CROP: An Efficient Cross-Platform Event Popularity Prediction Model for Online Media

Mingding Liao[1], Xiaofeng Gao[1](✉), Xuezheng Peng[2], and Guihai Chen[1]

[1] Shanghai Key Lab of Scalable Computing and Systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
liao.mingding@gmail.com, {gao-xf,gchen}@cs.sjtu.edu.cn

[2] Baidu, Inc., Beijing, China
pengxuezheng@baidu.com

Abstract. The popularity analysis of social media is crucial for monitoring the spread of information, which is beneficial to public concerns track and decision-making for online platforms. Numerous studies concentrate on the trend analysis on single platform, but they neglect the data correlation between different platforms. In this paper, we propose CROP, a cross-platform event popularity prediction model to forecast the popularity of events on one platform based on the information of the auxiliary platform. We first define the cross-platform event popularity prediction problem. Then we clean the data and explore the slot matching of event time series in diverse platforms. Moreover, we first define the aggregated popularity for the feature construction of event popularity prediction model. Finally, extensive experiments based on events data show that CROP achieves great improvement for predicting accuracy over other baseline approaches.

Keywords: Cross-platform · Event popularity prediction
Support vector regression

1 Introduction

In today's world, the Internet has become the main information dissemination media with large user base. Therefore, the analysis of information dissemination in the online media have become an important research topic [7]. The online media are the complex system consisting of diverse platforms including social networks, search engines, online group, online forum, etc. A large number of studies focus on the event popularity analysis on a single platform [11–14,16], but they neglect the data correlation between different online platforms. However, event information is generally spread through the multiple platforms. For example, an event would be introduced on the website, be searched on the search engine, be relaid upon the social network and be talked in the forum. In fact,

S. Hartmann et al. (Eds.): DEXA 2018, LNCS 11030, pp. 35–49, 2018.
https://doi.org/10.1007/978-3-319-98812-2_3

particular properties of different platforms, such as the users communities and the speed of propagation, demonstrate the potential for cross-platform research for event popularity prediction, which plays a crucial a role in understanding the process of event propagation [19].

Therefore, we design CROP, an efficient **CRO**ss-**P**latform event popularity prediction model for online media. To measure the event popularity, CROP employs and improves the Term Frequency-Inverse Document Frequency (TF-IDF) based method which is developed in [19]. Then CROP studies the correlation of time series and employs Dynamic Time Warping method and improves it with the penalty and compound distance to generate sequence alignment matches. Next, we give the novel definition of aggregated popularity for the purpose of the application of data correlation and CROP extracts features on the basis of aggregated popularity. Last, Support Vector Regression is employed for event popularity prediction.

The contribution of this paper are summarized as follows: (1) We design an efficient scheme called CROP for cross-platform event popularity analysis and prediction with only post and query information; (2) We first study on and make use of the correlation of the cross-platform data for event popularity prediction; (3) We first improve Dynamic Time Warping method for time series matching and then define the aggregated popularity for feature construction on the basis of time series correlation; (4) We conduct extensive experiments on several datasets, which demonstrate that CROP performs best among the baselines.

The rest of this paper is organized as follows. In the Sect. 2, we introduce the related work of CROP. The problem statement is given in Sect. 3. In the Sect. 4, we introduce the overview and main components of CROP. The experiments are showed in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related Work

2.1 Event Popularity Analysis on Individual Platform

A event could be defined as a single word or a coherent set of semantically related terms which summarizes the majority of related documents or other semantics items [12,17,24]. Event popularity in online media is usually defined as the number of posts, reposts queries or comments. There are numerous studies regarding event popularity prediction, and machine learning method is the major approach.

Leskovec et al. designed a meme-tracking approach and studied the coherent representation of the popularity in the new cycle. However, this paper emphasizes the periodicity and did not propose an accurate numerical method [10]. Armed with this research, Yang et al. [23] proposed the K-Spectral Centroid (K-SC) clustering algorithm with the wavelet-based incremental version to solve the problem. Wang et al. [22] also improved the K-Spectral Centroid method by selecting orthogonal polynomial function and using wavelet transformation to decrease the dimensionality of data. Bandari et al. [1] constructed a multi-dimensional feature space and employed regression and classification for popularity prediction. Then Wang et al. [21] aimed at predicting the time of appearance of the burst and then reduced it into a classification problem. Miao et al. [12] predicted the event popularity based on template vectors of popularity and speeded up the time series prediction method by setting representative users.

