Introduction to Machine Learning (CSCI-UA 473): Fall 2021

Lecture 2: Overview of Machine Learning

Sumit Chopra

Courant Institute of Mathematical Sciences
Department of Radiology - Grossman School of Medicine
NYU

Slides derived from materials from Benjamin Peherstorfer, Kyunghyun Cho, Andrew Gordon Wilson and a few other books.

Lecture Outline

Logistics of the Course

Overview of Machine Learning

Homework 0

Course Logistics

- Lectures
  - Days: Tuesdays and Thursdays
  - Times: 2:00 PM - 3:15 PM
  - Mix of theory and labs
  - Most Thursdays will be labs
  - Most theory lectures will be delivered through slides (available online). Some parts might on a white board.
- Graders and Assistants
  - Umang Sharma (Grader and Tutor)
- Office Hours
  - Sumit Chopra: Thursday from 4:00 PM - 5:00 PM
  - Umang Sharma: Fridays from 3:00 PM - 4:00 PM
- Grading
  - Homework assignments: 70% (7 homeworks over the course of semester carrying equal weight)
  - Final exam: 30%
  - Assignments should be written in Latex, submitted as pdfs. Hand written notes will not be accepted
  - Late homework assignments will not be accepted
  - You **must pass** the final exam in order to pass the course

Course Logistics

- Course is on Brightspace. All the material and other information will be posted there
- Books and Reading Materials
  - The Elements of Statistical Learning: Trevor Hastie, Robert Tobshirani, and Jerome Friedman
  - Learning From Data: Yaser S. Abu-Mostafa, Malik Magnon-Ismail, and Hsuan-Tien Lin
  - Pattern Recognition and Machine Learning: Christopher M Bishop
  - Lecture Notes by Kyunghyun Cho
- Prerequisite
  - Linear Algebra and Vector Calculus
  - Probability Theory
  - Proficiency in Python 3 and PyTorch
- Lab Sessions
  - Installation instruction for Jupyter
  - Installation instruction for Python packages
  - Another option is to use Anaconda to install necessary packages
  - You can also use a virtual environment
  - Lab materials will be distributed via Brightspace before the session
  - They will lead by the tutor

Expectations from Y'all

- **Attendance:** it's a fast paced course. Come to the lectures and participate in the discussion. Don't just merely show up!
- **Theory is important:** unlike many other subjects Machine Learning is a highly practical subject. But understanding the underlying theoretical concepts is extremely important if you want to be excellent at applying ML models in your jobs.
- **Get your hands dirty:** actively participate in the labs. Play around with the code. Try to break it and see what happens. That is how you'll really learn
- **Read:** ML is a vast and growing field. A single semester is not nearly sufficient to teach everything. Many topics will be left out. Even for the topics we will discuss many details will be omitted. Strongly encourage you to read the relevant materials I will point in the class.
- **Homeworks:** strongly encourage you to discuss/brainstorm with peers. However when it comes to doing, **please do it on your own. Mention the name(s) of peers you discussed the homework with in your submission.**
- **Cheating:** zero tolerance policy on cheating. The goal of this class is for you all to learn about machine learning. Getting good grades is **not** the end goal. Be honest to yourself. If you are stuck on a problem, think harder —> research —> ask
- **Plagiarism:** zero tolerance policy on plagiarism. When you borrow material from an external source, always make sure to cite the source (text/images/datasets/anything!)

Overview of Machine Learning

What is Machine Learning

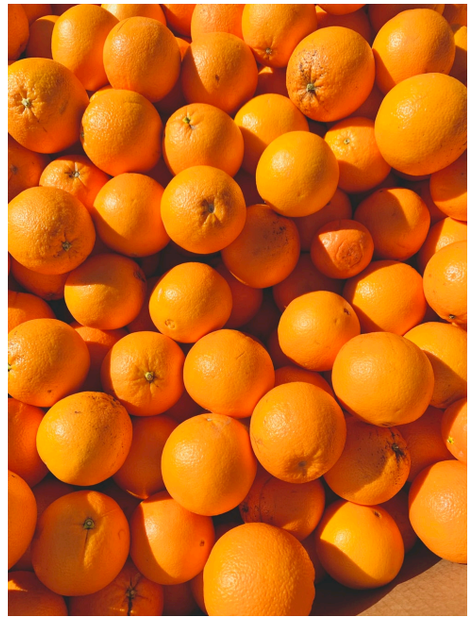
Pertains to teaching computers to **infer patterns** from **the data** without the need to **design** an **explicit program**

What is Machine Learning

Pertains to teaching computers to **infer patterns** from **the data** without the need to **design** an **explicit program**

Example Problem

Suppose you want to build a conveyor belt with a robot arm and a camera, that can automatically sort apples from oranges and put them in different piles

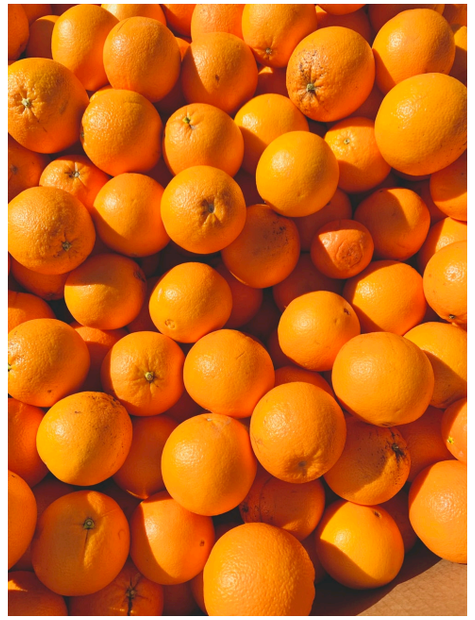


What is Machine Learning

Pertains to teaching computers to **infer patterns** from **the data** without the need to **design** an **explicit program**

Example Problem

Suppose you want to build a conveyor belt with a robot arm and a camera, that can automatically sort apples from oranges and put them in different piles



Design a Solution

Write a program to explicitly describe an apple (red/green color with streaks — might have a small stem — circular with flat top and bottom) and an orange (orange color — circular — small bumps on surface)

Run the images from camera through the program to infer the type of fruit

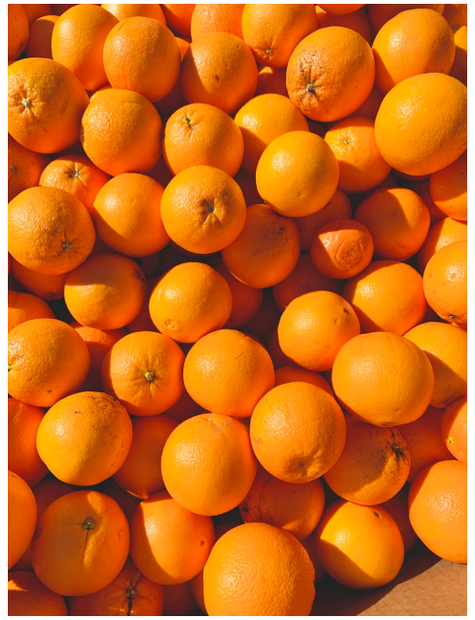
Guide the robot arm to pick up the fruit and put it in the correct basket

What is Machine Learning

Pertains to teaching computers to **infer patterns** from **the data** without the need to **design** an **explicit program**

Example Problem

Suppose you want to build a conveyor belt with a robot arm and a camera, that can automatically sort apples from oranges and put them in different piles



Machine Learning (Learning from Data)

Collect 100,000 images of apples and 100,000 images of oranges

Use these images to **train a parametric function** which takes as input an image and infers whether the image is of an apple or an orange

Use this trained function on the images captured by the camera and guide the robot arm

Example Problems: Spam Filtering

data

prediction

Osman Khan to Carlos — show details Jan 7 (6 days ago) — Reply

sounds good
+ok

Carlos Guestrin wrote:
> Let's try to chat on Friday a little to coordinate and more on Sunday in person?
>
> Carlos

Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Miche — show details 3:15 PM (8 hours ago) — Reply

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only $5.95 for shipping mfw rlk Spam | X

Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: anç — show details 9:52 PM (1 hour ago) — Reply

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

* Rapid WeightL0SS
* Increased metabolism - BurnFat & calories easily!
* Better Mood and Attitude
* More Self Confidence
* Cleanse and Detoxify Your Body
* Much More Energy
* BetterSexLife
* A Natural Colon Cleanse

