Mini-project 1

Predicting the popularity of a reddit comment

Thomas Porta (260663215), Florence Min (260704265), Saifullah Elsayed (260733168)

January 31, 2019

Abstract

We set out to predict the popularity of a reddit comment using two methods of linear regression, comparing a closed-form approach to gradient descent, and explored the effects of editing and adding additional features to each model on its predictive performance. Our best performing model on the test set was a closed form linear regression model using the 60 most common words in the training data. This model exhibited a mean squared error of $MSE_{best} = 1.2850$ on the test set and, before adding two new features, exhibited a mean squared error of $MSE_{before} = 1.28769$. We found that in general the gradient descent method had non-ideal convergence and finding the correct learning decay became non-trivial when the feature space increased to high dimensions. Additionally, the closed-form method is preferable when there is non-co-linearity in the features and the data set is not too large. Finally, we used learning curves to hypothesize whether collecting more data for training would help make our model perform better, finding that that was not the case; the problem in our predictions was most likely because of non-predictive features, the model under-fitting the data, or a combination of both factors.

1 Introduction

Reddit is a popular social media platform used by many - users can post comments that can be upvoted or downvoted by others, adding or subtracting to their overall "karma." We aimed to determine the popularity score of said comments by training a linear regression model on a set of comments with known popularity scores. Two paths were used to build the model weights: the closed form solution, $\hat{w} = (X^T X)^{-1} X^T y$; and the gradient descent solution, which had a step for the weights of $w_i = w_{i-1} - 2\alpha(X^T X w_{i-1} - X^T y)$. In both cases X was the data matrix, including the bias terms of 1, y was the array of target variables, and α was the learning rate, chosen to be decaying.

While we used standard numeric features, such as the number of children the comment had, we also parsed each comment and ran basic natural language processing (NLP) on its text. We will explain how our dataset was treated before discussing the performance and comparison of the two methods of computing the linear regression model. We will end with a discussion that analyzes these results in more detail.

2 Dataset

The dataset was constructed from a dictionary of reddit comments where each entry was a different feature: is_root, a boolean value for whether the comment was the child of another comment (0 if false, 1 if true); children, the number of children comments the comment had; controversiality, the "controversial" score of the comment as calculated by Reddit; text, the text of the comment; and the popularity score. We separated the dataset of 12,000 instances into 10,000 instances for training, 1,000 for validation and model testing, and 1,000 for final testing of our model. Since the popularity score was our target variable, we separated this entry from the training, validation, and testing matrices into its own array.

Each instance's text was pre-processed by stripping its capitalization and splitting it into a list of words by white space. After, the top N most common words in the training set were found. The number of occurrences for each of these words were counted for each comment, creating N new features, one for each top occurring word, for every datapoint.

We later added two custom features to improve the performance of our model using sentiment analysis from the NLP library, TextBlob. TextBlob provides a subjectivity value as part of its sentiment analysis, representing it as a float in the range $[0, 1]$, with 0 being very objective and 1 being very subjective. This value was used on its own as a feature, while a second feature was engineered with the equation: $f_{new} = subjectivity * isroot * children$.

In the end, each instance consisted of $6 + N$ features: the bias term, is_root, children, controversiality, two custom features, and N word count features.

3 Results

We compared the runtime, stability, and performance of the closed-form linear regression and the gradient descent approaches using different N most common words. Various learning rates were also tested to check for the best convergence in gradient descent. Table 1 shows the average runtime for both approaches, the average Mean Squared Error (MSE over 3 separate runs) of both approaches, and the variances for the gradient descent. In Figure 1, we plotted the MSE of the validation and training sets for each approach as a function of word count N. The $N = 60$ closed-form model returned the lowest MSE and is later referred to as our best model.

	Word Count	Avg. Runtime (s)	Avg. MSE	Variance (Runtime)	Variance (MSE)
Closed-Form	0	0	1.020327	-	-
	60	0.0024	0.983940	-	-
	160	0.13	0.995069	-	-
Gradient Descent	0	5.22	1.0293	7.49	0.000056
	60	16.39	1.0978	1.45	0.0006
	160	82.6	1.3634	11.55	0.0022

Table 1: Table of the runtime, MSE on the validation set, and variance of the different approaches and word counts.

