CPSC 340:
Machine Learning and Data Mining

Fundamentals of Learning

Bonus slides

Golden Rule of Machine Learning

- Even though what we care about is test error:
  - THE TEST DATA CANNOT INFLUENCE THE TRAINING PHASE IN ANY WAY.

Tom Simonite
June 4, 2015

Why and How Baidu Cheated an Artificial Intelligence Test

Machine learning gets its first cheating scandal.

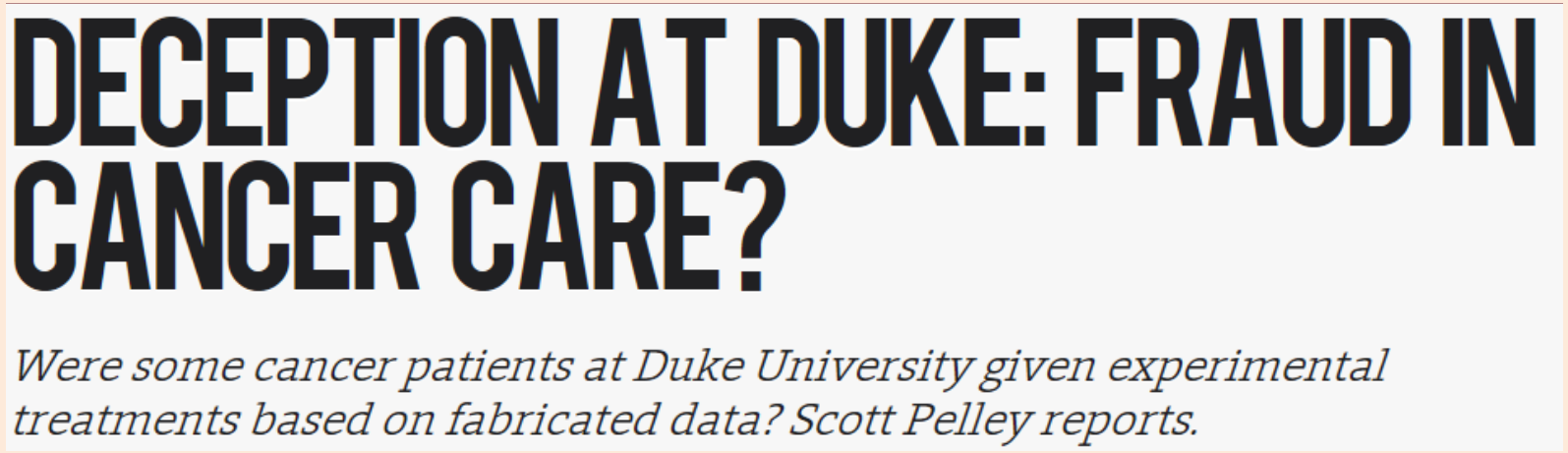
The sport of training software to act intelligently just got its first cheating scandal. Last month Chinese search company Baidu announced that its image recognition software had inched ahead of Google's on a standardized

http://www.technologyreview.com/view/538111/why-and-how-baidu-cheated-an-artificial-intelligence-test/

Golden Rule of Machine Learning

- Even though what we care about is test error:
  - THE TEST DATA CANNOT INFLUENCE THE TRAINING PHASE IN ANY WAY.

- You also shouldn't change the test set to get the result you want.



  - http://blogs.sciencemag.org/pipeline/archives/2015/01/14/the_dukepotti_scandal_from_the_inside

https://www.cbsnews.com/news/deception-at-duke-fraud-in-cancer-care/

Digression: Golden Rule and Hypothesis Testing

- Note the golden rule applies to hypothesis testing in scientific studies.
  - Data that you collect can't influence the hypotheses that you test.

- EXTREMELY COMMON and a MAJOR PROBLEM, coming in many forms:
  - Collect more data until you coincidentally get significance level you want.
  - Try different ways to measure performance, choose the one that looks best.
  - Choose a different type of model/hypothesis after looking at the test data.

