CSCI 181 / E-181 Spring 2014 Practical 2

Kaggle Team "No Comment"

David Wihl
davidwihl@gmail.com

Zack Hendlin
zgh@mit.edu

February 26, 2014

Warm-Up

Baseline

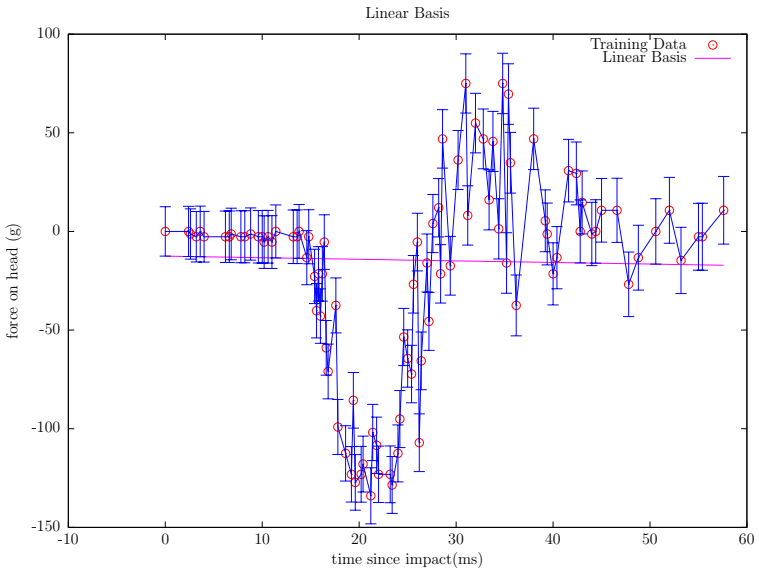

Figure 1: Warmup: Linear Basis

As a baseline, we first created a simple linear gradient descent with a flat slope and intercept.

We also used a polynomial basis, iterating with polynomials from n^2 up to n^{12} and selecting the lowest error. Unsurprisingly, n^{12} had the lowest error rate, but is likely highly overfit.

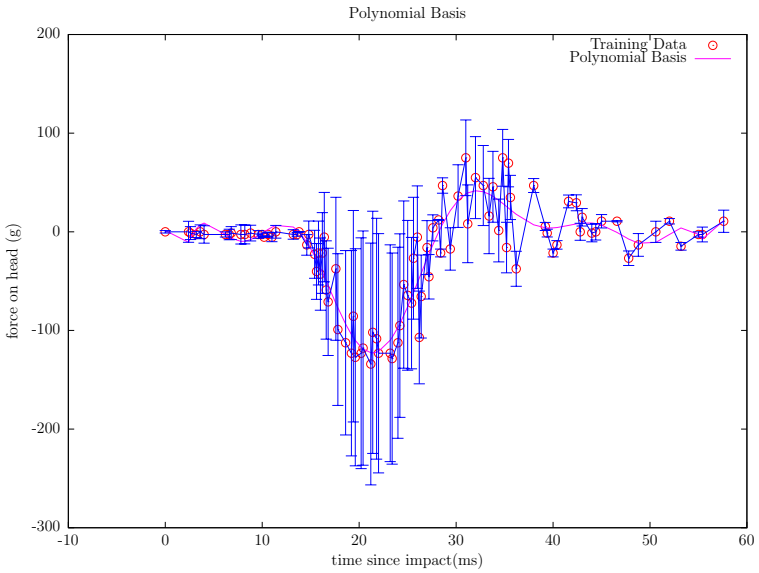

Figure 2: Warmup: Polynomial Basis n^{12}

Bayesian Linear Regression

Using Gaussian Likelihood and Prior, we solved for W using Moore Penrose.

$$W_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t \tag{1}$$

This was simple to implement, especially in Octave/Matlab. However, without normalization the error rate was close to the baseline linear basis and significantly worse than the polynomial.

Locally Weighted Linear Regression

Locally Weighted Linear Regression (LWLR)[1] provided the lowest cost overall and a smooth fit to the data without overfitting given the profile of this dataset. A variety of K values were attempted. 0.001 never converged. Values from 0.5, 1.0, 5.0 and 10.0 did converge with 1.0 seemingly providing the best balance between fit and smoothness.

LWLR is an expensive operation. Since the dataset here was small and did not match a typical straight line or polynomial pattern, it was appropriate to attempt LWLR.

