Regression-enhanced Random Forests with Personalized Patching for COVID-19 Retweet Prediction

Guangyuan Piao
Maynooth International Engineering College
Department of Computer Science
Maynooth University
Maynooth, Co Kildare, Ireland
guangyuan.piao@mu.ie

Weipeng Huang
Insight Centre for Data Analytics
School of Computer Science
University College Dublin
Dublin, Ireland
weipeng.huang@insight-centre.org

ABSTRACT

In this report, we describe an ensemble approach with a set of enhanced random forest models for COVID-19 retweet prediction challenge at CIKM Analyticup 2020 held by the 29th ACM International Conference on Information and Knowledge Management. The proposed approach is based on a global model and a set of personalized models. The global model consists of a set of random forests enhanced by three different types of models such as linear regression, feed-forward neural networks, and factorization machines. In addition to this global model, we trained a number of personalized models for users that exist in both training and test sets and have a sufficient number of tweets for training. Our approach obtained a MSLE (Mean Squared Log Error) value of 0.149997 on the test set of the challenge and ranked 4th on the final leaderboard.

KEYWORDS

COVID-19, Random Forests, Neural Networks, Factorization Machines, Deep Learning, Retweet Prediction, Twitter

1 INTRODUCTION

Retweeting or reposting, a function to repost a post such as a tweet with followers, is one of the most crucial functionalities in many popular social media platforms such as Twitter[1] or Weibo[2] as it enables information spreading on those platforms. Understanding retweet behavior is useful for many applications such as political audience design [8] or fake news spreading and tracking [9]. Therefore, understanding and modeling retweet behavior has been an active research area and might be particularly helpful during times of crisis, such as the current COVID-19 pandemic.

In this regard, the COVID-19 retweet prediction challenge held in conjunction with the 29th ACM International Conference On Information and Knowledge Management was launched to better understand retweet behavior in the context of COVID-19. The challenge has two phases including validation and testing where 51 teams participated the validation phase and 20 teams participated the testing pahse. In this report, we present our proposed approach for the retweet prediction task in the challenge, which ranked 4th place on the final leaderboard after the testing phase.

[1]https://twitter.com
[2]https://weibo.com

In: Dimitar Dimitrov, Xiaofei Zhu (eds.): Proceedings of the CIKM AnalytiCup 2020, 22 October, 2020, Gawlay (Virtual Event), Ireland, 2020, published at http://ceur-ws.org.

1.1 COVID-19 Retweet Prediction Challenge

The retweet prediction challenge is based on the TweetsCOV19 dataset [2] — a publicly available dataset containing more than 8 million tweets related to COVID-19, spanning the period October 2019 to April 2020. On top of the TweetsCOV19 dataset, the dataset provided by the challenge and the problem and evaluation metric are given as follows.

Dataset. The dataset of the challenge consists of 8,151,524 COVID-19 related tweets for training, 961,182, and 961,183 tweets for validation, and testing, respectively. In addition, the challenge also provides a set of features for each tweet, such as:

- *TweetID* for each tweet from Twitter
- *Username*, i.e., the author of a tweet
- *Timestamp* of a tweet in the UTC time zone
- *#Followers(No. of followers)* which indicates the number of followers of the author of a tweet
- *#Friends(No. of friends)* which indicates the number of friends of the author of a tweet
- *#Favorites(No. of favorites)* which indicates the number of favorites of a tweet
- *Entities* and their scores extracted from each tweet using FEL library [1]
- *Sentiment* scores of each tweet extracted from SentiStrength[3]
- *Mentions* of other user accounts in each tweet
- *Hashtags* in each tweet
- *URLs* in each tweet
- ***#Retweets(No. of retweets)*** which indicates the number of retweets of a tweet. This is the target variable for prediction on the validation and test datasets.

Problem. Given the set of features for a tweet from TweetsCOV19, the task is to predict the number of times it has been retweeted.

Evaluation metric. Consider the predicted results $\hat{\boldsymbol{y}}$ and the actual retweet counts $\boldsymbol{y}$ on the test set, which are both of length M. The performance is evaluated by MSLE (Mean Squared Log Error):

$$\text{MSLE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{M}\sum_{i=1}^{M}(\ln(1+y_i) - \ln(1+\hat{y}_i))^2 \quad (1)$$

2 PROPOSED APPROACH

Our approach consists of two main components by splitting users into two groups based on whether the user exists in training, validation, and test sets. Figure 1 shows an overview of the approach.