2.2 Cross-Platform Popularity Analysis

Few studies noticed the potential of cross-platform popularity analysis and attempted make use of the correlation of cross-platform data. Giummole et al. first studied the trending events in Twitter and Google and found that most Twitter trends would cause the later occurrence of similar Google trends [5]. On the basis of it, the Bipartite Graph of the trend was introduced to handle the similar keywords [6]. Then Tang et al. confirmed that the correlation also existed in Baidu and Sina Weibo [19]. Tolomei et al. noticed that Wikipedia had the Entity Linking technology, which could be linked by Twitter and provided the feasibility to predict the popularity of Wikipedia article based on Twitter data [20]. Keneshloo et al. pointed out the fact that the posts in the social network which mentioned an article could enhance the popularity of the article [8]. Chen et al. paid attention to two commonly used information acquisition platforms: Google and Stack Overflow. This study stated that the correlation between the Google Trends and Stack Overflow Data Dump and provided the dataset for others' future research [2].

3 Definition and Problem Statement

For an event, the raw data should be the semantic relevant items, for example, microblogs in social networks and queries in search engines. The time span of the corresponding event is divided into n periods and $T = [t_1, t_2, \ldots, t_n]$. For different platforms, $\mathbf{R^p}$ is defined as the collection of all information about the event in the platform p. $\mathbf{R_i^p}$ is the collection of all information on the platform p at the time slot t_i and $r_{i,j}^p$ is the j-th records in the R_i^p. In details, we organize the records as a sequence of words: $r_{i,j}^p = \langle d_{i,j,1}^p, d_{i,j,2}^p, \ldots, d_{i,j,s_{i,j}^p}^p \rangle$, where $s_{i,j}^p$ is the length of the sequence.

The event popularity on the platform p is labeled as pop^p, and pop_i^p is the popularity in the platform p at the time slot t_i. CROP provides two different definition of event popularity in terms of text length. On the one hand, the event popularity of short-text platform is defined as Eq. (1). For example, most queries in the search engine are extremely short with only one or two words and the purpose of queries is for acquaintance of events, so all of the queries should be treated equally.

$$pop_i^{p_1} = \frac{|\mathbf{R_i^{p_1}}|}{\max_{i=j}^{n} |\mathbf{R_j^{p_1}}|} \tag{1}$$

One the other hand, we employ a cross-platform event dissemination trend analysis approach [19] based on TF-IDF method in the long-text platform and make the necessary modification to improve the generality of the model, for example, Twitter. The popularity of an event is measured with the help of the hot words $HW = \{hw_1, hw_2, \ldots, hw_m\}$, where m is the number of hot words. Thus, the definition of the event popularity in the long-text platform is described as Eqs. (2) (3) (4), where nhw_j^i is the number of occurrences of hot word hw_i at the time slot t_j. In detail, $freq_i^{hw_j}$ is the term frequency of the hot word hw_j at the time slot t_j and $rec_i^{hw_j}$ is corresponding inverse record frequency. Different from the standard TF-IDF algorithm, we only consider the data before the time slot t_j as the total records instead of all data when computing the inverse record frequency to improve the applicability because the total records could be difficult to achieve.

$$pop_i^{p_2} = \sum_{j=1}^{m} rec_i^{hw_j} \times freq_i^{hw_j} \tag{2}$$

$$freq_i^{hw_j} = \frac{nhw_j^i}{\sum_k nhw_k^i} \tag{3}$$

$$rec_i^{hw_j} = \log \frac{|\bigcup_{k \leq i} \mathbf{R_i^{p_2}}|}{|\{x|hw_i \in x, \in \bigcup_{k \leq i} \mathbf{R_i^{p_2}}\}| + 1} \tag{4}$$

Both two definitions of popularity are independent for data, which ensures the applicability of CROP. Based on the previous definitions, the problem of the cross-platform event popularity prediction could be defined as Definition 1.

Definition 1. *Given the relevant raw documents R^{p_1} and R^{p_2} about a event and the division of the time span T. Provide the most proper values of popularity of the event at the next time slot on the target platform p_1 with the help of data on the assistant platform p_2.*

4 Cross-Platform Popularity Prediction Model

4.1 Overview

Figure 1 illustrates the structure of CROP. CROP consists of five steps: data preprocessing, hot word extraction, popularity analysis, time series matching and popularity prediction.

The first part is data preprocessing. Then the hot words are selected from the preprocessing raw data in the word extraction part. After the calculation of hot word dynamic frequency and the recursive document frequency, the popularity of the event is measured on the two platform at each time slot and is denosing by Discrete Wavelet Transform. In the next step, the two time series are matched by the Dynamic Time Warping method and the aggregated popularity is defined on the basis of the matching of popularity time series. Finally, the novel features constructed from aggregated popularity are used in the Support Vector Regression methods for event popularity prediction.

