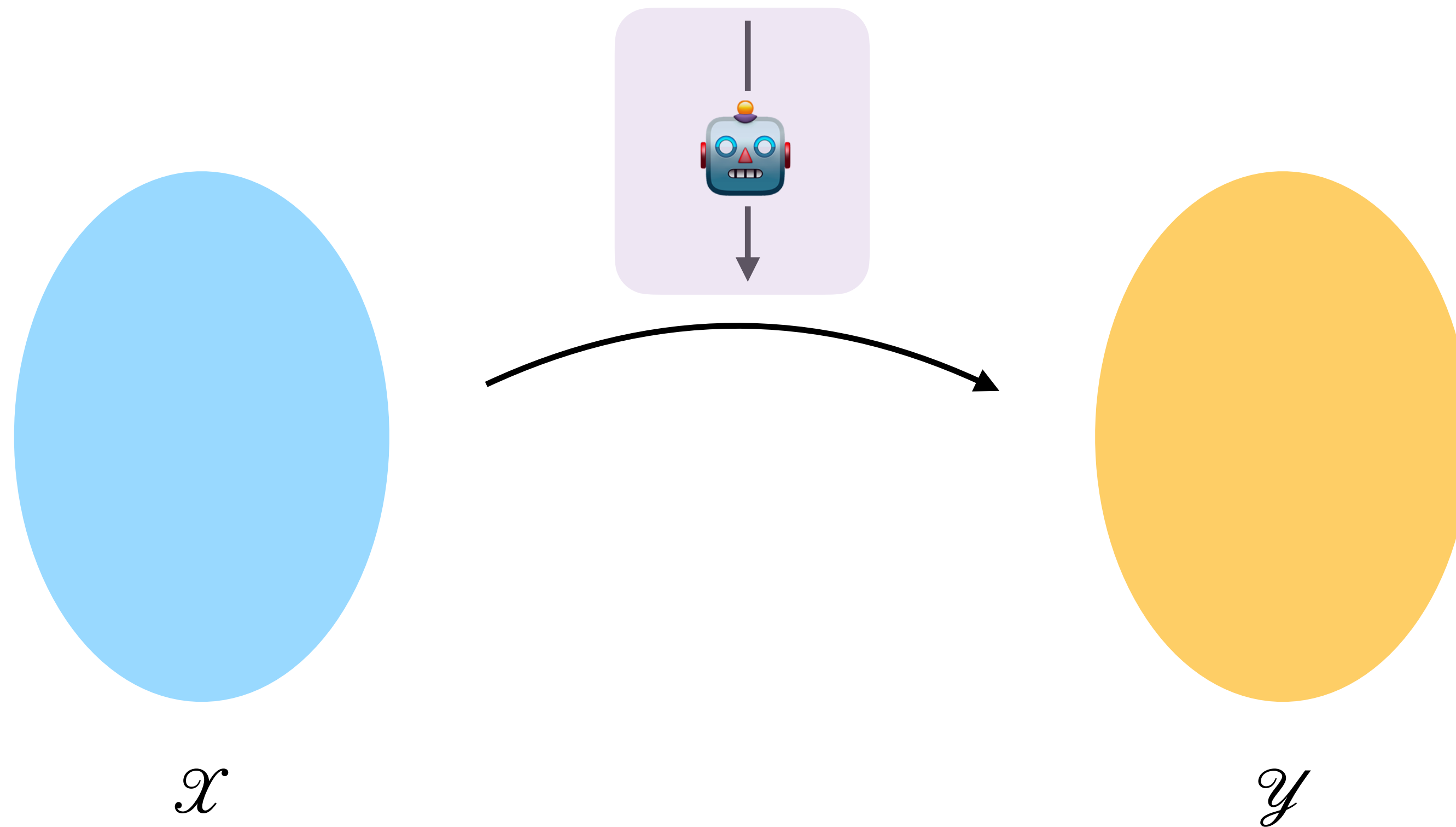
$\mathcal{Y}$

# Supervised Machine Learning

## A Definition

$$D_n := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

The study of *making predictions* from *data*.



# Supervised Learning

## Basic Pipeline

1. Collect training dataset, a collection of labeled input-output pairs.

2. Decide on the template of the hypothesis mapping that will map inputs to actions.

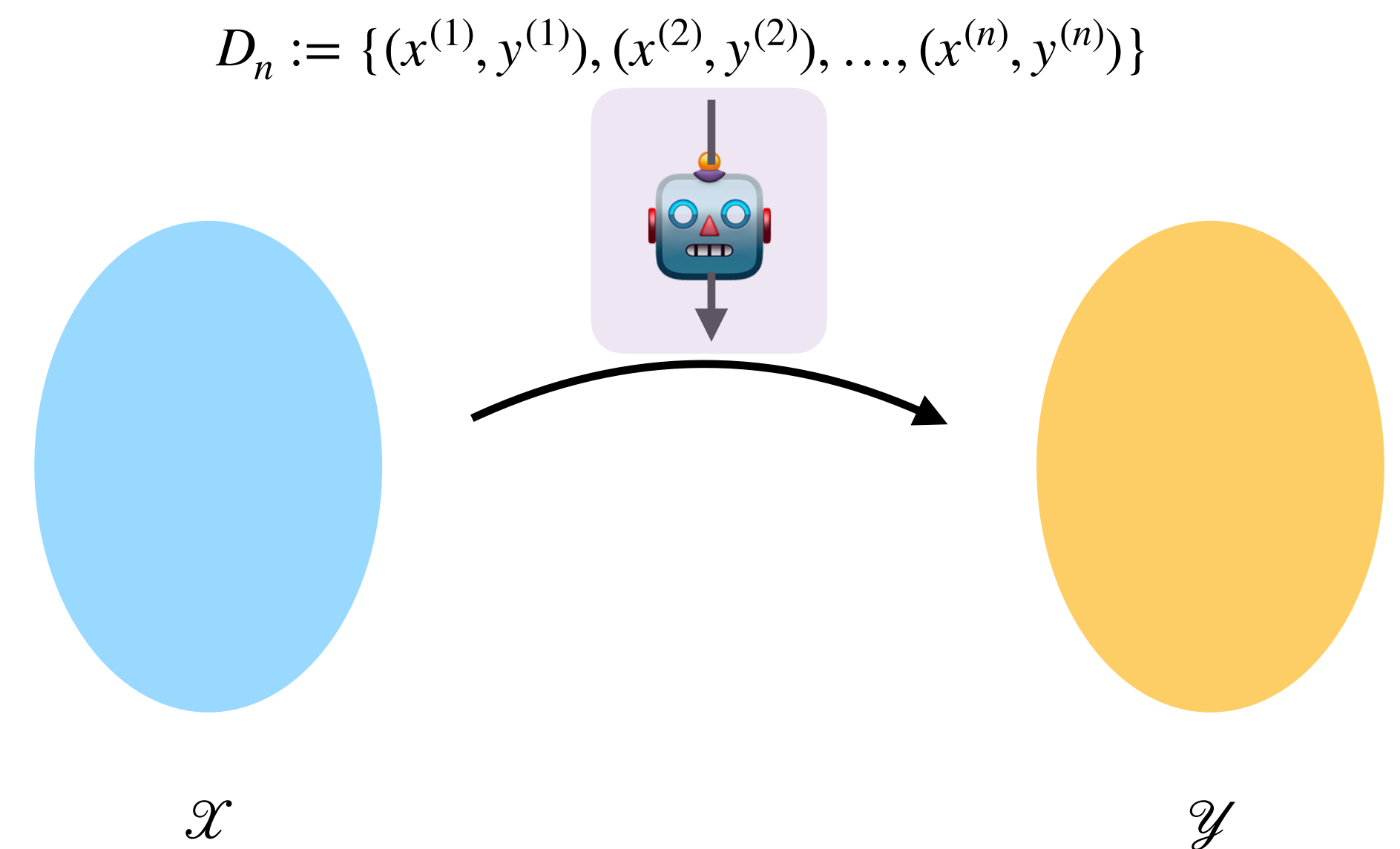
3. A learning algorithm takes the labeled training data as input and outputs a hypothesis.

4. The hypothesis predicts on new, unseen data which we hope it does well on, under a notion of loss.

Representation

Optimization

Generalization



# Outline

Course Overview and Logistics

Introduction to Machine Learning

**Statistical Learning Setup**

Statistical Learning: Bayes Risk

Statistical Learning: Empirical Risk and ERM

Statistical Learning: Hypothesis Class

Excess Risk Decomposition and Three Types of Error

# Inputs, Outcomes, and Evaluation

## The Basic Prediction Problem

The “template” of the problems we care about follow this structure:

1. Observe an input  $x \in \mathcal{X}$ .
2. Take an action  $a \in \mathcal{A}$ .
3. Observe the true outcome  $y \in \mathcal{Y}$ .
4. Evaluate the actions in relation to the outcome.



# Inputs, Outcomes, and Evaluation

## Input Space

$\mathcal{X}$  is the input space (aka feature space), where  $x \in \mathcal{X}$  is an input.

In many cases,  $\mathcal{X} = \mathbb{R}^d$ ,  $d$ -dimensional Euclidean space.

Example: Measurements in an individual's medical exam (height, weight, BP, etc.)

Example: Pixels in a 1,024x1,024 image.

Example: Words in a document of English text.

The task of finding good features for a task is known as feature engineering.

Neural networks (latter half of semester) can be seen as "automated feature engineers."

# Inputs, Outcomes, and Evaluation

## Outcome Space

$\mathcal{Y}$  is the outcome space (aka label space), where  $y \in \mathcal{Y}$  is outcome/label.

In many cases,  $\mathcal{Y}$  can be encoded as a single number.

Example:  $\mathcal{Y} = \{-1, +1\}$  (e.g. yes/no, cat/dog, etc.) in binary classification.

Example:  $\mathcal{Y} = \{1, 2, \dots, k\}$  (e.g. English word, breed of cat, etc.) in multiclass classification.

Example:  $\mathcal{Y} = \mathbb{R}$  (e.g. day's temperature, stock price, etc.) in regression.

# Inputs, Outcomes, and Evaluation

## Action Space

$\mathcal{A}$  is the action space, where  $a \in \mathcal{A}$  is an action.

Generic term for what is produced by our system (in many cases, a **prediction**).

In many cases, we will set  $\mathcal{A} = \mathcal{Y}$ .

*Example:* Produce a  $-1/1$  classification (binary classification).

*Example:* Reject hypothesis that  $\theta = 0$  (classical statistics).

*Example:* Prediction of storm location in 3 hours.

*Example:* Written English text (image captioning, speech recognition, translation).

# Inputs, Outcomes, and Evaluation

## Evaluation (Loss Functions)

A loss function  $\ell : \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$  measures the “badness” of action  $a$  with respect to  $y \in \mathcal{Y}$ .

$$(a, y) \mapsto \ell(a, y)$$

By convention, smaller loss is better, and loss is usually non-negative.