[3]http://sentistrength.wlv.ac.uk/

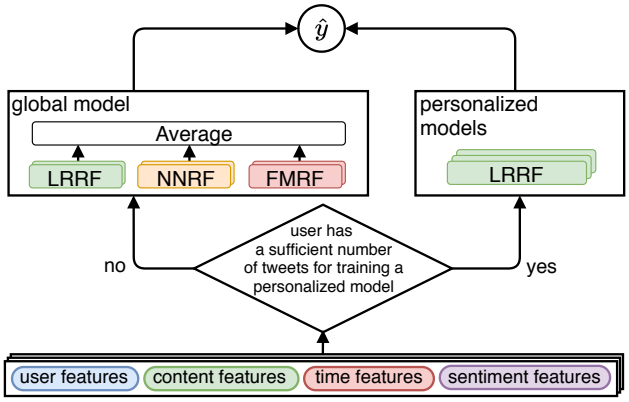

Figure 1: Overview of our proposed approach based on LRRF (Linear Regression-enhanced Random Forest), NNRF (Neural Networks-enhanced Random Forest), and FMRF (Factorization Machine-enhanced Random Forest) which are introduced in Section 2.1.

The first group of users consists are the ones who exist in both training and the test (and validation) sets with a sufficient number of tweets for training. The rest of users fall into the second group.

First, for the second group of users, we build a global model which is an ensemble of random forest models enhanced by linear regression, feed-forward neural networks, and factorization machines. Secondly, for each user in the first group, we build a personalized model for each user using a random forest enhanced by a linear regression model. Next, we discuss the global and personalized models in detail.

2.1 Global model

The global model is a collection of regression-enhanced random forests (RERF), which has been introduced recently in [10] to cope with the extrapolation problem of random forests where predictions on the test set are required at points out of the domain of the training dataset. In contrast to the definition of RERF with a specific regression model (Lasso) in [10], we use a general definition of RERF in this work as follows:

Given a training dataset $\boldsymbol{C} = \{C_i = (\boldsymbol{x}_i, y_i) : i = 1, \ldots, N\}$ where N is the size of the training set. Also, $\boldsymbol{y} = \{y_i : i = 1, \ldots, N\}$ is the set of targeted feature values and $\boldsymbol{X} = \{\boldsymbol{x}_i : i = 1, \ldots, N\}$ refers to the final set of features (e.g., after manual engineering, transformation, scaling, adding high-order, or interaction):

Step 1: Train a regression model $g(\boldsymbol{X})$ using the training set, and let $\epsilon^{\lambda} = \boldsymbol{y} - g(\boldsymbol{X})$ be the residual from $g(\boldsymbol{X})$. Here, $g(\boldsymbol{X})$ can be any regression model such as linear, Lasso, Ridge, neural networks, or factorization machines, except a tree-based regressor. We then create a new training dataset $\boldsymbol{C}^{\lambda} = \{C_i^{\lambda} = (\boldsymbol{x}_i, \epsilon_i^{\lambda}) : i = 1, \ldots, N\}$.

Step 2: Train a random forest model $f(\boldsymbol{X})$ using the new training set $\boldsymbol{C}^{\lambda}$. The hyper-parameters can be predefined or determined with grid search and cross-validation.

Step 3: Given the trained model $g(\cdot)$ and $f(\cdot)$, the RERF prediction $\hat{\boldsymbol{y}}$ for the response at $\hat{\boldsymbol{X}}$ is given by $\hat{\boldsymbol{y}} = g(\hat{\boldsymbol{X}}) + f(\hat{\boldsymbol{X}})$.

We use ★★RF to refer to a RERF depending on which regression model is used for enhancing a random forest. The global model consists of three types of RERFs with 16 models in total where the final prediction is the mean of predicted values from those models.

- A LRRF (Linear Regression-enhanced Random Forest) which denotes a simple linear regression-enhanced random forest model. We used a simple linear regression without an intercept and regularization given the large number of examples in the training set. For the corresponding random forest model, we used one with a maximum depth of 20 which consists of 500 estimators/trees.
- Ten NNRFs (Neural Networks-enhanced Random Forests) where each NNRF uses feed-forward neural networks with different hyper-parameters (e.g., the number of hidden layers and neurons) for enhancing the corresponding random forest model. For the corresponding random forest model, we used one with a maximum depth of 18 which consists of 500 estimators.
- Five FMRFs (Factorization Machine-enhanced Random Forests) where four of them are DeepFM (Deep Factorization Machine) [3] models with different hyper-parameters (e.g., the number of iteration or seed) and one xDeepFM [4] for enhancing the corresponding random forest model. The random forest model consists of 500 estimators and has a maximum depth of 16 and maximum features of 50%.

For training, the input of each RERF is a set of feature values (we will discuss the features in Section 2.3) regarding a tweet and the number of retweets of it. Given MSLE as the evaluation metric of the challenge, we further log transformed the set of feature values and the number of retweets for each tweet for training a RERF. Those RERFs are implemented using scikit-learn [5] and DeepCTR [7] Python packages. The implementation details can be found in our github repository[4].