→

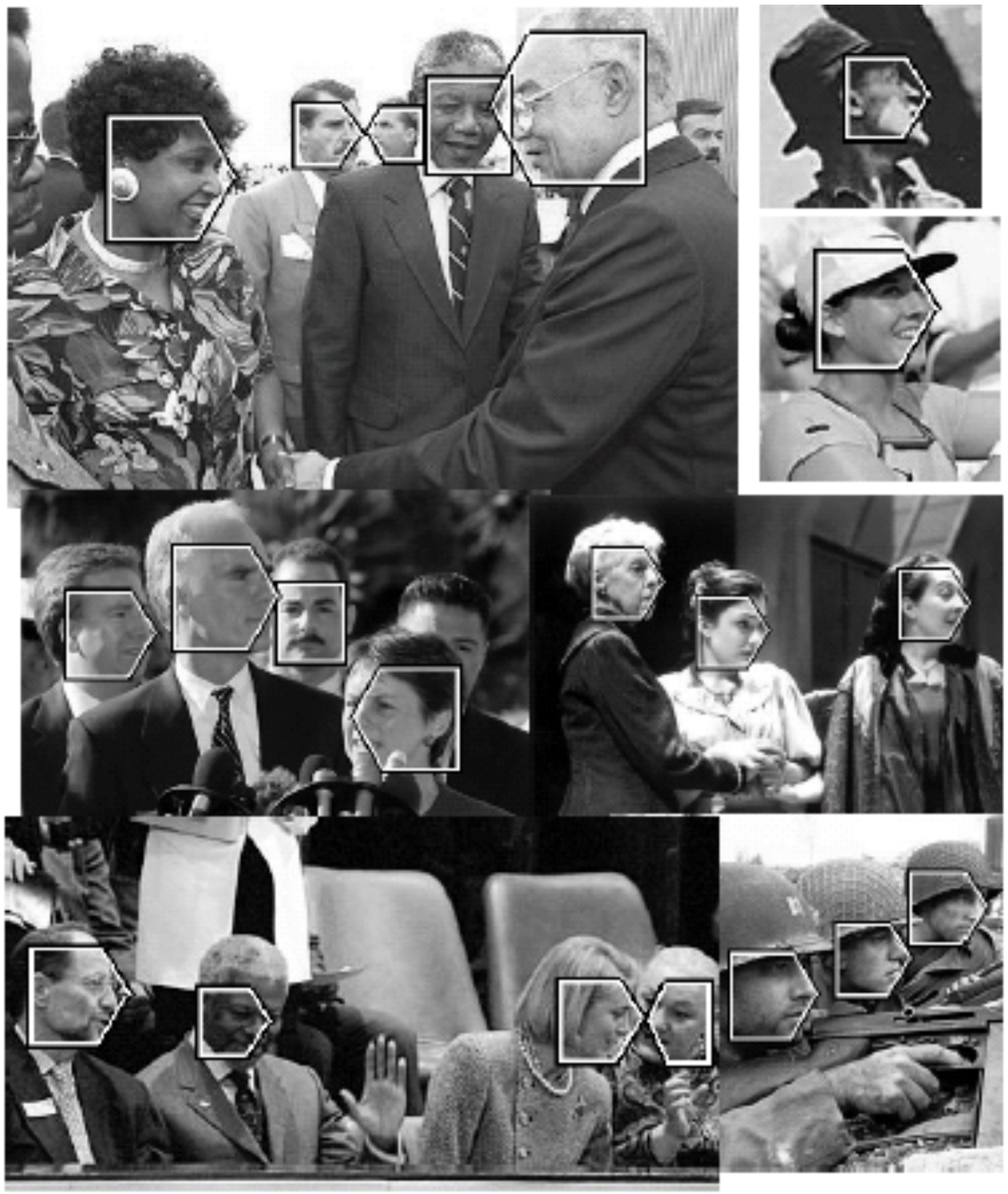
Spam

vs.

Not Spam

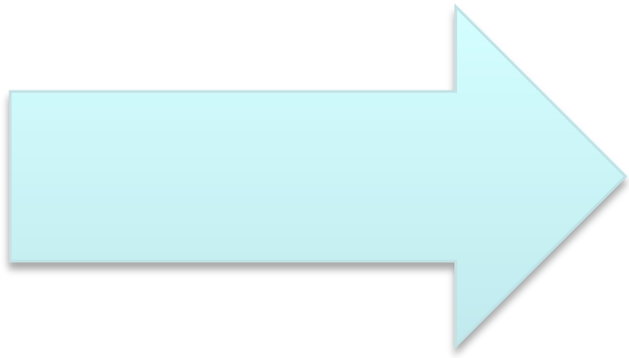
Training examples will consist of a collection of spam and non-spam emails

Example Problems: Face Recognition



Training examples will consist of a collection of facial images of people you want to recognize

Example Problems: Weather Prediction



Training examples will
consist of a collection of satellite images
labeled with the actual weather pattern observe

Example Problems: Stock Price Prediction

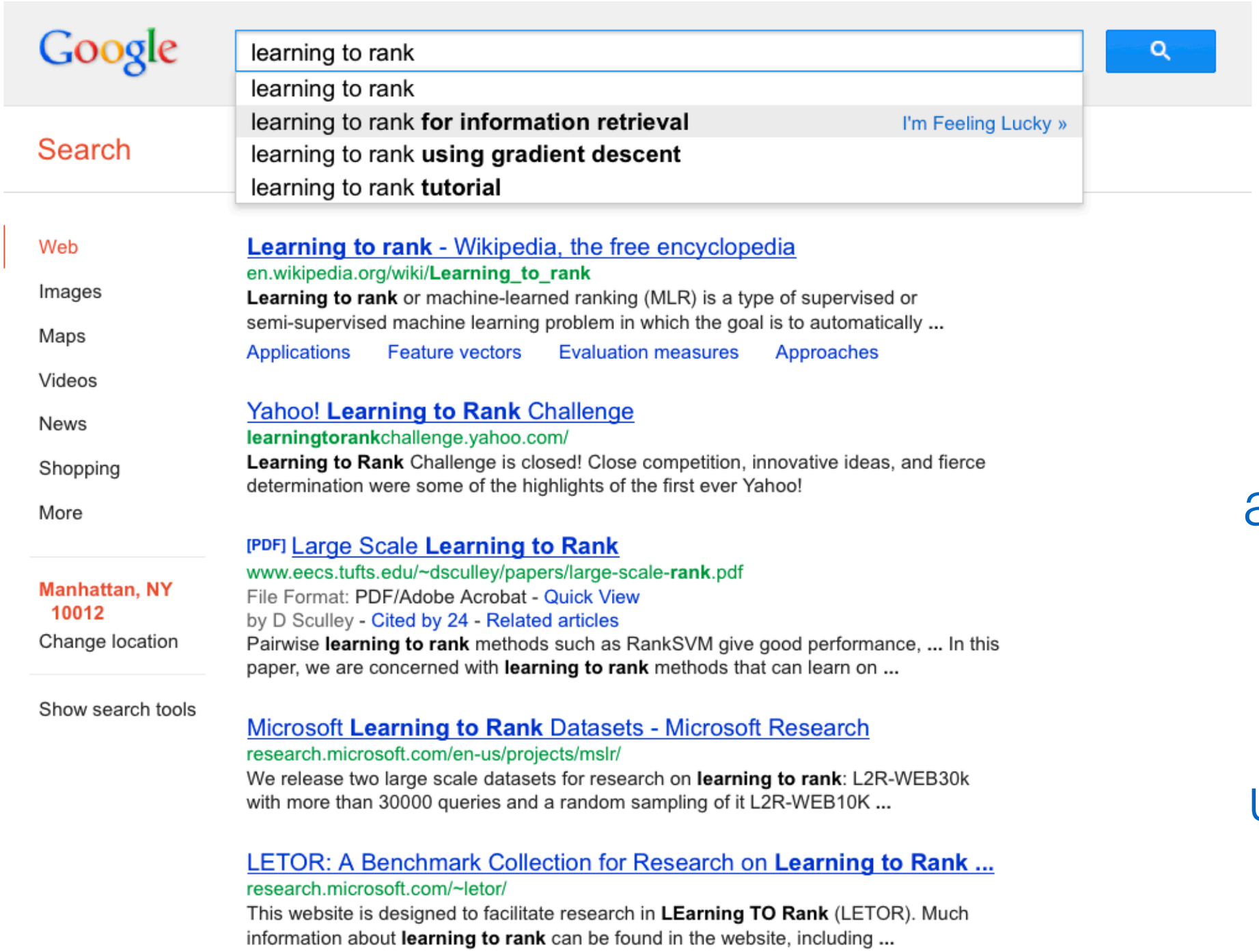
Training examples will consist of the historic price of the stock under consideration. It could also consist of the prices of other stocks which could influence the price of the stock under consideration. In short training set could be complex.

Example Problems: Stock Price Prediction

What type of dataset to collect is a part of ML designer's responsibility.

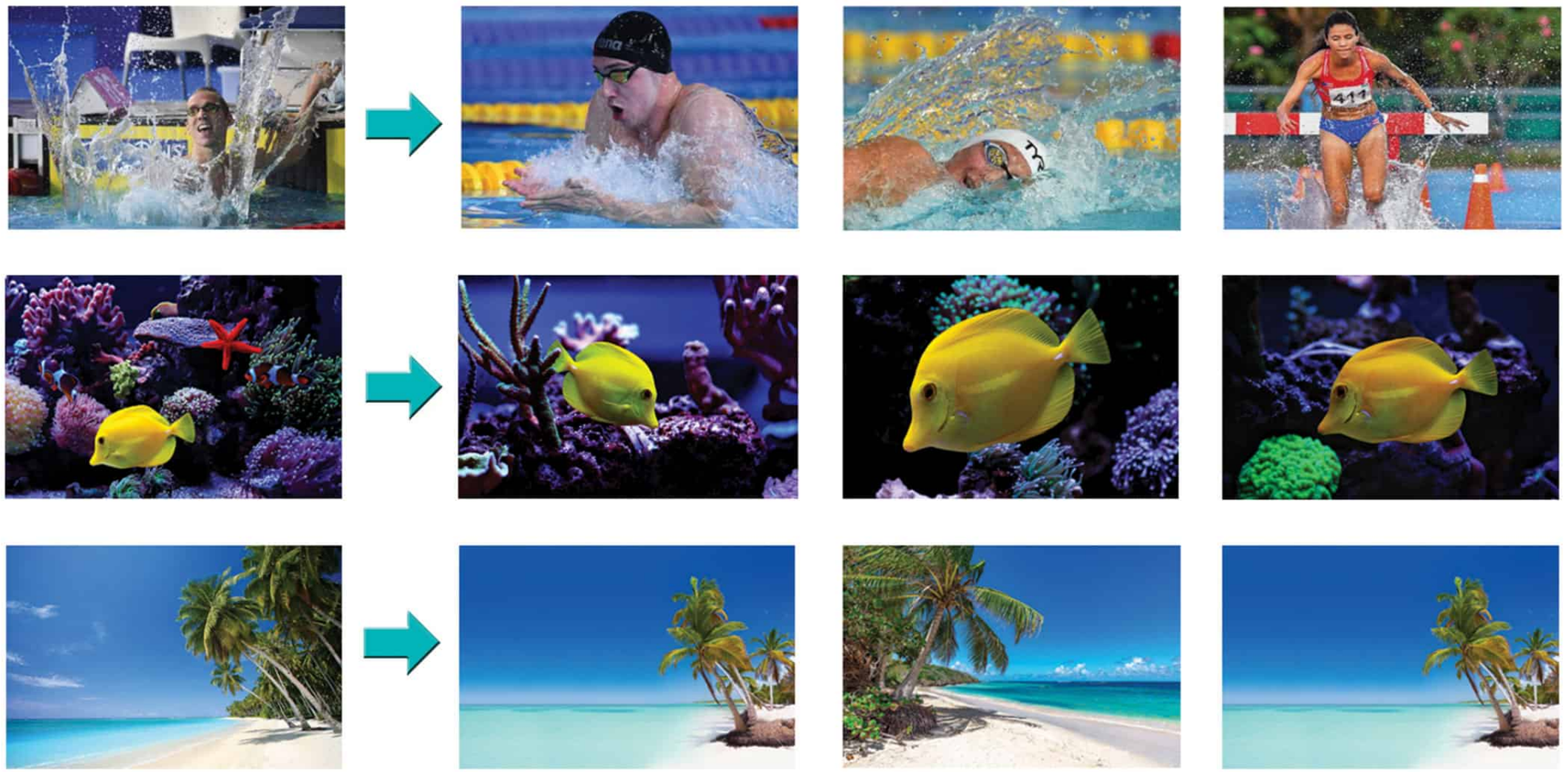
Training examples will consist of the historic price of the stock under consideration. It could also consist of the prices of other stocks which could influence the price of the stock under consideration. In short training set could be complex.

