CPSC 540: Machine Learning

More Fundamentals of Learning

Winter 2021

Last Time: Violating the Golden Rule?

- Usual strategy for hyper-parameter tuning:
  - Optimize performance on a validation set.

- This can lead to overfitting to the validation set.

- We showed a bound on the amount of overfitting:

$$p(|E_{test} - E_{valid(\lambda)}| > \varepsilon \text{ for } \underline{\text{any}}\ \lambda) \leq k\, 2\exp(-2\varepsilon^2 t)$$

  - Probability of overfitting increases linearly in ‘k’:
    - “Number of hyper-parameter values you are optimizing over.
  - Probability of overfitting decreases exponentially in ‘t’:
    - Number of IID examples in validation set.
- You can violate the golden rule, even quite a bit, with a big validation set.

Generalization Error

- An alternative measure of performance is the generalization error:
  - Average error over the set of x^i values that are not seen in the training set.
  - "How well we expect to do for a *completely unseen* feature vector".
- Test error vs. generalization error when labels are deterministic:

$$E_{test} = \mathbb{E}[|\hat{y}^i - \tilde{y}^i|] \qquad E_{generalize} = \frac{1}{t}\sum_{\tilde{x}^i \notin \{\text{train set}\}} |\hat{y}_i - \tilde{y}_i|$$

Labels are deterministic, but we still take expectation over data distribution

number of x^i values not in training set

average error over unseen x^i values.

“Best” and the “Good” Machine Learning Models

- Question 1: what is the “best” machine learning model?
  - The model that gets lower generalization error than all other models.
- Question 2: which models always do better than random guessing?
  - Models with lower generalization error than “predict 0” for all problems.

- No free lunch theorem:
  - There is **no** “best” model achieving the best generalization error for every problem.
  - If model A generalizes better to new data than model B on one dataset, there is another dataset where model B works better.

No Free Lunch Theorem

- Let's show the "no free lunch" theorem in a simple setting:
  - The x^i and y^i are binary, and y^i being a deterministic function of x^i.
- With 'd' features, each "learning problem" is a map from $\{0,1\}^d \rightarrow \{0,1\}$.
  - Assigning a binary label to each of the 2^d feature combinations.

Feature 1	Feature 2	Feature 3
0	0	0
0	0	1
0	1	0
…	…	…

y (map 1)	y (map 2)	y (map 3)	…
0	1	0	…
0	0	1	…
0	0	0	…
…	…	…	…

- Let's pick one of these 'y' vectors ("maps" or "learning problems") and:
  - Generate a set training set of 'n' IID samples.
  - Fit model A (convolutional neural network) and model B (naïve Bayes).

No Free Lunch Theorem

- Define the "unseen" examples as the (2^d – n) not seen in training.
  - Assuming no repetitions of x^i values, and $n < 2^d$.
  - Generalization error is the average error on these "unseen" examples.
- Suppose that model A got 1% error and model B got 60% error.
  - We want to show model B beats model A on another "learning problem".

- Among our set of "learning problems" find the one where:
  - The labels y^i agree on all training examples.
  - The labels y^i disagree on all "unseen" examples.
- On this other "learning problem":
  - Model A gets 99% error and model B gets 40% error.

No Free Lunch Theorem

- Further, across all "learning problems" with these 'n' examples:
  - Average generalization error of **every** model is 50% on unseen examples.
    - It's right on each unseen example in exactly half the learning problems.
  - With 'k' classes, the average error is (k-1)/k (random guessing).

- This is kind of depressing:
  - For general problems, no "machine learning" is better than "predict 0".

(pause)

Limit of No Free Lunch Theorem

- Fortunately, the world is structured:
  - Some "learning problems" are more likely than others.

- For example, it's usually the case that "similar" x^i have similar y^i.
  - Datasets with properties like this are more likely.
  - Otherwise, you probably have no hope of learning.

- Models with right "similarity" assumptions ("bias") can beat "predict 0".