2.2 Patching personalized models

Although the global model captures the overall relationship between the set of features and the retweet count of a tweet, the relationship would vary depending on the author of a tweet [6]. Figure 2 shows an example of the variance of the relationship between the number of favorites and the number of retweets for two different users in a log scale. Therefore, for the first group of users who are in both training and test sets and have at least 10 tweets for training, a personalized LRRF model for each user is trained, and the prediction using the global model will be patched/updated with the prediction from a personalized model.

One challenge of training a personalized model is the number of tweets for a user can be limited, and using all features used for training the global model can result in overfitting. To cope with this problem, we only used *#Favorites* as a single feature to learn a personalized model for each user. Also, as tweets having zero values in either *#Favorites* and *#Retweets* are not useful to learn a personalized model, we further limit users who have more than six tweets having none zero values in both *#Favorites* and *#Retweets*. Overall, 236,240 tweets in the test set belong to this category.

[4] https://github.com/parklize/cikm2020-analyticup

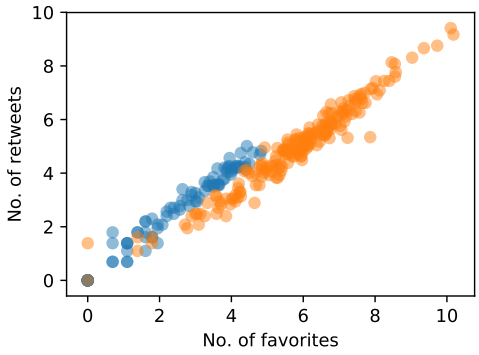

Figure 2: The relationship between the number of favorites and retweets for two different users in a log scale.

On one hand, the above-mentioned personalized LRRFs using a single feature might resolve the problem of overfitting for users with a small number of tweets. On the other hand, we found that those LRRFs can result in underfitting for user who have a large number of tweets for training. Therefore, for the group of users who have more than η tweets having nonzero values in both *#Favorites* and *#Retweets*, we use RidgeRFs (or LRRFs with L2 regularization) with all features that have been used for the global model where the penalty term is set to 5. We empirically found that $\eta = 160$ achieves the best results. Overall, 70,821 tweets in the test set belong to this category.

2.3 Features

On top of the features provided by the challenge for each tweet which has been introduced in Section 1, we extracted 30 features which are described in detail in Table 2. The features we used for training models in Section 2.1 and 2.2 can be classified into four categories: (1) user features, (2) content features, (3) time features, and (4) sentiment features.

User features denote a set of features related to the user/author of a tweet. In addition to the number of followers and friends of a user, we also included the ratio of those two numbers and the total number of tweets posted by the user in the training, validation, and test datasets. The total number of tweets shows the activity level of a user and we found that it helped to improve the prediction performance.

Content features include a set of features related to tweet content to capture different characteristics of the content. For example, the number of favorites that a tweet has, the popularity of entities, hashtags, mentions, and URL domain in a tweet. The popularity of an entity can be estimated by how many times an entity in a tweet appeared in all tweets in the training, validation, and testing datasets. We also noticed that a tweet could be retweeted more when a popular account (e.g., *@WHO*) is mentioned in the tweet. To incorporate popularity of mentioned users in a tweet, we used the maximum number of followers and friends of mentioned users where the number of followers and friends for each mentioned user has been obtained via the Twitter API.

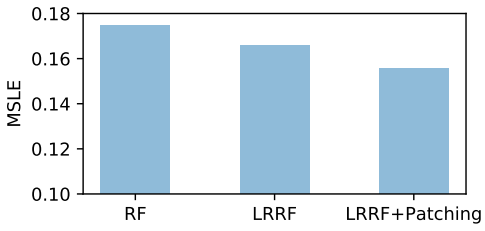

Figure 3: Improvement of the performance in terms of MSLE using an regression-enhanced random forest and personalized patching compared to using a random forest model on the validation set.

Time features consist of features that capture relevant information related to the time when a tweet is posted such as whether the tweet is posted on a weekend, or on which day of the week.

Sentiment features refer to both positive and negative sentiment scores of a tweet provided by the SentiStrength, and their interaction (e.g., the sum of positive and negative scores).

3 RESULTS

Table 1 shows the results of the top six teams (semi-finalists) according to the MSLE score in the testing phase. As we can see from the table, our team (PH) achieved the MSLE score of 0.149997 on the test set and ranked 4th among 20 teams.

To investigate whether a regression-enhanced random forest or personalized patching (i.e., updating with personalized models) improves the prediction performance, we tested the prediction results on the validation set using a random forest model, LRRF, and applying personalized patching for users who have a sufficient number of tweets for training a personalized model as we described in Section 2.2. Figure 3 shows that the MSLE decreases when using LRRF as well as applying personalized patching, which clearly shows the contribution of each component of our approach.