# Inputs, Outcomes, and Evaluation

## Loss Function Examples

Example.  $\mathcal{Y} = \{-1, +1\}$  or  $\mathcal{Y} = \{1, \dots, k\}$  and  $\mathcal{A} = \mathcal{Y}$ . A reasonable loss is zero-one loss.

$$\ell(a, y) = \begin{cases} 1 & \text{if } a \neq y \\ 0 & \text{otherwise} \end{cases} \quad \text{or, shorthand: } \ell(\hat{y}, y) := \mathbf{1}\{a \neq y\}$$

Example.  $\mathcal{Y} = \mathbb{R}$  and  $\mathcal{A} = \mathcal{Y}$ . A reasonable loss is the squared loss.

$$\ell(\hat{y}, y) = (a - y)^2.$$

# Inputs, Outcomes, and Evaluation

## The Basic Prediction Problem

The “template” of the problems we care about follow this structure:

1. Observe an input  $x \in \mathcal{X}$ .
  2. Take an action  $a \in \mathcal{A}$ .
  3. Observe the true outcome  $y \in \mathcal{Y}$ .
  4. Evaluate the actions in relation to the outcome.
- We will construct prediction functions to do this.
-

# Hypothesis

## Definition & Goal

A hypothesis (aka predictor/prediction function) is a function  $h : \mathcal{X} \rightarrow \mathcal{A}$  that takes inputs/features  $x$  and maps to an action  $h(x)$ .

The loss of action  $a$  in context of  $y$ :  $\ell(a, y)$

The loss of action  $h(x)$  in context of  $y$ :  $\ell(h(x), y)$

**Goal.** Turn our prediction problem into an *optimization problem*.

**Question:** How do we evaluate a prediction function  $h$  *as a whole*?

# Data Generating Distribution

## Definition & Goal

$\ell(h(x), y)$  is the quality of  $h$  for a single  $(x, y)$ .

But how can we evaluate  $h$  over *all* of  $\mathcal{X} \times \mathcal{Y}$ ?

We will assume that there exists a data-generating distribution  $P_{\mathcal{X} \times \mathcal{Y}}$  over  $\mathcal{X} \times \mathcal{Y}$ .

Any input/output pair  $(x, y)$  is assumed generated i.i.d. (independent and identically distributed) from  $P_{\mathcal{X} \times \mathcal{Y}}$  (this is the i.i.d. assumption).

*In machine learning,  $P_{\mathcal{X} \times \mathcal{Y}}$  is assumed to be unknown!*



# Data Generating Distribution

Considering what is random

🛑 and consider: *What is random in this problem?*

Input/output pairs  $(x, y)$  are random variables from joint distribution  $P_{x \times y}$ .

The inputs  $x$  are random variables from marginal distribution  $P_x$ .

For any given  $x$ , the  $y$  are random variables from the conditional distribution  $P_{y|x}$ .

For a fixed hypothesis  $h$ , the loss  $\ell(h(x), y)$  is a random variable

# Evaluation, Overall

## Definition of Risk

$\ell(h(x), y)$  is the quality of  $h$  for a single  $(x, y)$ .

But how can we evaluate  $h$  over *all* of  $\mathcal{X} \times \mathcal{Y}$ ?

The risk of a hypothesis  $h : \mathcal{X} \rightarrow \mathcal{A}$  is the expected loss of  $h$  over  $P_{\mathcal{X} \times \mathcal{Y}}$ :

$$R(h) := \mathbb{E}_{(x,y) \sim P_{\mathcal{X} \times \mathcal{Y}}} [\ell(h(x), y)]$$

*Our ultimate goal will typically be to minimize this quantity!*

# Statistical Learning Setup

## Summary of Characters So Far

1. Observe an input  $x \in \mathcal{X}$ .
2. Predict an action  $a \in \mathcal{A}$ .
3. Observe the true outcome  $y \in \mathcal{Y}$ .
4. Evaluate the actions in relation to the outcome.

$\mathcal{X}$  is the input space (e.g.  $\mathbb{R}^d$ , pixels, words).

$\mathcal{Y}$  is the output space (e.g.  $\{0,1\}$  or  $\mathbb{R}$ ).

$\mathcal{A}$  is the action space (e.g. prediction of  $y$ , some decision).

$h : \mathcal{X} \rightarrow \mathcal{A}$  is a hypothesis to generate action  $h(x)$ .

Evaluate  $h$  with loss function  $\ell : \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$ .

$\ell(h(x), y)$  evaluates  $h$  on  $(x, y)$ .

$R(h) = \mathbb{E}_{(x,y) \sim P_{\mathcal{X} \times \mathcal{Y}}}[\ell(h(x), y)]$  is risk of  $h$ .

# Outline

Course Overview and Logistics

Introduction to Machine Learning

Statistical Learning Setup

**Statistical Learning: Bayes Risk**

Statistical Learning: Empirical Risk and ERM

Statistical Learning: Hypothesis Class

Excess Risk Decomposition and Three Types of Error

# Minimizing Risk

What's the smallest possible risk?

$$R(h) := \mathbb{E}_{(x,y) \sim P_{x \times y}} [\ell(h(x), y)]$$

*Our ultimate goal will typically be to minimize this quantity!*

# Bayes Risk

## Definition

The Bayes hypothesis  $h^* : \mathcal{X} \rightarrow \mathcal{A}$  is a function that achieves the *minimal risk* among all possible functions

$$h^* \in \arg \min_h R(h)$$

where the minimum is taken over all possible functions from  $\mathcal{X}$  to  $\mathcal{A}$ .

The risk of  $h^*$  is called the Bayes risk.

# Bayes Risk

## Example: Binary Classification

Binary classification:  $\mathcal{Y} = \{0,1\}$  and  $\mathcal{A} = \{0,1\}$ .

Zero-one loss:  $\ell(\hat{y}, y) = \mathbf{1}\{\hat{y} \neq y\} := \begin{cases} 1 & \hat{y} \neq y \\ 0 & \text{otherwise} \end{cases}$  (when  $\mathcal{A} = \mathcal{Y}$ , use  $\hat{y} \in \mathcal{A}$  as shorthand)

$$R(h) = \mathbb{E}[\mathbf{1}\{\hat{y} \neq y\}] = 1 \cdot \Pr(h(x) \neq y) + 0 \cdot \Pr(h(x) = y)$$

$$\implies R(h) = \Pr(h(x) \neq y).$$

Therefore, the Bayes hypothesis returns the most likely label:

$$h^*(x) = \begin{cases} 1 & \text{if } \Pr(y = 1 \mid x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

# Bayes Risk

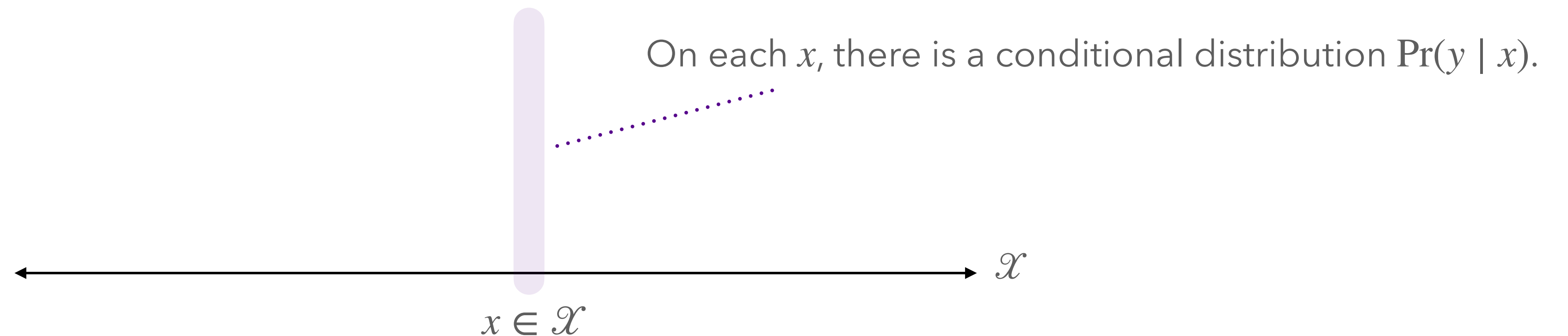
## Example: Binary Classification

$$\implies R(h) = \Pr(h(x) \neq y).$$

Therefore, the Bayes hypothesis returns the most likely label:

$$h^*(x) = \begin{cases} 1 & \text{if } \Pr(y = 1 \mid x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Minimizing  $R(h)$  over every possible function allows us to define  $h^*$  "pointwise" for  $x \in \mathcal{X}$ .





# Bayes Risk

Example: Squared Loss Regression

Regression:  $\mathcal{Y} = \mathbb{R}$  and  $\mathcal{A} = \mathbb{R}$ .

Squared loss:  $\ell(a, y) = (a - y)^2$

$$R(h) := \mathbb{E}[(h(x) - y)^2]$$

Can show that the Bayes hypothesis is:

$$h^*(x) = \mathbb{E}[y \mid x]$$

# Bayes Risk

## Example: Binary Classification

$$\implies R(h) = \Pr(h(x) \neq y).$$

Therefore, the Bayes hypothesis returns the most likely label:

$$h^*(x) = \begin{cases} 1 & \text{if } \Pr(y = 1 \mid x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Problem: We don't know what  $P_{x \times y}$  is in a machine learning problem!

# Minimizing Risk

What's the smallest possible risk?

$$R(h) := \mathbb{E}_{(x,y) \sim P_{\mathcal{X} \times \mathcal{Y}}} [\ell(h(x), y)]$$

*Our ultimate goal will typically be to minimize this quantity!*

Problem: We don't know what  $P_{\mathcal{X} \times \mathcal{Y}}$  is in a machine learning problem!

# Outline

Course Overview and Logistics

Introduction to Machine Learning

Statistical Learning Setup

Statistical Learning: Bayes Risk

**Statistical Learning: Empirical Risk and ERM**

Statistical Learning: Hypothesis Class

Excess Risk Decomposition and Three Types of Error

# Minimizing Risk

What's the smallest possible risk?

$$R(h) := \mathbb{E}_{(x,y) \sim P_{\mathcal{X} \times \mathcal{Y}}} [\ell(h(x), y)]$$

*Our ultimate goal will typically be to minimize this quantity!*

Problem: We don't know what  $P_{\mathcal{X} \times \mathcal{Y}}$  is in a machine learning problem!

But we assume that we have a dataset of i.i.d. samples:

$$D_n := \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$$

# Law of Large Numbers

## Reminder

If  $z_1, \dots, z_n$  are i.i.d. random variables with expected value  $\mathbb{E}[z]$ , then

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n z_i = \mathbb{E}[z], \text{ with probability 1.}$$

If  $D_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$  is drawn i.i.d. from  $P_{\mathcal{X} \times \mathcal{Y}}$ , then for a fixed  $h : \mathcal{X} \rightarrow \mathcal{A}$  and  $\ell : \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$ ,

$$\ell(h(x^{(1)}), y^{(1)}), \dots, \ell(h(x^{(n)}), y^{(n)})$$

are all random variables...

# Empirical Risk

## Definition

Let  $D_n := \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$  be drawn i.i.d. from  $P_{\mathcal{X} \times \mathcal{Y}}$ .

The empirical risk of  $h : \mathcal{X} \rightarrow \mathcal{A}$  with respect to  $D_n$  is

$$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \ell(h(x^{(i)}), y^{(i)}).$$

By the strong law of large numbers,

$$\lim_{n \rightarrow \infty} \hat{R}_n(h) = R(h) \text{ almost surely.}$$

But, in practice, we only have a finite sample.



# Empirical Risk Minimization

## Definition

The empirical risk of  $h : \mathcal{X} \rightarrow \mathcal{A}$  with respect to  $D_n$  is

$$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \ell(h(x^{(i)}), y^{(i)}).$$

The empirical risk minimizer (ERM) (over all functions  $h : \mathcal{X} \rightarrow \mathcal{A}$ ) is a function  $\hat{h}$  satisfying

$$\hat{h} \in \arg \min_h \hat{R}_n(h).$$

Is this a good proxy?

In an ideal world, we want the Bayes hypothesis:

$$h^* \in \arg \min_h R(h).$$

# Empirical Risk Minimization

## Example

$P_{\mathcal{X}} = \text{Unif}([0,1])$  and  $Y = 1$  always.

