Predicting the Popularity of Reddit Comments with Linear Regression

McGill, COMP 551, Mini-project 1

Yu-Cheng Cho (260722992), Navpreet Singh (260764696) Mila Urosevic (260723644)

1. Abstract

The purpose of this project is to predict the popularity of Reddit comments using linear regression models. We compared both closed-form and gradient descent implementations and tested various combinations of input features. We found that the gradient descent approach was slower than the closed-form approach and that its performance depended upon the initialization of hyperparameters. The basic model was created from considering the following features: 'children', 'controversiality', 'is_root', and the frequencies of the most frequent words. In addition, three additional data exploration attempts were made to improve the model's performance: (1) dropping the punctuations when calculating the frequencies of the top 160 most frequent words, (2) implementing feature selecting algorithm on the top 160 words to find the optimal features, and (3) constructing new features utilizing the basis expansion and transformation of 'children'. Ultimately, our best model obtained a Mean Squared Error (MSE) of 1.2717 on the test data.

2. Introduction

Reddit is a popular discussion website with millions of visitors, where posts and comments containing images, links and texts can be submitted to the website. In this project, we attempted to predict the popularity comments on Reddit by implementing the linear regression method. Although there have been similar attempts in the past, they mostly focused on predicting the popularity of Reddit posts or threads, whereas comment popularity may depend on different variables. Moreover, typically post popularity prediction is done using classification or other methods, whereas we would like to explore the success of a linear regression method.[1,2] The input dataset includes a broad range of numerical and binary features as well as the comment's raw text. The targeted popularity scores have numerical values in the range of -6.85094 to 8.37335. The prediction was studied first by considering the three non-text features: 'children', 'controversiality' and 'is_root', and a Mean-Square-Error (MSE) of 1.0203 in the validation set was calculated. By further utilizing the text, dropping the punctuations and implementing a feature selecting algorithm, the MSE in the validation set can be reduced to 0.9734. Furthermore, our studies on the datasets also show that among the given features 'children' is strongly correlated to the 'popularity score'. By adding the basis expansion and transformation of 'children' into the features, we obtained an MSE of 0.9522.

3. Dataset

The data set consists of 12,000 comments on a Reddit post was split into a training set (10000 data points), a validation set (1000 data points) and a test set (1000 data points). Each data point can be described by the following non-text features: popularity ('popularity_score'), number of replies ('children'), 'controversiality' and whether or not this comment is the root comment of the discussion thread ('is_root'). The correlation matrix in Figure 1 demonstrates the relationships between the different input features and the popularity score output. It can be seen that the popularity score is most correlated with the feature 'children'.

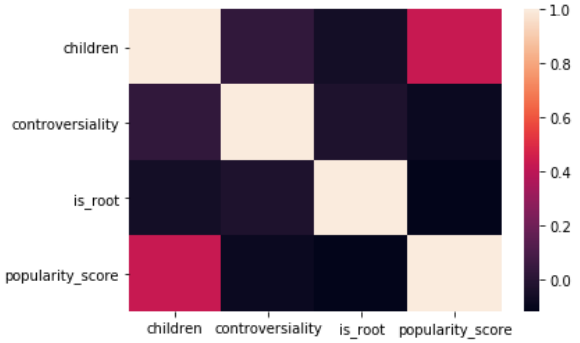

Figure 1. *Matrix showing the correlation between the predicted variable (popularity_score) and the three non-text features.*

In addition to the non-text features, the text content of the comments was also processed. The text was first lower-cased, and whitespace was used to define different words. Also, the number of times that a word

from the top 160 (or top 60) most frequent words occurred in each comment was computed. This added a total of 160 new features to the dataset. To improve the model's performance, processed the text by removing the punctuation and then re-calculating the top most frequent words. Furthermore, we tried employing simple forward or backward stepwise feature selection algorithms by either starting with one feature and only adding features that lower the MSE or by starting with all features and removing those that increase the MSE.

In addition, we created new features by using a basis expansion (square term) and the log transformation of the existing feature 'children', because it has a strong correlation with the popularity score.

Although the dataset and prediction that we are handling right now is rather simple, we should always be aware of the power of having the ability to access and analyze a huge amount of data. For example, predicting the popularity of comments can translate to an ability to spread biased information towards certain group of people on the internet, or even manipulate the results of an election in a democratic country. Additionally, taking information from public social media sites and storing it indefinitely in a model without the explicit permission of the people who posted the content is not completely ethical.

4. Results

4.1 Comparison of the Closed-Form and Gradient Descent Methods

For the purpose of comparing stability, runtime, and performance, we ran both the closed-form and the gradient descent models on only the three non-text features.

