Some practical advice for term project

Monday, February 12, 2018 10:10 PM

Suppose you have implemented a model (e.g. logistic regression) and find that the error on the test set is large.

Now, what should you do?

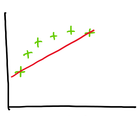
Options

1. try adding new features ($x^{(i)} \mapsto \phi(x^{(i)})$)
2. try changing regularization parameter λ
3. switch to SVMs with a Mercer kernel
4. get more data
5. others ?

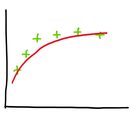
Before deciding on these options, you first need to determine if your model is overfitted or underfitted (i.e., whether you have a variance or a bias problem).

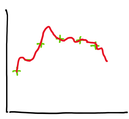
Overfitting : $J_{train}(\theta)$ is low ; $J_{test}(\theta)$ is high

Underfitting : $J_{train}(\theta)$ is high ; $J_{test}(\theta)$ is high

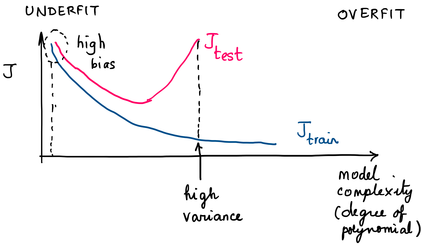


HIGH BIAS
UNDERFIT



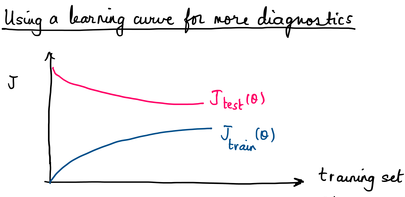


HIGH VARIANCE
OVERFIT

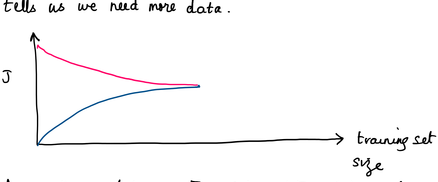


In logistic regression, as you increase λ, we zero out higher order coefficients and enter the high bias regime. Conversely, as λ decreases, we enter the high variance regime. Sweet spot for model complexity is the lowest point of the J_{test} curve. $J_{test}(\theta) \approx J_{train}(\theta)$ and both should be low.

Using a learning curve for more diagnostics



A large gap between $J_{train}(\theta)$ and $J_{test}(\theta)$ tells us we need more data.



A small gap between $J_{train}(\theta)$ and $J_{test}(\theta)$ and both are high, suggests need for more features.

Overfitting : get more examples, increase λ
Underfitting : get more features, decrease λ

A simple protocol

1. Start with a simple (perhaps, linear) model - build & test on your data set.
2. Plot learning curves and determine whether your model underfits or overfits. Then either get more examples and increase regularization (overfit), or get more features and decrease regularization (underfit)
3. Do careful error analysis - manually examine errors (false +, false -) and see if there are systematic trends in the misclassified examples.
4. Do A/B testing to check which option to pursue - i.e. use validation set to test whether an idea (e.g. add a specific feature) is good or not.
5. Increase model complexity gradually moving to more complex models as dictated by your error analysis and A/B testing.

Also look at Alex Hayes' tips on doing ML projects posted on Piazza.