Example Problems: Ranking



Training data consists of the search query accompanied by the search results which were clicked by users in the past

Example Problems: Ranking



Training data consists of the query images accompanied by the images which users think are visually similar to the query image

Example Problems: Recommender Systems

Training examples will consist of a collection of users and the items they have bought/liked in the past.

Components of a Learning System

Example Problem Setting

You are a bank and you receive thousands of credit card applications everyday. For every application you want to answer two questions:

1. Whether to extend a credit card to the applicant?
2. If the answer is “yes” then what should be the credit limit?

Components of a Learning System

Example Problem Setting

You are a bank and you receive thousands of credit card applications everyday. For every application you want to answer two questions:

1. Whether to extend a credit card to the applicant?
2. If the answer is “yes” then what should be the credit limit?

You have lots of historical data: your current customers, their detailed information, whether or not you’ve made money from them after offering the credit card, and how much money you’ve made.

Components of a Learning System

Customer information (age, gender, income etc): x (the Input)

Binary credit decision (you made money or not): y (the output)

Unknown target function: $f : \mathcal{X} \rightarrow \mathcal{Y}$

$\mathcal{X}$: the entire space of inputs (potentially infinite)

$\mathcal{Y}$: the entire space of outputs (binary "yes/no" or [+1, -1] in this case)

Customer data (input-output pairs): $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ (**Training data**)

The data is such that: $y_i = f(x_i) : \forall i$

The input-output pairs are often called "data points" or "examples"

The goal of learning is to use the dataset D to find a function $g : \mathcal{X} \rightarrow \mathcal{Y}$ such that:

g approximates the function f

Learning algorithm chooses g from a set of candidate functions called Hypothesis Set $\mathcal{H}$

For a new customer the learning algorithm will base its decision on the output of g

Decision will make sense so long as the learnt g is faithful to f on the training data

Components of a Learning System

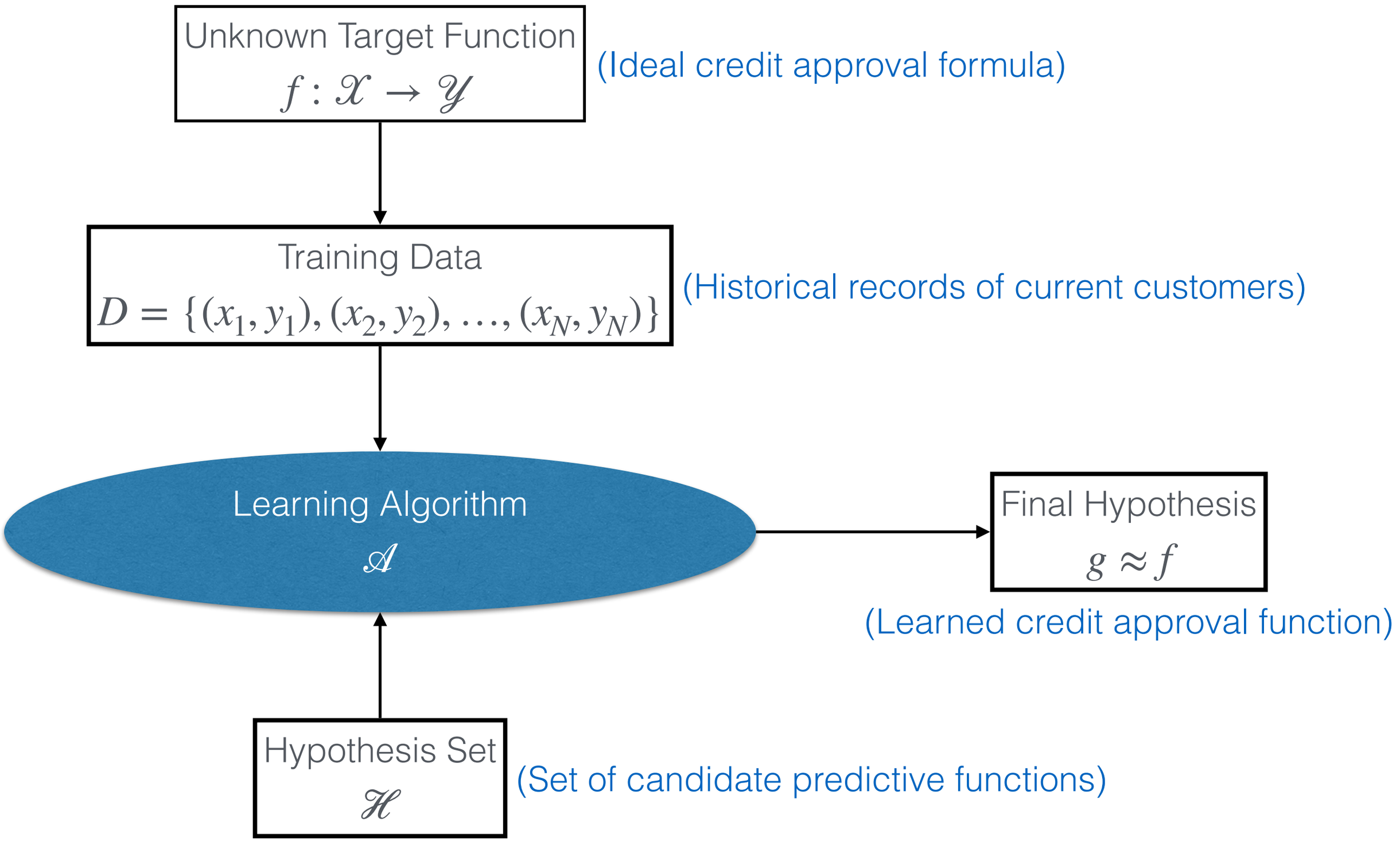


Figure adapted from the book Learning from Data: A Short Course

Components of a Learning System

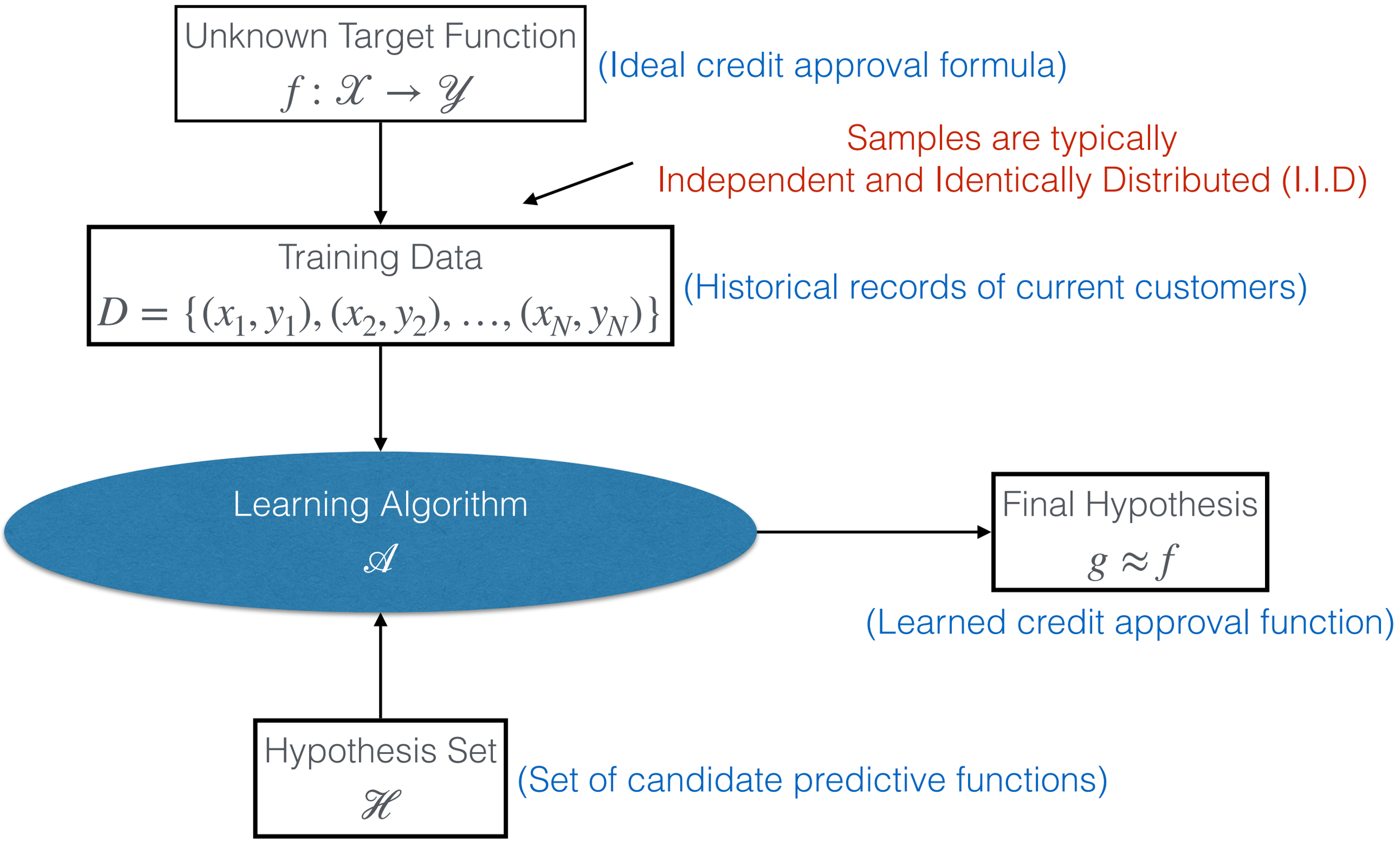


Figure adapted from the book Learning from Data: A Short Course

A Simple Learning Model

Let $\mathcal{X} = \mathfrak{R}^d$ be the input space, where $\mathfrak{R}^d$ is the d dimension Euclidean space

