# Program Popularity Prediction Approach for Internet TV Based on Trend Detecting

Chengang Zhu[1,2], Guang Cheng[1,2(✉)], and Kun Wang[3]

[1] School of Computer Science and Engineering, Southeast University, Dhaka, Bangladesh
{cgzhu,gcheng}@njnet.edu.cn
[2] Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Dhaka, Bangladesh
[3] School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China
kwang@njupt.edu.cn

**Abstract.** Predicting program popularity precisely and timely is of great value for content providers, advertisers, as well as Internet TV operators. Existing prediction methods usually need large quantity of samples and long training time, while the prediction accuracy is poor for the programs that experience a high peak or sharp decrease in popularity. This paper presents our improved prediction approach based on trend detecting. First, we apply a dynamic time warping (DTW) distance based k-medoids algorithm to group programs popularity evolution into 4 trends. Then, 4 trend-specific prediction models are built separately using random forests (RF) regression. According to the features extracted from electronic program guide (EPG) and early view records, newly published programs are classified into the 4 trends by a gradient boosting decision tree. Finally, combining forecasting values from the trend-specific models and classification probability, our proposed approach achieves better prediction results. The experimental results show that, compared to the existing prediction models, the prediction accuracy can increase more than 20%, and the forecasting period can be effectively shortened.

**Keywords:** Internet TV · Popularity prediction · Dynamic time warping
Random forests regression · Gradient boosting decision tree

## 1 Introduction

With the maturity and popularity of high-definition (HD) and 3D technology, IP video traffic will be a major part of all consumer Internet traffic. According to data published by Cisco Visual Networking Index at July 2016 [1], Internet video to TV will continue to grow at a rapid pace, which will be 26% of consumer Internet video traffic by 2020. Nevertheless, users' attention is not uniformly distributed among all programs. Only a few of programs can attract massive user's attention, and the rest will be left without anybody to care for it. Take Tencent video for example [2], the cumulative requests for the top 50 programs is 45 billion, which is more than 80% of the total requests.


Y. Chen and T. Q. Duong (Eds.): INISCOM 2017, LNICST 221, pp. 142–154, 2018.
https://doi.org/10.1007/978-3-319-74176-5_14

In this context, it is of great importance to predict the popularity of Internet TV programs. First, using the program popularity prediction results, audience will save much time to find more valuable TV programs from mass video resource. Secondly, based on program popularity forecasting data, commercial company is able to maximize advertising effect by choosing the TV programs with highest potential. Finally, with the help of popularity prediction model, the Internet TV operator can optimize configuration of network in advance by deploying enough transmission and storage resource to distribution hot programs.

However, accurately predicting the popularity of Internet TV programs is a quite challenging task. First of all, there are a lot factors to influence TV program popularity which is hard to measure, such as the quality of the program and the interests of audience. Then the relationship between hot events in real world and TV programs cannot be easily introduced in the prediction model. Last but not least, there is a massive gap between the popularity evolution of different programs, which should be considered when designing the prediction model. In this paper, cooperating with an Internet TV operator, we analyze massive user behavior data and present our improved method to predict the popularity for Internet TV programs. The main contribution of our work is as follows:

1. We apply a dynamic time warping (DTW) distance based k-medoids algorithm to group programs with similar popularity into 4 evolution trends, which has the ability to capture the inherent heterogeneity of program popularity. It's computationally more efficient than previous methods used to delineate popularity evolution trends, such as K-Spectral Clustering (KSC) models [3]. Computation in these models is always extensive due to model training and the transformation of features in semantic spaces. By contrast, the DTW distance based k-medoids algorithm is directly driven by raw data. Our method can be implemented without much human intervention, and has a much lower computational cost.
2. We build trend-specific prediction models using random forests (RF) regression, which have an overall higher predictive performance than a single model trained from the entire data set. The popularity prediction model trained separately from different popularity trends data sets can focus on particular types of programs, reduce the effects of noise. To our knowledge, we are the first to tackle the inherent challenges of predicting Internet TV program popularity combining forecasting values from the trend-specific models and classification probability.

The proposed method is evaluated with the data collected from Jiangsu Cloud-media TV, one of the largest Internet TV platforms in China. The rest of this paper is organized as follows. Section 2 discusses related work, whereas Sect. 3 formally presents our new Internet TV program popularity prediction model. Our evaluation methodology and main results are discussed in Sect. 4. Section 5 concludes this paper.

