Predicting Popularity of Reddit Comments Using Linear Regression

Gabriele Dragotto[a], Zhiyue Jiang[b], Jiaqi Liang[a]

[a]CERC Data Science for Real-Time Decision-Making, Polytechnique Montreal.
{gabriele.dragotto,jiaqi.liang}@polymtl.ca
[b]McGill University
zhiyue.jiang@mail.mcgill.ca

COMP551 (Winter 2019) - Project 1

Abstract

Reddit is a famous social aggregation and discussion website. For the scope of this work, we apply a multivariate linear regression to a set of comments in order to predict a popularity metric. The dataset contains 12000 observations, and each of them contains the full text among with 3 simple attributes. We first compute the estimated weights $\boldsymbol{w}^*$ both with a closed-form exact approach and a Gradient Descent (GDS). We compare the performances of the two approaches in terms of MSE and computing times considering the 3 simple features. According to the size of the training-set and features, the closed form approach performs better both in times and error rates. After extracting the most popular words in the training set, we test two models based on the absolute occurrences of the k most popular words ($k \in \{60, 160\}$). In particular, models $W60$ and $W160$ respectively introduce such occurrences as extra features. Model $W160$ overfits if solved in closed-form, while other models have lower MSE compare to the baseline (increase of +6.8% at most) for both closed-form and GDS solutions. Finally, we propose a new regression based on several new features (eg, $TF * IDF$ for the most 8 popular words, sentiment analysis, logarithmic transformations of attributes). The proposed model achieves the best results, lowering the error rate of 2.8% in respect to the baseline.

1. Introduction

Reddit is one of the most famous discussion and aggregation website. Users can start threads in sections, and the interaction with the community leverages on comments and replies. For the scope of this work, we are trying to estimate a popularity metric for comments, given the text and 3 attributes. In particular, we work on a data-set of 12000 comments, performing the training with the first 10000 samples. We apply a multivariate linear regression in order to estimate the popularity score. In the first instance, we compare the performances (MSE and time) of the closed-form solution with the Gradient Descent (GDS) one, considering only the 3 attributes as features (Model $3-Simple$). In the second instance, we compute the most k ($k \in \{60, 160\}$) popular words in the training-set, and introduce k new features based on their occurrences in each comment. Models with $k = 60$ and $k = 160$ (namely $W - 60$ and $W - 160$) worsen the MSE over the validation set compared to $3-Simple$. Finally, we present a new model which improves the performances of $3-Simple$. We employ several metrics of the language processing library NLTK (Bird et al., 2009), and adapt such metrics for the model according to the lower covariance between our new features.

Theoretical background

We give a brief introduction to the mathematics for multivariate regression. We reference to Bishop (2006) for a detailed review. Consider as $\boldsymbol{X}$ the $n \times (m + 1)$ matrix of n observations, and $\boldsymbol{y}$ as the $n \times 1$ column vector of predictions. We assume as m the number of features, to which a dummy all-1 column is added in order to represent the intercept term. Under the iid assumption (independent and identically distributed random variables), the aim is to estimate the vector $\boldsymbol{w}$ of coefficients that both interpolates data-point and minimizes a given loss function in (1).

$$MSE(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \qquad (1)$$

The target interpolating function (2) is the one minimizing the loss function.

$$f(w, \boldsymbol{x}) = \arg\min_{w} \sum_{j=1}^{n} (y_j - \boldsymbol{w}_w^{\boldsymbol{T}} \boldsymbol{x}_j)^2 \qquad (2)$$

