Chapter 1 Advice for Applying Machine Learning

This chapter is about how to implement powerful algorithm.

Sometimes, a better algorithm is better than more data. But usually, we should try :

- more training examples
- smaller sets of features
- additional features
- polynomial features
- changing λ

We need **Machine Learning Diagnostic** to test whether we can gain insight that if it works on it. A diagnostic will take time to implementm, but it will be worth the time.

1.1 Evaluating a Hypothesis

Less error is not always meaning better, for it's possible to generalize to new examples.

We can divide dataset into training set and test set. Normally training set is of 70%.

Learn from training set and compute via test set. You can apply the cost or misclassification error.

$$\text{err}(h_\theta(x), y) = \begin{cases} 1, \text{if } h_\theta(x) \geq 0.5, y = 0 \\ 1, \text{if } h_\theta(x) \leq 0.5, y = 1 \\ 0, \text{else} \end{cases}$$

And the total cost is

$$\text{Test Err} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \text{err}(h_\theta(x_{test}^{(i)}), y^{(i)})$$

Via this, we can simply check whether the algorithm is proper or not.

1.1.1 Model Selection and Train/Validation/Test Sets

How to decide the degree of polynomial or regularized parameters?

Once the parameters are overfitting a dataset, the error will be lower than it should be.

We can do model selection over the degree of the polynomial functions. But the cost in training is not a fair estimate of the model generalization. We should divide the dataset into 3 parts rather than 2: training set(60%), cross validation set(20%), test set(20%). And get error for each part, then treat the loss of cross validation set as a quality of generalization.

1.2 Bias vs. Variance

1.2.1 Diagnosing Bias vs. Variance

High bias turns to underfitting and high variance turns to overfitting. It's important to determine which kind of problem we are facing.

We can see them from the training error and cross validation error. As degree of polynomial increasing, the error will decrease. And the cross validation error will be higher than it, like **Figure ??**.

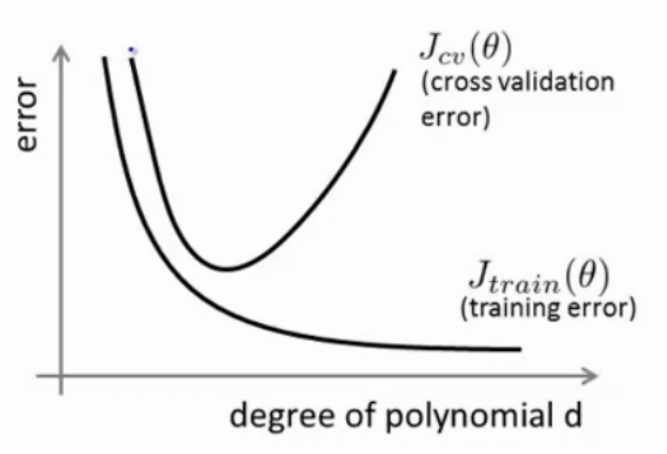

Figure 1.1: The error with degree of polynomial

If the bias is too big, the $J_{train}(\theta) \approx J_{cv}(\theta)$ will be high; If the variance is too big, $J_{train}(\theta) \ll J_{cv}(\theta)$

1.2.2 Regularization and Bias/Variance

If we have applied regularization to our algorithm, we will get an extra λ. If the λ is too large, the parameters will be rather small and then the bias gets too big then underfitting. If too small, similarly, overfitting.

So, we need to try different λ, and observe the relationship between loss and λ.

1.2.3 Learning Curves

Learning curves are easy to plot in learning and could show the learning condition. It's an (error-m(training set size)) curve. As the set size increasing from 0, the error would increase and the speed to increase in decreasing. At the same time, the cv-error would decrease.

If trapped in high bias, finally we will get $J_{cv} = J_{train}$. So, if algorithm is suffering from high bias, more data will not help so much/

If trapped in high variance, more data will help.

1.2.4 Decide What to Do next

All we mentioned told us what's useful for our ML work. Let's have a summary:

- Get more training examples to fix high variance
- Try less sets of features to fix high vairan
- Try more features to fix high bias
- Try more polynomial features to fix high bias
- Try decreasing λ to fix high bias
- Try increasing λ to fix high variance

Small NNs has fewer parameters and more prone to underfitting. Large ones have more parameters and likely t overfitting and **MORE EXPENSIVE**. Try regularization to address overfitting.

Low order polynomials have high bias and load variance, while high order polynomials have high variance and low bias.

1.3 Build a Classifier

It's a problem about 1 and 0. We need x as the features of mails and y is the class of the mails.

First choose some features that indicative of spam or not.

How to make it have low error ?

- collect lots of data
- develop sophisticated features based on email routing info from email header.
- develop sophisticated features for the message
- develop algorithm for misspellings

1.3.1 Error Analysis

Recommended approach:

- Start with a simple algorithm that can implement very quickly.
- Plot learning curves to decide if more data, more features that could help.
- Error analysis, manually examine the examples.

Usually over the evaluation set.

1.3.2 Error Metrics for Skewed Classes

If the data is skewed to a side, the model's precision is not so impressive.

Use precision and recall. Like true positive, false positive, false positive and true negative. You get the precision and recall to see the prediction and the actual fact.

Word

sad Avenues