4.1.1 Stability

The gradient descent model depends on multiple parameters, requiring the user to set them when running the model. We tried varying the hyperparameter initializations and observed the effects on performance. The first parameter sets the initial weights of the features and while varying it did not affect the final result, it did affect the number of iterations required for the gradient descent. In fact, the relationship between the number of iterations and the difference between the initial and final weights is non-linear, as can be seen in Figure 2.

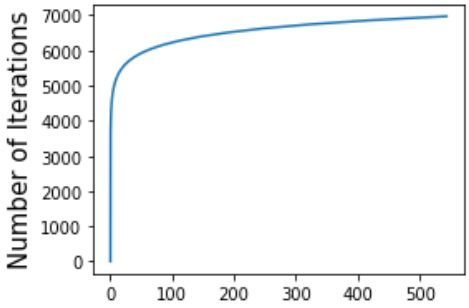

Figure 2. *Relationship between the number of iterations of the gradient descent loop and the discrepancy between the initial and final (optimal) weights. The initial weights were set as a scalar multiple of the known optimal weights.*

The next two parameters η_0 and β_0 determine the learning rate according to the following formula $\alpha = \frac{\eta_0}{1+\beta_i}$. β_i is updated after each iteration according to $\beta_i = 10 * \beta_{i-1}$ so that the learning rate will decrease with each iteration. We varied both η_0 and β_0, but the weights only changed by a maximum of 0.012%. The MSE did not change, only the number of iterations was affected. We also tried a model with a fixed learning rate, but here any rate over 0.0001 gave an overflow error.

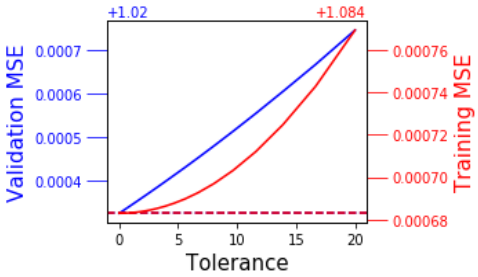

Figure 3. *The relationship between tolerance and MSE. The dotted lines show the closed form MSE values.*

The final parameter ε indicates the minimum gradient of the loss function that must be obtained before the gradient descent is stopped, also known as the tolerance [1]. A sufficiently low tolerance will give the same weights as the closed form solution (to the sixth decimal) but a tolerance that is too high will result in higher MSEs for training and validation. On the other hand, if the tolerance is too low then many iterations are required and the runtime is prolonged. Hence we calculated the optimal tolerance, obtaining $\varepsilon = 0.0163$. This corresponds to the highest tolerance that gives the same validation MSE as the closed form solution

(to the sixth decimal). The relationship between the tolerance and both training and validation MSEs are shown in Figure 3.

We further explored the effect of changing the number of data points in the training set. We did so by taking a subset of the training set but keeping the validation set the same. If we split the training set into two and trained a model on both halves, then both models had a validation MSE that was ~0.05% greater than the MSE of the full training set, and their respective validation MSEs differed by 0.4%. Hence, the MSE depends on both the number of data points in the training set and on the precise contents of the training set. The relationship between the number of data points and the validation and training MSEs can be seen in Figure 4. This result is true for both closed form and gradient descent approaches.

4.1.2 Runtime

We found that the closed form had a runtime of $1.67ms \pm 77.5\mu s$ whereas the gradient descent had a runtime of $2.1s \pm 33.7ms$ when optimal parameters were used. This corresponds to a 200% difference in runtimes, indicating that the closed form method is much more efficient.

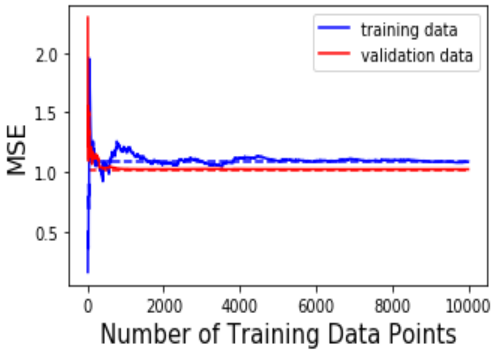

Figure 4. *The MSE values obtained for a model trained with a subset of the 10,000 initial data points.*

4.1.3 Performance

Both models are capable of obtaining a minimum training MSE of 1.084683 and a validation MSE of 1.020327. However, while the closed form approach will always yield this result, the gradient descent approach will only do so for certain hyperparameter initializations.

4.2 Finding the Optimal Feature Combination with the Closed Form Model

For the given task of comparing the closed form by including top 60 words and top 160 words, we observed that including top 60 words give better results. See Table 1. We consistently obtained a lower validation MSE than training MSE, however if we partition the validation set into 5 groups, their individual MSEs (for the 164 feature implementation) have a wide range from 0.6923 to 1.1650.