The overall time-complexity for computing the estimators with a closed-form approach is $O(m^3 + nm^2)$, and can hence be computationally challeinging with large datasets and/or features. The step-based *Gradient Descent* (*GDS*) provides an alternative for converging towards an estimation of vector $\boldsymbol{w}$. The *GDS* exploits the information given by the derivate of the loss-function w.r.t $\boldsymbol{w}$ in order to modify $\boldsymbol{w}$ itself. $\delta MSE(\boldsymbol{w})/\delta \boldsymbol{w}$ is the partial derivate of the *MSE* w.r.t vector $\boldsymbol{w}$ of incumbent weights. Given an initial vector of weights $\boldsymbol{w_0}$, the matrix $\boldsymbol{X}$ and predictions $\boldsymbol{y}$, the *GDS* modifies vector $\boldsymbol{w}$ proportionally to the partial derivate and a learning rate of $\alpha = f(\beta, \eta)$.

```
1: **Input:** $\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w_0}, \beta, \eta, \epsilon$
2: **Output:** $\boldsymbol{w}_a^*$
3: $i = 0; Improve$=true
4: **while** $Improve$ =True **do**
5:    $\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - 2\frac{\eta}{1+i\beta}(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_{i-1} - \boldsymbol{X}^T\boldsymbol{y})$
6:    **if** $||\boldsymbol{w}_i - \boldsymbol{w}_{i-1}||_2 < \epsilon$ **then**
7:       $Improve$=false
8:       $i + +$
9:    **end if**
10: **end while**
11: **return** $\boldsymbol{w}_i$
```

Objectives

In the scope of this work, we compare the stability and performances of the *GDS* and the *closed form* approach considering 3 simple features ($3 - Simple$) provided among with the dataset (see Section 2). Three update methods for the learning rates α are tested. In the second instance, we extract the most $k \in \{60, 160\}$ popular words from the training-set and model their occurrence in each comment as a feature. The 60 or 160 features are combined with the 3 simple ones in the two models $W - 60$, and $W - 160$. The MSE for $3 - Simple$, $W - 60$ and $W - 160$ is compared by estimating $\boldsymbol{w}$ both in the closed-form and with *GDS*. Finally, we create new features and we show how the performances on the validation-set are improved compared to the baseline ($3 - Simple$).

2. Dataset

The dataset contains 12000 samples sourced from the Reddit and provided by Hamilton (2019). Each sample has the following attributes:

- *text*: the comment's full-text.
- *children* $\in \mathbb{N}$: number of replies to the comment.
- *controversiality* $\in \{0, 1\}$: a Reddit's proprietary metric.
- *is_root* $\in B = \{T, F\}$: a flag set to 1 iff the comment is the root in a thread.
- *popularity* $\in \mathbb{R}^+$

The regression's target is the *popularity* attribute. The training-set is built on the first 10000 samples. Both the validation and test sets account for 1000 samples.

Features selection

We introduce a new set of features that improves the baseline of 2.8%. In the first instance, we perform a better preprocessing exploiting the NLTK framework (Bird et al., 2009). In particular, sentences are tokenized with the *TweetTokenizer*, and English stopwords are removed. The *PorterStemmer* from NLTK stems the words in order to restrict the number of terms in comments. Punctuation is removed, with the exception of symbols $s \in \{\#, !, ?\}$. Our idea is to lead back words to their morphological root, and generalize a larger class of words with simpler ones. Therefore, the following features are constructed:

- *TF*IDF* $\in \mathbb{R}$: compute the $TF * IDF$ for the most k popular word.
- *popFrequency* $\in [0, 1]$: the relative frequency of the k most popular words in the comment.
- *log*$(children^2 + 1) \in \mathbb{R}^+$: a transformation for the children attribute.
- *controversiality* $\in \{0, 1\}$
- *popScore* $= \sum_{i=1}^{k} \frac{1}{i} o_i \in \mathbb{R}^+$: where o_i is the absolute frequency of the $i - th$ most popular word in the comment.
- *is_root* $\in \{0, 1\}$.
- *log*$(|se|^2 + 1) \in \mathbb{R}^+$: a transformation of the *sentiment* attribute generated with $NLTK$
- $o_! \in \mathbb{N}$: the number of exclamation marks in the sentence.

- $o_{\#} \in \mathbb{N}$: the number of hashtags in the sentence.

The $TF * IDF$ for the $i-th$ most popular word captures the word's relative importance compared to all the words in the training set. Segall and Zamoshchin (2012) exploited the $TF * IDF$ metric to infer the importance of words between titles and bodies of Reddit's comments. The sentimental analysis (Bird et al., 2009) captures with a numerical score how positive a sentence is. Its value is taken in the absolute value and processed through a logarithm. We suppose that either strongly negative or strongly positive comments influence the popularity of a post.