Different coordinates of the input vector $x = [x_1, x_2, \ldots, x_d] \in \mathfrak{R}^d$ correspond to different attributes of the user (income, age, debt etc)

Let the output space $\mathcal{Y} = \{+1, -1\}$ corresponding to a “yes”/“no” decision

Let us choose the hypothesis space $\mathcal{H}$ to be the set of all linear functions. That is:

$$h(x) = x \,.\, w = \sum_{i=1}^{d} x_i \,.\, w_i,$$

$w = [w_1, w_2, \ldots, w_d] \in \mathfrak{R}^d$ are the parameters of the hypothesis $h(\,.\,)$

Thus any hypothesis $h \in \mathcal{H}$ gives different weights to different input attributes.

Weighted attributes are then combined to give a “score”.

The decision algorithm can be written as:

If $\sum_{i=1}^{d} x_i \,.\, w_i > threshold$ then approve the credit application

If $\sum_{i=1}^{d} x_i \,.\, w_i < threshold$ then reject the credit application

A Simple Learning Model

$$h(x) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + b\right)$$

The bias b determines the *threshold* since according to the equation on previous slide, credit is approved if

$$\sum_{i=1}^{d} w_i x_i > -b$$

and $sign(s) = +1$ if $s > 0$; $sign(s) = -1$ if $s < 0$

The Decision Algorithm

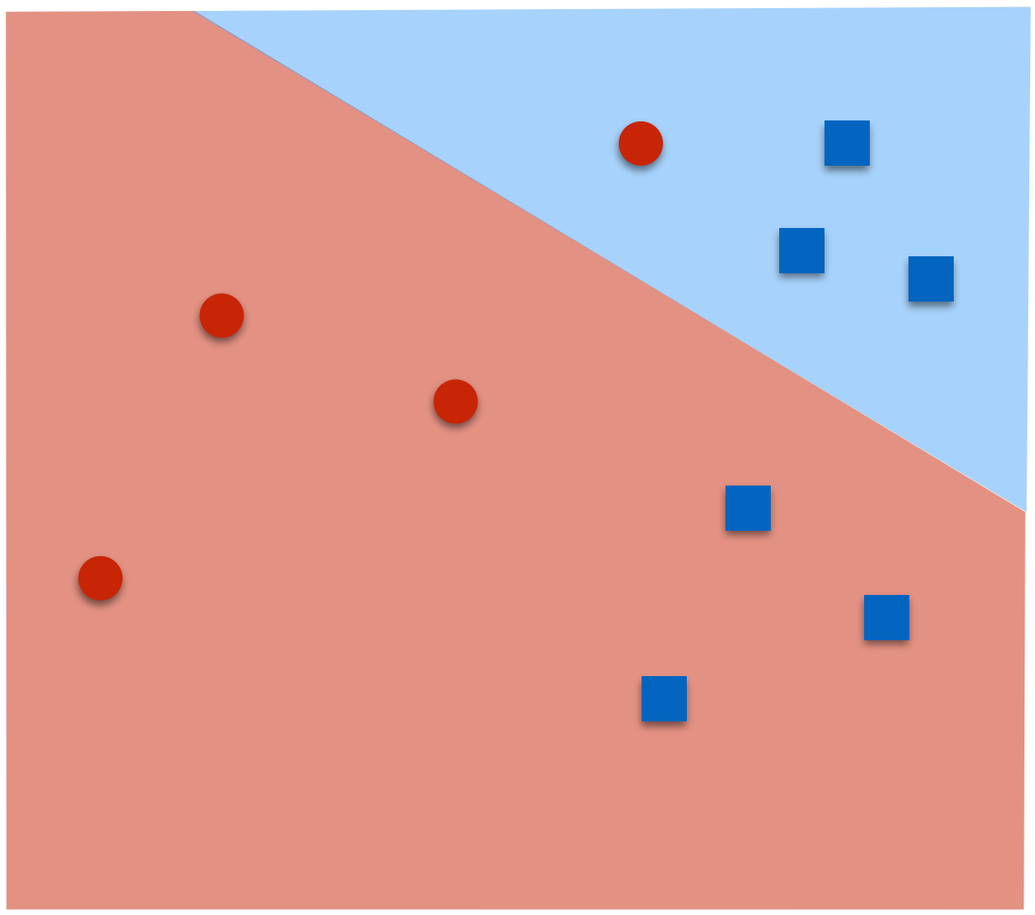
$h(x) = +1$ ==> approve credit

$h(x) = -1$ ==> reject credit

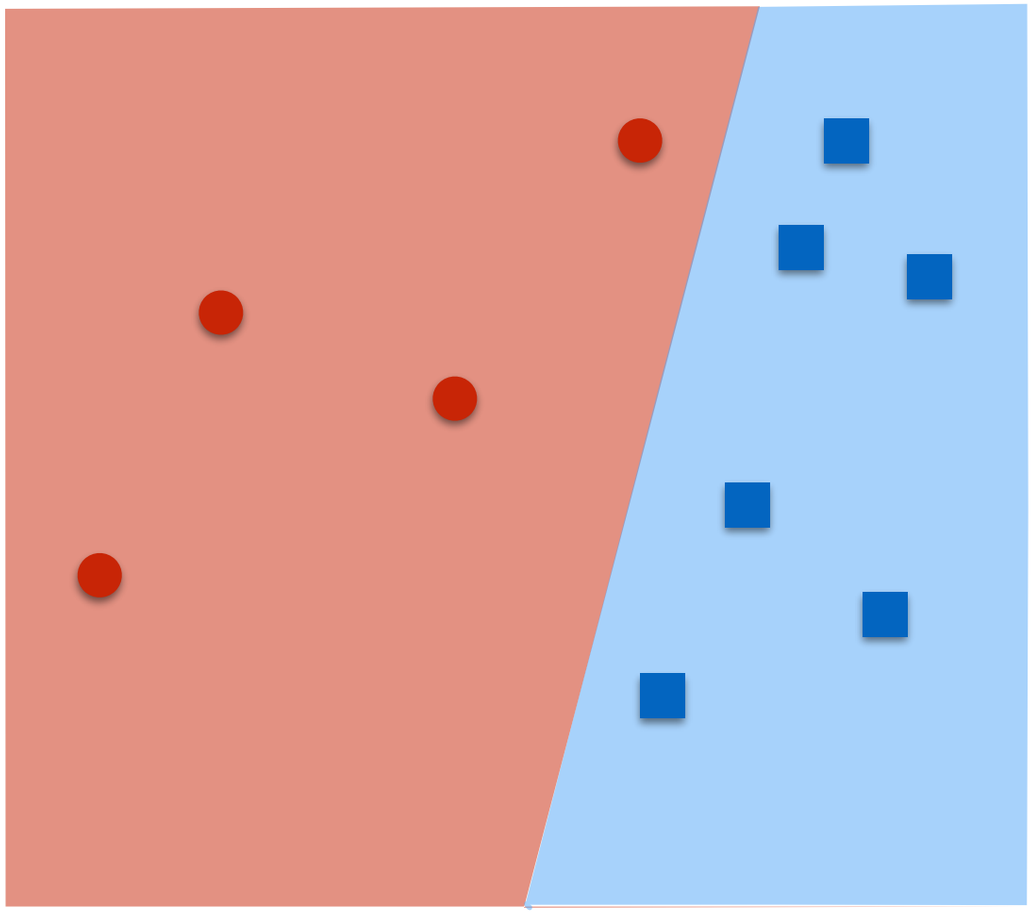
This simple model is called the ***Perceptron***