- With assumptions like this, you can consider consistency:
  - As 'n' grows, model A converges to the optimal test error.

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 further decompose E_{test}:

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

- This is similar to the bias-variance trade-off (bonus slide):
  - E_{approx} measures *how sensitive we are to training data* (like "variance").
  - E_{model} measures *if our model is complicated enough to fit data* (like "bias").
  - E_{best} measures how low can **any** model make test error ("irreducible" error).

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 further decompose E_{test}:

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

- This is similar to the bias-variance trade-off (bonus slide):
  - You need to trade between having low E_{approx} and having low E_{model}.
  - Powerful models have low E_{model} but can have high E_{approx}.
  - E_{best} does not depend on what model you choose.

Consistency and Universal Consistency

- A model is consistent for a particular learning problem if:
  - E_{test} converges to E_{best} as 'n' goes to infinity, for that particular problem.

- A model is universally consistent for a class of learning problems if:
  - E_{test} converges to E_{best} as 'n' goes to infinity, for all problems in the class.

- Class of learning problems is usually be "all problems satisfying":
  - A continuity assumption on the labels y^i as a function of x^i.
    - E.g., if x^i is close to x^j then they are likely to receive the same label.
  - A boundedness assumption of the set of x^i.

K-Nearest Neighbours (KNN)

- Classical consistency results focus on k-nearest neighbours (KNN).
- To classify an object $\tilde{x}_i$:
  1. Find the ‘k’ training examples x_i that are “nearest” to $\tilde{x}_i$.
  2. Classify using the most common label of “nearest” examples.

Egg	Milk	Fish
0	0.7	0
0.4	0.6	0
0	0	0
0.3	0.5	1.2
0.4	0	1.2

X

Sick?
1
1
0
1
1

y

Egg	Milk	Fish
0.3	0.6	0.8

$\tilde{x}_i$

Sick?
?

$\tilde{y}_i$

3 "nearest" neighbours

most common label is "sick"

Consistency of KNN (Discrete/Deterministic Case)

- Let's show universal consistency of KNN in a simplified setting.
  - The x^i and y^i are binary, and y^i being a deterministic function of x^i.
    - Deterministic y^i implies that E_{best} is 0.

- Consider KNN with k=1:
  - After we observe an x^i, KNN makes right test prediction for that vector.
  - As 'n' goes to ∞, each feature vectors with non-zero probability is observed.
  - We have $E_{test} = 0$ once we've seen all feature vectors with non-zero probability.

- Notes:
  - No free lunch isn't relevant as 'n' goes to ∞ here: we eventually see everything.
    - There are 2^d possible feature vectors, so might need a huge number of training examples.
  - It's more complicated if labels aren't deterministic and features are continuous.

