Evaluating a Learning Algorithm

Debugging

Suppose we have implemented regularized linear regression to predict housing prices with the following cost function:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) + \lambda\sum_{j=1}^{m}\theta_j^2\right]$$

If we test test the hypothesis on a new set of houses, and it ends up making predictions with huge errors, we can try the following:

- Get more training examples - Fixes high variance
- Try smaller set of features - Fixes high variance
- Try getting additional features - Fixes high bias sometimes
- Try adding polynomial features ($x_1^2, x_2^2, \cdots, x_1 \cdot x_2$, etc) - Fixes high bias usually
- Try decreasing λ (Regularization Parameter) - Fixes high bias
- Try increasing λ (Regularization Parameter) - Fixes high variance

Machine Learning Diagnostic

It is a test that we can run to gain insight on what is/isn't working with a learning algorithm, and gain guidance as to how best to improve its performance. It can take time to implement and understand, but doing so can be very useful.

Evaluating a Hypothesis

For problems with only one feature, we can just plot and look at our hypothesis function and get an intuition on what's going wrong. But if we have multiple features, it isn't really viable to plot the hypothesis function.

So, a good practice would be:

- Split the dataset into 70% training set and 30% test set
- If the dataset is sorted, mix it up and make sure the 70-30 split is completely random
- For Linear Regression
    - Learn the parameter θ from the training data [minimizing training error $J(\theta)$]
    - Compute the test set error:

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

- For Logistic Regression:
    - Learn parameter θ from the training data\
    - Compute the test set error:\

$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$$

- Misclassification error (0/1 misclassification error):

$$err(h_\theta(x), y) = \begin{cases} 1 & \text{if } h_\theta(x) \geq 0 \text{ and } y = 0 \\ 1 & \text{if } h_\theta(x) < 0 \text{ and } y = 1 \\ 0 & \text{if classified correctly} \end{cases}$$

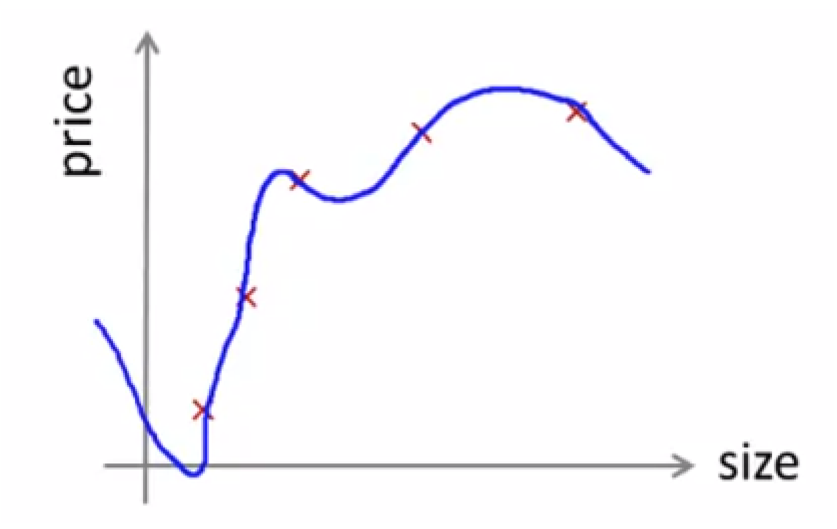
  Therefore,

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

Model Selection & Train/Validation/Test

Let's consider an overfitting example. Once parameters $\theta_0, \theta_1, \cdots, \theta_4$ were fit to some sort of data (training set), the error of the parameters as measured on the data [the training error $J(\theta)$] is likely to be lower than the actual generalization error.

$$h_\theta(x) = \theta_0 + \theta_1(x) + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$



Model Selection:

- $d = 1$ where $h_\theta(x) = \theta_0 + \theta_1 x$ and minimize to get parameter vector $\theta^{(1)}$
- $d = 2$ where $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ and minimize to get parameter vector $\theta^{(2)}$
- $d = 3$ where $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$ and minimize to get parameter vector $\theta^{(3)}$
- $\vdots$
- $d = 10$ where $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$ and minimize to get parameter vector $\theta^{(10)}$

Here, d is the degree of polynomial.

Now, take the parameter vectors obtained above to get the test set errors:

- $\theta^{(1)} \rightarrow J_{test}(\theta^{(1)})$
- $\theta^{(2)} \rightarrow J_{test}(\theta^{(2)})$
- $\vdots$
- $\theta^{(10)} \rightarrow J_{test}(\theta^{(10)})$

Now check which model has the lowest test set error. Suppose we choose $J_{test}(\theta^{(5)})$ which turns out to be the lowest. The problem is that it is likely to be an optimistic estimate of generalization error, i.e; our extra parameter (d = degree of polynomial) is fit to test set. Therefore our hypothesis is likely to do better on our test set, than on newer examples which it hasn't seen before.

Best Practice:

- Split the dataset into three sets
  - Training Set (60%)
  - Cross Validation Set (20%)
  - Test Set (20%)
- Find the errors
  - Training Error:

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

  - Cross Validation Error:

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

  - Test Error:

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

- Use Cross Validation set to select the model (explained above), get the cost validation errors and pick the lowest one.
- Estimate generalization error for test set using the above selected model.