Comp 551 Project 1 Group 65: Matthew Etchells, Brendon Keirle, Jonathan Pearce

Abstract

Analysis and prediction of trends in social media is an increasingly important task to individuals and businesses. Identifying factors that contribute to popularity of comments on Reddit.com may be valuable to those hoping to improve their outreach on the site.

We trained and tested linear regression models using a dataset of 12000 reddit comments, each assigned a popularity score. We found that a closed form approach to estimating parameters gave the lowest mean squared error (MSE) and had the quickest execution despite greater asymptotic complexity than gradient descent approaches. Features included contraversiality (0 or 1), whether the comment is from the thread root (0 or 1), the number of child comments (positive integer) and a set of word features. The n most common words throughout the dataset were computed, and the counts of each of these present in a given sample composed its set of word features. We varied the number of word features n throughout the range (1, 160) and found that the model performed best at n = 61. We also found that the model improved when using (no. children)^2 or (no. children)$*$(length of comment), but not when using both. In our final model we included the 3 basic features, 61 word features and the product of (no. children)$*$(length of comment) to achieve our best result of 1.2980 MSE on the test set.

Introduction

Popularity on social media has become increasingly important in culture and in advertising. As a result, the task of analyzing social media content to discover trends and predict users' reactions has become increasingly relevant. Reddit.com is a popular community forum that provides a clear metric for popularity of posts and comments by taking the net of "upvotes" and "downvotes". Reddit post and comment popularity has been previously studied with machine learning techniques. It has been viewed as a binary classification problem in which posts are either popular or unpopular, estimated using Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA).[1] Also, research into the variation of post and comment popularity across different reddit communities (subreddits) has been performed using Neural Networks. It is suggested that features that predict popularity in one community may not well generalize to another.[2] In this research project we look specifically at the "AskReddit" community and aim to predict popularity scores of comments using linear regression.

A dataset was prepared containing 12000 reddit comments. Each of these comments had an associated popularity score. This research project aims to train a linear regression model to predict the popularity score of new reddit comments based off of the text content of the comment, a binary variable indicating controversiality and the comments parents and children (ie. comments in a reddit thread form a tree structure).

The training set, validation set and test set were split into 10000, 1000 and 1000 data points respectively. Linear regression models were trained and parameters estimated using the following approaches: (1) closed form solution, (2) stochastic gradient descent (SGD), (3) SGD with momentum and (4) adaptive moment estimation (Adam). The trained models were evaluated on their mean-squared error (MSE) when predicting on both the training and validation sets.

The closed form approach resulted in the lowest MSE. Among gradient descent approaches, Adam was both the fastest to train and had the lowest MSE. We chose to continue with the closed form approach, then looked at ways to enhance our model by altering features. The number of each of the most common n words (throughout the entire dataset) contained in a post was used to represent its text content. We varied n in the range of (1, 160) and found that using the 61 most common words provided the model with lowest validation MSE. We also found that the following additional features decreased

the MSE of the model: (1) (number of children)^2, (2) (number of children)$*$(length of comment), and (3) both 1 and 2. Among these, (number of children)^2 seemed to provide the greatest improvement without overfitting.

Dataset

The dataset used in this project contained 12,000 data points, each representing a comment from reddit. Each data point contained the original text from the comment, as well as 3 basic features: a binary variable to indicate whether the comment was the root of a comment chain, a binary variable to indicate whether the comment was controversial, and an integer representing the number of replies a comment received. The final feature in each data point was the response feature, which was the popularity score of the comment itself. The data was partitioned into 3 sets, a training, validation and test set. The training set contained 10,000 data points ($\sim$ 83% of the total data), while the validation and test sets were each provided with 1,000 of the remaining data points ($\sim$ 8% of the total data in each). The raw text from the comments was used to create new text features. This involved calculating the n most frequently used words in the training set comments (n varied from 1 to 160 in our experiments). Using these most frequently occurring words we went through each raw text comment and counted how many times each frequently occurring word appeared in a comment. The last step was to construct a n-dimensional vector containing these counts for each data point. We also constructed two new original features. Our first custom feature was simply the children feature (number of replies) squared. Our second custom feature was an interaction term between the length of a comment (computed ourselves using the raw text data) and the children feature, specifically the product of these two features.