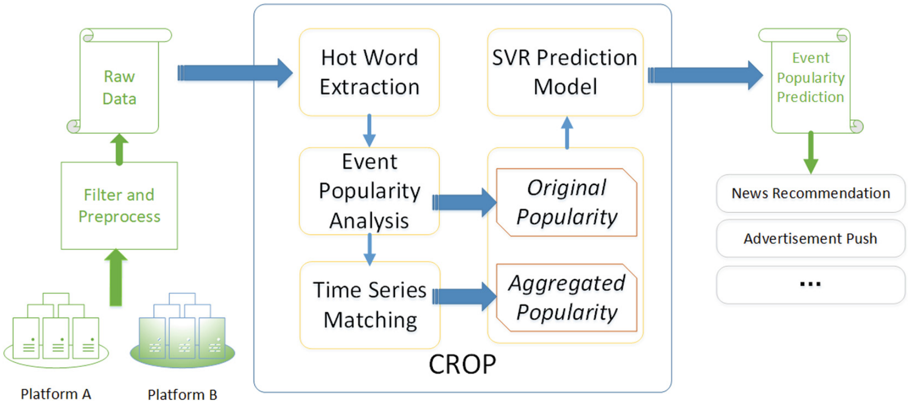


Fig. 1. Overview of CROP

4.2 Data Preprocessing

In the process of data acquirements, we utilize three methods to filter data. First, we provide related words for the corresponding event and only the records containing the given words or the similar words would be selected. Second, considering the enormous size of relevant data in the online media, the datasets used are sampling from the whole data. Last, the meaningless part of the raw data, such as URL, would be filtered.

4.3 Hot Word Extraction and Popularity Analysis

Before designing the model for popularity prediction, we have to provide a popularity measuring method. To make the process of popularity analysis convenient, the word extraction method is employed to extract the words and find the hot words. All of the documents are split into words based on a open-source project called jieba[1]. In the process of word extraction, we filter the stop words, which are the function words without actual meaning.

The hot words are selected based on the result of words extraction by the library in jieba, and then calculate the popularity by the definition mentioned in Sect. 3.

Because of the randomness of user behavior on the Internet, the event popularity series may contain noise. We novelly employ the Discrete Wavelet Transform method to denoise the data of the popularity. We utilize the Python library PyWavelets[2] to implement the process. We employ the Haar wavelet to implement the discrete wavelet transform [18].

[1] https://github.com/huaban/jieba-analysis.
[2] https://pypi.python.org/pypi/PyWavelets/.

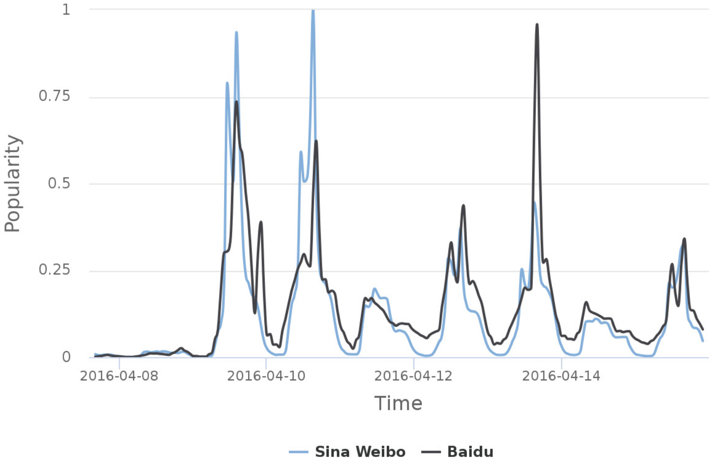


Fig. 2. The example of event popularity

4.4 Time Series Matching

Figure 2 is the popularity result of the event "Lee Sedol v.s. AlphaGo" in Sina Weibo and Baidu, which is introduced in Sect. 5.1, from the 16:00, March 7th, 2016 to the 00:00, March 16th, 2016. It demonstrates that the time series of two platforms are not matched slot by slot. To solve the problem, we novelly employ Dynamic Time Warping (DTW) method which is a popular dynamic programming method in the field of speech pattern recognition [3,15]. Dynamic Time Warping problem could be defined as the following: given two time series $P = \{p_1, p_2, \ldots, p_n\}$, $Q = \{q_1, q_2, \ldots q_m\}$,and find a matching from P to Q with the lowest cost. To simply the problem, it is assumed that each items p_i in P is matched with a continuous subsequence (called $M(p_i)$) of Q.

In term of the actual condition of the cross-platform popularity prediction, an obvious fact should be noticed that the time alignment of the two popularity time series should not be large. Therefore, we propose two ideas to improve the basic form of algorithm of the Dynamic Time Warping: (1) We define the penalty function to improve the similarity calculation based on the time difference; (2) We compress the state space of dynamic programming, which is the main process of Dynamic Time Warping. In the following part of this section, we describe the details of improvement.