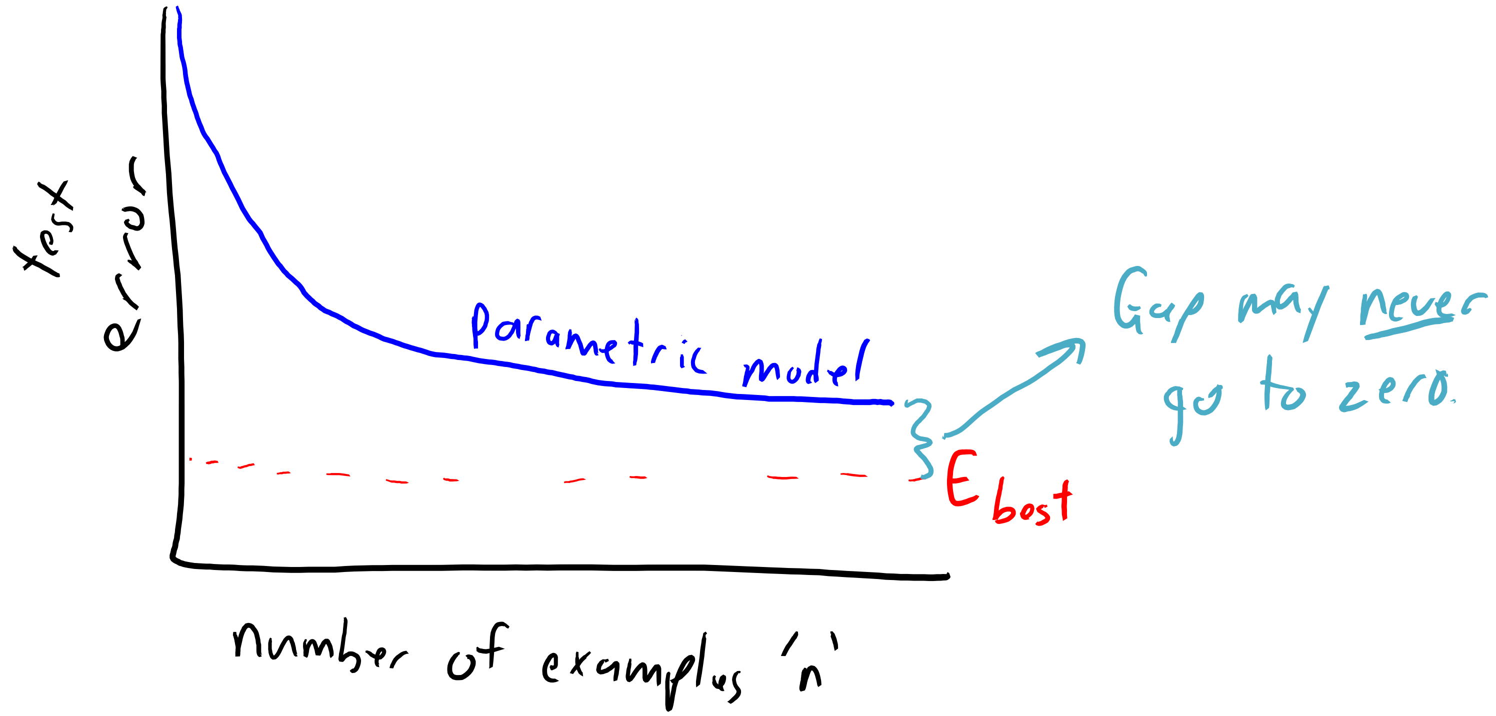
Consistency of KNN (Continuous/Non-Deterministic)

- KNN consistency properties (under reasonable assumptions):
  - As 'n' goes to ∞, $E_{test} \leq 2E_{best}$.
    - For fixed 'k' and binary labels.
- Stone's Theorem: KNN is "universally consistent".
  - If 'k' converges to ∞ as 'n' converges to ∞, but k/n converges to 0, E_{test} converges to E_{best}.
    - For example, k = O(log n).
    - First algorithm shown to have this property.
- Consistency says nothing about finite 'n'.
  - See "Dont Trust Asymptotics".