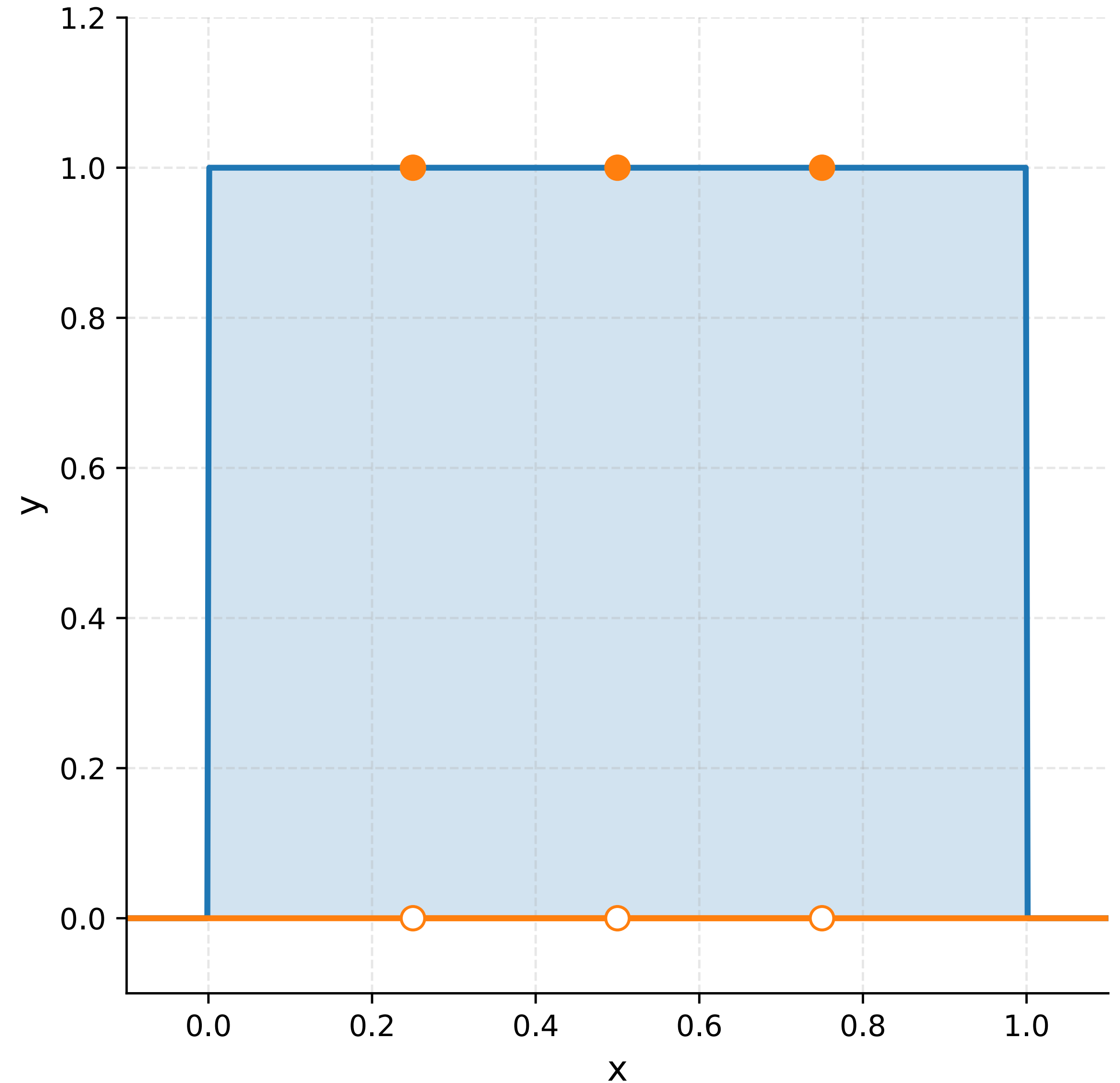
Draw i.i.d. sample of size  $n = 3$ :

$$D_n = \{(0.25,1), (0.5,1), (0.75,1)\}.$$

Under  $\ell(\hat{y}, y) = \mathbf{1}\{\hat{y} \neq y\}$  (zero-one loss):

$$\hat{h}(x) = \begin{cases} 1 & \text{if } x \in \{0.25, 0.5, 0.75\} \\ 0 & \text{otherwise} \end{cases}$$

This is an ERM.



# Empirical Risk Minimization

## Example

$P_{\mathcal{X}} = \text{Unif}([0,1])$  and  $Y = 1$  always.

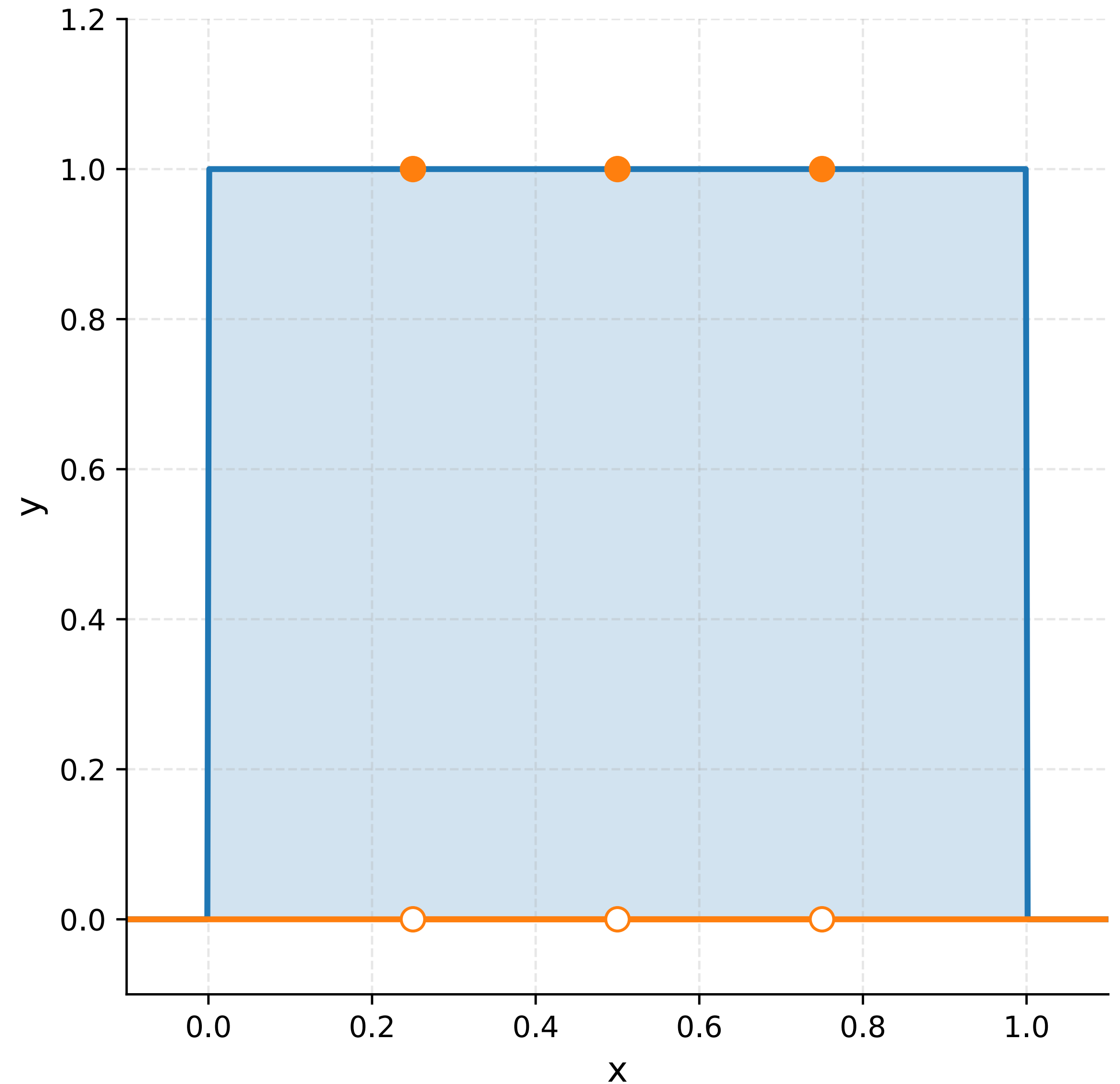
Draw i.i.d. sample of size  $n = 3$ :

$$D_n = \{(0.25, 1), (0.5, 1), (0.75, 1)\}.$$

Under  $\ell(\hat{y}, y) = (\hat{y} - y)^2$  (squared loss):

$$\hat{h}(x) = \begin{cases} 1 & \text{if } x \in \{0.25, 0.5, 0.75\} \\ 0 & \text{otherwise} \end{cases}$$

This is an ERM.



# Empirical Risk Minimization

Example: Gap with true risk

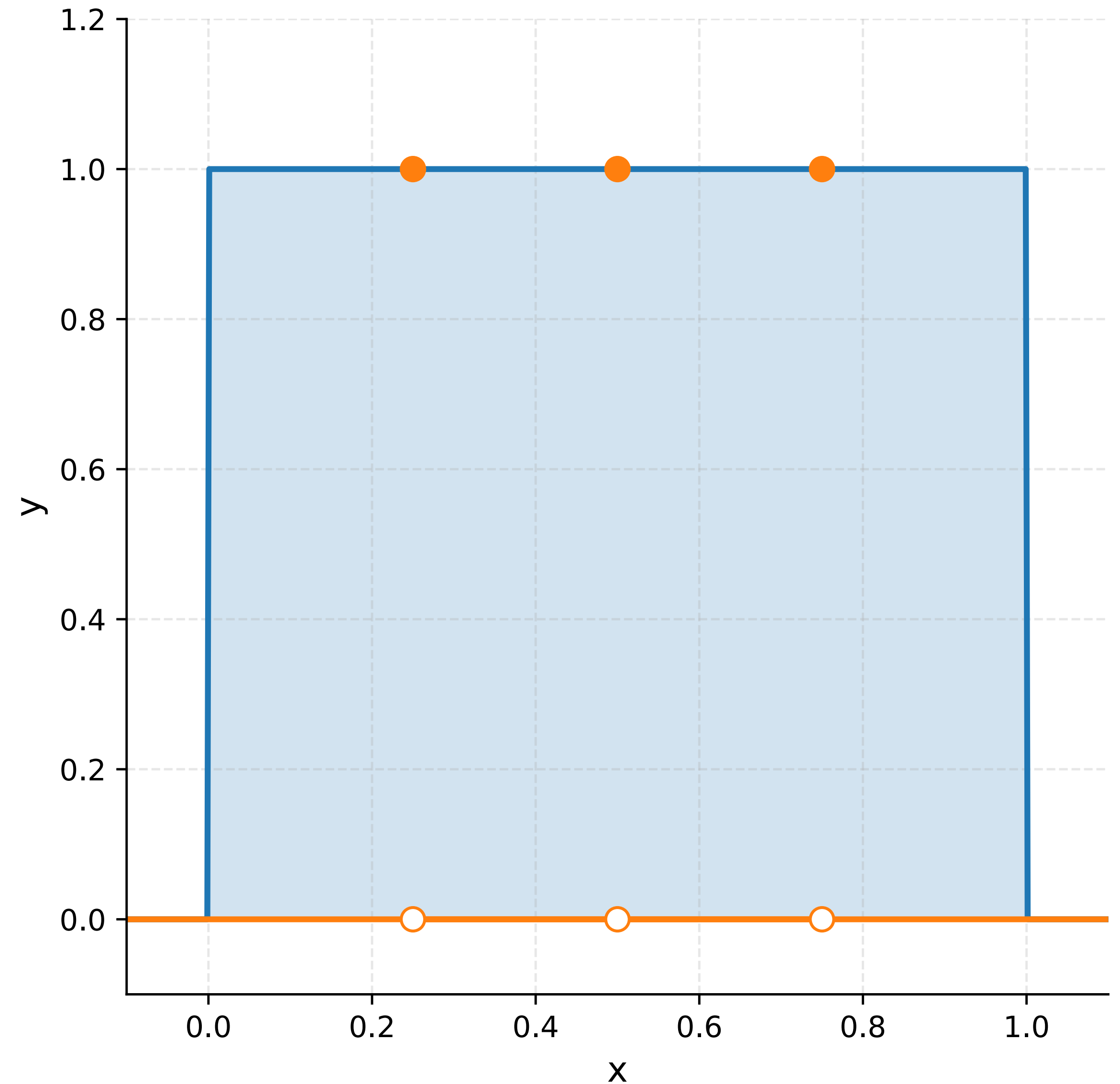
$$\hat{h}(x) = \begin{cases} 1 & \text{if } x \in \{0.25, 0.5, 0.75\} \\ 0 & \text{otherwise} \end{cases}$$

