Data For Machine Learning

There have been studies of using different classification algorithms on data

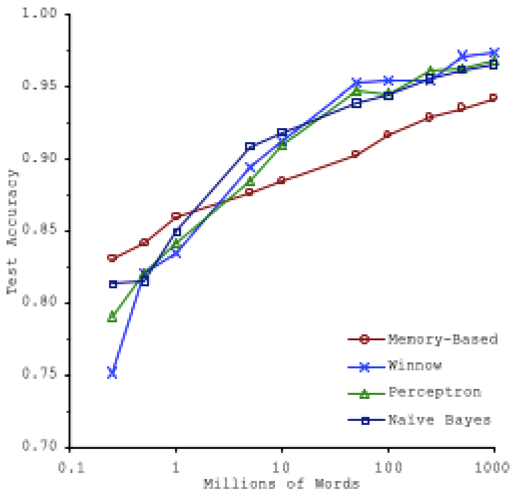
Designing a high accuracy learning system

- E.g. Classify between confusable words.
  - cofusing words: {to, two, too}, {then, than}

Algorithms:

- Perceptron (Logistic Regression)
- Winnow
- Memory-based
- Naives Bayes

Varied training set size and tried algorithms on a range of sizes



What can we conclude

- Algorithms give remarkably similar performance
- As training set sizes increases accuracy increases
- Take an algorithm, give it more data, should beat a "better" one with less data

Large Data Rationale

- Assume feature $x \in \mathbb{R}^{n+1}$ has sufficient information to predict y accurately.
  - Example: For breakfast I ate **two** eggs (Confusing words: two, too, to).
  - Concretely, if we can correctly assume that features x have enough information to predict y accurately, then more data will probably help
- **Counterexample:** Predict housing price from only size (feet^2) and no other features (the number of rooms in the house, how nicely furnished the house is?, or if the house is new or old).

Useful test: Given the input x, can a human expert confidently predict y?

Let's say we use a learning algorithm with many parameters (e.g. logistic regression/linear regression with many features; neural network with many hidden units), and we use features that have enough information to predict the value of y. These are powerful learning algorithms with many parameters which can fit complex functions, such algorithms are low bias algorithms.

- $J_{train}(\theta)$ will be small.

Use a very large training set (unlikely to overfit)

- $J_{train}(\theta) \approx J_{test}(\theta)$
- Unlikely to over fit with our complex algorithms (low variance)
- So the test set error should also be small

Another way to think about this is we want our algorithm to have low bias and low variance

- Low bias --> use complex algorithm
- Low variance --> use large training set

And fundamentally it's a key ingredients of assuming that the features have enough information and we have a rich class of functions that's why it guarantees low bias, and then it having a massive training set that's what guarantees low variance.

Video Question: Having a large training set can help significantly improve a learning algorithm’s performance. However, the large training set is **unlikely** to help when:

> The features x do not contain enough information to predict y accurately (such as predicting a house’s price from only its size), and we are using a simple learning algorithm such as logistic regression.

- We are using a learning algorithm with a large number of features (i.e. one with “low bias”).

> The features x do not contain enough information to predict y accurately (such as predicting a house’s price from only its size), even if we are using a neural network with a large number of hidden units.

- We are not using regularization (e.g. the regularization parameter λ = 0).