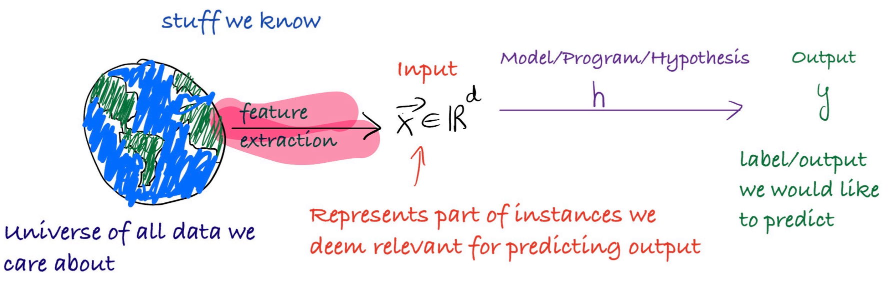
CS 3/5780: Introduction to Machine Learning

Lecture 2: Supervised Learning Setup

Supervised Learning



Feature Vectors

What feature vectors would you use for the following tasks:

1. Classify if a given email is spam or not
2. Predict the microsoft stock price for the next day
3. Where will jupiter be in the night sky tomorrow?
4. Is there a pedestrian in front of the car?

Finding a Good Hypothesis

- Traditional approach: Pay a programmer to program h
- Machine Learning: Learn h from examples or Data

Supervised Learning Setup

- Data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ measuring cost of error
- Model class $\mathcal{H}$ consisting of hypotheses or models $h : \mathcal{X} \mapsto \mathcal{Y}$
- Learning algorithm that picks hypothesis $h \in \mathcal{H}$ based on data $D+$

Data

- $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where x_i is the d-dimensional input feature vector for the i'th example and $y_i \in \mathcal{Y}$ is the output for i'th example
- Scenarios for $\mathcal{Y}$
  - Binary classification $\mathcal{Y} = \{0, 1\}$
  - Multiclass Classification $\mathcal{Y} = \{1, 2, \ldots, K\}$
  - Regression $\mathcal{Y}$ is the set of real numbers
  - Ranking $\mathcal{Y}$ is ordered set of choices
  - Structured prediction $\mathcal{Y}$ Eg. protein structure
- Feature vectors
  - Text: Bag of words features
  - Images: Raw pixels flattened to a vector

Loss Function

- Classification loss $\ell(y, y') = \mathbf{1}\{y' \neq y\}$
- Squared loss $\ell(y', y) = (y' - y)^2$
- Absolute loss $\ell(y', y) = |y' - y > |$

Generalization

- Idea: Find a model with low loss on Data D
- Does this always work?
- Eg. Algorithm Memorizer

Test/Train Split

- Split Data into training set D_{trn}, test set D_{tst} and validation set D_{val}
- Choose h based off of D_{trn} (+ D_{Val})
- Evaluate the algorithm on D_{tst}
- How should we split data?

Risk or Population Loss

- We often model data at deployment time as being drawn from a distribution P over input output pair
- iid draw from P often makes sense for non-temporal data
- Risk of a hypothesis:

$$\mathcal{E}(h) = \mathbb{E}_{(x,y)\sim P}\left[\ell(h(x), y)\right]$$

- Goal of ML algorithm is to pick hypothesis h based on data D to minimize risk $\mathcal{E}(h)$

Purpose of Split

- What is a good proxy for Risk?
- We use D_{trn} to learn/pick hypothesis h, and use D_{tst} to evaluate this hypothesis
- Why do we need D_{val}?

No Free Lunch!

- You must make assumptions in order to learn
- Corollary: No algorithm works in all settings
- For every method you encounter, question to ask: what is the underlying assumption needed to make this algorithm work?

Question 1: Spam filter you are given spam mail and labels for the past year

How would you split to test and train?

Question 2: Given hypothesis h learnt using D_{trn} I claim that

$$\mathcal{E}_{Dts}(h) = \frac{1}{|D_{tst}|} \sum_{(x,y)\in D_{tst}} l(h(x), y)$$

Is an unbiased and good estimate (when size of training set is large) for risk, why?