Empirical risk under zero-one loss:

$$\hat{R}_n(\hat{h}) = \frac{1}{3} \sum_{i=1}^3 \mathbf{1}\{\hat{h}(x^{(i)}) \neq y^{(i)}\} = 0$$

True risk under zero-one loss:

$$R(\hat{h}) = \mathbb{E}[\mathbf{1}\{\hat{h}(x) \neq y\}] = \Pr(\hat{h}(x) \neq y) = 1$$



# Empirical Risk Minimization

What went wrong?

$$D_n = \{(0.25,1), (0.5,1), (0.75,1)\}$$

$$\hat{h}(x) = \begin{cases} 1 & \text{if } x \in \{0.25, 0.5, 0.75\} \\ 0 & \text{otherwise} \end{cases}$$

This failed spectacularly because  $\hat{h}$  just memorized the data.

In ML, we want our hypotheses to **generalize** from training data to new data.

In order to do this, we need to smooth things out:

*Model how information is structured in input space  $\mathcal{X}$  to unobserved parts of  $\mathcal{X}$ !*

# Outline

Course Overview and Logistics

Introduction to Machine Learning

Statistical Learning Setup

Statistical Learning: Bayes Risk

Statistical Learning: Empirical Risk and ERM

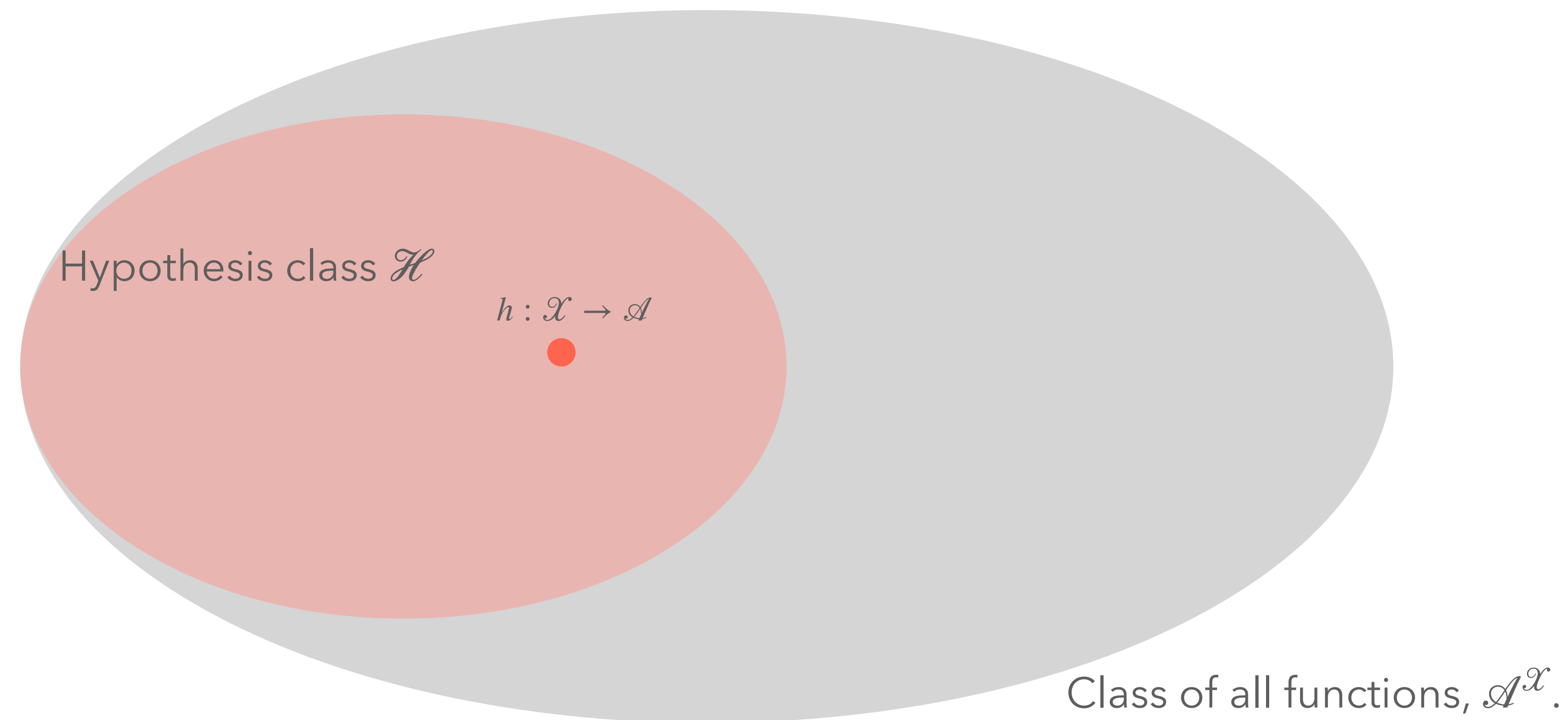
**Statistical Learning: Hypothesis Class**

Excess Risk Decomposition and Three Types of Error

# Hypothesis Class

## Definition

A hypothesis class is a set of functions  $\mathcal{H} \subseteq \mathcal{A}^{\mathcal{X}}$  where we will search for  $h$ .



# Hypothesis Class

## Example

$\mathcal{X} = \mathbb{R}^3$ , with  $x \in \mathcal{X}$  encoded as  $x = (\text{midterm}, \text{hours studied}, \text{hours slept})$ .

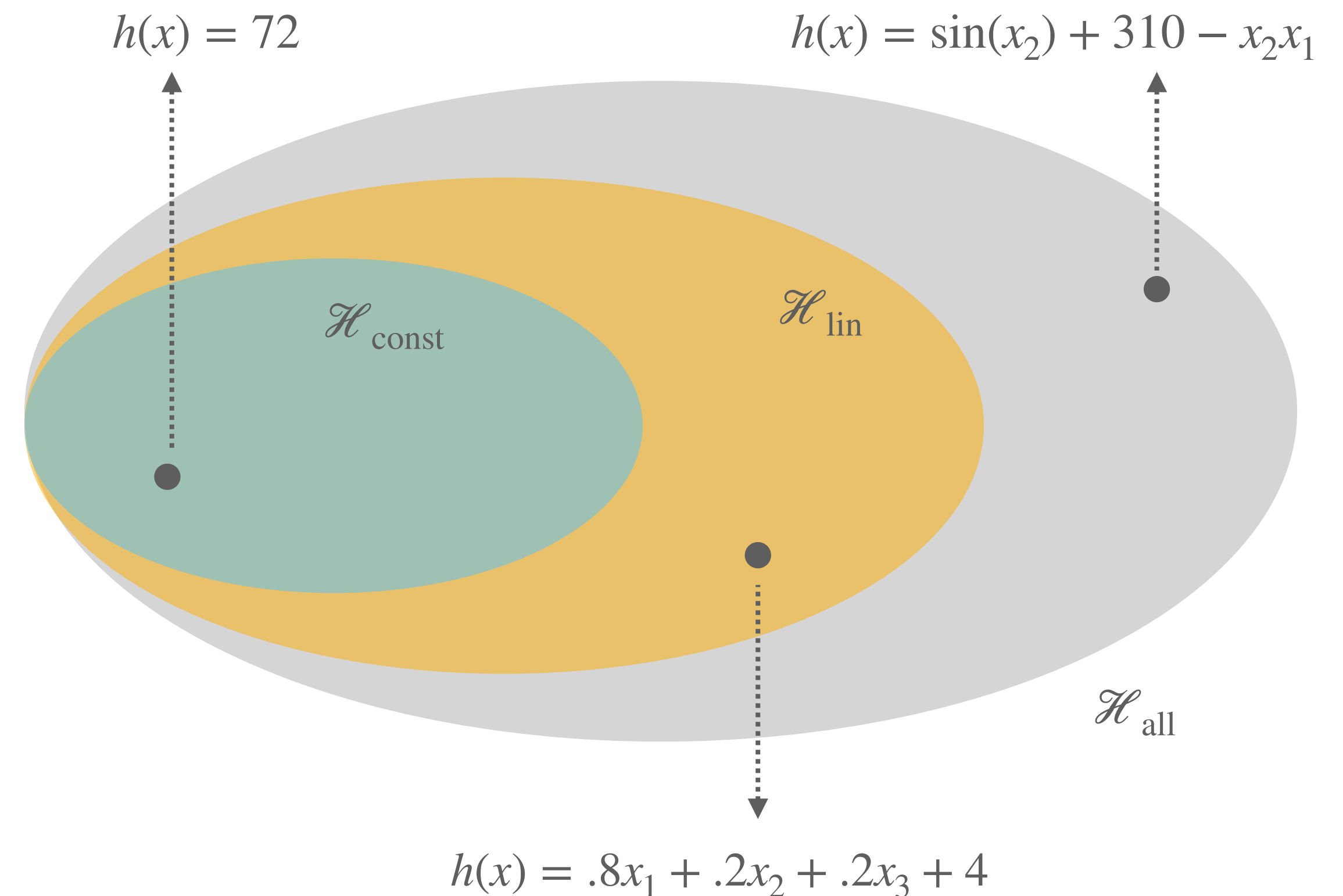
$\mathcal{A} = \mathbb{R}$ , where  $a \in \mathcal{A}$  is final exam score.

