Introduction

Machine Learning

Introduction

Definition (Machine Learning)

Spicy statistics/signal processing. Methods, algorithms, theorems, etc. to make inferences from data.

Three (main) types of learning:

- **Supervised learning.** Given data set of points and labels, find a prediction function that, given any point, finds a corresponding label.
  - Useful in: a lot. Analytics, autopilot, security, face recognition, etc.
- **Unsupervised learning.** Given data set of *only* points *(labels hidden to us)*, find a prediction function.
  - Useful in: cases where you only need similarity between datapoints, or have too much training data to label by hand. Collaborative filtering, data visualization, etc.
- **Reinforcement learning.** Given *no* data, but rather an environment, collect data from environment and learn to do a task optimally. Will skip this in this course.

So What Are We Actually Studying?

Mainly results/intuition/algorithms. Very few proofs, if any. (Take an actual class!)
Course organization: Data-Driven. Two types of data:

- **Unstructured Data.** (Useful for e.g. analytics, security, other traditional statistical tasks.)
    - Probabilistic classifiers.
    - Linear/logistic regression.
    - Regularization.
    - Dimensionality reduction.
    - Kernels.
    - Clustering.
    - Decision trees.
    - Boosting and ensembles.
- **Structured Data.** (Useful for e.g. image/text/audio processing.)
    - Sparse vector/matrix recovery.
    - Sparse vector/matrix completion.
    - Sparse/robust dimensionality reduction.
    - Dictionary learning.
- **But wait, where are neural networks?**
    - Last few lectures: neural networks for structured/unstructured data.

Notation/Vocabulary Dump

Different Spaces We Talk About

- $\mathcal{X}$ is **input space**.
  - For a list of numbers of length d, $\mathcal{X} = \mathbb{R}^d$.
  - For a 128×128 grayscale image, $\mathcal{X} = \mathbb{R}^{128\times128}$.
- $\mathcal{Y}$ is **output space**.
  - For a classification task into k classes, $\mathcal{Y} = \{1, 2, \ldots, k\} = [k]$.
  - For a regression task (predicting a continuous label), $\mathcal{Y} = \mathbb{R}$.
  - For an unsupervised representation learning task (embedding data into $\mathbb{R}^d$), $\mathcal{Y} = \mathbb{R}^d$.
- $\mathcal{Z}$ is **data space**.
  - In a supervised problem, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.
  - In an unsupervised problem, $\mathcal{Z} = \mathcal{X}$.
- $\mathcal{H}$ is a **hypothesis class** ; a class of functions $\mathcal{X} \to \mathcal{Y}$ we should care about

Technical note:
In general, we assume that these spaces have lots of structure: a rich topology, a measure on the Borel σ-algebra, and so on. This is fine because we can usually embed these spaces in $\mathbb{R}^d$ for large enough d and use the structure of $\mathbb{R}^d$ to give a structure to $\mathcal{X}$ and $\mathcal{Y}$. The question of $\mathcal{H}$ is a bit harder and requires functional analysis.

Notation/Vocabulary Dump

Definition, Assumptions (Data)

- The data point x_i is a random variable which has distribution μ_X on $\mathcal{X}$. Has density p_X.
  - Use x to denote arbitrary data point with distribution μ_X.
- The label y_i is a random variable which has distribution μ_Y on $\mathcal{Y}$. Has density p_Y.
  - Use y to denote arbitrary label with marginal μ_Y.
- The data z_i is a random variable which has distribution μ on $\mathcal{Z}$. Has density p_Z.
  - In a supervised learning problem, $z_i = (x_i, y_i)$.
  - In an unsupervised learning problem, $z_i = x_i$ and $\mu = \mu_X$.
  - Use z to denote arbitrary data with distribution μ.
- The data $\{z_i\}_{i=1}^n$ are independent and identically distributed (i.i.d.) as μ.

Definition (Sample)

A **sample** is n data points: $s = \{z_1, \ldots, z_n\} \sim \mu^n$. Has density p_S.
Sometimes we use the notation $s_X = \{x_1, \ldots, x_n\}$ or $s_Y = \{y_1, \ldots, y_n\}$.

Notation/Vocabulary Dump

Definition (Loss)

A **loss function** $\ell \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ quantifies how bad our prediction is. High loss = bad.

Example (Classification Loss)

A popular classification loss is **multi-class cross-entropy loss**:

$$\ell(y_{\text{true}}, y_{\text{pred}}) = -\sum_{i=1}^{k} y_{\text{true},i} \log\left(y_{\text{pred},i}\right)$$

Example (Squared-Error Loss)

A popular regression loss is **squared error loss**:

$$\ell(y_{\text{true}}, y_{\text{pred}}) = \left\| y_{\text{true}} - y_{\text{pred}} \right\|_2^2$$

Notation/Vocabulary Dump

Definition (Loss of A Hypothesis)

Given a hypothesis h, we can define the loss $\ell_h \colon \mathcal{Z} \to \mathbb{R}$.
In supervised learning, $\ell_h(z) = \ell_h((x, y)) = \ell(y, h(x))$.
Also define the loss over a sample s using L_s or distribution ν using L_ν:

$$L_s(h) = \mathop{\mathsf{E}}_{z \sim \mathrm{Uni}(s)} [\ell_h(z)] = \frac{1}{n} \sum_{i=1}^{n} \ell_h(z_i)$$

$$L_\nu(h) = \mathop{\mathsf{E}}_{z \sim \nu} [\ell_h(z)]$$

Machine Learning Problem

A machine learning *problem* is specified by the following:

- An **input space** $\mathcal{X}$.
- An **output space** $\mathcal{Y}$.
- An **loss function** ℓ.

(The hypothesis class $\mathcal{H}$ is a function of the algorithm. If we have prior knowledge what $\mathcal{H}$ *should* be, we can choose an algorithm that gets this $\mathcal{H}$.)

Machine Learning Solutions

How do we come up with solutions? Here's a standard workflow.

1. **Collect a sample** s. Split it into a training set s_{train} and a validation set s_{val}.
2. **Run a machine learning algorithm** $h = A(s_{\text{train}})$.
3. **Evaluate the output on the validation set.** Compute $L_{s_{\text{val}}}(h)$.
4. **Is your accuracy** $L_{s_{\text{val}}}(h)$ **good enough?**
    - If yes, you're done! Store the model somewhere for use, or keep improving it.
    - If not, either collect more data, improve data quality, or change the algorithm.

Data collection/quality is out of scope, but we can tune algorithms.

- Tune algorithm parameters (these are **hyperparameters** and usually set by hand). See k-fold cross-validation for how to efficiently put this into the workflow.
- Change the algorithm completely.