Statistics 231/CS229T Overview: What is this course about?

John Duchi

What is machine learning?

1. Statistics?
2. Optimization?
3. Computing?
4. Throw in some big datasets and competitions?

Goal of this course

Uncover common statistical principles underlying diverse array of machine learning techniques.

- Linear algebra
- Probability
- Optimization

A thought experiment

We train a Naive-Bayes classifier using bag-of-words features to predict the topic of a document (sports/politics/technology). It achieves 8% training error on $n = 1000$ training documents, and 13% on a held-out test set of 1000 documents

1. How reliable are these numbers? If we reshuffled the data, would we get the same answer?
2. How much should we expect the test error to change if we double the number of examples?
3. What if we double the number of features? What if our features or parameters are sparse?
4. What if we change the regularization?
5. Should we change the model and use an SVM with a polynomial kernel or a neural network?

The basic recipe in machine learning

Given a task

1. Choose a data representation/hypothesis class
2. Pick a loss
3. Get some data, minimize your loss on it

Example: Binary classification of news articles

Asymptotics

Classical statistics and the central limit theorem

Concentration and uniform convergence

Failures of asymptotic analysis?

- Smoothness
- Fixed dimension and $n = \infty$, essentially

Reconsider supervised learning with data pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, hypothesis class $\mathcal{H}$, where $h : \mathcal{X} \to \mathcal{Y}$, and loss $\ell(h; (x, y))$

- Learning algorithm chooses

$$\widehat{h} := \operatorname*{argmin}_{h \in \mathcal{H}} \widehat{L}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h; (X_i, Y_i))$$

- How does $\widehat{h}$ perform on future data?

Concentration and uniform convergence

Uniform convergence results tell us about gaps between $\widehat{L}_n$ and $L(h) := \mathbb{E}[\ell(h; (X, Y))]$ over *all* $h \in \mathcal{H}$

Online learning

Setting: data arrives sequentially

1. Learner receives new x_t
2. Learner makes prediction $\widehat{y}_t$
3. Learner suffers loss based on true label y_t
4. Learner updates parameters

Kernel methods, data representations, and more?

The dirty secret of machine learning: logistic regression with *one* better feature will beat your 10^6 horsepower model without it

Say we want to predict y from x via $h(x)$

- Typical approach: $h(x) = \theta^T \phi(x)$ and make ϕ complex

- Instead, use kernel $K(x, x')$ measuring "similarity" and use

$$h(x) := \sum_{i=1}^{n} \alpha_i K(x, x_i)$$

Kernel methods, data representations, and more?

A few issues with kernel methods: they are expensive (often n^2 even at test)

- Random features:

- Neural networks?

What you should get out of this class

1. Theoretical analysis will *usually not* tell you that something is going to beat one thing or another

2. Theoretical analysis *can* give you general insights, and explain how you might make changes to improve your algorithms