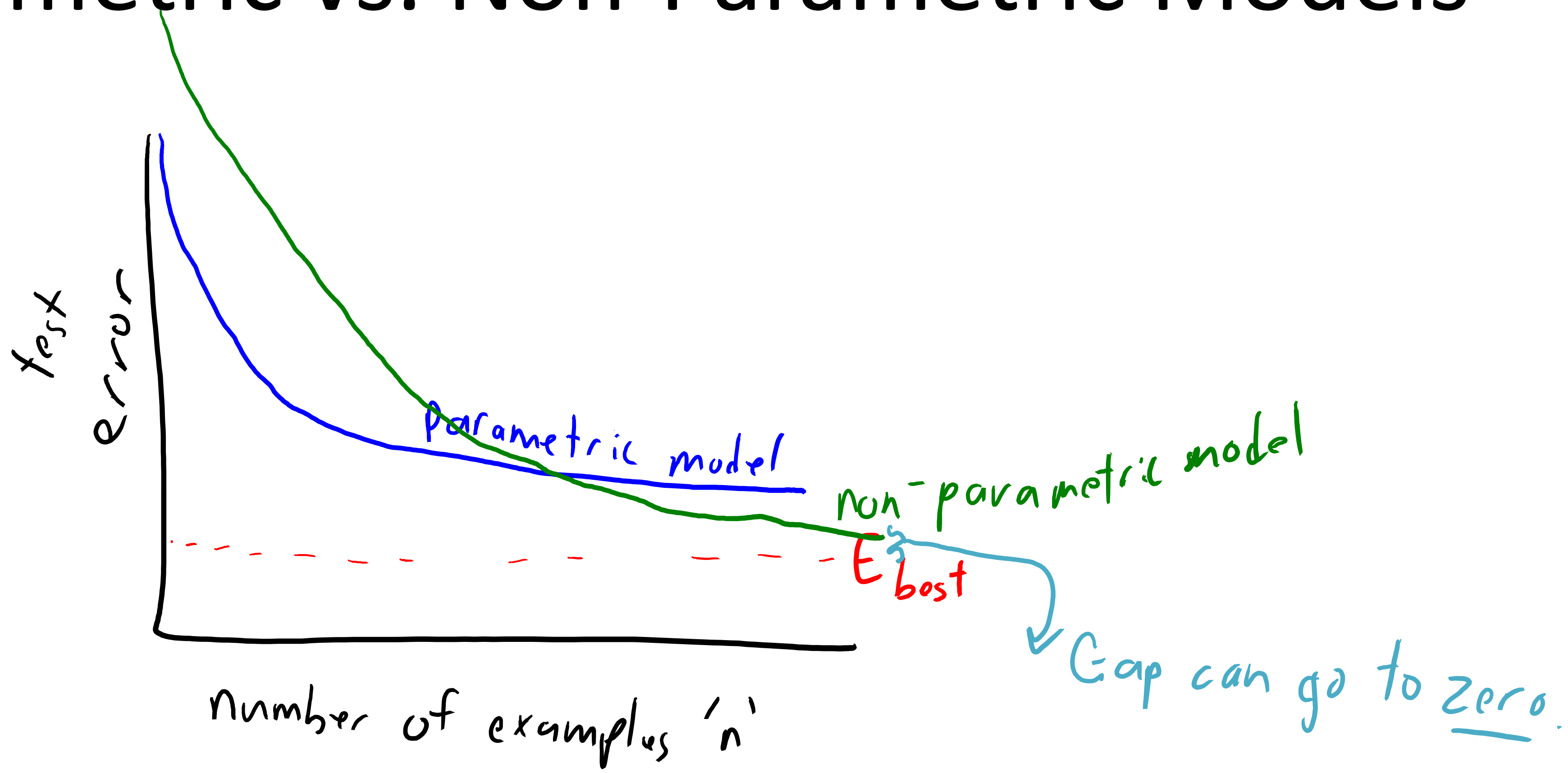
Consistency of Non-Parametric Models

- Universal consistency can be been shown for a variety of models:
  - Linear models with polynomial basis.
  - Linear models with Gaussian RBFs.
  - Neural networks with one hidden layer and standard activations.
    - Sigmoid, tanh, ReLU, etc.
- It's non-parametric versions that are consistent:
  - Size of model is a function of 'n'.
  - Examples:
    - KNN needs to store all 'n' training examples.
    - Degree of polynomial must grow with 'n' (not true for fixed polynomial).
    - Number of hidden units must grow with 'n' (not true for fixed neural network).

Parametric vs. Non-Parametric Models



Parametric vs. Non-Parametric Models



Summary

- No free lunch theorem:
  - There is no “best” or even “good” machine learning models across all problems.
- Universal consistency:
  - Some non-parametric models can solve any continuous learning problem.

- Next time:
  - More about convexity than you ever wanted to know.