Machine Learning 101

Group 11:

Bide Xu (260711367)

Mengjia Gu (260790140)

Robin Luo (260851506)

Course Number: COMP 551

Course Name: Applied Machine Learning

February 4, 2019

Abstract

This project was purposed to assess the performance of linear regression models for predicting comment popularity on Reddit. Given the pre-crawled dataset, we applied two approaches of linear regression models, closed-form solution and gradient descent, to prediction models with and without text features. Although we modified value of hyperparameters in gradient descent and tried to tune performance, the closed-form solution actually runs faster and presents more accurate results. While considering the major downside of the gradient descent is its lacking of accuracy, we promote an improved version of gradient descent algorithm with additional parameter "distance rate" that could achieve almost ideal accuracy as the closed form algorithm, and spends acceptable more runtime. Furthermore, new text feature based on 60 or 160 most high frequency words in the training data is implemented, with significant improvement for the prediction of popularity score. Other new implemented text feature such as "average words per sentence" indeed reduced MSE (mean square errors) of models. Another new numeric feature – the "cubic of children" is also implemented. Both of these new designed features perform well in reducing MSE w.r.t both training data and validation data, and thus they are selected in out final model, instead of selecting other attempted features.

Introduction

Discussion board is frequently used in academic communication, product review and other public message exchange. This variety of information, which serves as a data source of the predicting technique, contributes to commercial development in recommendation system design(Rohlin 2016). Reddit is a classic and popular discussion platform, connecting communities with various topics. This project is proposed to construct and compare linear regression models in purpose of predicting comment popularity on Reddit. We are given some prepared data of selected comments, which still require preprocessing before import into experiments. Three features (children, controversiality, is_root) and one output variable are directly from the original dataset. Other input variables, such as text features and word counts, were extracted or derived from text content of comments.

Given the pre-crawled dataset, we preprocessed data, extracted features, and built models for both closed-form solution and gradient descent. Not as the theoretical expectation, out experiment results illustrated that, compared to gradient descent method, closed-form solution is better in both runtime and accuracy. An improved version of gradient descent algorithm is implemented, by introducing a new parameter called "distance rate", which helps to keep the learning rate a reasonable distance from the epsilon value, so that the learning rate would not decay too quick. Therefore, this "distance rate" enables the algorithm to run a reasonable enough time in order to improve the accuracy of the result. This "distance rate" cooperates well with the Robbins Monroe logic, so could assist to achieve a well balance between run-time and accuracy. Compared models differently implemented with text features, we noticed that involving text content of a comment into models obviously improved the accuracy of prediction on popularity. By adding new input variables like "average number of words per sentence", we deduced the significance of comment content in prediction of popularity. By introducing new numeric feature based on expansion of original numeric feature such as "children" into "$children^3$", the prediction accuracy also increased obviously, which indicates that popularity of a comment does not merely have linear relation w.r.t the number of replies it receives.

Dataset

The original dataset consists of 12,000 data points in format of dictionary. Each dictionary contains five keys with different types of value. Data in each type of information forms one or more variables in linear regression models. We translated some of them in format better support linear regression models, as described below:

- popularity_score: numeric output variable, larger value means more popular comment
- children: non-negative integers indicate number of replies received
- controversiality: binary metric, 0 means not controversial and 1 means controversial comment
- is_root: binary, True represents a direct comment instead of being a reply, and contrary for False
- text: text content of a comment, which is convert to lower case and split by space, to generate new features

Besides the popularity_score representing how popular each comment is, other information are taken as input variables that may affect prediction results. Text features are drawn from a wordlist of most frequently used words in all given comments. This wordlist contains 160 words ranking by occurrences in text content of comments. We as well count "total number of words of comment", "average word per sentence in each text", "total number of sentence" and "average length per word" values in order to add them as new features for improving prediction.