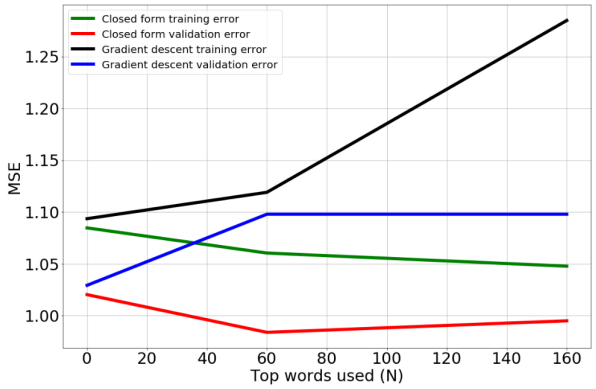

Figure 1: Mean Squared Error of the closed-form approach compared to the gradient descent using the validation and training sets for various word counts.

Figure 2: Learning curve for the closed form linear regression model.

On the validation set, our best model excluding the two custom features returned $MSE = 0.9839$, while our best model including the two features returned $MSE = 0.9784$; this was an improvement of $\Delta = 0.0055$. We then constructed learning curves for the training and validation errors of our best model, shown in Figure 2, where behaviour was tested while incrementing the number of instances up to 10,000. Finally, we ran our best model with the two custom features on the final test set which returned $MSE_{best} = 1.2850$.

4 Discussion and Conclusion

The closed-form approach was overall better than the gradient descent in our experiments. As closed-formed linear regression usually provides a more accurate model than gradient descent, it follows the MSE of the closed-form models were lower than the gradient descent models'. The scaling of closed-form's computational cost makes it impractical with larger datasets; however, our dataset was small enough no slowdown was observed and the runtime of the closed-form models were also shorter than the gradient descent models'. Since there was no significant difference in MSE between the training and validation sets we can likely conclude that our models were neither overfitting nor underfitting. However, it is statistically anomalous to have a lower validation error for almost all of our tests. It is possible this is from random noise and that using cross validation would fix this problem.

Adding more features to the model obviously increased the runtime. The MSE, meanwhile, shows more interesting behaviour. Figure 1 shows that convergence to the true weights becomes tricky with a high number of features. The $N = 160$ gradient descent model exhibited a substantial Euclidean distance from the exact weights found from the closed-form solution, indicating the error function most likely has multiple local minima and additional extraneous behaviour.

Our best model was determined to be the $N = 60$ closed-formed linear regression at $MSE_{60} = 0.9839$ on the validation set. The two additional features also significantly improved our model, lowering the MSE by $\Delta = 0.0055$ to $MSE = 0.9784$. The subjectivity of a Reddit comment likely impacts its popularity by influencing responses to the comment, i.e. whether someone upvotes or downvotes it. This impact would also depend on the comment's visibility; even if a comment would be well-received, if it was buried as a child or had few children itself to boost its visibility then it would not receive as much attention as it might have had otherwise. This explains how our second feature also improved our MSE. In the end, this model with the two added features was our best performing model and returned an MSE of $MSE_{best} = 1.2850$ on the test set.

The learning curves show using a small training set almost negates the statistically anomalous behaviour mentioned earlier; at $x = 2000$ the training error is lower than the validation error. Also, as the data set size increases, the validation error begins to stagnate as adding data does not reduce the error. This may indicate our model or its features are fundamentally weak predictors of popularity. It would be interesting to test additional models and verify whether more accurate predictions are possible with other types of features.

5 Statement of Contributions

Florence Min wrote the code to pre-process the data and added the two custom features, Thomas Porta wrote the code for the closed-form solution and gradient descent, and Saifullah Elsayed ran the tests with the help of Thomas Porta for the completed code and models. All three worked on and edited the report together.