Artificial Intelligence

CSC 665

tyler dae devlin

Machine Learning I

4.11.2024

- **Search:** make decisions by looking ahead
- **Logic:** deduce new facts from existing facts
- **Constraints:** find a way to satisfy a given specification
- **Probability:** reason quantitatively about uncertainty
- **Learning:** make future predictions from past observations

Learning — a different approach

- **So far:** studied methods that attempt to reproduce reasoning patterns of intelligent agents (especially humans)
- But
  - Humans use **different reasoning patterns** in different contexts
  - Often, a **combination** of such patterns is required
  - Often, it's **difficult to explicitly articulate** what patterns we're using
  - A significant amount of processing happens **subconsciously**
- All this makes it **difficult to program** a general-purpose agent that replicates human-level behavior and decision making

“Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child’s? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child.”

A. M. Turing, “Computing Machinery and Intelligence” (1950)

Reasons to learn

- **Can solve more problems:** even tasks that don't seem to take much intelligence can be very hard to explicitly program
  - It's easy for you to immediately say whether a photo is of a cat vs. dog
  - It's very hard to write a program that does the same thing
- **Can adapt:** systems that learn can react to future changes, unanticipated scenarios
- **Turing's idea:** easier to engineer a baby's brain and let it learn, than to build an adult's brain







Types of learning

- **Supervised:** given labeled data (input-output pairs) in a training set, find a function that maps inputs to outputs and can be used to predict outputs on new unseen inputs. *E.g. find a function that classifies dog vs. cat pictures.*
- **Unsupervised:** given unlabeled data, find structure in the data without any explicit feedback. *E.g. given a library of images, cluster them into groups — later we might find that one of these clusters consists mostly of cat pictures.*
- **Reinforcement:** take a sequence of actions and receive rewards and punishments; try to learn what actions will maximize rewards. *E.g. play Tetris.*

Supervised learning has been the most popular and successful of these.

Framework: empirical risk minimization

Goal: reproduce a mapping from inputs to outputs given a training set of example input-output pairs.

- **Representation:** How will we model the relationship between inputs and outputs?
- **Cost:** How will we evaluate whether we're successfully modeling the input-output relationship?
- **Optimizer:** How will we find the best possible (cost-minimizing) model among all possible choices given our representation?