After preparing features, we divided dataset into three parts for different purposes: first 10000 for data training, next 1000 as a validation set, and last 1000 to test model. Considered the limitation of data source, out project may lose accuracy lacking data of more features like post topics and timeline. Failures in measuring popularity score could also lead to missing objectiveness and reliability of our data source. Moreover, user preference is always difficult to detect, especially that Reddit targets on public audience with various backgrounds.

Results

All our experiment results are stored as .json file, which will be submitted with the source codes and other documents. The performance for different added features are summerized in Table 1, and discussed as following:

Closed-Form Solution v.s. Gradient Descent

Simply with non-text features directly extracted from dataset, closed-form solution without extra hyperparameter stands out for its stability and smaller mean square errors of both training and validation sets. Besides input variables, three hyperparameters affected results of gradient descent. Firstly, we studied how often should we change the learning rate: suppose the learning rate change every T iteration. Then in experiment, we set the value of ϵ to 10^{-6}, the value of η to 1. We found that MSE does not affected by the value of T, but running time is the lowest when T equals to 10^2, therefore, we set the value of T to 10^2. Secondly, we try to determine the value of ϵ, MSE does not change much after ϵ is less than 10^{-5}. However, the running time is increasing as the value of epsilon decreasing. Therefore, we set the value of ϵ to 10^{-5}. Finally, we determined the value of η, while eta increasing, MSE is decreasing. The maximum η is 1, therefore we set the value of η to 1. Lowering learning rate took longer time to reduce mean square errors, while decreasing run time ended experiment before reaching small mean square errors, as illustrated in Figure 1.

The new implemented hyperparameter "distance rate" was added to introduce some balance between speed and accuracy, although it could not outperform the ideal model with closed-form solution, it indeed provided a well balanced solution to solve the conflict between runtime and accuracy. Compared to the normal version of Gradient Descent method, it bring better accuracy, almost 10^{-10} in MSE regarding the ideal weight vector [[-0.22627679][-1.08584747][0.37536403][0.82092517]] obtained by the closed form method, as detailed in Figure 2.

Word Count Features

Implemented with the closed-form solution, models using text features from the high frequency words list took longer time to present lower MSE on both training and validation sets. Our experiment results show that the 60 words version could improve the MSE of validation data from 1.0203 to around 0.9693, while the improvement of 160 words version(s) are not as stable as the 60 words version(s), which have similar improvement in MSE for training set and validation set as the 60 words version(s), but do not perform well in the testing set.

New Features

The "average number of words per sentence" feature obiviously reduced the MSE of linear regression models in predicting comment popularity, from 1.0203 to around 1.0127 for the validation data. This text feature is selected due to its performance is better than some other text features we have attempted, such as “total number of words in the comment”, “total number of sentence” and “average length per word”. Although the “total number of words” feature does bring some degree of stable optimization, but its improvement is less than 0.005 in MSE, and therefore is not selected in our final model.

Also, we implemented word filters to remove punctuations, this feature does reduce MSE in some degree, but the improvement is not significant enough. However, we find that it is interesting when combine this feature together with the high frequency word count feature as well as other text features, for it could help to increase the stability of the prediction accuracy improvement of other text features.

The new numeric feature of “$children^3$” is also helpful to optimize the accuracy, the improvement is from 1.0203 to 1.0024 in MSE for the validation data. All these two new added features also work well for the training data.

Performance for Combined features

When the word frequency, the “average number of words per sentence” and the “$children^3$” features are integrated together, our experiment shows that their improvement can be combined well, which improve the MSE of the validation data from 1.0203 to around 0.9473 in total. This overall improvement is also verified by the testing data, whose MSE can be reduced from 1.2975 to 1.2631.

Discussion and Conclusion

This project indicated better results of linear regression models with closed-form solution than which with gradient descent, which against the theoretical suggestion of better performance by gradient descent. One possible explanation is that small size of dataset, requiring only few operations, cannot reflect the advantage of gradient descent in dealing with huge data and complex operations.