σ	#2	#3	#4	#5	#6
O	0.4016	0.5886	0.6238	0.6961	0.7030
TF*IDF	-0.0081	-0.0169	0.0184	0.0390	0.1590
TF	-0.0361	-0.0763	0.0077	0.0207	0.1445

σ	#7	#8	#9	#10
O	0.3215	0.5106	0.6058	0.4581
TF*IDF	-0.0555	-0.0075	0.0235	0.0000
TF	-0.0352	-0.0161	0.0961	-0.0082

Table 1: *Covariance of the feature for the 1-st most popular word and features for 2nd to 10th most popular words*

The hint for all these features is provided by the covariance analysis (see Table 1) between popular words and 3 attributes (Occurrence O, $TF * IDF$ and TF). The original features of occurrence are highly dependent with each others, since the σ covariances are bigger than the ones for TF (frequency of a popular word in the comment) and $TF * IDF$. Hence, we tried to lower the mutual dependence of features by replacing the number of occurrences O with the $TF * IDF$. The TF has been implemented through the *popFrequency* which envelopes the absolute frequencies of the k most popular words by ranking them with a paraboloid.

3. Results

Closed form vs *GDS*

In the scope of this work, the training set was relatively small as well as the number of features. Therefore, the closed for approach provided the exact result with a faster computing time (see Table 2). Closed-form ran 100 times and the average metrics are reported. *GDS* for

#	MSE	Time (ms)
GDS^a	1020.5	614
EXACT	1020.3	20

Table 2: *Exact Approach vs GDS($\eta = 2 \cdot 10^{-5} \beta = 8 \cdot 10^{-3} \epsilon = \cdot 10^{-10}$) on the validation set*

results in Table 2 and Figure 1 implements an inverse-time update method $\alpha = \eta/(1 + i\beta)$. Moreover, the *GDS* is suitable for large matrixes $\boldsymbol{X}$, hence with large training sets and/or a large set of features. In term of stability, both the algorithms produce similar results with different portions of the training set (disjoint subsets).

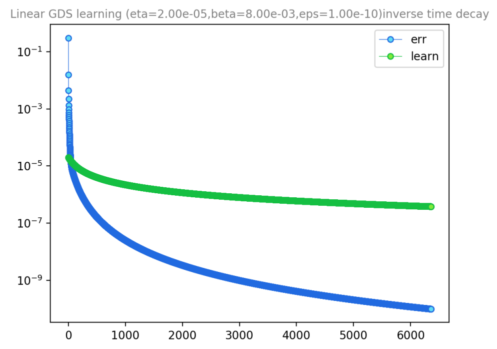

Figure 1: *Learning rate α and $err = ||\boldsymbol{w}_i - \boldsymbol{w}_{i-1}||_2$ for the GDS in Table 2. Y-Axis is logarithmic*

.

Two new learning rate update methods are applied. The notation i refers to the *step*, namely the number of iteration being performed.

- cosine (see Figure 2): $\alpha = \eta[(1/2)(1-\beta)(1 + cos(\frac{\pi \cdot i}{10^3}) + \beta])$
- exponential (see Figure 3): $\alpha = \eta \cdot \beta^{\frac{i}{10^3}}$