Possible hypothesis classes:

$$\mathcal{H}_{\text{const}} = \{x \mapsto b : b \in \mathbb{R}\}$$

$$\mathcal{H}_{\text{lin}} = \{x \mapsto w^\top x + b : w \in \mathbb{R}^3, b \in \mathbb{R}\}$$

$$\mathcal{H}_{\text{all}} = \{\mathbb{R}^3 \mapsto \mathbb{R}\}$$





# Empirical Risk Minimization

Example:  $\mathcal{H}_{\text{const}}$

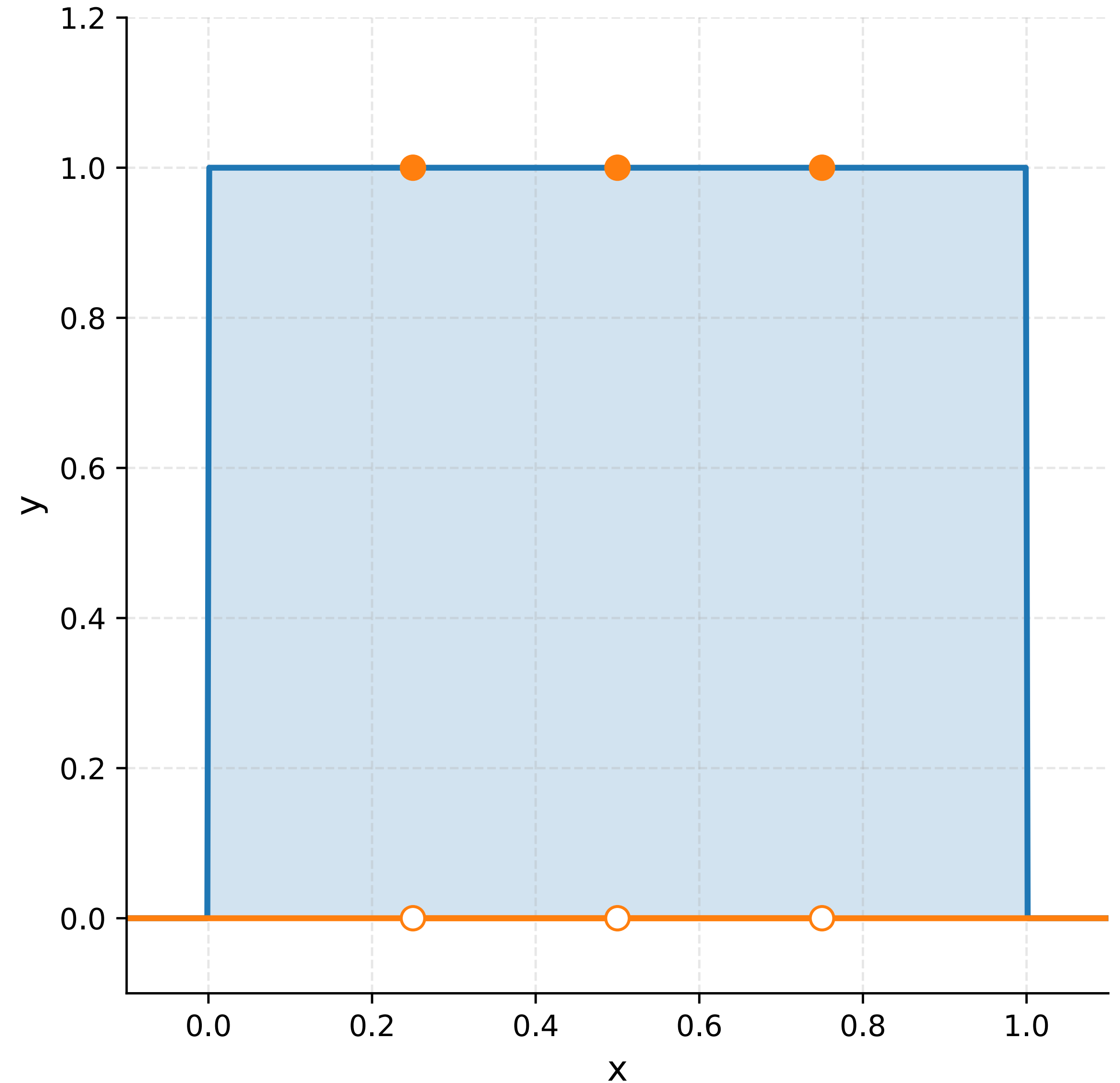
$P_{\mathcal{X}} = \text{Unif}([0,1])$  and  $Y = 1$  always.

$$D_n = \{(0.25,1), (0.5,1), (0.75,1)\}.$$

$$\hat{h}(x) = \begin{cases} 1 & \text{if } x \in \{0.25, 0.5, 0.75\} \\ 0 & \text{otherwise} \end{cases}$$

ERM over  $\mathcal{H}_{\text{const}} = \{x \mapsto b : b \in \mathbb{R}\}$ :

$$\hat{h}(x) = 1$$



# Hypothesis Class

## Definition

A hypothesis class is a set of functions  $\mathcal{H} \subseteq \mathcal{A}^{\mathcal{X}}$  where we will search for  $h$ .

Fixed *before* the learning process.

Encodes assumptions about the relationship of  $x$  to  $y$ .

Should be easy to work with (i.e. we have efficient algorithms to search over  $\mathcal{H}$ ).

# Risk Minimization

With a hypothesis class

The empirical risk minimizer (ERM) in  $\mathcal{H}$  is a function  $\hat{h}$  satisfying

$$\hat{h} \in \arg \min_{h \in \mathcal{H}} \hat{R}_n(h).$$

The risk minimizer in  $\mathcal{H}$  is a function  $\hat{h}$  satisfying

$$h_{\mathcal{H}}^* \in \arg \min_{h \in \mathcal{H}} R(h)$$

The Bayes hypothesis  $h^*$  is a function with *minimal risk* among all functions

$$h^* \in \arg \min_h R(h)$$

# Outline

Course Overview and Logistics

Introduction to Machine Learning

Statistical Learning Setup

Statistical Learning: Bayes Risk

Statistical Learning: Empirical Risk and ERM

Statistical Learning: Hypothesis Class

**Excess Risk Decomposition and Three Types of Error**

# Excess Risk

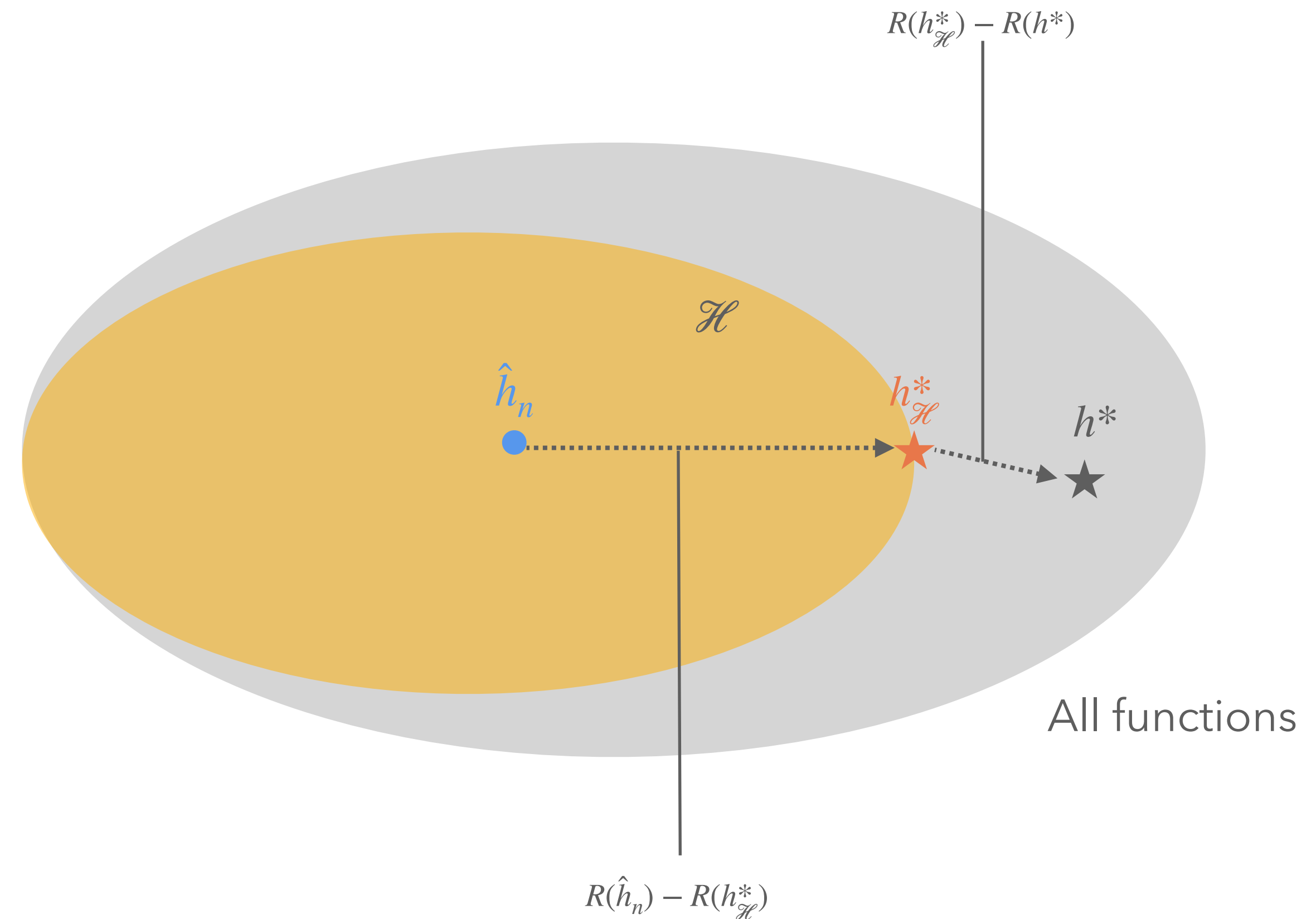
## Definition

$$h^* \in \operatorname{argmin}_h \underbrace{\mathbb{E}_{(x,y) \sim P_{x \times y}} [\ell(h(x), y)]}_{R(h)}$$

$$h_{\mathcal{H}}^* \in \operatorname{argmin}_{h \in \mathcal{H}} \underbrace{\mathbb{E}_{(x,y) \sim P_{x \times y}} [\ell(h(x), y)]}_{R(h)}$$

$$\hat{h}_n \in \operatorname{argmin}_{h \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(h(x^{(i)}), y^{(i)})}_{\hat{R}_n(h)}$$

The excess risk of  $h$  is how far  $h$  is from  $h^*$ :  
 $R(h) - R(h^*)$ .



# Excess Risk

## Decomposition

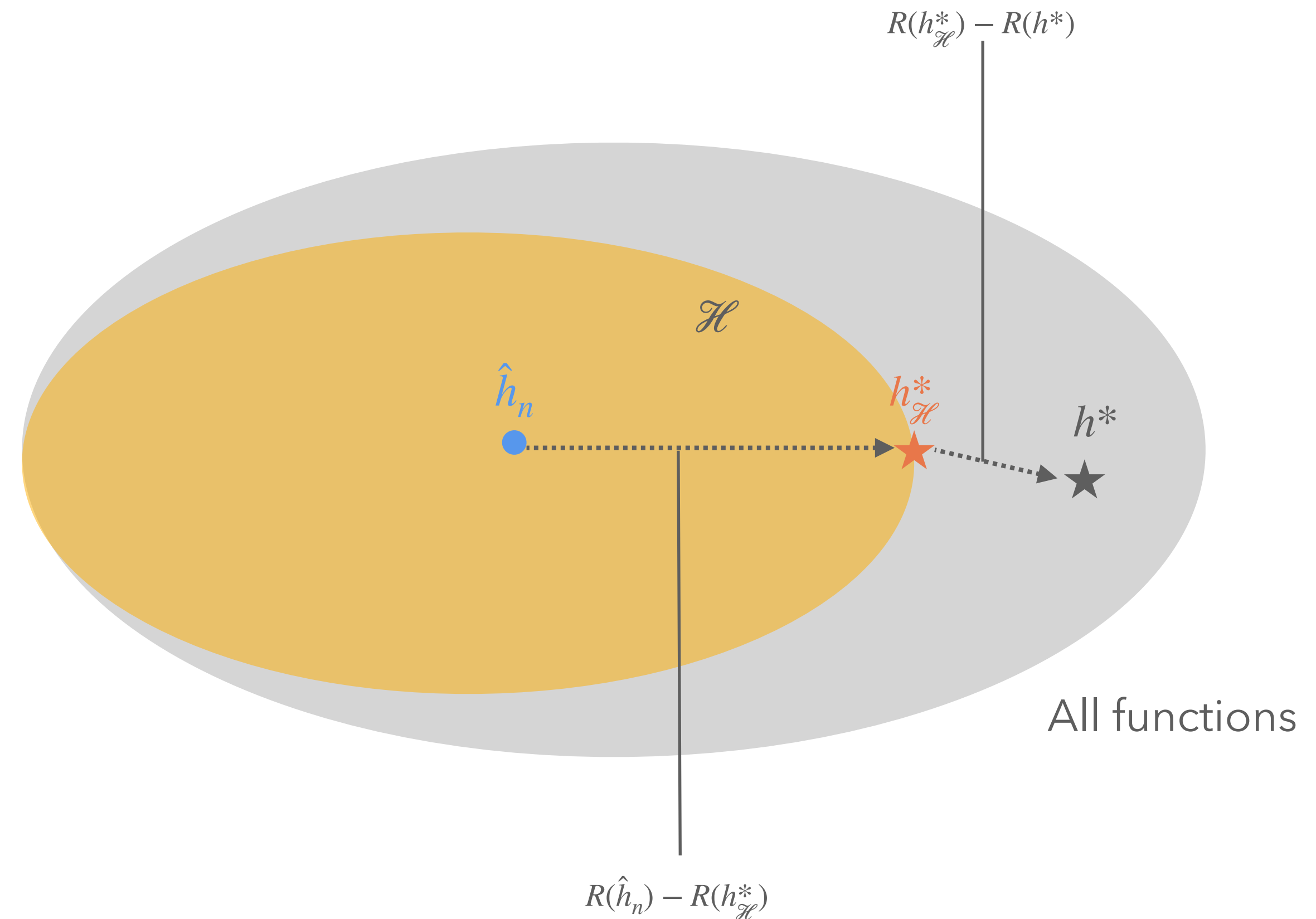
The excess risk of  $h$  is how far  $h$  is from  $h^*$ :  
 $R(h) - R(h^*)$ .