- If you want to modify your hypotheses, you need to test on new data.
  - Or at least be aware and honest about this issue when reporting results.

Digression: Golden Rule and Hypothesis Testing

- Note the golden rule applies to hypothesis testing in scientific studies.
  - Data that you collect can't influence the hypotheses that you test.

- EXTREMELY COMMON and a MAJOR PROBLEM, coming in many forms:
  - "Replication crisis in Science".
  - "Why Most Published Research Findings are False".
  - "False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant".
  - "HARKing: Hypothesizing After the Results are Known".
  - "Hack Your Way To Scientific Glory".
  - "Psychology's Replication Crisis Has Made The Field Better" (some solutions)

Approximation Error for Selecting Hyper-Parameters

- From the 2019 EasyMarkit AI Hackathon:
  - "We ended up selecting the hyperparameters that gave us the lowest approximation error (gap between train and validation) as opposed to the lowest validation error. This was quite a difficult decision for our team since we were only allowed one submission. However, the model with the lowest validation error had a very high approximation error, which felt too risky, so we went with a model with a slightly higher validation error and much lower approximation error. When the results were announced, the reported test accuracy was within 0.1% of what our model predicted with the validation set."

- This is the type of reasoning you want to do.
  - A high approximation error could indicate low validation error by chance.

“A visual Introduction to machine learning”

- The “housing prices” example is taken from this website:
  - http://www.r2d3.us/visual-intro-to-machine-learning-part-1
- They also have a “Part 2” here:
  - http://www.r2d3.us/visual-intro-to-machine-learning-part-2
- Part 2 covers similar topics to what we covered in this lecture.

Bounding E_{approx}

- Let's assume we have a fixed model 'h' (like a decision tree), and then we collect a training set of 'n' examples.
- What is the probability that the error on this training set (E_{train}), is within some small number ε of the test error (E_{test})?

- From "Hoeffding's inequality" we have:

$$P\left[\left|E_{train}(h) - E_{test}(h)\right| > \varepsilon\right] \leq 2\exp(-2\varepsilon^2 n)$$

- This is great! In this setting the probability that our training error is far from our test error goes down exponentially in terms of the number of samples 'n'.

Bounding E_{approx}

- Unfortunately, the last slide gets it backwards:
  - We usually don't pick a model and then collect a dataset.
  - We usually collect a dataset and then pick the model 'w' based on the data.
- We now picked the model that did best on the data, and Hoeffding's inequality doesn't account for the optimization bias of this procedure.

- One way to get around this is to bound ($E_{test} - E_{train}$) for *all* models in the space of models we are optimizing over.
  - If we bound it for all models, then we bound it for the best model.
  - This gives looser but correct bounds.

Bounding E_{approx}

- If we only optimize over a finite number of events 'k', we can use the "union bound" that for events $\{A_1, A_2, \ldots, A_k\}$ we have:

$$p(A_1 \cup A_2 \cup \cdots \cup A_K) \leq \sum_{i=1}^{k} p(A_i)$$

- Combining Hoeffding's inequality and the union bound gives:

$$p\left(|E_{train}(h_1) - E_{test}(h_1)| > \varepsilon \cup |E_{train}(h_2) - E_{test}(h_2)| > \varepsilon \cup \cdots \cup |E_{train}(h_K) - E_{test}(h_K)| > \varepsilon\right)$$

$$\leq \sum_{i=1}^{k} p\left(|E_{train}(h_i) - E_{test}[h_i]| > \varepsilon\right)$$

$$\leq \sum_{i=1}^{k} 2\exp(-2\varepsilon^2 n)$$

$$\leq 2k\exp(-2\varepsilon^2 n)$$

Bounding E_{approx}

- So, with the optimization bias of setting "h*" to the best 'h' among 'k' models, probability that ($E_{test} - E_{train}$) is bigger than ε satisfies:

$$p(|E_{train}(h^*) - E_{test}(h^*)| > \varepsilon) \leq 2k\exp(-2\varepsilon^2 n)$$

- So optimizing over a few models is ok if we have lots of examples.
- If we try lots of models then ($E_{test} - E_{train}$) could be very large.
- Later in the course we'll be searching over continuous models where k = infinity, so this bound is useless.
- To handle continuous models, one way is via the VC-dimension.
  - Simpler models will have lower VC-dimension.