Cost Function with Penalty. The cost function with penalty, called as $C(p_i, q_j)$, is the cost to match p_i and q_i. The basic idea is measuring the matching cost by Euclidean distance of p_i and q_j [15]. [9] showed the deviation of the time series could provide simple and robust measure result. Thus, we ulitize the geometric mean to integrate the Euclidean distance of the popularity value and the deviation of that, which is showed in Eq. (5).

$$C_{base}(p_i, q_j) = \sqrt{|p_i - q_i| \times |(p_i - p_{i-1}) - (q_i - q_{i-1})|} \tag{5}$$

Morever, we design the penalty coefficient for the constraint that the two time slots away from each other would not be matched. Logistic function is employed to define the penalty as the Eq. (6), where β is a parameter to control the effort of C_{pena} and b is an expectation of time slot bias.

Based on the Eqs. (5) and (6), the final formula of the cost function with penalty is Eq. (7).

$$C_{pena}(p_i, q_j) = \frac{1}{1 + e^{-\beta \times (|i-j|-b)}} \tag{6}$$

$$C(p_i, q_j) = C_{base}(p_i, q_j) \times C_{pena}(p_i, q_j) \tag{7}$$

Limited State Space. The naive DTW method suffers from high executing time. Limitation of the state space is proposed to solve the efficiency problem, which is inspired from the fact that we need not consider the matching with two time slots away from each other.

$LTC(P_{1..i}, Q, k)$ is defined as the minimal of the total matching cost of $P_{1..i}$ and $Q_{1..(i-k)}$ and the state space could be limited by the value of k. In detail, P is the popularity time series on target platfrom and Q is the popularity time series on assistant platform.

The recursive formula to calculate LTC is showed in Eq. (8).

$$LTC(P_{1..i}, Q, k) = \max \left\{ \begin{array}{c} LTC(P_{1..i-1}, Q, k-1) + C(p_i, q_j) \\ LTC(P_{1..i}, Q, k+1) + C(p_i, q_j) \\ LTC(P_{1..i-1}, Q, k) + 2 \times C(p_i, q_j) \end{array} \right\} \tag{8}$$

In the first case of Eq. (8), we give the limitation that $i > 0, k > 0$, and $k < len$ in the second case, and $i > 0, i - k > 1$ in the third case. Also, we let $k > 0$ in all cases, because we can only use the data of the assistant platform which is monitored before the predicting time.

To facilitate the following process of CROP, $ORI(P_{1..i}, Q, k)$ is set to record how the $LTC(P_{1..i}, Q, k)$ is calculated. It could assist us to find the previous state conveniently and know the best matching between $P_{1..i}$ and $Q_{1..(i-k)}$. The $ORI(P_{1..i}, Q, k)$ would be set from 1 to 3, which corresponds to the three cases of the origin of $LTC(P_{1..i}, Q, k)$.

4.5 Prediction Model Establishment

CROP reduces the event popularity prediction problem into the regression problem and construct the feature space based on the known event popularity on the target and assistant platforms. However, the different event propagation trends of two platforms inspire the special features. Considering the result of time series matching, the several time slots in the event popularity time serial on the assistant platform could be matched with the same time slot in the event popularity time serial on the target platform. The complicate relationship makes it arduous to employ data correlation for machine learning model. Therefore, CROP defines novelly aggregated popularity for the feature construction. The aggregated popularity $agpop_j^{p_2}$ on the assistant platform p_2 is defined as the average of the event popularity on p_2 at the time slot which is matched with the time slot t_j on target platform p_1. Algorithm 1 shows the calculation of aggregated popularity.

Algorithm 1. Aggregated Popularity Calculation

```
Input: popularity time series $P[1..n], Q[1..m]$, the parameters $vis$
Output: $agpop$ which is the aggregated popularity of $Q$
Invoke the Dynamic Time Warping method with the input of $P$, $Q$ and $vis$ and
get $LTC$ and $ORI$;
for $1 = 1; i \leq n; i \leftarrow i + 1$ do
    $index \leftarrow i - \arg\min_j(LTC(i, j))$;
    $now \leftarrow i$;
    $direc \leftarrow ORI(\arg\min_j(LTC(i, j)))$;
    for $j \leftarrow 0; j < vis; j \leftarrow j + 1$ do
        $tot \leftarrow 0$;
        $sum \leftarrow 0$;
        while $direc = 2$ do
            $tot \leftarrow tot + 1$;
            $sum \leftarrow sum + q_{index}$;
            switch the value of $direc$ do
                case 1
                    do not change $index$;
                    $now \leftarrow now - 1$;
                    $direc \leftarrow ORI(\arg\min_j(LTC(now, now - direc)))$;
                case 2
                    $index \leftarrow index - 1$;
                    do not change $now$;
                    $direc \leftarrow ORI(\arg\min_j(LTC(now, now - direc)))$;
                case 3
                    $index \leftarrow index - 1$;
                    $now \leftarrow now - 1$;
                    $direc \leftarrow ORI(\arg\min_j(LTC(now, now - direc)))$;
        $pop_{i-j}^{i} \leftarrow sum/tot$;
return $pop$;
```