## 2 Related Works

Due to the rich variety and strong timeliness, semantic understanding of Internet TV programs is more difficult than that of news, microblogging and other web content. An ideal prediction model for Internet TV programs not only needs high prediction accuracy, but also needs good calculation performance, which means prediction result should be given before audience interest fades. At present, there is little research on program popularity prediction of Internet TV. The existing popularity prediction methods are for other media forms, but can be used as a reference. Commonly used web content popularity prediction methods include cumulative growth, temporal analysis and evolution trends.

**Cumulative growth.** Some researchers studied the cumulative growth of attention, such as the amount of attention that one item received from the moment it was published until the prediction moment. Kaltenbrunner et al. [4] proposed that depending on the time of the publication, news stories followed a constant growth. A log-linear model was proposed by Szabo and Huberman [5], outperformed the constant growth models in terms of mean squared error (MSE). A different approach was proposed by Lee et al. [6]. They used survival analysis model to detect the threads that would receive more than 100 comments in MySpace with 80% accuracy. Tatar et al. [7] used a simple linear regression based on the early number of comments to predict the final number of comments for news articles. Kim et al. [8] used a linear model on a logarithmic scale to predict popularity ranges for political blog posts. Predicting the popularity of web content, based on the aggregate user behavior, has also been addressed as a classification problem. Jamali and Rangwala [9] trained different classification methods to predict the popularity class of a Digg story with an accuracy of 80%. Wang et al. [10] proposed a local data processing architecture to solve the problem of rapid analysis and processing of massive data.

**Temporal analysis.** Some researchers performed a temporal analysis of how content popularity evolved over time until the prediction moment. Pinto et al. [11] relied on a multivariate linear regression model to predict the popularity of YouTube videos. Maass et al. [12] built a large recurrent neural network that could consider more complex interactions between early and late popularity values. Gürsun et al. [13] observed that the daily number of views could be modeled through a time series prediction model using Autoregressive Moving Average (ARMA). Wang et al. [14] presented a novel model for enhancing classification precision and reducing network overload

**Evolution trends.** Other researchers used clustering methods to find web items with similar popularity evolution trends. Crane and Sornette [15] observed that a Poisson process could describe the attention around the majority of videos and the remaining ones followed three popularity evolution trends. Ahmed et al. [16] proposed a model that used a more granular description of the temporal evolution of content popularity, which showed a significant improvement over the log-linear model. Wang et al. [17] proposed a context-aware system architecture to realize accurate detection of the leak points in large-scale petrochemical plants.

Most of the previous studies focus on building a general model to predict popularity of certain content in some media, but neglect the fact that there is massive gap between contents popularity evolution progresses. That leads those methods are generally ineffective for programs popularity prediction of Internet TV, especially when predicting programs with early peaks and later bursts of accesses. To our knowledge no other work has studied the predictive power of features extracted from electronic program guide. In summary, we are first to detect different popularity evolution trend of Internet TV automatically, and develop an integrated predicting model combining forecasting values from the trend-specific models and classification probability.

# 3 Methodology

## 3.1 Problem Statement

The program popularity prediction problem can be defined as follows. Let $c \in C$ be an individual program from a set of programs $C$ observed during a period $T$. We use $t \in T$ to describe the age of a program (i.e., duration since the time it was published) and mark two important moments: indication time $t_i$, representing the time we perform the prediction and reference time $t_r$, the moment of time when we want to predict program popularity. Let $N_c(t_i)$ be the popularity of $c$ from the time it was published until $t_i$ and $N_c(t_r)$ be the value that we want to predict, i.e., the popularity at a later time $N_c(t_r)$. We define $\hat{N}_c(t_i, t_r)$ at the prediction outcome: the predicted popularity of program $c$ at time $t_r$ using the information available until $t_i$. Thus, the better the prediction, the closer $\hat{N}_c(t_i, t_r)$ is to $N_c(t_r)$.

## 3.2 Method Overview

Our method follows 3 steps. The first step is to detect popularity evolution trends. We calculate the DTW distances between historical-record time series and try to cluster popularity evolution into optimal trends. 11 static features extracted from EPG are introduced to strengthen the result of clustering. A few trials are performed to determine an appropriate value for the number of popularity trends (k) in our case study (Fig. 1).

The second step is to build trend-specific prediction model using RF regression. We spilt the view records into 4 groups according to the above trends, and feed them to the RF regression model together with static features.

The third step is to use gradient boosting decision tree (GBDT) to classify newly published programs popularity into the trends, and get the final prediction results based on the prediction values of 4 models and the probability of classification.