A Simple Learning Model

$$h(x) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + b\right)$$



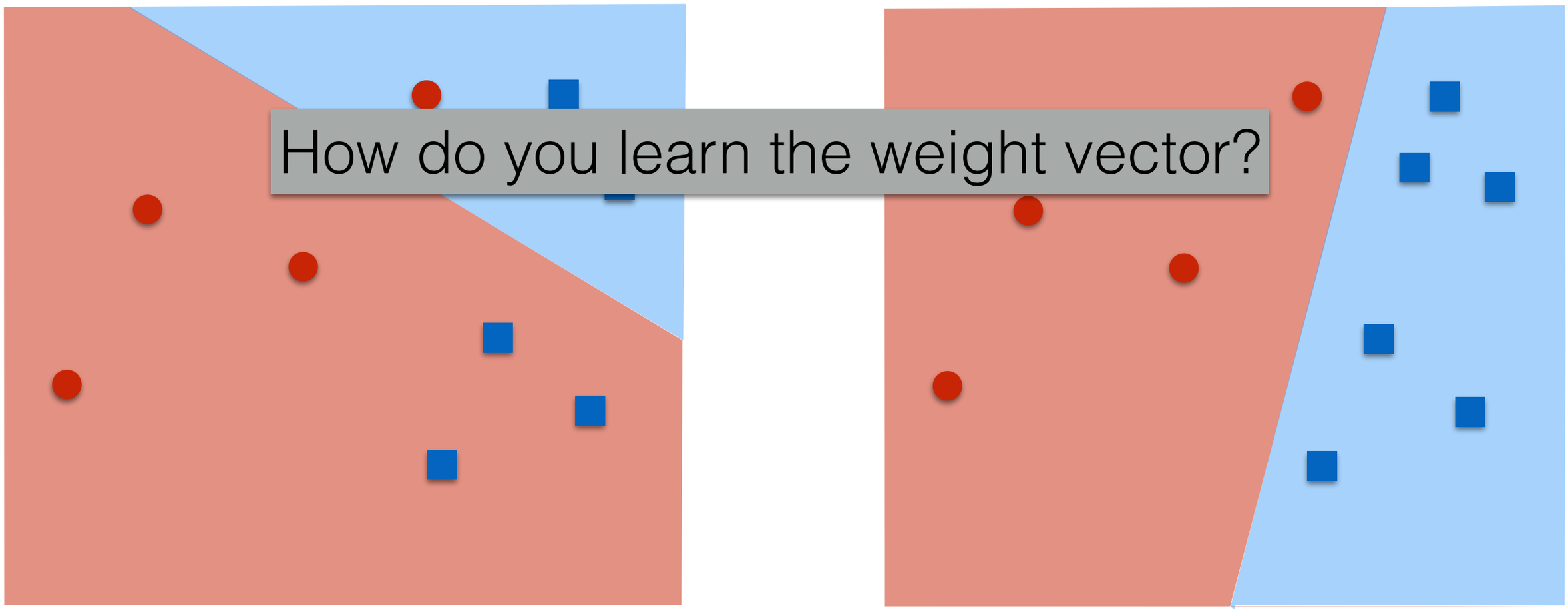
Random Weight Vector



Learned Weight Vector

A Simple Learning Model

$$h(x) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + b\right)$$



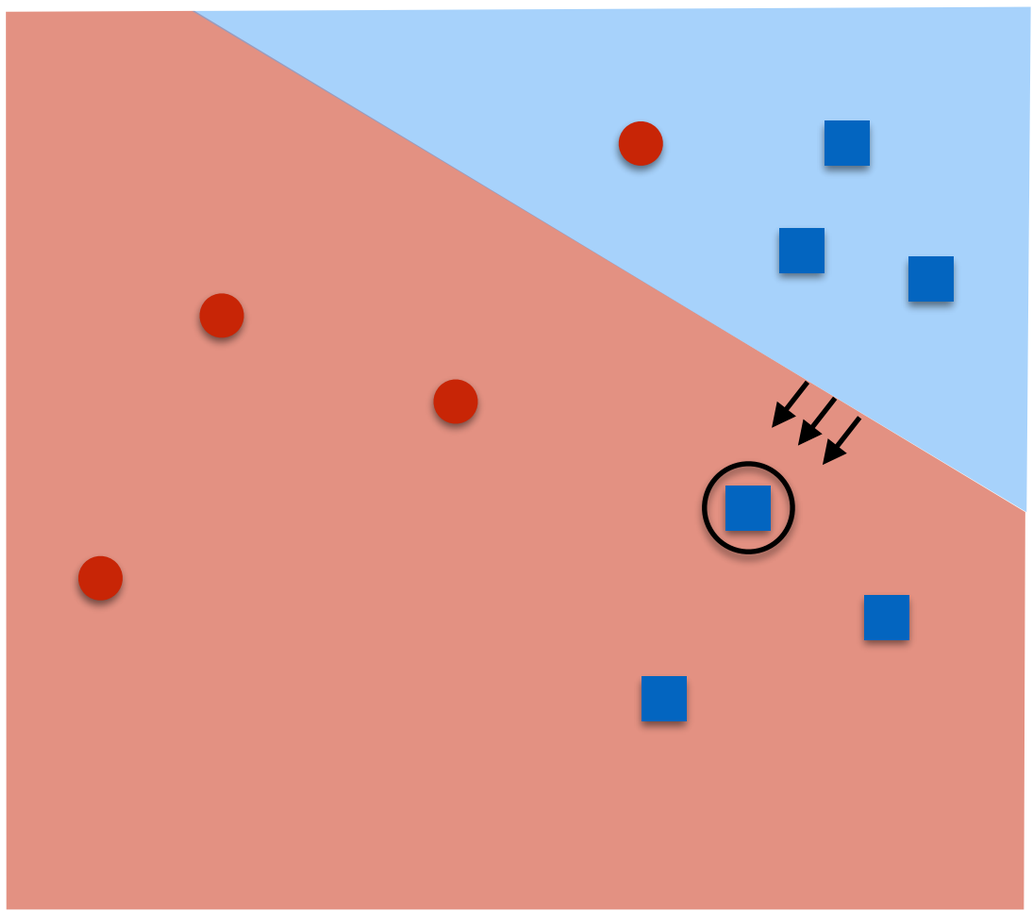
Random Weight Vector

Learned Weight Vector

Perceptron Learning Algorithm (PLA)

$$h(x) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + b\right) \longrightarrow h(x) = sign\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}\right)$$

$$\mathbf{w} = [b, w_1, w_2, \ldots, w_d] \;\; and \;\; \mathbf{x} = [1, x_1, x_2, \ldots, x_d]$$



Random Weight Vector

Let $w(t)$ be the weight vector at iteration t

for $t = 0$ until no example is misclassified

1. Pick a random sample $(x(t), y(t))$ from the set D which is misclassified
2. Update the weight vector with the following update rule

$$w(t+1) \leftarrow w(t) + y(t)x(t)$$

Note that since the example is misclassified

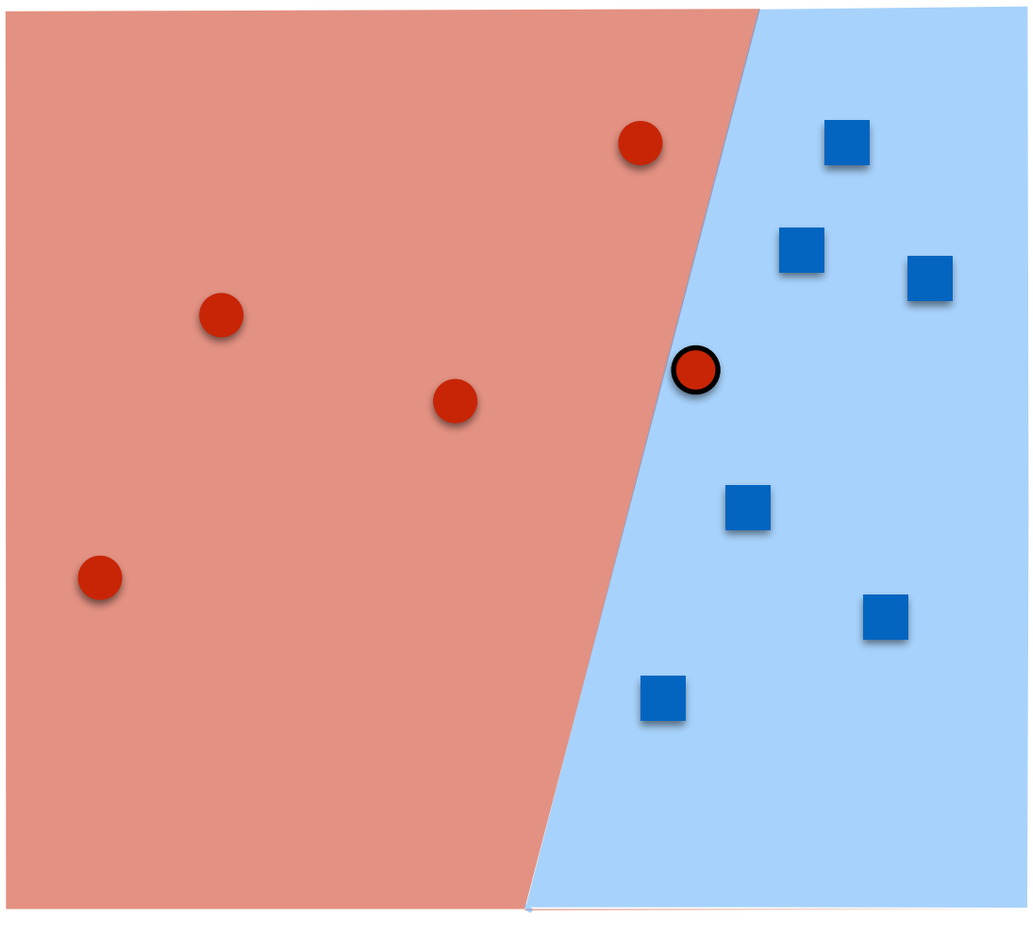
$$y(t) \neq sign(w(t)^T x(t))$$

One can prove that so long as the data is linearly separable the above algorithm will find a separating hyperplane

Will This Actually Work on Unseen Data?

$$h(x) = sign\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}\right)$$

$$\mathbf{w} = [b, w_1, w_2, \ldots, w_d] \;\; and \;\; \mathbf{x} = [1, x_1, x_2, \ldots, x_d]$$



Random Weight Vector

Note that the actual target function f is unknown

We only approximated it with g by looking at the limited amount of data D

What is the guarantee that g is actually a good approximation of f and will also work on unseen examples?