Therefore, when considering the time slot t_i, $pop_i^{p_1}$ is treated as the label where p_1 is the target platform, and $\{pop_{i-vis}^{p_1}, \ldots, pop_{i-1}^{p_1}\}$ can be used as a part of feature, where vis is set to represent the length of time slot used for the feature establishment in the regression model. Moveover, the aggregated popularity $\{pop_{i-vis}^{p_2}, \ldots, pop_i^{p_2}\}$ on the assistant platform also be utilized as features. In all, the feature vector of time slot t_i is $\langle pop_{i-vis}^{p_2}, \ldots, pop_i^{p_2}, pop_{i-vis}^{p_1}, \ldots, pop_{i-1}^{p_1}\rangle$.

After the label and feature construction, the Support Vector Regression model is employed for event popularity prediction, which is one of the most efficient model for regression.

The time complexity of Algorithm 1 is $O(n \times (vis + len))$, where vis is the size of feature space and len is the length of limited state space of dynamic programming. Line 2 and Line 6 enumerates i and j, so the low bound of the complexity is $O(n \times vis)$. The running time from Line 9 to Line 25 is not trivial to derive, but every time the loop executes, either now or $index$ would decrease 1. In fact, now is equal to j and $index$ is the time slot matched with now, so $now \leq index - len$. Thus, for each i, Line 9 to Line 25 is only executed at most $2 \times vis + len$ times. Therefore, the time complexity is $O(n \times (vis + len))$.

5 Experiments

In this section, we present the experimental results of CROP and analyze the effectiveness. All of the experiments are implemented by Python 2.7 and running on a PC with the Intel(R) Core(TM) i5-6500 CPU @ 3.20 GHz, 8 GB RAM on the Ubuntu 16.04 operating system. The experiments are based on the datasets of three hot events in the Baidu and Sina Weibo, which are most popular search engine and social network in China.

5.1 Experiment Setup

Dataset Description. In this paper, we concentrate on the search engine (Baidu) and the social network (Sina Weibo) as the object of research, because they are the most important parts of social media and they have the unique properties. The social network is the platform of initiative information sharing, but the search engine only provides the information based on the users' queries, which lead to the hysteresis of the popularity time series of hot events in the search engine comparing with that in the social network [19]. Thus, the Search Engine is the object platform and the Sina Weibo is the target platform.

The dataset used in our experiment is the data of three hot events in Baidu, which are provided by research institute of Baidu Online Network Technology Company through internal interface and the corresponding data in Weibo through crawler. The datasets of Baidu consist of massive inquire records with inquire words and cryptographic user identity with data masking. The datasets of Sina Weibo consist of massive post records and cryptographic post information. Table 1 shows the details of the datasets. All of the dataset are the hot events from 2015 to 2016 in China. The "Lee Sedol v.s. AlphaGo" event is about the go competition between one of the best human go player and the artificial intelligence AlphaGo. "Brexit" is the event that majority of British people wanted to vote to leave the European Union in the wake of the migrant crisis. The event "The Capsizing of a Big Ship" was started by the accident that a ship called "Easten Star" capsized on Yangtze River with hundreds missing, and became the hot event because the plenties of later news and talks.

Table 1. Overview of the experimental dataset

Topic name	Date	Data of Sina Weibo	Data of Baidu
Lee Sedol v.s. AlphaGo	02/20–03/29 (2016)	654.89 MB	406.83 MB
The Capsizing of a Big Ship	06/01–06/29 (2015)	320.59 MB	401.48 MB
Brexit (Britain + Exit)	06/21–06/30 (2016)	715.51 MB	392.32 MB

For each dataset, we use 1 h as a unit of the time slot and divide them into discrete time series, because 1 h is a common time unit for social content process. For example, Sina Weibo maintains the hot post list by 1 h. To simplify the examination of our model, we arbitrary select a subsequence of the time series in 300 h. The data in the first 200 h is treated as the training set, and the other data is used as testing data.

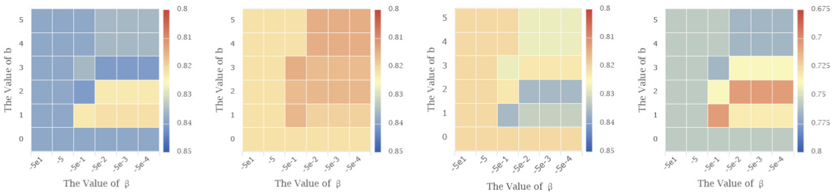
Metric Measures. In our experiments, the adjusted Mean Absolute Error (called $aMAE$) and the R-square (called $R\text{-}square$) are utilized to measure the predicting accuracy of the models. R-square is a common metric in regression model. Adjusted Mean Absolute Error is the metric improved from the Mean Absolute Error. The formula of these two metrics are Eq. (9), where y' is the label value of the item in the testing set and $\hat{y'}$ are the predicting result of y' and $\bar{y'}$ is the average of all y'. The predicting accuracy is higher if the adjust Mean Absoluted Error is smaller and the R-square is closer to 1.