A pictorial representation is shown in Fig. 3. Compared to existing methods, our approach incorporates multiple trend-specific forecasting models, combines the predicting results with probability of each program belonging to each tend, and leads to better results.

## 3.3 Popularity Trend Detecting

In this section, we describe the details of our method for k-medoids [18] clustering of program popularity time series with DTW [19] distance. DTW distance is an accurate similarity measurement which is generally used for time series data. An optimal

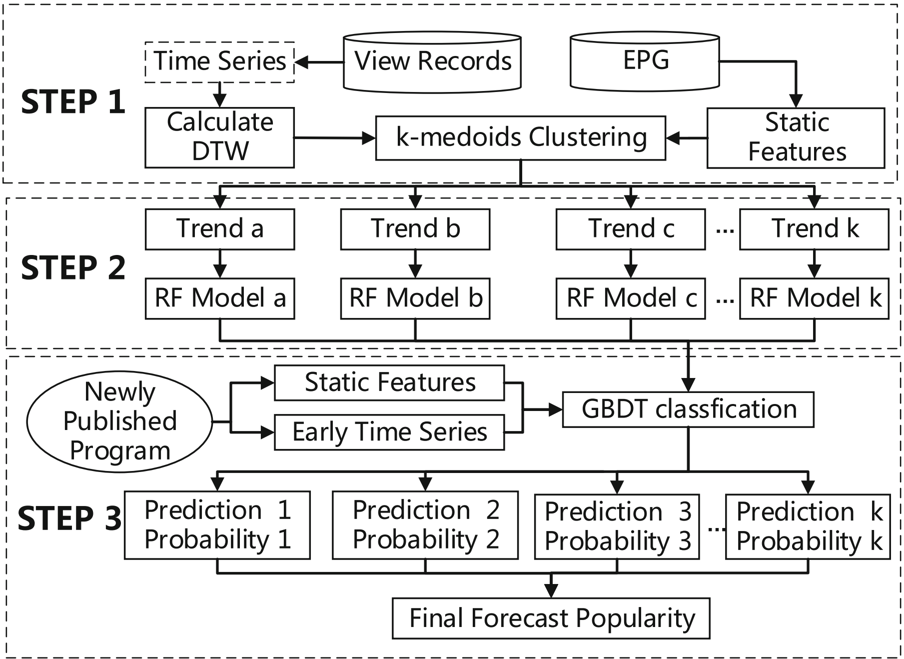


**Fig. 1.** Overview of Internet TV program popularity prediction method

alignment and distance between two sequences $P = (p_1, p_2 \ldots p_n)$ and $Q = (q_1, q_2 \ldots q_m)$ can be determined as follows:

$$DTW(P, Q) = \sqrt{dist(p_n, q_m)} \tag{1}$$

$$dist(p_i, q_j) = (p_i - q_j)^2 + \min\begin{cases} dist(p_{i-1}, q_j) \\ dist(p_i, q_{j-1}) \\ dist(p_{i-1}, q_{j-1}) \end{cases} \tag{2}$$

DTW distance is calculated through dynamic programming to discover the minimum cumulative distance of each element in $n \times m$ matrix. In addition, the warping path between two sequences can be found by tracing back from the last cell. Wang et al. [20] proposed an interests-based reduced variable neighborhood search queue architecture to process large amounts of data. In this work, DTW distance is used to measure the similarity between each program popularity time series data and cluster centers to give more accurate results.

The k-medoids algorithm is similar to the well-known k-means for performing clustering analysis. However, these two methods differ in how they update the center location for a certain cluster. In the k-means approach, the center of a cluster is indeed virtual, because it denotes the mean position of the members that are currently within the cluster. However, the k-medoids method treats the center as the median of the cluster, which thus coincides with one of the actual members. Owing to this difference, k-medoids is more robust in responding to the outliers in the dataset.

The k-medoids based on DTW algorithm is described briefly as in Algorithm 1. First, we arbitrarily choose $k$ programs in $D$ as the initial medoids, and assign each remaining program to the cluster with the nearest medoids. Then we randomly select a non-medoid program to compute a new DTW distance of the trends. If the new DTW distance is less than the previous one after the swapping, we swap to form a new set of $k$ medoids. Above steps are repeated until there is no change of programs in each trend.