Table 1: Comparison of closed-form approach with non text features only, with non-text features plus the occurrence of top 60 most frequent words and with non text features plus the occurrence of top 160 most frequent words.

Description	MSE on training set	MSE on validation set
No Text	1.0846	1.0203
Using top 60 word	1.0602	0.9846
Using 160 words	1.0469	0.9923

For further feature selection, we tried to remove the punctuation to refine the text. On implementation, we found out that removing punctuation doesn't help. On the contrary, it worsens the MSE in the case of top 60 words (though we observed marginal improvement in the case of the top 160 words).

Our next step was to select the optimum number of top words to lower the MSE. The intuition behind this implementation was that using top 60 words gives better results as compared to the top 160. We carried out a forward feature selection on the number of top words to be used as features to train the model. We were able to get the best results when the number of top words used were 104.

To add an extra feature, we tried to use the basis expansion of one of the features. We tried the quadratic form of the feature 'children' and observed an improvement of 0.02 in the absolute value of MSE on the validation set. For another extra feature, we tried the transformation $\log(children + 10^{-7})$ as one of the

features. On training the model with this extra feature, we got an improvement of 0.003 in the absolute value of MSE on the validation set. Based on the numbers reported, the model that performs the best on the validation set includes the following features: '**is_root**', '**children**', '**controversiality**', '**top 140 words**', '$\mathbf{(children)^2}$' $\mathbf{log}(\boldsymbol{children} + \mathbf{10^{-7}})$'. The **lowest MSE** on the validation set is **0.95224.**

Table 2: The effect of further text processing and of additional special features on the MSE. These results were obtained using the closed form method.

Description	Total number of features	MSE Training Set	MSE Validation Set
[Closed Form] Three Non-Text features	3	1.0847	1.0203
[Gradient Descent] Three Non-Text features	3	1.0847	1.0203
[Closed Form] 3 features + top 60 words	63	1.0603	0.9847
[Closed Form] 3 features + top 160 words	163	1.0470	0.9924
[Closed Form] 3 features + top 60 words (DP)	63	1.0608	0.9872
[Closed Form] 3 features + top 160 words (DP)	163	1.0462	0.9863
[Closed Form] 3 features + top 104 words (DP) (FS)	107	1.0541	0.9734
[Closed Form] 3 features + top 104 words (DP) (FS) + basis expansion of 'children' feature	108	1.0088	0.9566
[Closed Form] 3 features + top 104 words (DP) (FS) + basis expansion and transformation of 'children' feature	109	0.9998	0.9522

**** Bias term not included in the number of features**

(DP): dropping punctuations, (FS): forward feature selection (to select the optimum number of top words)

6. Discussion and Conclusion

Overall, after testing different types of linear regression implementations as well as various combinations of features, we found that the closed form implementation was faster and more stable than the gradient descent implementation. We also observed that adding more features didn't always decrease the MSE, instead the optimal MSE corresponded to a certain set of features that requires careful selection from the most frequent features.

That being said, our best model consisted of not only the three non-text feature but also (1) dropping the punctuations when calculating the frequencies of the top 160 most frequent words, (2) implementing feature selecting algorithm on the top 160 words to find an optimum number of words, and (3) creating features using the basis expansion of 'children'. This helped us to lower the MSE on the validation set from 1.0203 to 0.9522. The best-implemented model gave an MSE of 1.2727 on the test data. The performance of our model indicates that it is indeed possible to predict the popularity of Reddit comments, to some extent. This could have potential applications to the way that the Reddit site is structured, for example by pushing to the top not only comments with the most votes but also new comments that have a higher chance of being popular. However, this model can be further improved; in the future, we would like to explore whether extracting other features such as the time that the comment was posted, the sub-Reddit on which the comment was posted and the user's number of Karma can make our model better. Furthermore, we believe that we can advance our implementation of feature selection by using mixed stepwise selection instead of just forward or just backward selection.[3]

7. Statement of Contribution

Yu-Cheng Cho: Code – Closed form and gradient descent. Report – Abstract and Introduction.
Navpreet Singh: Code – Closed Form. Report – Dataset and Results
Mila Urosevic: Code – Gradient Descent. Report – Abstract, Results, and Conclusion

8 References

[1] Rohlin, Tracy, "Popularity Prediction of Reddit Texts" (2016). Master's Theses. 4704. http://scholarworks.sjsu.edu/etd_theses/4704

[2] He, J., Ostendor, M., He, X., Chen, J., Gao, J., Li, L., & Deng, L. (2016). Deep Reinforcement Learning with a Combinatorial Action Space for Predicting Popular Reddit Threads. *Empirical Methods in Natural Language Processing*.

[3] Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3*.