Modeling

Inference

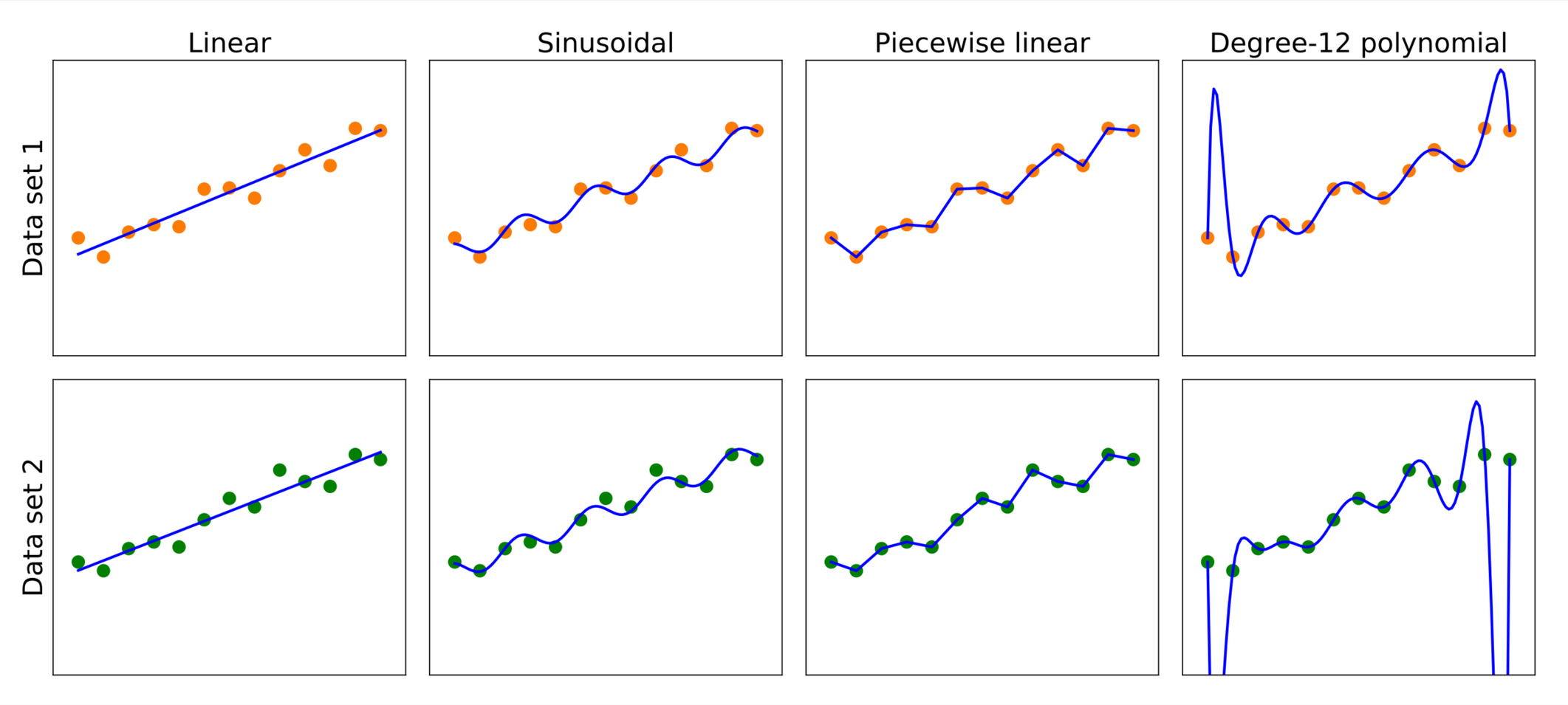
ERM, mathematically

Goal: approximate a function $f : \mathcal{X} \to \mathcal{Y}$ from features to labels given a training set of examples $(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_n, f(x_n))$.

- **Representation:** The hypothesis class $\mathcal{H}$ consisting of all possible candidates to approximate f. E.g. the set of all linear functions $\mathcal{H} = \{x \mapsto wx + b \mid w, b \in \mathbb{R}\}$.
- **Cost:** A function $C : \mathcal{H} \to \mathbb{R}$ that scores elements of the hypothesis class (higher cost is worse). Often the sum of pointwise costs on the data set, i.e. $\sum_{i=1}^{n} c(h(x_i), f(x_i))$ for some function $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. E.g. squared error $c(\hat{y}, y) = (\hat{y} - y)^2$.
- **Optimizer:** An algorithm for finding a hypothesis $h \in \mathcal{H}$ that minimizes $C(h)$. $\mathcal{H}$ is usually infinite, so you can't check all hypotheses. E.g. gradient descent.

Modeling

The hypothesis class matters



Picking the hypothesis class

Want a hypothesis class that

- can fit the training data reasonably well — low bias, doesn't underfit
- isn't too sensitive to the particulars of a specific training set — low variance, doesn't overfit

There is often a tradeoff between these two goals:

- "bias" vs. "variance"
- "approximation" vs. "generalization"