4 CONCLUSION

In this report, we presented an approach using regression-enhanced random forests with personalized patching for the task of COVID-19 retweet prediction. Regression-enhanced random forests with different types of regression models improved the performance of prediction compared to using a single regression-enhanced random forest. In addition, personalized patching for those users having

Table 1: Results of MSLE (Mean Squared Log Errors) for semi-finalists of the challenge.

User (Team)	MSLE
vinayaka (BIAS)	0.120551 (1)
mc-aida (MC-AIDA)	0.121094 (2)
myaunraitau	0.136239 (3)
parklize (PH)	0.149997 (4)
JimmyChang (GrandMasters)	0.156876 (5)
Thomary	0.169047 (6)
⋮	⋮

Table 2: Details of used features in our approach. The features are classified into four categories (with the number of features in each category) in the table.

Category	Feature	Description
User (4)	No. of followers	Number of followers that a user has
	No. of friends	Number of friends that a user has
	No. of friends / No. of followers	The ratio of those two numbers
	Number of tweets	No. of tweets posted by a user
Content (20)	No. of favorites	Number of favorites that a tweet has
	No. of favorites / No. of followers	The ratio of those two numbers
	Has entity	1 or 0 to denote whether a tweet contains any entity
	Has hashtag	1 or 0 to denote whether a tweet contains any hashtag
	Has mention	1 or 0 to denote whether a tweet mentions other users
	Has URL	1 or 0 to denote whether a tweet contains any URL
	No. of entities	The total number of entities extracted from a tweet
	No. of hashtags	The total number of hashtags in a tweet
	No. of mentions	The total number of mentions in a tweet
	No. of URLs	The total number of URLs in a tweet
	Entity popularity	How many times an entity in a tweet appeared in all tweets (Take the maximum value of all entities in a tweet)
	Hashtag popularity	How many times a hashtag in a tweet appeared in all tweets
	Mention popularity	How many times a mentioned user in a tweet appeared in all tweets
	URL domain popularity	How many times the domain of a URL in a tweet appeared in all tweets
	Tweet length	The total number of entities, hashtags, mentions, as well as URLs
	No. of top 20 entities	Number of top 20 entities from all tweets of a day
	No. of top 20 hashtags	Number of top 20 hashtags from all tweets of a day
	No. of top 20 mentions	Number of top 20 mentioned users from all tweets of a day
	Maximum No. of followers of mentioned users	The maximum number of followers of mentioned users in a tweet
	Maximum No. of friends of mentioned users	The maximum number of friends of mentioned users in a tweet
Time (3)	Time segment	The time segment of a tweet $\{1 \cdots 24\}$ indicating when it is posted
	Weekend	1 or 0 to indicate whether a tweet is posted on a weekend or not
	Day of week	A value from $\{1 \cdots 7\}$ to indicate the n^{th} day of a week
Sentiment (3)	Positive sentiment	A score for positive (1 to 5) sentiment for a tweet
	Negative sentiment	A score for negative (-1 to -5) sentiment for a tweet
	Overall sentiment	The sum of positive and negative sentiment of a tweet

a sufficient number of tweets for training a personalized model further improved the performance.

ACKNOWLEDGEMENTS

We pay our highest respect to numerous healthcare professionals and volunteers battling the COVID-19 pandemic on the front lines. W. Huang is supported by Science Foundation Ireland under grant number SFI/12/RC/2289_P2.

REFERENCES

[1] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and Space-Efficient Entity Linking in Queries. In *Proceedings of the Eight ACM International Conference on Web Search and Data Mining* (Shanghai, China) *(WSDM 15)*. ACM, New York, NY, USA, 10.

[2] Dimitar Dimitrov, Erdal Baran, Pavlos Fafalios, Ran Yu, Xiaofei Zhu, Matthäus Zloch, and Stefan Dietze. 2020. TweetsCOV19 - A Knowledge Base of Semantically Annotated Tweets about the COVID-19 Pandemic. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 2991–2998. https://doi.org/10.1145/3340531.3412765

[3] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[4] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[6] Guangyuan Piao and John G Breslin. 2018. Learning to Rank Tweets with Author-Based Long Short-Term Memory Networks. In *International Conference on Web Engineering*. Springer, 288–295.

[7] Weichen Shen. 2018. DeepCTR: Easy-to-use,Modular and Extendible package of deep-learning based CTR models. https://github.com/shenweichen/deepctr.

[8] Stefan Stieglitz and Linh Dang-Xuan. 2012. Political communication and influence through microblogging–An empirical analysis of sentiment in Twitter messages and retweet behavior. In *2012 45th Hawaii International Conference on System Sciences*. IEEE, 3500–3509.

[9] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

[10] Haozhe Zhang, Dan Nettleton, and Zhengyuan Zhu. 2019. Regression-enhanced random forests. *arXiv preprint arXiv:1904.10416* (2019).