This dataset is a good example of an appropriate social media dataset because the usernames have been removed and therefore the comments are anonymous. A large concern with social media datasets is privacy. Many people have usernames that are their exact name or are something very close to their name. This means that if a social media dataset is not anonymous it is possible for people to link comments to specific people, this could have significant negative consequences for the person making the comments. To ensure people whose online activity contributes to social media datasets are protected via anonymity, it is vital for usernames to not be included with the dataset.

Results

We compared the run times of training models using the following 4 approaches: (1) closed form solution, (2) stochastic gradient descent (SGD), (3) SGD with momentum, and (4) Adaptive Moment Estimation (AdaM). We ran 100 iterations of training on each model. The results are presented in Figure 1. The closed form solution was trained with the fastest mean run time of 15.0467ms, and SGD the slowest at 46.29ms. The asymptotic complexity of gradient descent is $O(n^2m)$ which is faster than the closed form solution's complexity of $O(m^3 + n^2m)$. However, given that $m \leq 164$ is small in comparison to $n = 10000$ in the training set, the closed form solution is able to perform faster than each of the gradient descent approaches. It is expected that gradient descent approaches would tend to execute faster than the closed form approach when m is larger relative to n. Further, the closed form solution is a stable algorithm, whereas the stability of gradient descent depends heavily on the hyperparameters (learning rate and decay rate) as well as the feature weight intializations. Because of these reasons, we decided to complete our model development and evaluation using the closed form solution.

To set a performance baseline we trained a model with only the 3 basic features from the dataset. The baseline model had a training set mean squared error of 1.0846 and validation set mean squared error of 1.0203. In order to improve the basic model we tried to add text features from the dataset to the original 3 basic features. We started with the 60 most common words as our text features, this model produced a training set mean squared error of 1.0604 and a validation set mean squared error

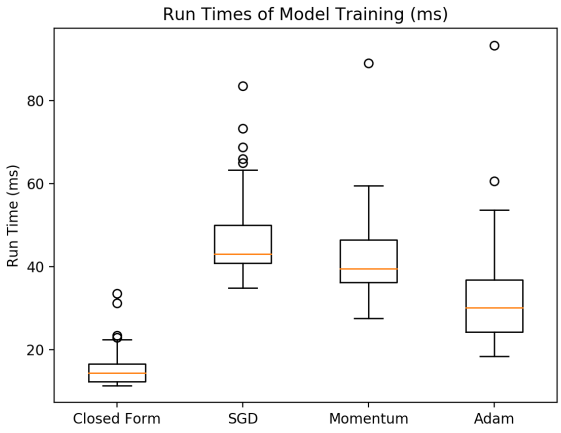

Figure 1

of 0.9839. Next, we increased the number of most common word text features to 160, this model had a training set mean squared error of 1.0478 and a validation set mean squared error error of 0.9951. As expected the more complex model had a lower training set error however the validation error had increased, this suggested that the model with 160 text features was overfitting the training set. We decided to investigate in detail, and find out how many text features the model could include before overfitting. Figure 2a. shows that as extra text features are added the training error decreases as expected, and that after 61 text features the model begins to overfit the data since the validation error begins to increase. Therefore we concluded that the model with the 3 basic features and 61 text features of the most common words produced the best model available, with a training set error of 1.0592 and a validation set error of 0.9698.

Figure 2a.

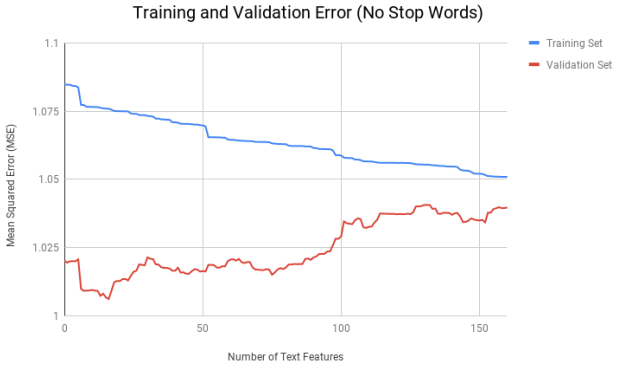

Figure 2b.