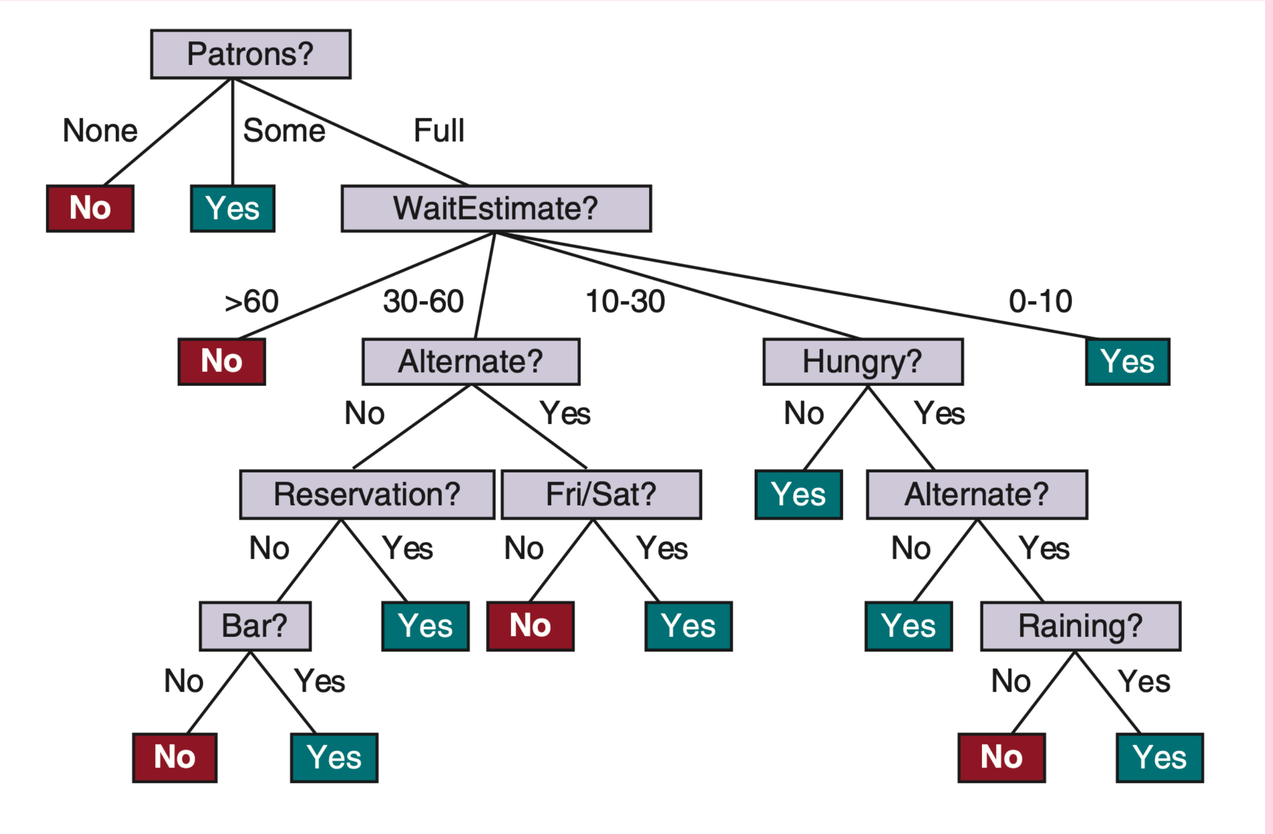
Will develop these ideas further later on

A learning example

Example	Input Attributes										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Decision trees

- First major hypothesis class: decision trees
- Internal nodes test a particular feature and branch based on the feature value
- Leaf nodes correspond to label predictions
- What types of hypotheses are contained in this $\mathscr{H}$?



Decision trees: expressivity

- For binary classification trees, can translate the tree into a logical formula
- $(\text{Output} = \text{Yes}) \iff (\text{Path}_1 \vee \text{Path}_2 \vee \ldots \vee \text{Path}_m)$
- Each Path_i is a conjunction representing a decision sequence in the tree leading to a label of "Yes"
  - $\text{Path}_1 = (\text{Patrons} = \text{Some})$
  - $\text{Path}_2 = (\text{Patrons} = \text{Full}) \wedge (\text{WaitEstimate} = \text{0-10})$
  - etc.
- So there is a bijective correspondence between decision trees and formulas in disjunctive normal form (DNF)
- But every Boolean function can be written in DNF, so binary classification trees can represent any Boolean function of the features