Probability will come to the rescue here

We have to make additional assumptions about what kinds of functions we want to consider —> need to posit structure that exists in both data we have seen and the data we have not seen (**a.k.a. inductive bias**)

There does not exist a “general inductive bias” that is the “right one” across all universe

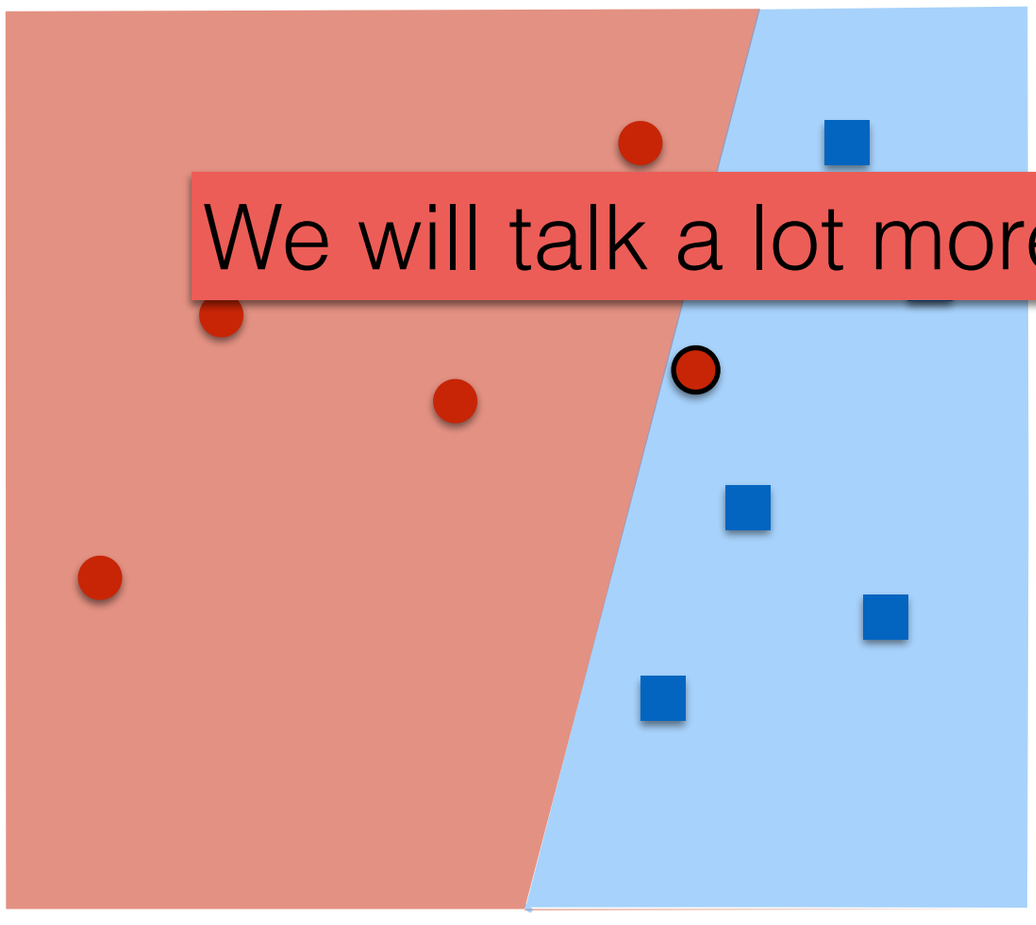
> **No Free Lunch Theorem** of Machine Learning
>
> For any set of target functions on which an inductive bias works well, it will work badly on the complement of that set

There is no Machine Learning without assumptions

Will This Actually Work on Unseen Data?

$$h(x) = sign\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}\right)$$

$$\mathbf{w} = [b, w_1, w_2, \ldots, w_d] \;\; and \;\; \mathbf{x} = [1, x_1, x_2, \ldots, x_d]$$



Random Weight Vector

Note that the actual target function f is unknown

We only approximated it with g by looking at the limited amount of data D

We will talk a lot more about this in the next lecture

tion of f

Probability will come to the rescue here

We have to make additional assumptions about what kinds of functions we want to consider —> need to posit structure that exists in both data we have seen and the data we have not seen (**a.k.a. inductive bias**)

There does not exist a “general inductive bias” that is the “right one” across all universe

No Free Lunch Theorem of Machine Learning

For any set of target functions on which an inductive bias works well, it will work badly on the complement of that set

There is no Machine Learning without assumptions

Learning Paradigms: Supervised Learning

You are explicitly provided with the input examples along with their annotations (labels)

E.g., a collection of images consisting of cats and dogs, with each image marked with what it contains

Online Learning

When the algorithm does not have access to all the data upfront. The the data is given to the algorithm one example at a time.

E.g., An autonomous robot learning how to drive on a new terrain

Active Learning

When the algorithm is allowed to query for the example (input and its label) for its training

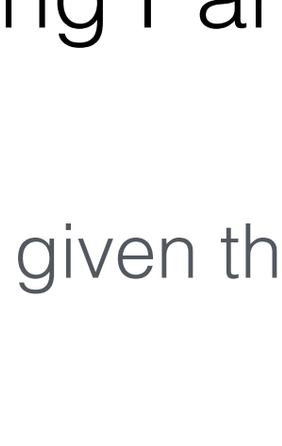
Typically this situation is employed when there is a high cost associated with getting labels for the input and one needs to train a model under a budget. A way to accomplish this efficiently is to enable the algorithm to query for examples which can provide the biggest bang for the buck

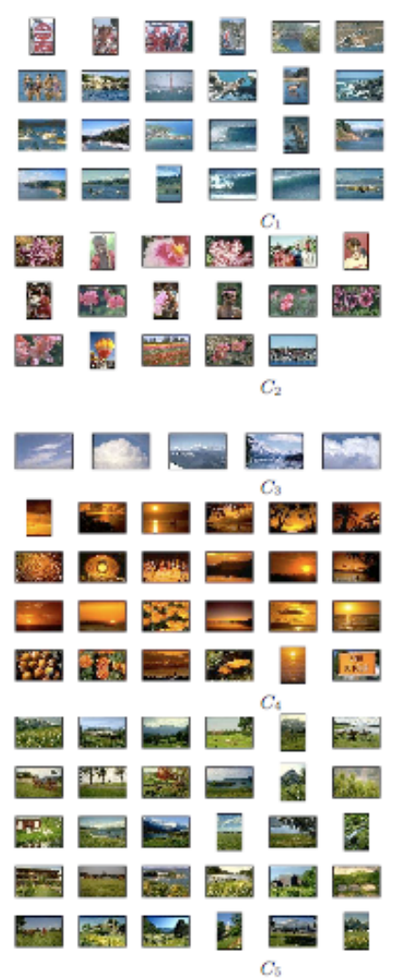
Learning Paradigms: Unsupervised Learning

You are only given the input examples and there are no annotations (labels)

While it might seem not be very useful to only have inputs without the labels, there are many useful things you can do with such a data

Clustering images

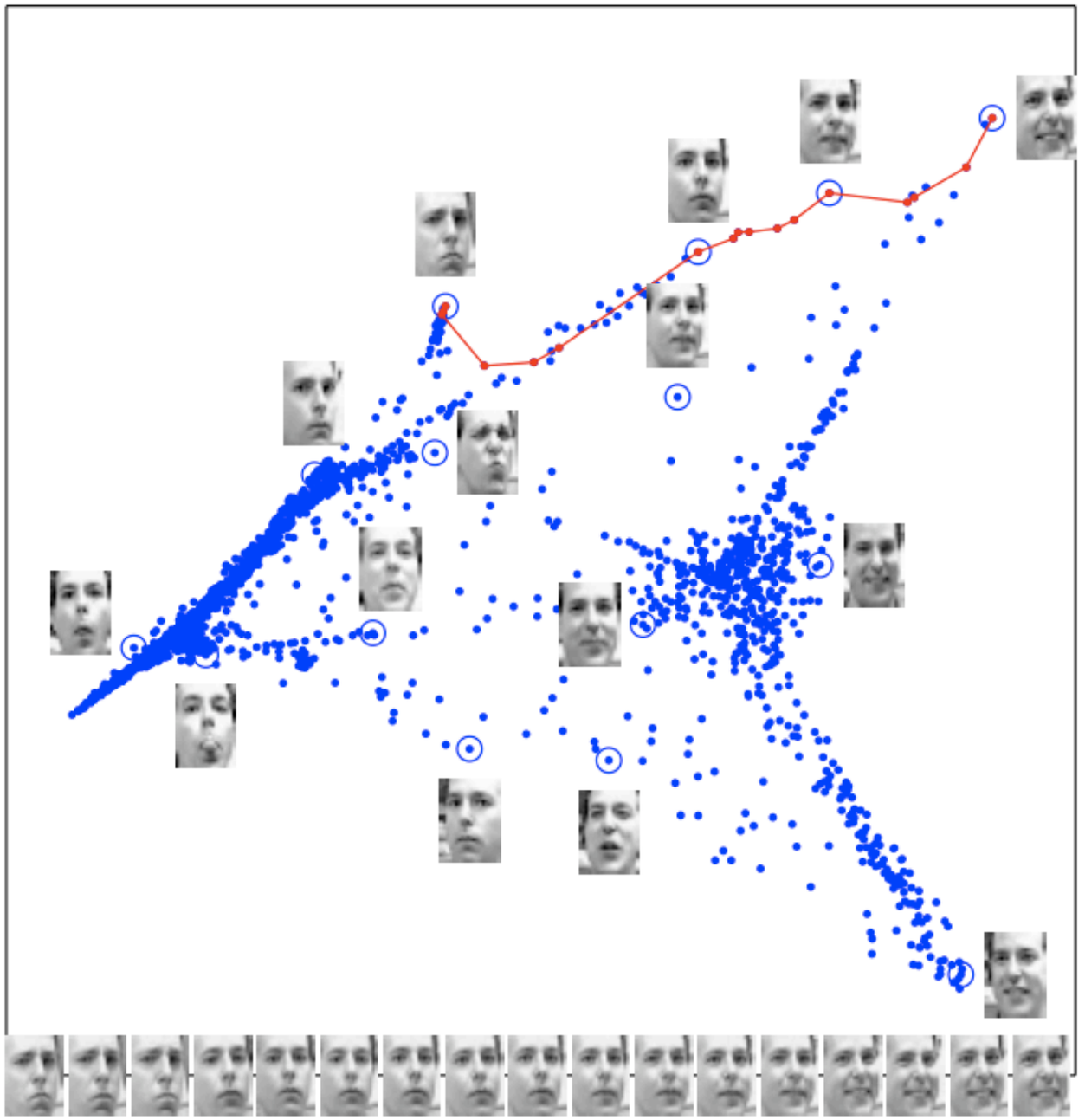




[Goldberger et al.]