Excess risk of ERM  $\hat{h}_n$ :

$$R(\hat{h}_n) - R(h^*) = \underbrace{R(\hat{h}_n) - R(h_{\mathcal{H}}^*)}_{\text{est. error}} + \underbrace{R(h_{\mathcal{H}}^*) - R(h^*)}_{\text{approx. error}}$$

Estimation error is from using finite training as a proxy for risk (a generalization issue).

Approximation error is from our choice of class  $\mathcal{H}$  (a representation issue).



# Estimation Error

## Details

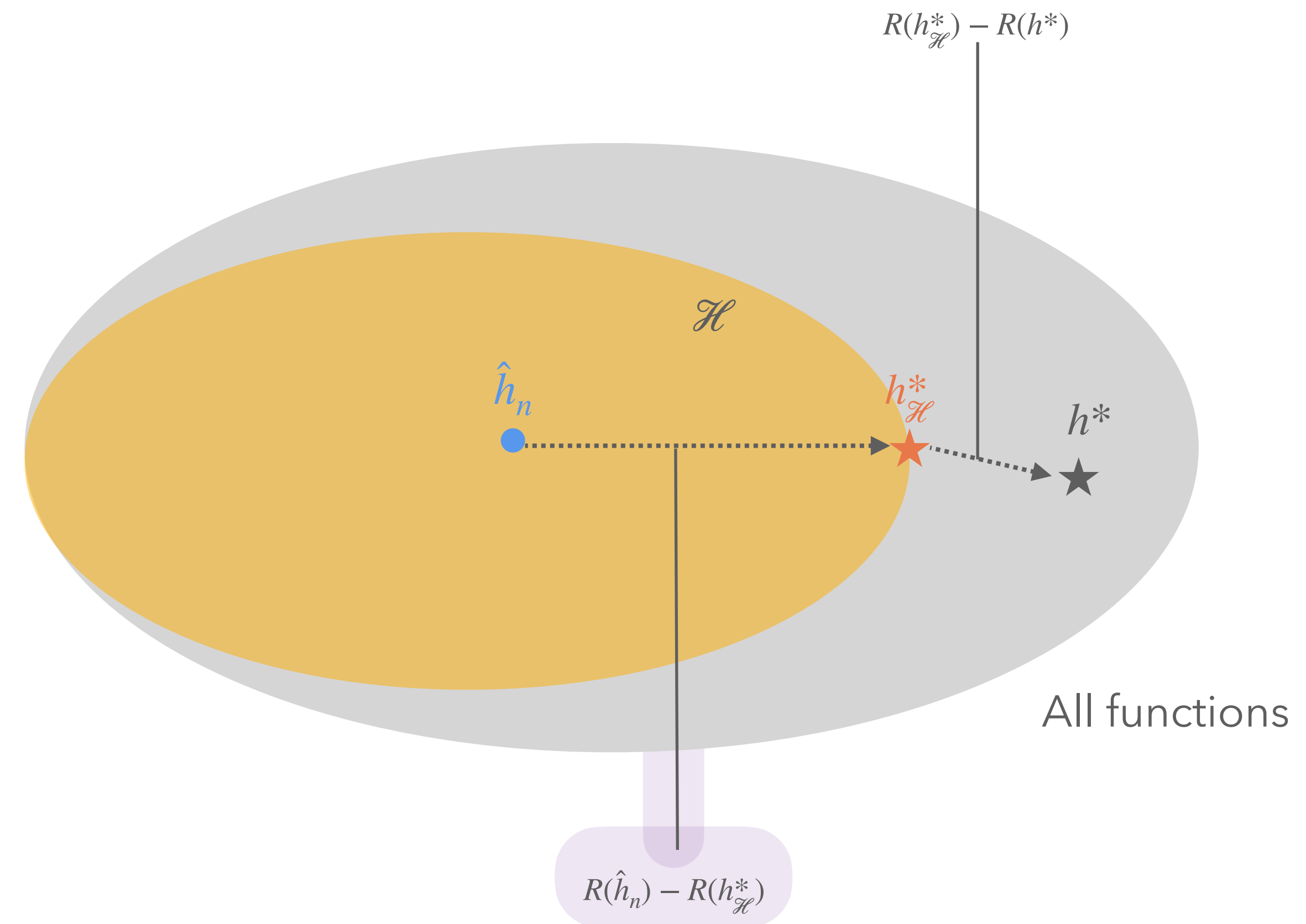
The estimation error  $R(\hat{h}_n) - R(h_{\mathcal{H}}^*)$  is the error incurred by using a finite sample  $D_n$  to obtain  $\hat{h}_n$ .

This is a random variable (why)?

Typically, when  $n \rightarrow \infty$  (infinite training data), the estimation error goes to zero.

We expect that estimation error increases with larger  $\mathcal{H}$ .

*Very rough intuition: a "variance" term.*



We will come back to the tension this has with modern machine learning practice!

# Approximation Error

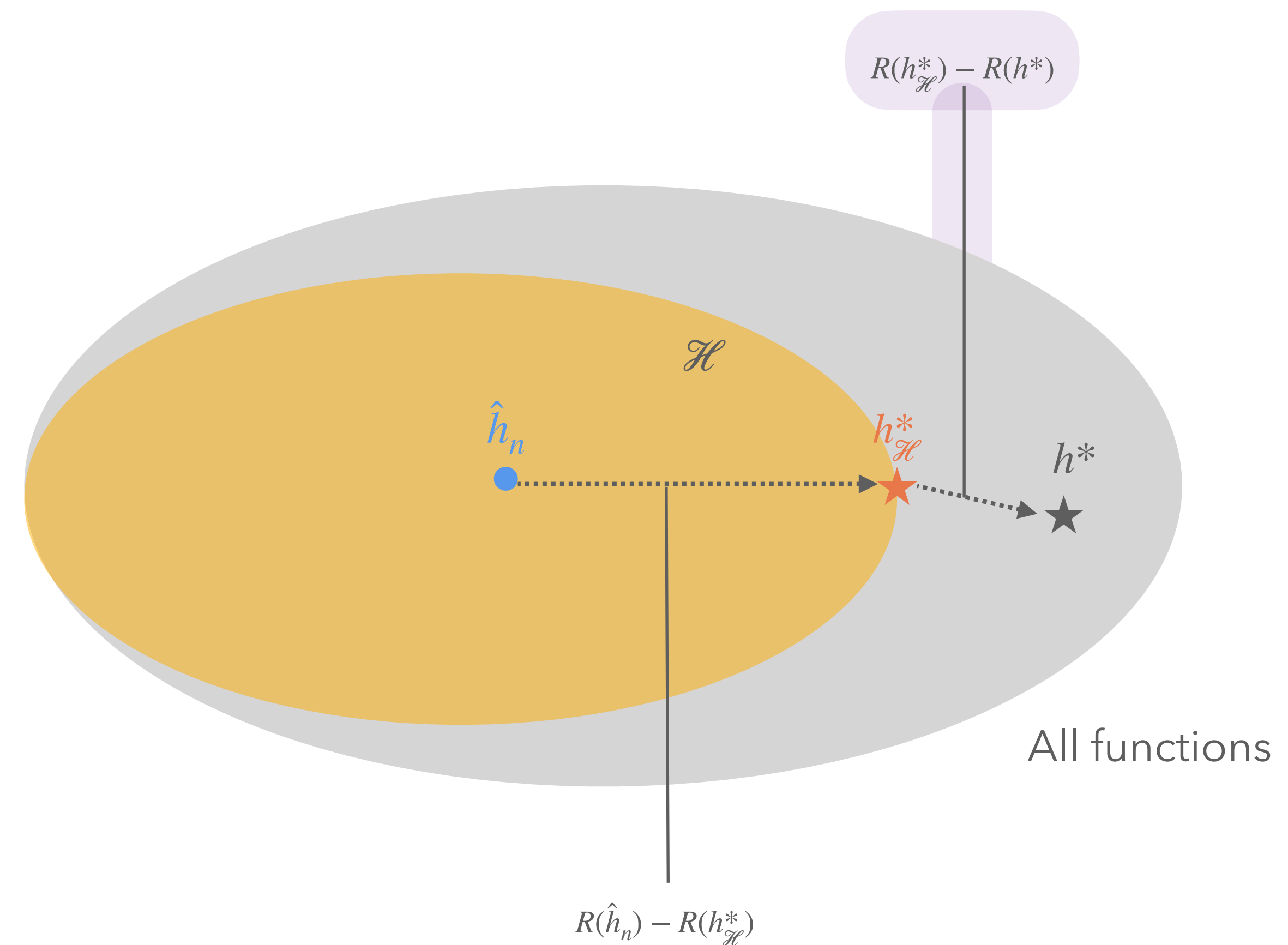
## Details

The approximation error  $R(h_{\mathcal{H}}^*) - R(h^*)$  is the error incurred by restricting to  $\mathcal{H}$ .

This is not a random variable (why)?

Typically, approximation error decreases with larger  $\mathcal{H}$ .

