Machine Learning and Data Science: Logistic Regression Implementation

Posted: on August 5, 2017 by Dr Dokchitpong

Share

Logistic Regression: Implementation and Evaluating a Trained Model

In a nice two-tier jupiter notebook on Logistic Regression: Logistic Regression Theory and Logistic and Linear Regression Regularization Theory pdf included general discussion of logistic regression and derivation of the equations with the purpose of giving insight as how it works. How this time to relate it work?

In this post I'll do a simple implementation of the logistic regression equations with Python. This will just be an "Outdoor" style implementation in this Jupyter notebook. Most statistics packages will include logistic regression functions. The Python modules scikit-learn and statsmodels both have implementations. The purpose of this blog post is to help myself relate how some of the functionality of these "black-box" tools works.

Before I read up the Python modules and some examples I want to discuss one more bit of "theory", the very important problem of deciding if your model is any good or not

This posts along with all of the others in this series were converted to html from Jupyter notebooks. The notebooks are available at https://github.com/dokchitpong/jupyter-notebooks

Evaluating the quality of the trained Logistic model

This is the last important concept that needs to be discussed before we get on to some code and examples.

In Linear Regression, we had used the value of the cost function and the statistic R^2 to evaluate the goodness of fit. We also used a hold-out CV to see how well the model function generalized and thus, fit predictive value. In Logistic Regression there isn't a direct substitute for R^2 but there are several statistics tests that are used. You will see discussion of Deviance (and variations on that) and Pearson's Chi-Square statistic. Deviance is a measure of how well the model fits the data. Pearson's Chi-Square test (one of the many things I want to get around to the importance of evaluating the predictive value of the model. The best way to find out is to see how well it works on a set of data that was not used during training (or validation) of the model.

Some simple measures to evaluate the quality of a model:

- **Cost Function J at the optimized value of the parameters** $\underline{\theta}$ This is basically the "log-likelihood" that is used in many of the statistical measures. It is therefore that is smaller value of the Cost Function indicates a better fit to the training data. The Cost Function is what is being minimized during the training optimization. Has your optimization converged? You do need to be careful about over-fitting! Also, note that the hypothesis form can be only needed during the optimization. For evaluating the fit it's probably better to leave that out.
- **Hold-out CV** Hold out a set of data from the training data. This can be computed as a percentage of correctly predicted values using the optimized model on the training data, when evaluation. Training accuracy is important to understand the data set that is being used. **It is good to look at Training Accuracy for the positive cases and the negative cases separately as well as the overall accuracy.**

> ...look carefully at the data! Says one New YORK data scientist but an only 10 of those data points are positive cases. That means that training a model that always predicts negative will have a 90% Training Accuracy! Garbage!! For a case like that it might be better to look at Accuracy Detection ... we'll get to that in a latter blog post.

- **Test/cross validation** Now well does the model work on a set of data samples that was NOT used during training.

It's the Test/cross validation that actually means something! Having a perfect fit to the data and a zero cost function is useless if your model has poor predictive value on a test-set.

The Cost function and Training accuracy are useful measures to look at when you are adjusting hyperparameters like λ and deciding when to stop training. Cross validation is also used for the model