$$aMAE = \frac{\sum_{y'} |y' - \hat{y'}|}{ave} \quad R\text{-}square = \frac{\sum_{y'} (y' - \bar{y'})^2}{\sum_{y'} (\hat{y'} - \bar{y'})^2} \tag{9}$$

Baseline. We set two baselines for experiments. The first baseline is SVR, which is the classical Support Vector Regression only based on the previous vis time slot popularity in the search engine. The second baseline is cro-SVR [6], which uses popularity at the previous vis time slot in the social network.

The difference between cro-SVR and CROP is that cro-SVR does not use the time series matching to eliminate the effect of time series alignment.

Parameters Setup. In this part, we set the value of the primary parameters in these experiments. We complement the preliminary experiments on "Lee Sedol v.s. AlphaGo" datasets. Figure 3 displays the predicting accuracy measured with R-square when selecting different β and b. In the corresponding preliminary experiments, vis is set as 2, 3, 4, 5. The experimental result shows that there is no option of β and b that performs best in all situations. Moreover, comparing Fig. 3(d) with other three figures, we could find the accuracy change in the Fig. 3(a), (b) and (c) is much smaller than that in Fig. 3(c). Therefore, it is vis instead of β and b that would influence the predicting accuracy.



(a) $vis = 2$ (b) $vis = 3$ (c) $vis = 4$ (d) $vis = 5$

Fig. 3. R-square w.r.t. β and b

Thus, it the following experiments, we set $\beta = -0.5$ and $b = 4$, which performs well in most situation, and we would concentrate on the influence of vis to the predicting accuracy.

5.2 Experiment Analysis

Analysis on Datasets. In this part, we would analyze the datasets about three hot event by the V/S Test and the Pearson Correlation Coefficient.

V/S test [4] is a typical method to distinguish whether the time series have long-term memory or short-term memory. The Hurst parameter H is the result of V/S Test. If $H > 0.5$, the sequence has long term memory. If $H \leq 0.5$, the sequence has short term memory.

Pearson Correlation Coefficient is used to reflect the degree of linear correlation between two variables. The greater the absolute value of Pearson Correlation Coefficient(PCC), the stronger the correlation.

Table 2 shows the V/S Test result of the datasets about event hot events the Pearson Correlation Coefficient between the popularity time series on Sina Weibo and Baidu.

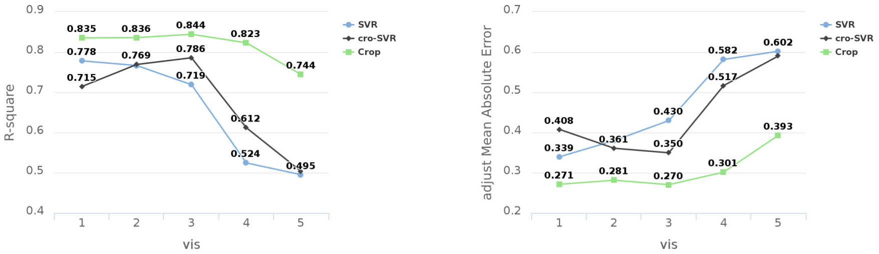
Table 2. Statistical Analysis of the Experimental Datasets

Topic name	Hurst of Baidu	Hurst of Sina Weibo	PCC value
Lee Sedol v.s. AlphaGo	0.389	0.379	0.839
The Capsizing of a Big Ship	0.375	0.322	0.762
Brexit (Britain + Exit)	0.422	0.388	0.859

Based on the result showed in the Table 2, we gain an intuitive understanding of about the popularity time series extracted from the datasets. The event popularity time series of hot eventS on Baidu and Sina Weibo have short-term memory, which means that it is reasonable to consider short term data only for event popularity prediction. Moreover, popularity time series of the same event have high correlation, which shows that the idea to predict the popularity of hot events on the search engine based on the data on the social network is feasible.

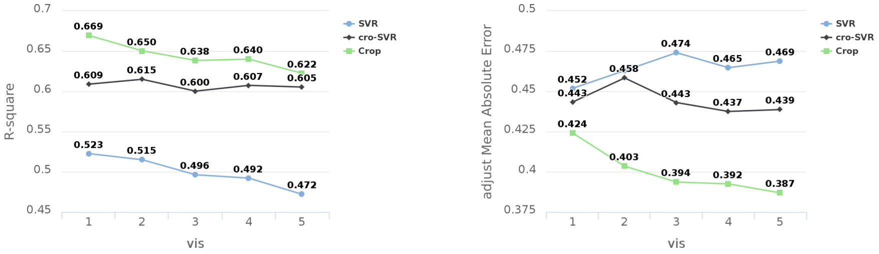
Experiment Results. In the part, we would provide the experiments on CROP and the other two baseline approaches SVR and cro-SVR in the datasets. The following figures show the experimental results in the event datasets of the hot event. In all, the performance of CROP is better than the other two baseline methods. In the following part, we will explain the experimental results in detail and analyze the reasons of these results.