We were also curious to see how the removal of stop words would affect the model's performance. In natural language processing it is common practice to remove stop words from text data as they are very common and usually add little information. After removing stop words we re-calculated the text features and examined the model performance. Similarly as before, in Figure 2b. the training set error decreases as more text features are added to the model. In this case the model begins to overfit data with fewer text features, specifically after 16 text features are added to the model it begins to overfit the training set and the validation error increases. This optimal model with 16 text features had a training set mean squared error of 1.0759 and a validation set mean squared error of 1.0063. We

concluded that the model with no stop words was worse than the model with stop words included, this is most likely because of the greater sparsity of non stop words in the reddit comments. For regression it appears stop words are beneficial to model performance.

We continued our model evaluation with the model containing the 3 basic features and the 61 text features (including stop word features). Our last idea to improve model performance was to incorporate our own custom features. These features included the children feature squared as well as an interaction term that was the product of the length of the comment and the number of replies to that comment (children feature). First we added these new features individually to the model in order to ensure each feature provided information that improved performance. After this we added both to the model and evaluated. The results of these experiments are in the table below.

Custom Feature(s) Added to Model	Training Set MSE	Validation Set MSE	Test Set MSE
No Custom Features	1.0592	0.9698	-
Feature 1 (children^2)	1.0126	0.9495	-
Feature 2 (children$*$length)	1.0577	0.9459	1.2980
Feature 1 and 2	1.0126	0.9507	-

In all 3 tests the validation set mean squared error was improved by more than 0.005 over the model with just the 3 basic features and the 61 text features. However, when both features are added to the model the validation error increases slightly compared to the validation error with only one of the new features, this suggests that including both new features caused the model to overfit on the training data. We concluded that the model with the 3 basic features, 61 text features and the interaction term between children and length was the best model we developed. We concluded our model development by evaluating our best model on the test set. This model had a mean squared error of 1.2980 on the test set.

Discussion and Conclusion

Building our final model has been a great exercise in machine learning model development. We worked on iteratively improving model performance with the addition of new features and were able to see how overly complex models can overfit the training data. We attempted to create a better model by excluding stop words from the raw text data, however the extreme sparsity of non stops words compared to stop words actually decreased the model performance and we kept stop words included for the remainder of our experiments. Initially we demonstrated that for a linear regression problem with this size of feature set utilizing the closed form solution was faster that gradient descent in execution time and as expected was more stable and produced better results. Building on our discussion in class about the complexity of the closed form solution we do understand that for problems with much larger feature sets gradient descent becomes very helpful. We expanded upon the lecture material and demonstrated that there are more efficient gradient descent methods than the standard algorithm discussed in class.

There are a few ways to continue this model development. More complex feature design could help improve model performance, an example would be running sentiment analysis on the comments and using the sentiment score as a new feature in the linear model. Another way would be to enrich the dataset with more information. One idea would be to include the time between the original post until the time of the comment, from our experience on reddit more popular comments are generally left soon after a post has been made, a comment that is published days later will hardly garner any attention. Therefore time from thread creation to comment creation could be a very powerful indicator of comment popularity.

Statement of Contributions

Matthew completed the preprocessing code, executed task 3.1, wrote the abstract and introduction and the first section of results. Brendon developed the gradient descent methods and the closed form solution, he also tuned the learning rate parameters for the descent methods. Jonathan executed task 3.2-3.4, wrote the dataset section, remainder of the results section and the discussion and conclusion.

References

[1] Rohlin, T. M. et al. **Popularity Prediction of Reddit Texts**. *San Jos State University.* 2016 May. https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8251context=etd_theses [Online; accessed January 31 2019]

[2] Ting, J. **A Look Into the World of Reddit with Neural Networks**. *Institute of Computational and Mathematical Engineering: Stanford University.* https://cs224d.stanford.edu/reports/TingJason.pdf [Online; accessed January 31 2019]