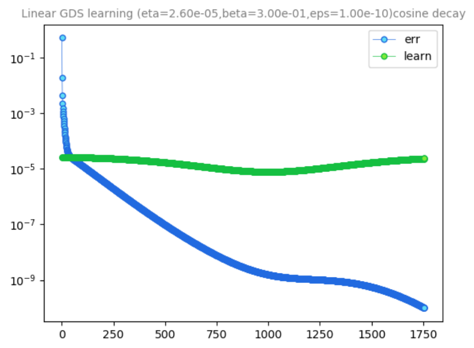

Figure 2: *cosine update method*

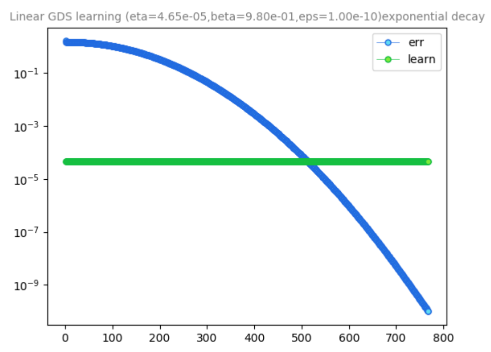

Figure 3: *exponential update method*

Features on vocabulary

We tested $3-Simple$, and compared its performances with $W-60$ and $W-160$. The benchmark is performed with both closed-form and GDS. The GDS can achieve either better or worse performances, depending on the parameters. $W-160$ is overfitting with the closed-form, while in all the other cases there is a good fit. In the reported GDS, more features lead to increased computing times but lower MSE (see Table 3).

#	C-form MSE	t (ms)	GDS MSE	t (ms)
3-Simple	1020.3	20	1089.4	269
W-60	1037.5	258	1087.3	1352
W-160	2188.4	798	1084.9	2968

Table 3: *Comparison of features selections over the training-set with closed form and GDS(inverse-time* $\eta = 7.5 \cdot 10^{-6} \beta = 5 \cdot 10^{-1} \epsilon = \cdot 10^{-7}$*)*

Improved features

The parameter k - namely the number of popular words - selected for our model is the one ($k = 8$) achieving the minimum MSE over the validation-set (see Figure 4). In particular, we tested the features selection with $k = 1$ popular words to $k = 10$. The proposed model improves the baseline of $3-Simple$ (see Table 4) of 2.8%. We found a singularity with $k = 11$, and therefore stopped our analysis.

4. Discussion

The results achieved by our custom model improves the baseline of $3-Simple$. Text analyisis methods provided insights for the features selection, since only 3 attributes were provided along with the comments. Sentiment analysis can be trained and tailored to the specific model. Moreover, POS analysis can provide additional information for new features. The logarithmic transformations we introduced for the features *children* and *sentiment* improve our MSE error. In order to extend the analysis of our model for larger values of k, the singularity can be removed. Starting from a non-singular set of z features, features for the $z+1$ word are introduced. If the new feature causes a singularity, the $z-th$ word is not included in the model. Otherwise, the training is performed and a new estimation for the MSE is given. Performances can improve by performing this extended analysis.

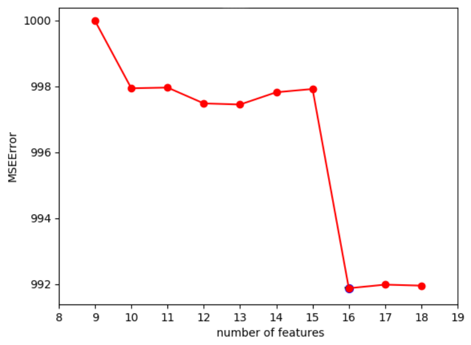

Figure 4: *Number of popular words compared to MSE over validation-set.*

.

#	MSE$_{tr}$	%	MSE$_{Test}$
160-W	2188.4	220.0%	-
3-Simple	1020.3	100%	-
Custom	991.9	97.2%	1261.0

Table 4: *Comparison between features selections. Custom refers to our proposed set of features.*

Statement of Contributions

1. Gabriele [45%]: features ($popFrequency$, $log(children^2+1)$, $popScore$, sentiment analysis, $o_{!,\#}$) LaTeXwriting, proof reading, code, literature review and results aggregations.
2. Zhiyue [10%]: popular words selection.
3. Jiaqi [45%]: features ($TF-IDF$, original TF, $popScore$), code, learning rate analysis, covariance analysis, graphs, and proof reading.

References

Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".

Bishop, C. M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

Hamilton, W., 2019. Comp551 lectures. McGill University.

Segall, J., Zamoshchin, A., 2012. Predicting reddit post popularity. nd): n. pag. Stanford University.