The result of first case is showed in Figs. 4 and 5.



(a) Comparison of R-square (b) Comparison of aMAE

Fig. 4. Predicting accuracy about "Lee Sedol v.s. AlphaGo"



(a) Comparison of R-square (b) Comparison of aMAE

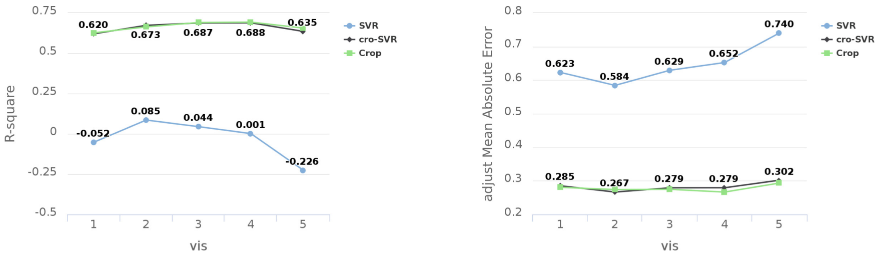
Fig. 5. Predicting accuracy about "Brexit"

Figure 4 shows the experimental results on the dataset about the event "Lee Sedol v.s. AlphaGo". In the two figures, the CROP performs best if comparing with the baselines when $vis = 2$. If setting $vis = 2$, CROP increases the predicting accuracy by 8.9% in terms of the R-square and by 21.9% in terms of adjust Mean Absoluted Error. The results demonstrate that the predicting accuracy decreases when the vis increases from 2, with respect to both the R-square and the adjust Mean Absoluted Error. It is reasonable because the bigger vis means that more history data could be utilized which result in high variance.

The same results also happen at the experiments based on the event "Brexit". The Fig. 5 shows the experimental results. The results show that the predicting accuracy is decreasing when the value of *vis* increases, which is similar to the trend in the experiment of "Lee Sedo v.s. AlphaGo". Moreover, when the *vis* is setting as 1, the predicting accuracy is increased by 9.8% in terms of R-squre and increased by 6.2% in terms of adjust Mean Absoluted Error. Another similarity of the experimental result of these two datasets is that the perfomance of cro-SVR is better than SVR but worse than CROP, which shows that the cross-platform data has the potential to improve the predicting accuracy even though the matching alignment between the two time series is not considered.

In all, CROP has the best predicting performance and SVR has the worst predicting performance on the datasets of "Lee Sedol v.s. AlphaGo" and "Brexit". Morever, the small *vis* leads to the best performance. The results are reasonable, since the hot event "Lee Sedol v.s. AlphaGo" and "Brexit" are guided by a series of latest news. Thus, people's behaviors are strongly influenced by the recent information, which lead to the result that small *vis* is better.

The result of second case is showed in Fig. 6.



(a) Comparison of R-square

(b) Comparison of aMAE

Fig. 6. Predicting accuracy about "The Capsizing of a Big Ship"

The Fig. 6 shows the experimental result on the dataset about the event "The capsizing of a Big Ship". The result is that CROP and cro-SVR perform much better than SVR in terms of both R-square and adjust Mean Absolute Error. The R-square of SVR is approximately from 0 to 0.1 but that of co-SVR and CROP is approximately from 0.6 to 0.7. The adjust Mean Absolute Error of SVR is about from 0.6 to 0.7, but that of co-SVR and CROP is about 0.3 The performance difference may result from the property about the event. The event "The capsizing of a Big Ship" is started by the accident which shocked a plenty of Chinese so they maybe knew about the news without much details in the online social networks, and then search for the details on the search engine.

Thus, it is concluded that CROP performs best for cross-platform event popularity prediction model compared with other baseline methods and the cross-platform data can provide significantly improvement of prediction accuracy.

6 Conclusion

In this paper, we have proposed the CROP, an efficient cross-platform event popularity prediction model. CROP measures the popularity based on the TF-IDF method, denoises the data by Discrete Wavelet Transform method, explores the correlation of cross-platform event popularity and predicts the event popularity based on Support Vector Regression. Specially, CROP employs and improves the Dynamic Time Warping method to match two time series in different platforms and novelly define the aggregated popularity to establish the feature space of CROP, which could provide the inspiration for follow-up research. The real time dataset of hot events in China from 2015 to 2016 are used to validate the effectiveness of CROP.