*Very rough intuition: a "bias" term.*

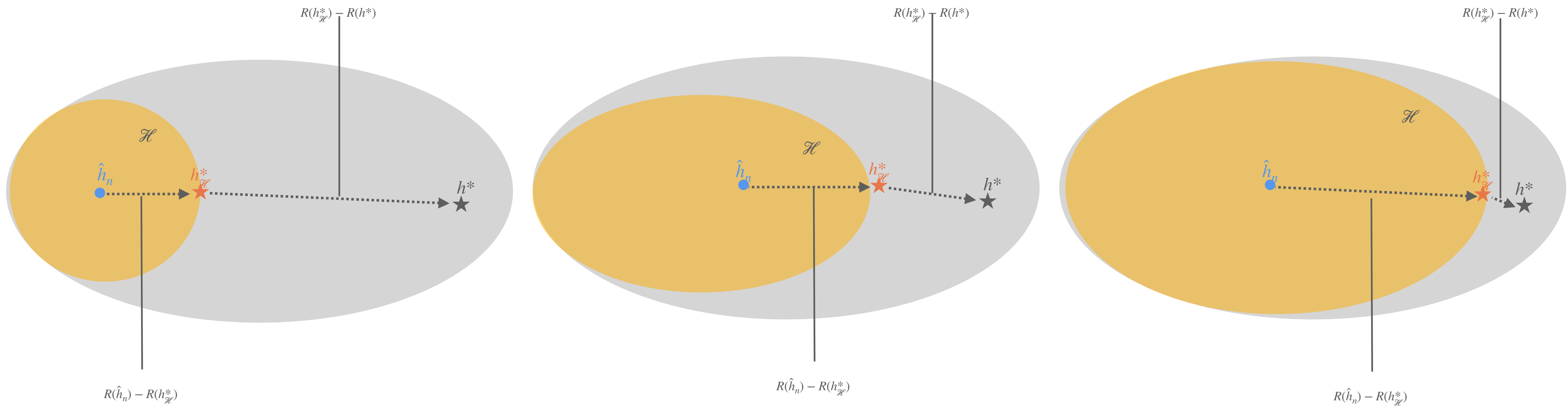




# Excess Risk

Intuition: Size of  $\mathcal{H}$

$$R(\hat{h}_n) - R(h^*) = \underbrace{R(\hat{h}_n) - R(h_{\mathcal{H}}^*)}_{\text{est. error}} + \underbrace{R(h_{\mathcal{H}}^*) - R(h^*)}_{\text{approx. error}}$$



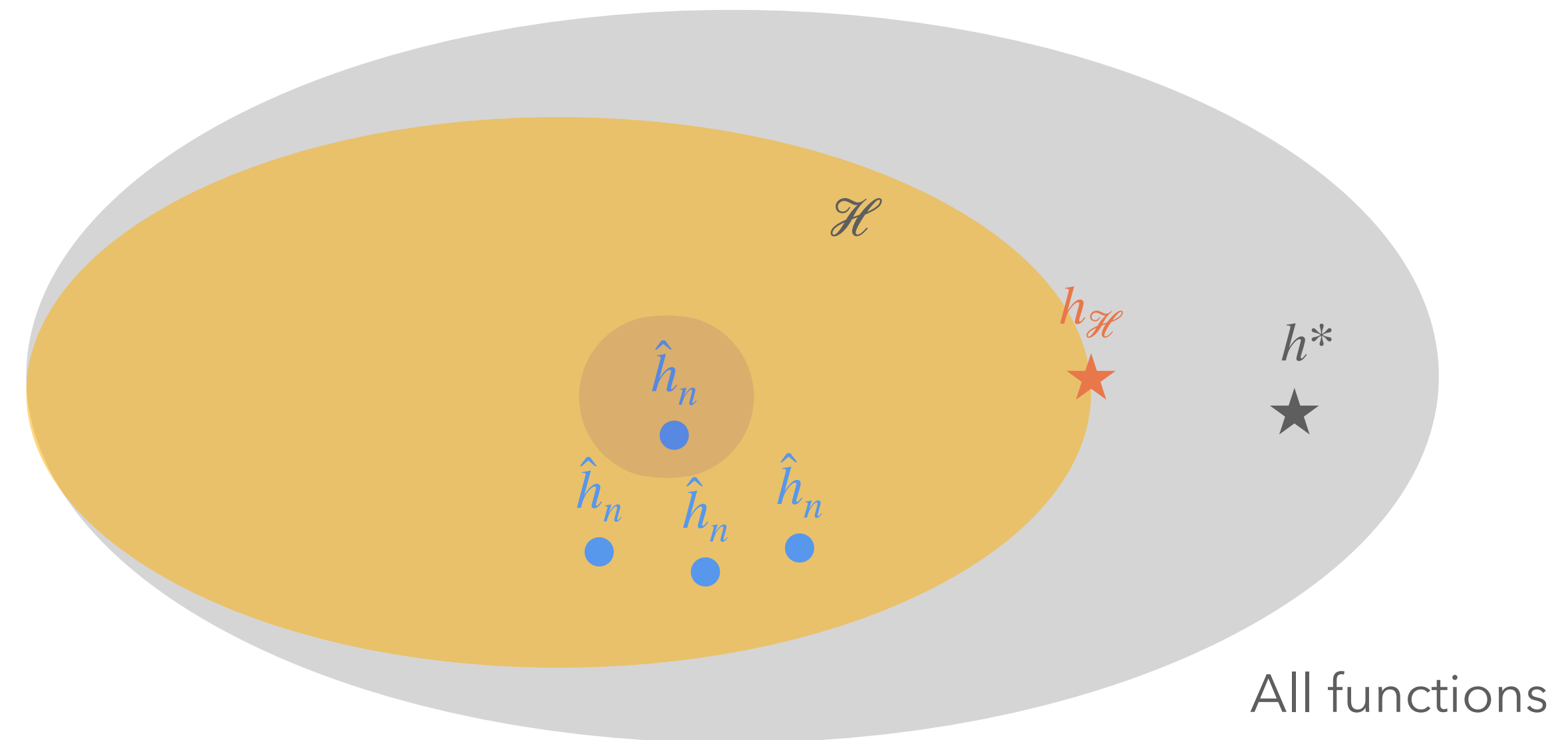
# Optimization Error

## Details

But how do we search for a hypothesis that minimizes empirical risk?

$$\hat{h}_n \in \underset{h \in \mathcal{H}}{\operatorname{argmin}} \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(h(x^{(i)}), y^{(i)})}_{\hat{R}_n(h)}$$

To search for one of them, we run a [learning algorithm](#) which typically uses a well-defined optimization procedure.



# Optimization Error

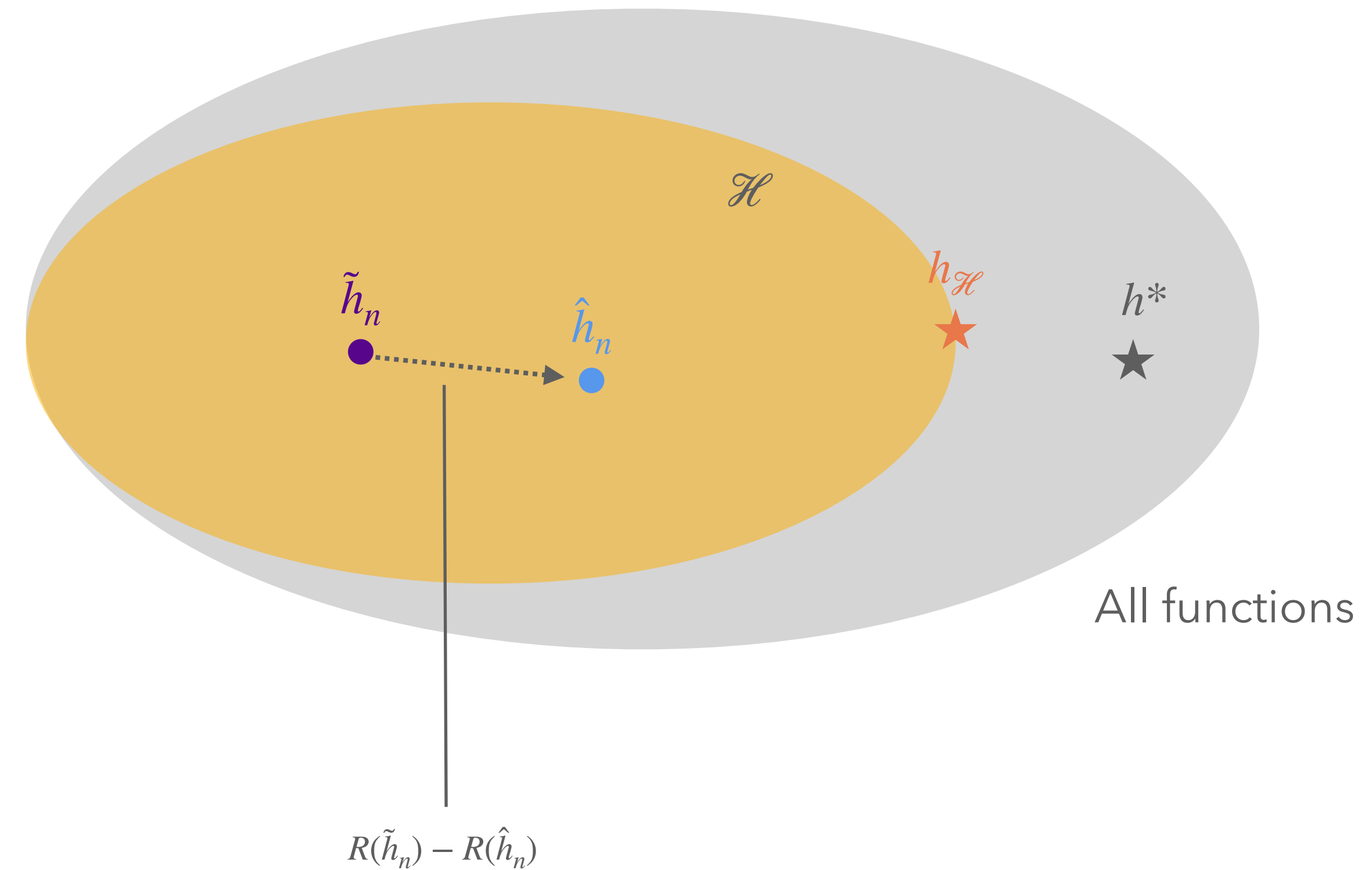
## Details

We might not find the ERM  $\hat{h}_n \in \mathcal{H}$ .

We instead find  $\tilde{h}_n \in \mathcal{H}$  via an algorithm, typically through optimization.