[1] *Machine Learning in Action* by Peter Harrington. © 2012 ISBN 978-1617290183

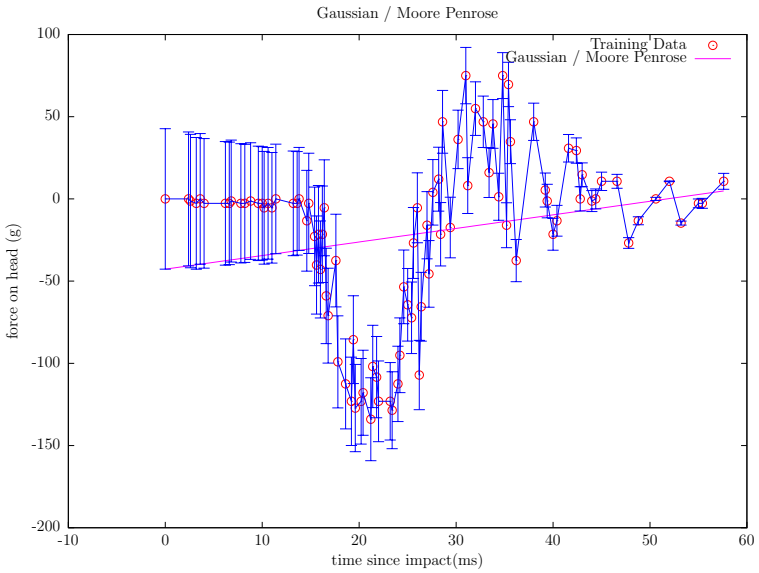

Figure 3: Warmup: Gaussian

Warmup Summary

Across all basis functions, overall error rate was calculated by sum-of-squares:

$$J = \frac{1}{2N}\sum_{i=1}^{N}(y_i - t_i)^2 \tag{2}$$

The following table summarizes our results. LWLR was reasonably simple to implement and provided the lowest cost. For this particular data set, it would be our basis function of choice.

Basis	Lowest Error
Linear Basis	1293.0
Gaussian Basis	1187.7
Polynomial Basis	211.9
LWLR Basis	185.6

Predicting Movie Opening Weekend Revenues

Preliminary Data Analysis

The training set consists of movie metadata and textual movie reviews. From the sample code provided in the problem, the initial set of features has a classic problem of too

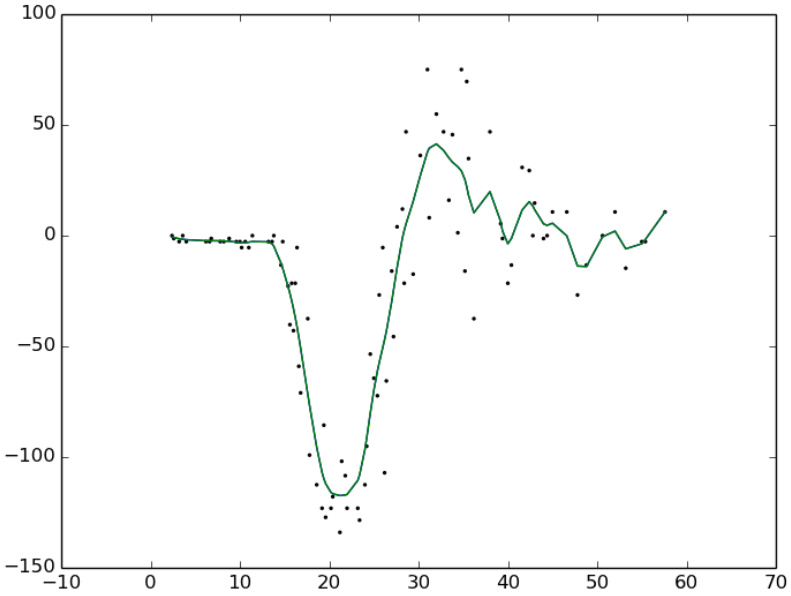

Figure 4: Warmup: Locally Weighted Linear Regression $K = 1$

many dimensions (105403) for too little data (1147). This is mostly due to the unigrams converting each word of each movie review and description into a different dimension. Eliminating the unigrams as a first step significantly improved classification results and reduced dimensionality to only(!) 11276 TODO update down just 2!

Using Cross-validation

To quickly evaluate the regression algorithms, we build a simple cross validation set, using 10% of the data and two folds. This enabled us to track J_{cv} vs J_{train}. By measuring the learning curve,[2] we could see our algorithms' progression.

TODO: expand to full CV

Subsection

NOTES follow:

Attempts made: lasso, ridge, normalizing data evaluating features threshold values polynomial values removing extra features, like words, dates examining w to see which

[2] *The Elements of Statistical Learning* by Hastie, et al. © 2009 ISBN 978-0-387848587

features were unusual or had significant weighting came down to two: number of screens, and production budget. We tried various polynomials of numbers of screens, with 4th power yielding minimum error production budgets ¿ $50,000,000 were deemed a "blockbuster" type as they had the largest errors, and therefore the largest penalties. Upon detailed analysis in Excel of the training data, we needed to apply a correction of 0.85 to this different category of movie. Given the large production budget, a studio can produce only a few blockbusters per year so by definition they are outliers compared to the majority of the movies. While our correction for large budget films decreased the overall error rate somewhat, it did not have as dramatic as effect as we'd hoped. Given most of the big budget films had lower openings relative to smaller budget films, we assume the studios were generally also disappointed.

Conclusion

Exercise in feature engineering - to quickly weed out which features are relevant and which are noise. More sophisticated algorithms like ridge and lasso are not necessarily better as they involve tweaking normalizing values.