Acknowledgement. This work is supported by the program of International S&T Cooperation (2016YFE0100300), the China 973 project (2014CB340303), the National Natural Science Foundation of China (Grant number 61472252, 61672353), the Shanghai Science and Technology Fund (Grant number 17510740200), and CCF-Tencent Open Research Fund (RAGR20170114).

References

1. Bandari, R., Asur, S., Huberman, B.A.: The pulse of news in social media: forecasting popularity. In: International AAAI Conference on Weblogs and Social Media (2012)
2. Chen, C., Xing, Z.: Towards correlating search on Google and asking on stack overflow. In: IEEE Annual Computer Software and Applications Conference (COMPSAC), vol. 1, pp. 83–92 (2016)
3. Gao, T., et al.: DancingLines: an analytical scheme to depict cross-platform event popularity. arXiv preprint arXiv:1712.08550 (2017)
4. Giraitis, L., Kokoszka, P., Leipus, R., Teyssière, G.: Rescaled variance and related tests for long memory in volatility and levels. J. Econ. **112**(2), 265–294 (2003)
5. Giummol, F., Orlando, S., Tolomei, G.: Trending topics on Twitter improve the prediction of Google hot queries. In: IEEE International Conference on Social Computing (SocialCom), pp. 39–44 (2013)
6. Giummolè, F., Orlando, S., Tolomei, G.: A study on microblog and search engine user behaviors: how Twitter trending topics help predict Google hot queries. Human **2**(3), 195 (2013)
7. Hoang, B.-T., Chelghoum, K., Kacem, I.: Modeling information diffusion via reputation estimation. In: Hartmann, S., Ma, H. (eds.) DEXA 2016. LNCS, vol. 9827, pp. 136–150. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44403-1_9
8. Keneshloo, Y., Wang, S., Han, E.H., Ramakrishnan, N.: Predicting the popularity of news articles. In: SIAM International Conference on Data Mining (ICDM), pp. 441–449 (2016)
9. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: SIAM International Conference on Data Mining (ICDM), pp. 1–11 (2001)
10. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 497–506 (2009)

11. Mathioudakis, M., Koudas, N.: TwitterMonitor: trend detection over the twitter stream. In: ACM SIGMOD International Conference on Management of Data (ICMD), pp. 1155–1158 (2010)
12. Miao, Z., et al.: Cost-effective online trending topic detection and popularity prediction in microblogging. ACM Trans. Inf. Syst. (TOIS) **35**(3), 1–36 (2016). Article no. 18
13. Rozenshtein, P., Anagnostopoulos, A., Gionis, A., Tatti, N.: Event detection in activity networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 1176–1185. ACM (2014)
14. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: ACM International Conference on World Wide Web (WWW), pp. 851–860 (2010)
15. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. **26**(1), 43–49 (1978)
16. Schubert, E., Weiler, M., Kriegel, H.P.: SigniTrend: scalable detection of emerging topics in textual streams by hashed significance thresholds. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 871–880 (2014)
17. Shang, C., Panangadan, A., Prasanna, V.K.: Event extraction from unstructured text data. In: International Conference on Database and Expert Systems Applications (DEXA), pp. 543–557 (2015)
18. Struzik, Z.R., Siebes, A.: The Haar wavelet transform in the time series similarity paradigm. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 12–22. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-540-48247-5_2
19. Tang, Y., Ma, P., Kong, B., Ji, W., Gao, X., Peng, X.: ESAP: a novel approach for cross-platform event dissemination trend analysis between social network and search engine. In: Cellary, W., Mokbel, M.F., Wang, J., Wang, H., Zhou, R., Zhang, Y. (eds.) WISE 2016. LNCS, vol. 10041, pp. 489–504. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48740-3_36
20. Tolomei, G., Orlando, S., Ceccarelli, D., Lucchese, C.: Twitter anticipates bursts of requests for Wikipedia articles. In: ACM Workshop on Data-Driven User Behavioral Modelling and Mining from Social Media (DUBMOD), pp. 5–8 (2013)
21. Wang, S., Yan, Z., Hu, X., Philip, S.Y., Li, Z., Wang, B.: CPB: a classification-based approach for burst time prediction in cascades. Knowl. Inf. Syst. (KIS) **49**(1), 243–271 (2016)
22. Wang, S., Kam, K., Xiao, C., Bowen, S., Chaovalitwongse, W.A.: An efficient time series subsequence pattern mining and prediction framework with an application to respiratory motion prediction. In: AAAI Conference on Artificial Intelligence (AAAI) (2016)
23. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: ACM International Conference on Web Search and Data Mining (WSDM), pp. 177–186 (2011)
24. Zheng, L., Jin, P., Zhao, J., Yue, L.: A fine-grained approach for extracting events on microblogs. In: Decker, H., Lhotská, L., Link, S., Spies, M., Wagner, R.R. (eds.) DEXA 2014. LNCS, vol. 8644, pp. 275–283. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10073-9_22