The optimization error is the gap between  $\tilde{h}_n$  (which our algorithm returns) and  $\hat{h}_n$  (the ERM):

$$R(\tilde{h}_n) - R(\hat{h}_n).$$



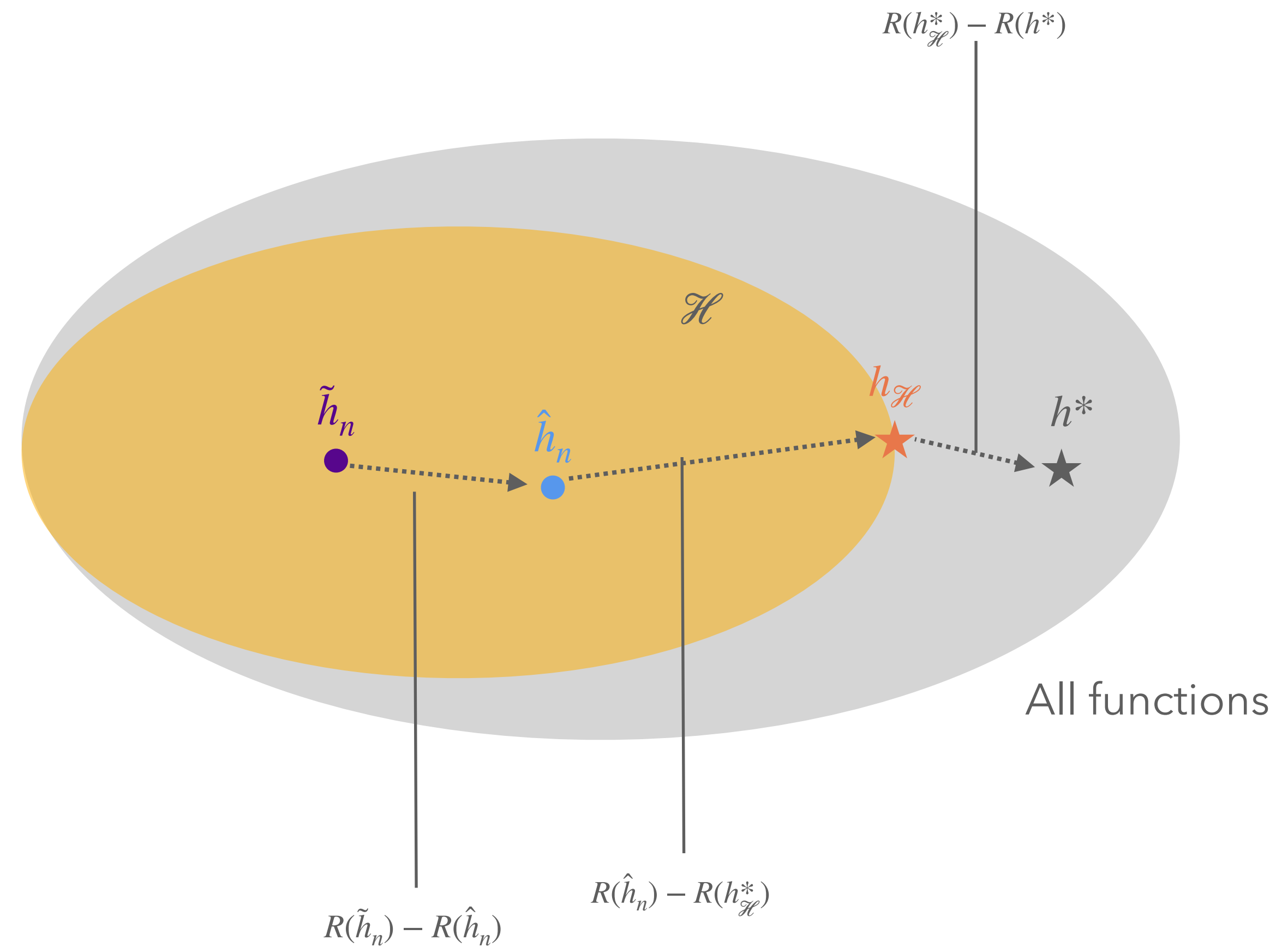
# Excess Risk

## Full Decomposition

We receive  $\tilde{h}_n$  from an algorithm.

Excess risk of  $\tilde{h}_n$ :

$$R(\tilde{h}_n) - R(h^*) = \underbrace{R(\tilde{h}_n) - R(\hat{h}_n)}_{\text{opt. error}} + \underbrace{R(\hat{h}_n) - R(h_{\mathcal{H}}^*)}_{\text{est. error}} + \underbrace{R(h_{\mathcal{H}}^*) - R(h^*)}_{\text{approx. error}}$$



# Supervised Learning

## Basic Pipeline

1. Collect training dataset, a collection of labeled input-output pairs.

2. Decide on the template of the hypothesis mapping that will map inputs to actions.

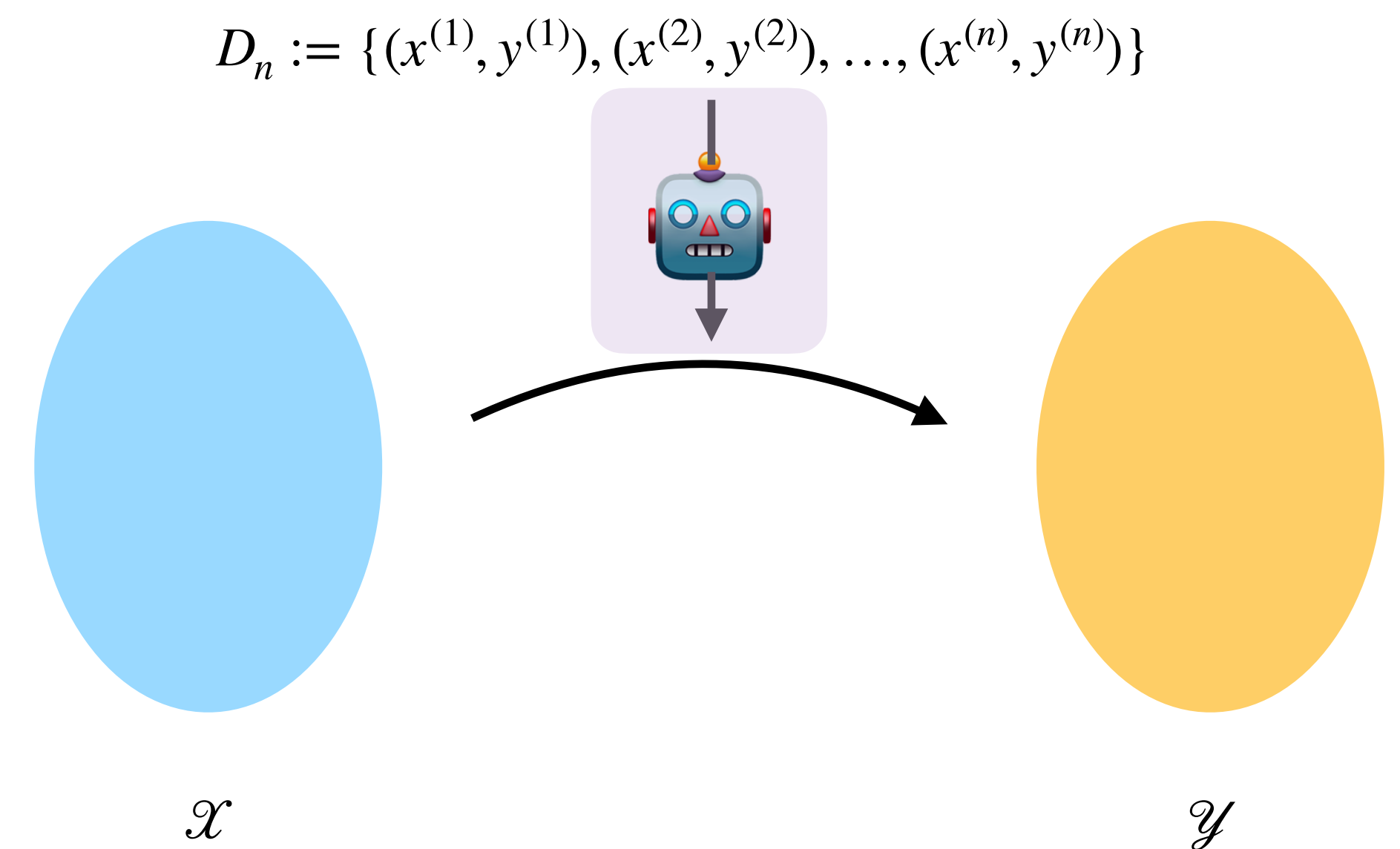
3. A learning algorithm takes the labeled training data as input and outputs a hypothesis.

4. The hypothesis predicts on new, unseen data which we hope it does well on, under a notion of loss.

Representation

Optimization

Generalization



# Supervised Learning

## Excess Risk Formalization

1. Collect training dataset, a collection of labeled input-output pairs.

2. Decide on the template of the hypothesis mapping that will map inputs to actions.

3. A learning algorithm takes the labeled training data as input and outputs a hypothesis.

4. The hypothesis predicts on new, unseen data which we hope it does well on, under a notion of loss.

Representation

Optimization

Generalization

We receive  $\tilde{h}_n$  from an algorithm.

Excess risk of  $\tilde{h}_n$ :

$$R(\tilde{h}_n) - R(h^*) =$$

$$\underbrace{R(\tilde{h}_n) - R(\hat{h}_n)}_{\text{opt. error}} + \underbrace{R(\hat{h}_n) - R(h_{\mathcal{H}}^*)}_{\text{est. error}} + \underbrace{R(h_{\mathcal{H}}^*) - R(h^*)}_{\text{approx. error}}$$

Optimization      Generalization      Representation

# Three Main Questions

Representation, Optimization, and Generalization

$$R(\tilde{h}_n) - R(h^*) =$$
$$\underbrace{R(\tilde{h}_n) - R(\hat{h}_n)}_{\text{opt. error}} + \underbrace{R(\hat{h}_n) - R(h_{\mathcal{H}}^*)}_{\text{est. error}} + \underbrace{R(h_{\mathcal{H}}^*) - R(h^*)}_{\text{approx. error}}$$

Optimization      Generalization      Representation

**Representation:** Which hypothesis class  $\mathcal{H}$  best models the relationship of  $\mathcal{X}$  to  $\mathcal{A}$ ?

**Generalization:** How well can we extrapolate from training data to new, unseen data?

**Optimization:** How can we efficiently and accurately solve the ERM optimization problem?

# The Main Cast

## Summary of the Problem

Examples from input space  $\mathcal{X}$  and output space  $\mathcal{Y}$ ; unknown distribution  $P_{\mathcal{X} \times \mathcal{Y}}$  over  $\mathcal{X} \times \mathcal{Y}$ .

Action space  $\mathcal{A}$  as the output (often, a *prediction*) of learned hypothesis/predictor.

We evaluate actions with a loss function  $\ell : \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$ .

Goal: Find a hypothesis  $h : \mathcal{X} \rightarrow \mathcal{A}$  to minimize the risk  $R(h) := \mathbb{E}[\ell(h(x), y)]$ .

We can approximate risk with the empirical risk over sample  $D_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$ :

$$\hat{R}_n(h) := \frac{1}{n} \sum_{i=1}^n \ell(h(x^{(i)}), y^{(i)}).$$



# The Main Cast

## Summary of the Problem

Goal: Find a hypothesis  $h : \mathcal{X} \rightarrow \mathcal{A}$  to minimize the risk  $R(h) := \mathbb{E}[\ell(h(x), y)]$ .

We can approximate risk with the empirical risk over sample  $D_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$ .

Choose a hypothesis class  $\mathcal{H}$  and find the empirical risk minimizer  $\hat{h}_n \in \mathcal{H}$ :

$$\hat{h}_n \in \operatorname{argmin}_{h \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(h(x^{(i)}), y^{(i)})}_{\hat{R}_n(h)}$$

Or find  $\tilde{h}_n$  that approximates  $\hat{h}_n$  well.

# The Main Cast

## Summary of the Problem

Goal: Find a hypothesis  $h : \mathcal{X} \rightarrow \mathcal{A}$  to minimize the risk  $R(h) := \mathbb{E}[\ell(h(x), y)]$ .

Overall quality (excess risk) of our produced  $\tilde{h}_n$  :

$$R(\tilde{h}_n) - R(h^*) = \underbrace{R(\tilde{h}_n) - R(\hat{h}_n)}_{\text{opt. error}} + \underbrace{R(\hat{h}_n) - R(h_{\mathcal{H}}^*)}_{\text{est. error}} + \underbrace{R(h_{\mathcal{H}}^*) - R(h^*)}_{\text{approx. error}}$$

Choose  $\mathcal{H}$  that balances approximation error and estimation error.

With more data, estimation error typically decreases, can use bigger  $\mathcal{H}$ .

Produce  $\tilde{h}_n$  via an algorithm that (approximately and efficiently) minimizes empirical error.

# Outline

Course Overview and Logistics

Introduction to Machine Learning

Statistical Learning Setup

Statistical Learning: Bayes Risk

Statistical Learning: Empirical Risk and ERM

Statistical Learning: Hypothesis Class

Excess Risk Decomposition and Three Types of Error