Web Traffic Time Series Forecasting

Jiayi Gao(jg4821), Ben Zhang(bz957), Xiaoyu Xue(xx761)

1. Introduction

Internet surfing is almost a daily routine for everyone in the society. When people visited a webpage, they left digital footprints where we can gather this information and analyze them. Our project is using the collected data, the number of visits for a collection of wiki pages, to predict the future traffic of these pages. The general application of the solution to this problem is beneficial and applicable to many other temporal and sequential data.

2. Data Set Description

The main data set is Wikipedia traffic data starting from July, 1st, 2015 up until September 1st, 2017. It's provided by a research prediction competition held by Kaggle. Moreover, during the exploration, we might use some other data sources to augment our main data, like Google Trend data, etc.

The dataset consists of approximately 145k time series. Each of these time series contains meta information and a number of daily views, including article id, article name, language, access, date and daily hit counts. More detailed features can be extracted from these raw features like weekday, holiday, article topics. Inevitably, the data set has some issues that need be handled before precasting. For example, the data set doesn't distinguish between traffic values of zero and missing values. Another issue is that about 8% values in this data set are missing, which is not trivial and needed to be taken into account in our analysis.

3. Evaluation Methods

The performance will be evaluated on SMAPE between forecasts and actual values. Since the prediction or true values are can be zero, we used smoothed differentiable SMAPE variant, which is well-behaved at all real numbers [1]:

$$\epsilon = 0.1$$

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{\max(|A_t| + |F_t| + \epsilon, 0.5 + \epsilon)/2}$$

The boundary of SMAPE is between 0 and 200%. The smaller the SMAPE is, the better our model performs. For the training purposes, we may use MAE loss on log1p(data) which is smooth almost everywhere and close enough to SMAPE.

4. Baseline Solutions

Linear Regression: We will use linear regression as one of our baseline models. It's straightforward a regression model could be used to work with time series data. A reported

mean absolute error with 70% of the data to train is more than 77.19 [2]. We will be building our own ridge regression model and evaluate with SMAPE.

Median Model: Simply calculating median number of visits in last 30/40/50 days and use it as a prediction.

ARIMA: ARIMA models can be used to make predictions for time series. But some of our data are seasonal and not stationary, so we will try Seasonal ARIMA which is an improvement version of ARIMA. SARIMA takes into account the seasonality of dataset and will provide us with a useful baseline to compare other methods to.

5. Methods

5.1 Data preprocessing:

For the missing value we may fill with 0, but it needs more discussion. Raw values can be transformed by log1p() to get more-or-less normal intra-series values distribution, instead of skewed one. All features are normalized to zero mean and unit variance.[3]

5.2 Feature generation：

For now, we selected some basic features including article id, article name, language, access, date, daily hit counts, median of views, mean of views, weekday and holiday. Some of them are directly given by the data set and some of them need some calculation and complementary information. Afterwards, we will also try creating some combinations of these features, for example, weekday median of views, holiday median of views, mean of same language/access/topics article views, etc. We hope these complicated features would gain us some performance score.

5.3 Models/Tools:

a)Recurrent Neural Networks (RNN)

We will train our data on RNN because it can use its internal state to process sequential data. The connection between data points allows it to display dynamic temporal behavior for a time sequence. We hope this would improve the accuracy of the prediction.

b)Prophet

Prophet is a open source procedure for forecasting time series data. It is based on an additive model where non-linear trends are fit with yearly and weekly seasonality, plus holidays. It works best with daily periodicity data with at least one year of historical data. Prophet is robust to missing data, shifts in the trend, and large outliers. Therefore, we consider it as a powerful candidate tool.

c)Convolutional Sequence to Sequence Learning (CNN-seq2seq)

Seq2seq is natural for time series and achieved good results according previous works. In traditional CNN model, computations over all elements can be fully parallelized but can not be sequenced. Thanks to the work by Jonas Gehring(2017), we can apply an entirely convolutional sequence to sequence modeling in our project. We will equip our model with gated linear units (Dauphin et al., 2016) and residual connections (He et al., 2015a).[4] We hope this combination enables us to tackle the large web-traffic dataset with high speed and accuracy.

6. Project Plan Timeline

Dates	Contents
03.24 - 03.30	Data preprocessing
03.31 - 04.06	Feature generation
04.07 - 04.17	Models and Selection
04.18	Second meeting with adviser
04.19-05.01	Adjust our model according the adviser's advice: Model tuning and optimization
05.02	Third meeting with adviser
05.03-05.10	Wrapping up: Code debug & Clean up. Final Report.
05.11	Final Project Reports Due to Advisers

7. Previous work and Reference

[1],[3] *1stplacesolution, Arthur Suilin,*

https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion/43795

[2] *Predictive Analysis with different approaches, Julien Heiduk,*
https://www.kaggle.com/zoupet/predictive-analysis-with-different-approaches

[4] *Convolutional Sequence to Sequence Learning. Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, Yann N. Dauphin*