Embedding images

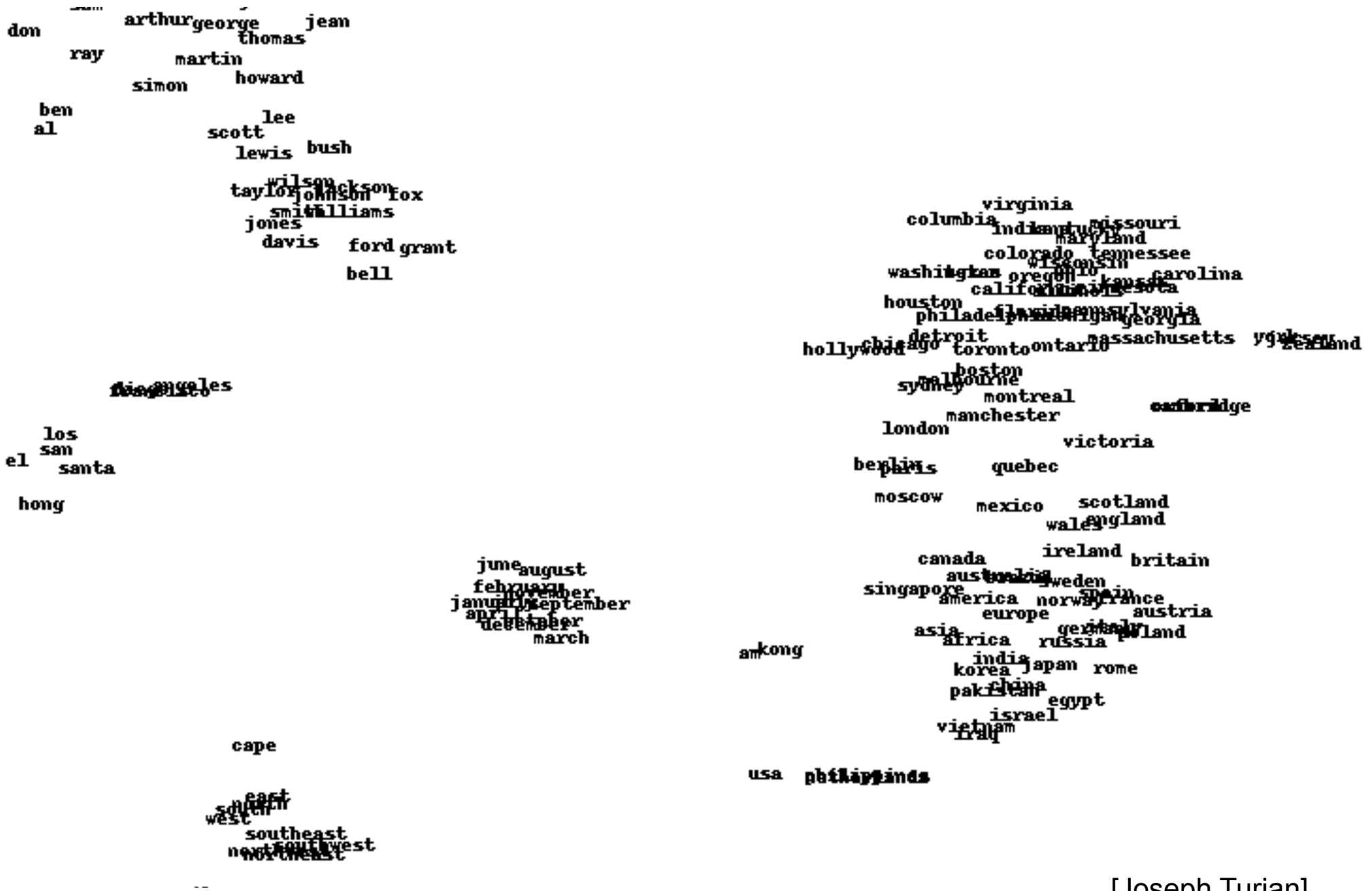
- Images have thousands or millions of pixels.
- Can we give each image a coordinate, such that similar images are near each other?



[Saul & Roweis ‘03]

Learning Paradigms: Unsupervised Learning

Embedding words



[Joseph Turian]

Learning Paradigms: Semi Supervised Learning

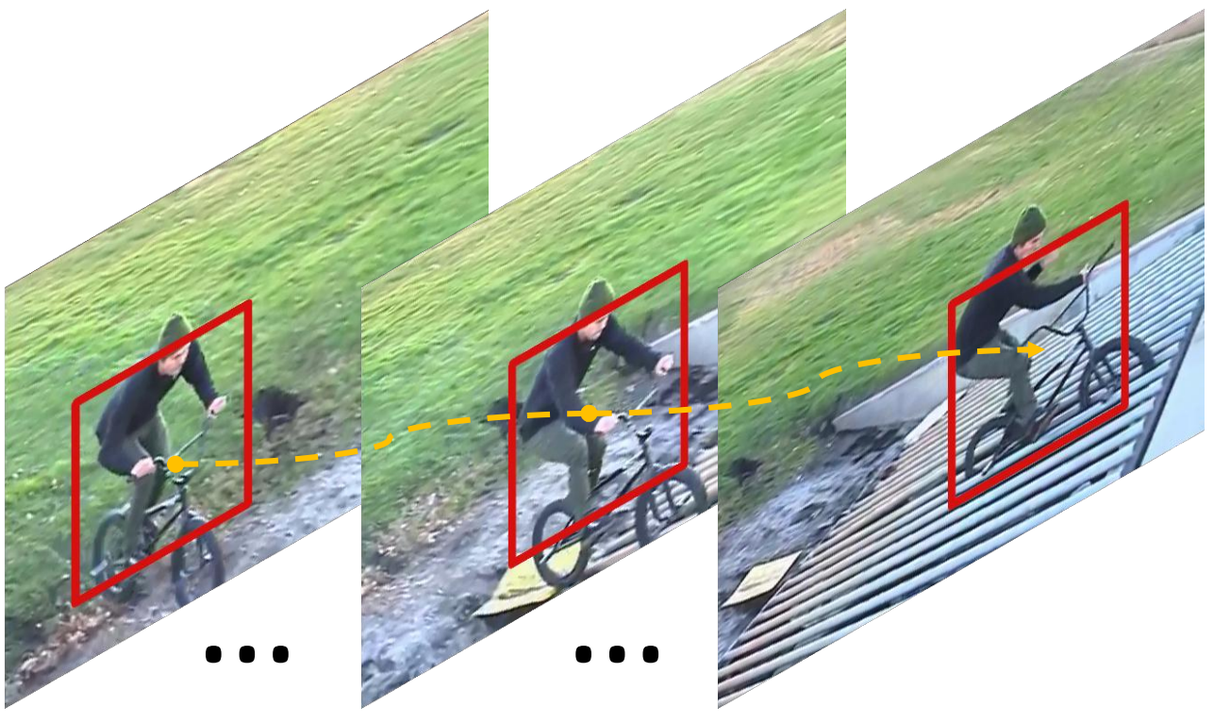
When you have labels for only a subset of input examples. One uses both the labeled and the unlabeled data to train the models

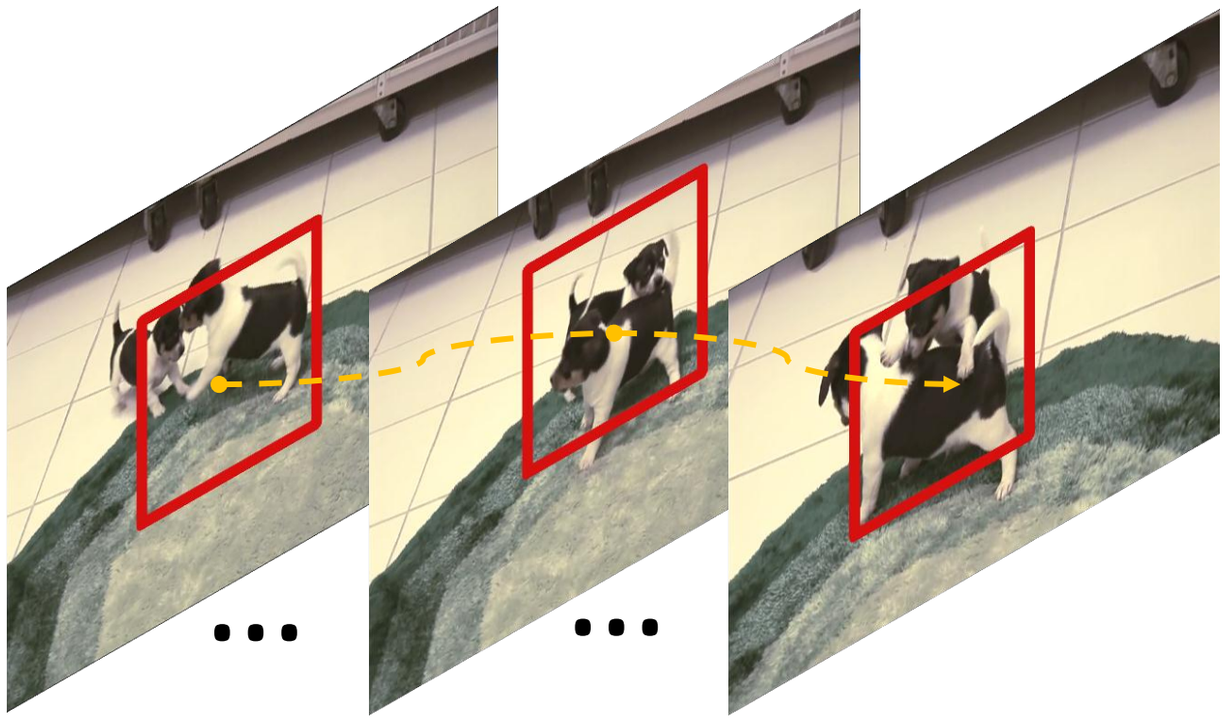
E.g., Label propagation

Learning Paradigms: Self-Supervised Learning

When you infer labels from a large collection of unlabeled data by exploiting other properties associated with the dataset

E.g., Exploiting the temporal correlations of video frames to deduce that images from nearby frames are similar to each other