More and better filtered text features improve the prediction accuracy. Other features derived from text data, like comment length and word counts per sentence, better predict the comment popularity as well. Newly added features are also related to text content of comments. Therefore, text content significantly affects the popularity of a comment. Meanwhile, more text features take models longer time to run experiments.

More features in reality should be considered as important variables affecting how popular a comment on Reddit is. The idea of predicting comment popularity facilitates in technical development in commercial area, especially through improving recommendation systems.

Contribution Statement

All members brainstormed for data preparation, feature extraction, and model construction.

References

- Bide Xu: main python coding
- Robin Luo: models improvement and graph plotting
- Mengjia Gu: results test and summarize write-up

All participated in whole progress and learnt through this project.

References

Domingos, Pedro. 2012. “A few useful things to know about machine learning.” *Communications of the ACM* 55 (10): 78–87.

n.d. 2005. The Conversation Starts Here.

Rohlin, Tracy. 2016. “Popularity prediction of Reddit texts.” *Master Theses*, vol. 4704.

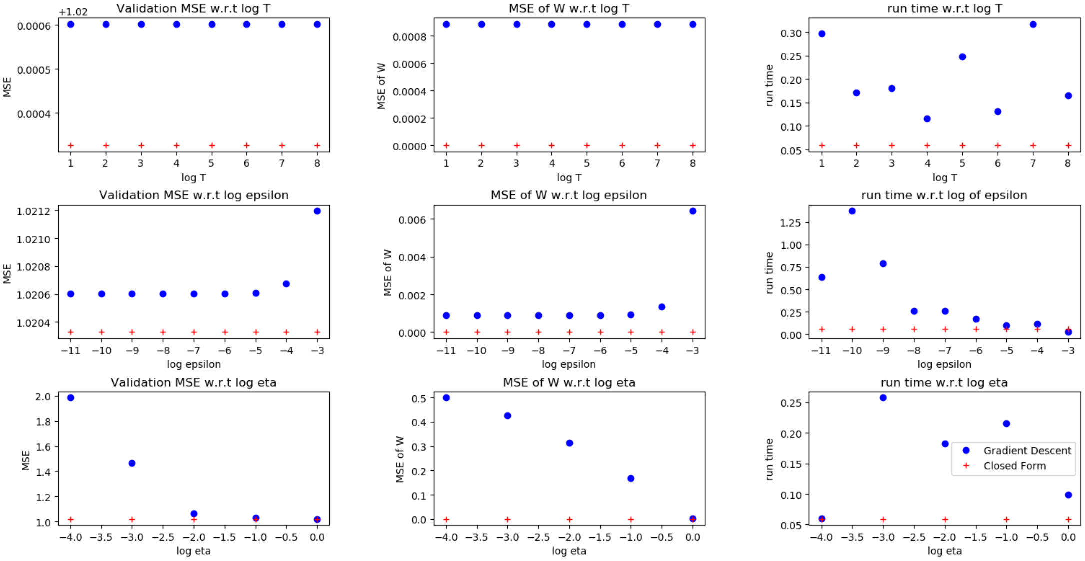

Figure 1: run time, accuracy performance comparsion between closed Form Alg and Gradient Descent Alg, the stability of Gradient Descent Alg varies w.r.t different initial settings of its hyperparameters, such as period (T) of Robbins Monroe, ϵ and η. While the performance of the Closed Form Alg is stable.