Refined Fundamental Trade-Off

- Let E_{best} be the irreducible error (lowest possible error for *any* model).
  - For example, irreducible error for predicting coin flips is 0.5.
- Some learning theory results use E_{best} to futher decompose E_{test}:

$$E_{test} = \underbrace{(E_{test} - E_{train})}_{\text{"variance"}} + \underbrace{(E_{train} - E_{best})}_{\text{"bias"}} + \underbrace{E_{best}}_{\text{"noise"}}$$

- This is similar to the bias-variance decomposition:
  - Term 1: measure of variance (how sensitive we are to training data).
  - Term 2: measure of bias (how low can we make the training error).
  - Term 3: measure of noise (how low can any model make test error).

Refined Fundamental Trade-Off

- Decision tree with high depth:
  - Very likely to fit data well, so bias is low.
  - But model changes a lot if you change the data, so variance is high.
- Decision tree with low depth:
  - Less likely to fit data well, so bias is high.
  - But model doesn't change much you change data, so variance is low.

- And degree does not affect irreducible error.
  - Irreducible error comes from the best possible model.

Bias-Variance Decomposition

- You may have seen "bias-variance decomposition" in other classes:
  - Assumes $\tilde{y}_i = \bar{y}_i + \varepsilon$, where ε has mean 0 and variance σ^2.
  - Assumes we have a "learner" that can take 'n' training examples and use these to make predictions $\hat{y}_i$.
- Expected squared test error in this setting is

$$E[(\tilde{y}_i - \hat{y}_i)^2] = E[(\hat{y}_i - \bar{y}_i)]^2 + (E[\hat{y}_i^2] - E[\hat{y}_i]^2) + \sigma^2$$

"test squared error" "bias" "variance" "noise"

  - Where expectations are taken over possible training sets of 'n' examples.
  - Bias is expected error due to having wrong model.
  - Variance is expected error due to sensitivity to the training set.
  - Noise (irreducible error) is the best can hope for given the noise (E_{best}).

Bias-Variance vs. Fundamental Trade-Off

- Both decompositions serve the same purpose:
  - Trying to evaluate how different factors affect test error.

- They both lead to the same 3 conclusions:
  1. Simple models can have high E_{train}/bias, low E_{approx}/variance.
  2. Complex models can have low E_{train}/bias, high E_{approx}/variance.
  3. As you increase 'n', E_{approx}/variance goes down (for fixed complexity).

Bias-Variance vs. Fundamental Trade-Off

- So why focus on fundamental trade-off and not bias-variance?
  - Simplest viewpoint that gives these 3 conclusions.
  - No assumptions like being restricted to squared error.

  - You can measure E_{train} but not E_{approx} (1 known and 1 unknown).
    - If E_{train} is low and you expect E_{approx} to be low, then you are happy.
      - E.g., you fit a very simple model or you used a huge independent validation set.

  - You can't measure bias, variance, or noise (3 unknowns).
    - If E_{train} is low, bias-variance decomposition doesn't say anything about test error.
      - You only have your training set, not distribution over possible datasets.
      - Doesn't say if high E_{test} is due to bias or variance or noise.

Learning Theory

- Bias-variance decomposition is a bit weird compared to our previous decompositions of E_{test}:
  - Bias-variance decomposition considers expectation over *possible training sets*.
  - But doesn't say anything about test error with *your* training set.

- Some keywords if you want to learn about learning theory:
  - Bias-variance decomposition, sample complexity, probably approximately correct (PAC) learning, Vapnik-Chernovenkis (VC) dimension, Rademacher complexity.

