DS-GA 1003 Machine Learning
Lecture 1

February 2, 2021

1 Introduction to Machine Learning

- In most of our examples of machine learning problems, the outcome/label does not depend on the prediction. However, this is not always the case, e.g.
  - Search result ranking (contextual bandit)
  - Automated driving (reinforcement learning)
  - Fraud detection (game theory)
- Even though rule-based systems are largely replaced by machine learning systems now, they still have certain advantages:
  - Interpretable and easy to debug (decision trees can be considered as learning a bunch of rules)
  - Easy to implement business logic, e.g. never delete the email if it is from John
- For many tasks (especially those with structured data), we can easily collect useful rules, e.g. relation extraction ("[A] is married to [B]" implies that A and B are spouses), object recognition (A cat is unlikely to be in the sea). How can we incorporate these rules into a machine learning system?
  - Feature engineering: Put the patterns from the rules in the input
  - Semi-supervised learning: Use rules to assign noisy labels to unlabeled data (Snorkel)
  - Regularization: Penalize predictions too far from the rule-based prediction

  This is usually helpful when we do not have access to large amounts of labeled data, e.g. in medical domains.

2 Statistical Learning Theory

- Recall that our framework builds on top of three spaces: input space, action space and outcome space. What are the corresponding spaces for the following models?
  - Linear regression (input $\mathbb{R}^d$, action $\mathbb{R}$, outcome $\mathbb{R}$)
  - Logistic regression (input $\mathbb{R}^d$, action $(0, 1)$, outcome $\{0, 1\}$)
- What are some examples of hypothesis spaces?

  - Linear functions from $\mathbb{R}^d$ to $\mathbb{R}$
  - Polynomials of degree less than n
  - Neural networks

- Excess risk decomposition (draw picture on board):

$$\text{approximation error} = R(f_{\mathcal{F}}) - R(f^*) \tag{1}$$

$$\text{estimation error} = R(\hat{f}_n) - R(f_{\mathcal{F}}) \tag{2}$$

$$\text{optimization error} = R(\tilde{f}_n) - R(\hat{f}_n) \tag{3}$$

  - Can the approximation error be negative? [no]
  - If we make the hypothesis space "larger", does the approximation error increase or decrease? [decrease]
  - Is approximation error random? [no]
  - Can the estimation error be negative? [no]
  - If we make the hypothesis space "larger", does the estimation error increase or decrease? [We *expect* it to increase given fixed amount of data.]
  - Is estimation error random? [yes, the training set is random, so the emprical risk minimizer and its risk are both random.]
  - Can the optimization error be negative? [yes]

- Given insights from the excess risk decomposition, how can we debug machine learning algorithms in practice?

  - Check the gap between training loss and validation loss. If the gap is large, then it's probably overfitting. If the gap is small but the accuracy is low, then it's probably underfitting.
  - Run the algorithm on a small set first and check if it can overfit. If not, then the hypothesis space may be too "small" or the optimizer is not working.
  - Debug optimization:
    - Plot the learning curve: is the loss going down?
    - Check if gradient norm is close to zero around the solution.
    - Construct synthetic data where we know the minimizer (or have another way to find the minizer).