Classification: Basics

Sanmay Das

George Mason University

CS 584: Spring 2022

A classification example: Credit approval

- You apply for a credit card
- Bank decides whether to approve or deny
- What are we trying to learn, and from what?

A classification example: Credit approval

- You apply for a credit card
- Bank decides whether to approve or deny
- What are we trying to learn, and from what?
  - What: The “ideal credit approval function”

A classification example: Credit approval

- You apply for a credit card
- Bank decides whether to approve or deny
- What are we trying to learn, and from what?
  - What: The "ideal credit approval function"
  - From: Past data on customers (demographic, income, personal data) – *features* or *attributes*
  - *Labels* (which we're trying to predict) are some relevant outcome (default, profit, etc)

What is a Tree?

What is a Tree?

A brown trunk coming up from the ground, with branches extending out?[1]

[1] Abu-Mostafa, Magdon-Ismail, and Lin, 2012.

Are These Trees?

Are These Trees?

- Hard to define: I know it when I see it!
- I've *learned* it from data!

The Supervised Learning Problem

- Unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$
  - Classification is when $\mathcal{Y}$ is categorical (e.g. binary)
- Training data $\mathcal{D} : (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_n, y_n)$ where $y_i = f(\mathbf{x}_i)$ (possibly noisy).
- Want to learn h "close to" f.
- Two central questions:
  - How do we learn h?
    - Key algorithmic question!
  - What can we say about how close h is to f?
    - Why is this hard?

Generalization Error

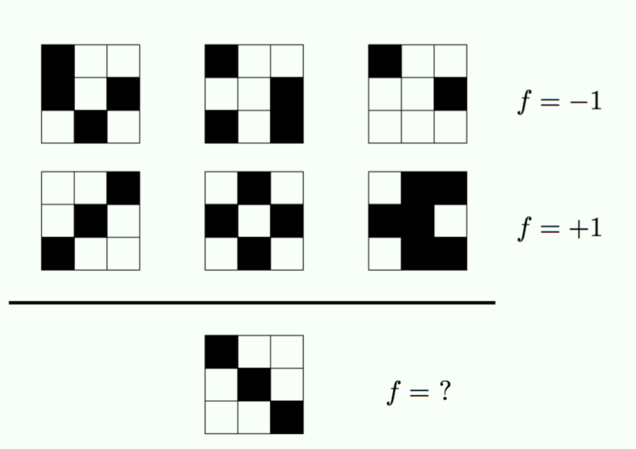
- Standard for closeness that we care about
  $E_{\text{out}}(h) = \Pr[h(\mathbf{x}) \neq f(\mathbf{x})]$
- In practice, we estimate E_{out} by evaluating on a (held-out) *test set*. We call this *test error*
- **Caution**: What happens when the sampling distribution for test data is not the same as that for $\mathcal{D}$?

How Do We Learn *f*?

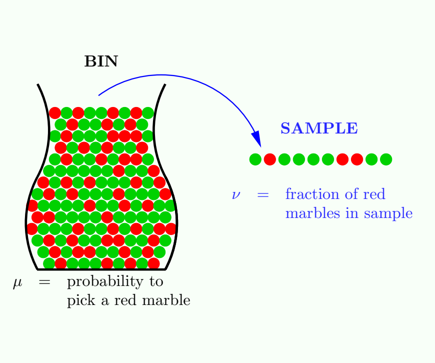
- Pick a *hypothesis set* $\mathcal{H} = \{h_1, h_2, \ldots,\}$
- Use a *learning algorithm* to select a hypothesis from $\mathcal{H}$ on the basis of $\mathcal{D}$.
- The choice of $\mathcal{H}$ and the learning algorithm are intertwined

How Do We Learn f?

- Pick a *hypothesis set* $\mathcal{H} = \{h_1, h_2, \ldots,\}$
- Use a *learning algorithm* to select a hypothesis from $\mathcal{H}$ on the basis of $\mathcal{D}$.
- The choice of $\mathcal{H}$ and the learning algorithm are intertwined
- No free lunch in machine learning



Probability to the Rescue



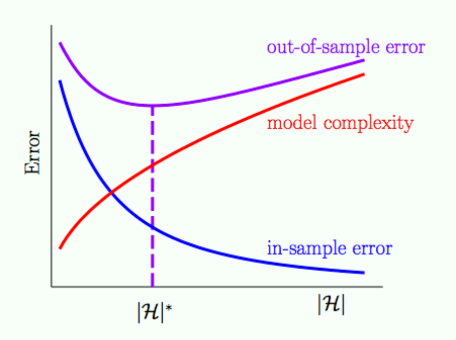
- What can we say about μ based on ν?
- There are many tools in probability theory for this, e.g. Hoeffding's inequality
- We can use similar arguments for ML algorithms, but have to be careful

Choosing h from $\mathcal{H}$

- First thought: Minimize **training error** $E_{\text{in}}(g) = \frac{1}{n}\sum_{i=1}^{n}[h(\mathbf{x}_i) \neq f(\mathbf{x}_i)]$
- Many algorithms can be thought of within this broad framework.
  - Linear regression: Find a weight vector **w** that minimizes $E_{\text{in}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - f(\mathbf{x}_i))^2$
  - Logistic regression: Find a linear function that minimizes $E_{\text{in}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\ln(1 + e^{-y_i\mathbf{w}^T\mathbf{x}_i})$
  - Decision trees: Find the tree that directly minimizes the above. Problem: Computationally intractable, so we use heuristics

Minimizing E_{out}

- But E_{in} is not really our objective. E_{out} is. A lot of theory and practice tells us that E_{out} is a combination of E_{in} and the complexity of the model you have learned



(from (Abu-Mostafa, Magdon-Ismail, and Lin, 2012))

- Gives us two objectives:
  - Control model complexity
  - Minimize E_{in}

The Central Problems

- There are deep relationships between the stability and variance of a learning algorithm, hypothesis complexity, and generalization ability.
- Bigger data $\rightarrow$ more complex hypothesis spaces can generalize better.
- Different ML algorithms arise from different choices related to two questions:
  - What $\mathcal{H}$ to search
  - What and how to optimize in the search process