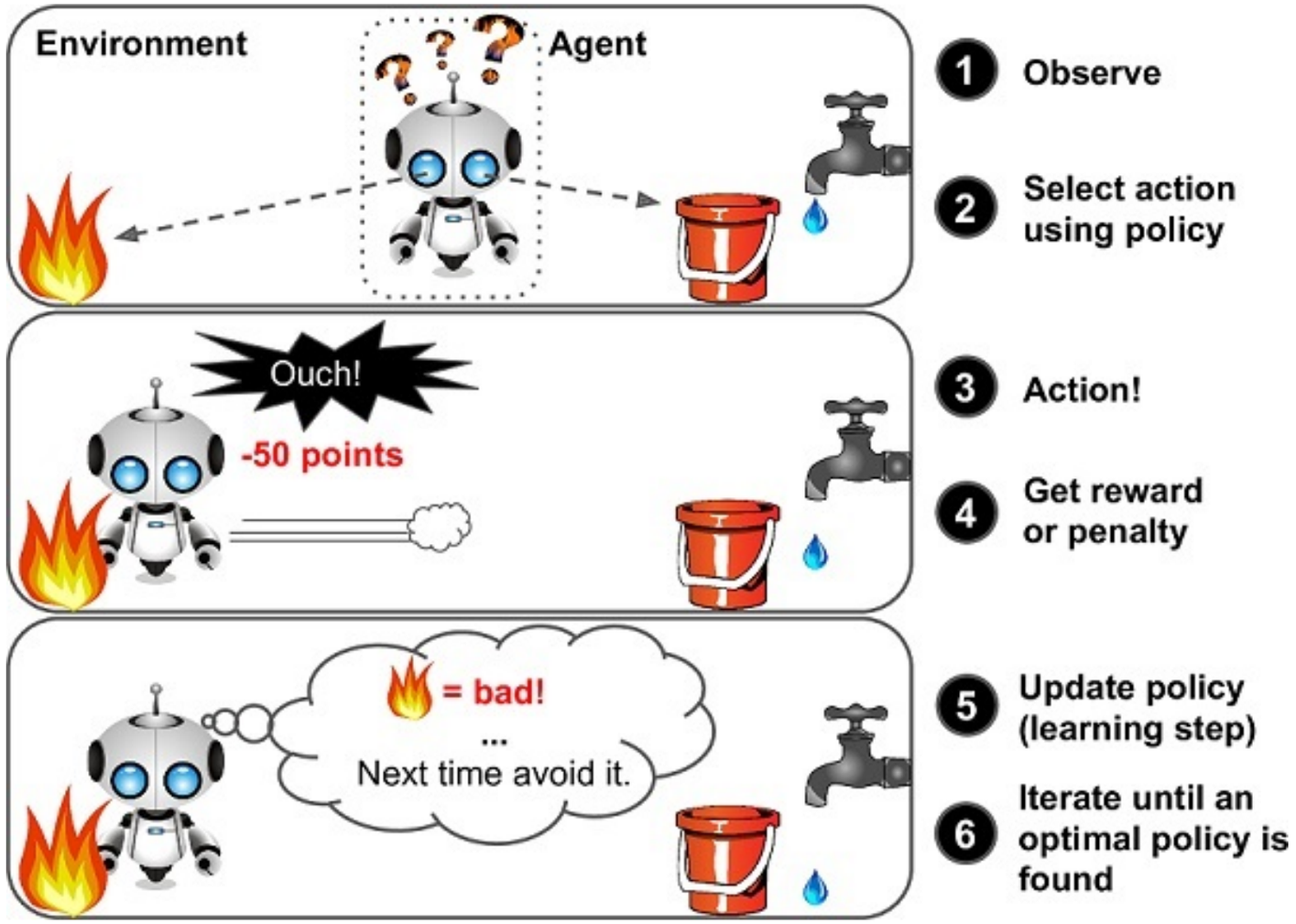




Learning Paradigms: Reinforcement Learning

When the training data does not contain an explicit label for each input. Instead the learning algorithm learns by trial and error. Given an input the algorithm makes a decision (takes an action) and gets feedback (+/-) from the environment. It uses feedback to improve (learn) itself.

Consider a toddler learning not to touch a hot cup. Has two options: 1) touch the hot cup or 2) not to touch the hot cup. **First action:** receives a huge negative reward (pain of burnt fingers). **Second action:** again receives a small negative reward (unsatisfied curiosity). After a few trials of touching and not touching (and accumulating rewards) the toddler eventually learns that it is better off not to touch



Important Questions in Machine Learning?

What data set to collect?

What labels to collect?

How to collect labels?

What is a parametric function?

How will you choose the function to train?

How will you actually train the parameters of the function?

How will you know when the function has been trained?

How will you measure the usefulness of the trained function?

What's the guarantee that a trained function that does well on the training data is actually working on unseen data when deployed?

And others..

Important Questions in Machine Learning?

What data set to collect?

What labels to collect?

How to collect labels?

What is a parametric function?

How will you choose the function to train?

All these questions will be answered in this course!

How will you know when the function has been trained?

How will you measure the usefulness of the trained function?

What's the guarantee that a trained function that does well on the training data is actually working on unseen data when deployed?

And others..

Course Overview

Machine Learning Overview: learning and generalization

Probabilistic interpretation of the concept

Linear parametric models

Non-linear parametric models

Unsupervised learning and Dimensionality Reduction

Mixture models and ensemble learning

Reinforcement Learning

Self-supervised learning

Components of a Learning System

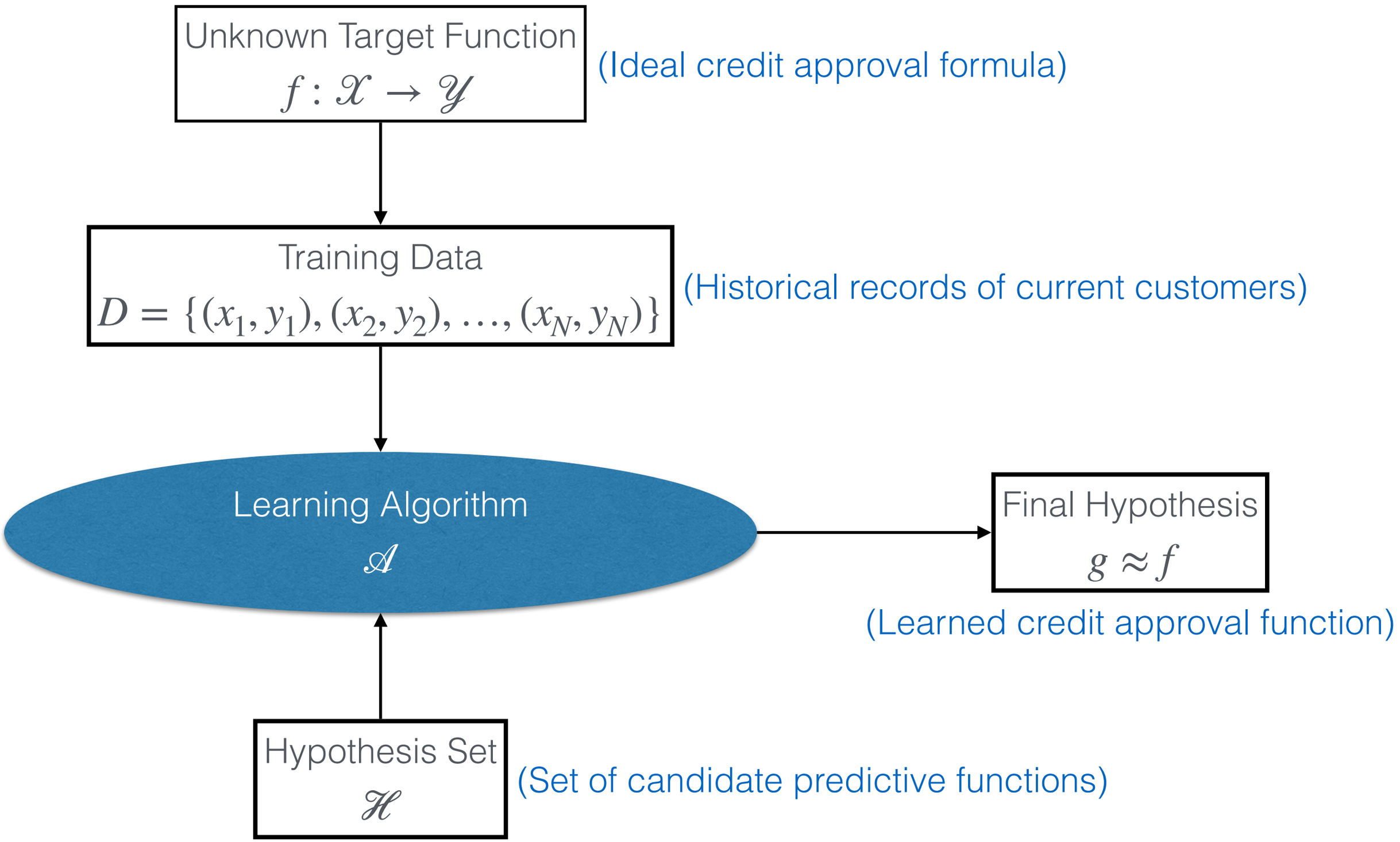


Figure adapted from the book Learning from Data: A Short Course

Hypothesis Space (a.k.a. Models)

Linear Parametric Models

Linear Models, Support Vector Machines

Non-Linear Parametric Models

Neural Networks, Convolutional Neural Network, Random Forests

Non-Parametric Models

Nearest Neighbors, Parzen Window Classifiers etc

Homework 0

- **Goal:** for you to calibrate yourself whether you have the right set of tools to succeed in this course.
- **Contents:** questions from probability, calculus, and linear algebra
- If you are not able to get a passing grade (6-7 out of 10 questions) then you should probably re-consider your decision. Unfortunately you will struggle otherwise. Don't want to spoil your GPA for graduate college.
- Again, no cheating please. The goal is to be honest to yourself and learn the subject!
- The assignment will be posted on Brightspace after the class.
- **Due date is: September 13th, 2021, 11:55 PM** (a day before last day of dropping classes)

See you next week!