- A gentle place to start is the "Learning from Data" book:
  - https://work.caltech.edu/telecourse.html

A Theoretical Answer to "How Much Data?"

- Assume we have a source of IID examples and a fixed class of parametric models.
  - Like "all depth-5 decision trees".
- Under some nasty assumptions, with 'n' training examples it holds that:
  E[test error of best model on training set] – (best test error in class) = O(1/n).

- You rarely know the constant factor, but this gives some guidelines:
  - Adding more data helps more on small datasets than on large datasets.
    - Going from 10 training examples to 20, difference with best possible error gets cut in half.
      - If the best possible error is 15% you might go from 20% to 17.5% (this does **not** mean 20% to 10%).
    - Going from 110 training examples to 120, error only goes down by ~10%.
    - Going from 1M training examples to 1M+10, you won't notice a change.
  - Doubling the data size cuts the error in half:
    - Going from 1M training to 2M training examples, error gets cut in half.
    - If you double the data size and your test error doesn't improve, more data might not help.

Can you test the IID assumption?

- In general, testing the IID assumption is not easy.
  - Usually, you need background knowledge to decide if it's reasonable.

- Some tests do exist, like shuffling the order of data and then measuring if some basic statistics agree.
  - It's reasonable to check if summary statistics of train and test data agree.
    - If not, your trained model may not be so useful.

- Some discussion here:
  - https://stats.stackexchange.com/questions/28715/test-for-iid-sampling

Wrong Decisions under false IID Assumption

There is a different narrative that one can tell about the current era. Consider the following story, which involves humans, computers, data, and life-or-death decisions, but where the focus is something other than intelligence-in-silicon fantasies. When my spouse was pregnant 14 years ago, we had an ultrasound. There was a geneticist in the room, and she pointed out some white spots around the heart of the fetus. "Those are markers for Down syndrome," she noted, "and your risk has now gone up to one in 20." She let us know that we could learn whether the fetus in fact had the genetic modification underlying Down syndrome via an amniocentesis, but amniocentesis was risky—the chance of killing the fetus during the procedure was roughly one in 300. Being a statistician, I was determined to find out where these numbers were coming from. In my research, I discovered that a statistical analysis had been done a decade previously in the UK in which these white spots, which reflect calcium buildup, were indeed established as a predictor of Down syndrome. I also noticed that the imaging machine used in our test had a few hundred more pixels per square inch than the machine used in the UK study. I returned to tell the geneticist that I believed that the white spots were likely false positives, literal white noise.

She said, "Ah, that explains why we started seeing an uptick in Down syndrome diagnoses a few years ago. That's when the new machine arrived."

We didn't do the amniocentesis, and my wife delivered a healthy girl a few months later, but the episode troubled me, particularly after a back-of-the-envelope calculation convinced me that many thousands of people had gotten that diagnosis that same day worldwide, that many of them had opted for amniocentesis, and that a number of babies had died needlessly. The problem that this episode revealed wasn't about my individual medical care; it was about a medical system that measured variables and outcomes in various places and times, conducted statistical analyses, and made use of the results in other situations. The problem had to do not just with data analysis per se, but with what database researchers call *provenance*—broadly, where did data arise, what inferences were drawn from the data, and how relevant are those inferences to the present situation? While a trained human might be able to work all of this out on a case-by-case basis, the issue was that of designing a planetary-scale medical system that could do this without the need for such detailed human oversight.

I'm also a computer scientist, and it occurred to me that the principles needed to build planetary-scale inference-and-decision-making systems of this kind, blending computer science with statistics, and considering human utilities, were nowhere to be found in my education. It occurred to me that the development of such principles—which will be needed not only in the medical domain but also in domains such as commerce, transportation, and education—were at least as important as those of building AI systems that can dazzle us with their game-playing or sensorimotor skills.

https://hdsr.mitpress.mit.edu/pub/wot7mkc1/release/8