References

Feature(s) w.r.t Performance	**MSE of Training Data**	**MSE of Validation Data**	**MSE of Testing Data**	**run time (in S)**	**pre-run time (in S) time for computing text/numeric features and generate related matrix**
Basic Model (3 numeric features)	1.084683	1.020326	1.297531	0.027982	3.207755
Model With:					
60 high frequency words	1.059316	0.969286	1.298629	0.026985	88.564367
60 high frequency words + remove punctuation	1.061179	0.983503	1.285445	0.149432	79.076935
160 high frequency words	1.047630	0.996385	1.318918	0.891486	185.632877
160 high frequency words + remove punctuation	1.046506	0.983375	1.318479	0.099944	193.501532
Model With:					
average words per sentence	1.081087	1.012782	1.288209	0.015989	3.331969
average words per sentence + remove punctuation	1.081173	1.013062	1.288329	0.000998	3.996707
Model with:					
$Children^3$	1.041047	1.002414	1.267450	0.001998	3.309102
$Children^3$ **+ average words per sentence**	1.037979	0.995746	1.258209	0.002000	4.707468
$Children^3$ **+ average words per sentence + remove punctuation**	1.038087	0.995989	1.258282	0.001997	3.371063
All text features:					
60 high frequency words + average words per sentence	1.057966	0.966220	1.294168	0.219871	67.993910
60 high frequency words + average words per sentence + remove punctuation	1.059723	0.979785	1.280751	0.023983	62.781974
All Combined: text & numeric features					
$Children^3$ **+ 60 high frequency words + average words per sentence**	1.015920	0.947381	1.263172	0.020987	69.911887
$Children^3$ **+ 60 high frequency words + average words per sentence + remove punctuation**	1.017403	0.959957	1.253317	0.021993	65.182590

Table 1: Performance Evaluation For Varies Implemented Features and Their Combinations

References

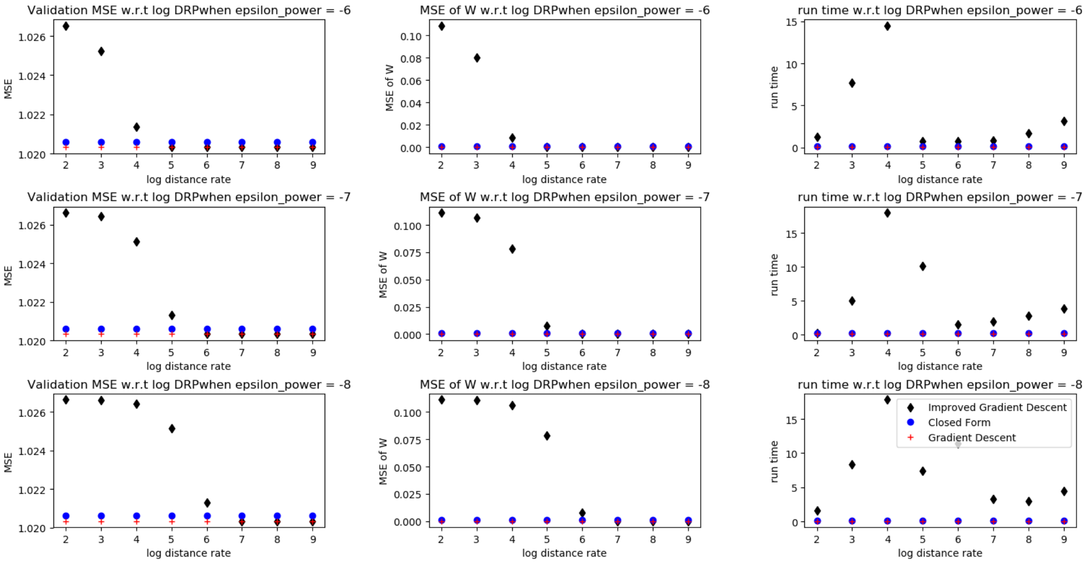

Figure 2: run time, accuracy performance comparsion between closed Form Alg, Gradient Descent Alg and Improved Gradient Descent Alg, with ϵ value around 10^{-6} to 10^{-8}. Those figures shows than, when the distance rate is more than 4, in other words, roughly 10^{4} larger then the ϵ value, the accuracy for the Improved Gradient Descent Alg starts to outperform the normal Gradient Descent Alg and become very near to the ideal results obtained by the closed form Alg. So the improved Gradient Descent Alg could be more accurate than the original Gradient Descent Alg, with acceptable more runtime.