**Algorithm** KMDTW($D$,$C$)

```
D: the data set containing program popularity time series
C: the number of trends
K: the set of trend centers
M: the set of popularity sequences in each trend
initialize C as trend centers of K
do
    for i = 1:size(D)
        for k = 1:K
            Dist_{Di,Ck} = DTW(D_i,C_k)
        end for
        if(Dist_{Di,Ck} is min)
            assign D_i into M_k
        end if
    end for
while(the cluster membership changes)
return K,M
```

## 3.4 Trend-Specific Prediction Models

In this section, we describe the details of training trend-specific prediction models using RF regression [21]. RF is an extension of bagging and a competitor to boosting. It uses either categorical (i.e., classification) or continuous (i.e., regression) response variables, and either categorical or continuous predictor variables. In RF modelling, the training parameters that needed specification were: (i) the number of trees to grow in the forest (*n* tree), the number of randomly selected predictor variables at each node (*m* try), and the minimal number of observations at the terminal nodes of the trees (node size). In our study, those were set to 1000, 12, and 5, respectively. The default of n tree was 500, but it has been observed that more stable results for estimating variable importance are achieved with a higher number. The training data that were left out of the bootstrap (i.e., Out-Of-Bag, OOB) samples were used to estimate prediction error and variable importance. In error estimation, the OOB samples were predicted by the respective trees and by aggregating the predictions, the mean square error of OOB was calculated by (3).

$$MSE_{OOB} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_{i_{OOB}}\right)^2 \quad (3)$$

Where $\hat{y}_{i_{OOB}}$ is the OOB prediction for observation $y_i$. Regarding variable importance, the values of a specific predictor variable were randomly permutated in the OOB data of a tree while the values of other predictors remain fixed. The modified OOB data were predicted, and the differences between the MSEs obtained from the permutated and original OOB data gave a measure of variable importance. In our dataset, we will use the attributes of the Internet TV programs, the first-7-day view records and some derivation as the predictor, and the 30th day records as $y$. For each cluster, we train a unique model to fit the dataset.

### 3.5 Classification of Newly Published Programs Popularity

Gradient Boosting Decision Trees (GBDT) [22] is an additive regression model consisting of an ensemble of decision trees. A single decision tree has the problem of over-fitting, however the GBDT algorithm can overcome this by combining hundreds of weak decision trees consisting of a few leaf nodes. GBDT owns a few advantages, including the ability to find non-linear transformations, the ability to handle skewed variables without requiring transformations, computational robustness and high scalability.

In this paper, we build decision trees to classify newly published programs into 4 popularity evolution trends. 11 attributes of program are extracted from electronic program guide, which are described in Table 1.

**Table 1.** Description of attributes extracted from EPG

| Attribute | Description |
|---|---|
| Time | The time when the program is firstly published |
| Name | The formal name or nickname of the program |
| Duration | Time length of the program |
| Actors | The names of main actors and actresses |
| Director | The names of directors for the program |
| Language | The language for dialogue and subtitle |
| Area | The country where the program is filmed |
| Category | The type of the program, such as news, cartoon |
| Rating | The scores given by douban.com |
| Publisher | The program production company name |
| Summary | Brief introduction of the program |

Those attributes and first-7-day view records are predictors and the 4 trends are the targets of classification. With the help of GBDT and RF prediction model, for each program, we get its probabilities $P_{cj}$ of belonging to each trend $k$ and the temporary popularity value $\hat{N}_{ck}(t_i, t_r)$ predicted by corresponding model. To maximize the information gain, we use Eq. 4 to calculate the final predicted popularity of program $c$ at time $t_r$ using the information available until $t_i$.

$$\hat{N}_c(t_i, t_r) = \sum_{k=1}^{4} P_{ck}\hat{N}_{ck}(t_i, t_r) \tag{4}$$

# 4 Experiments

## 4.1 Datasets

The experiment data originates from Jiangsu Cloud-media TV, one of the largest Internet TV platforms in China. The data set is summarized in Table 2. It contains Internet TV requests over 213 days between January 1st and July 31st 2016. During this period 423254 programs were requested. The data set includes more 1.3 million clients making more than 2 billion requests.

**Table 2.** The data sets in figures

| | Requests | Programs | Clients |
|---|---|---|---|
| Daily max | 2300747 | 20447 | 407133 |
| Daily min | 1066706 | 17552 | 225290 |
| Daily median | 1678506 | 19098 | 318052 |
| Total | 201420717 | 423254 | 1309381 |

By cleaning the RTSP (Real Time Streaming Protocol) packets from video server and the analysis of the EPG information, we obtained programs popularity time series and 11 static features for 110 thousand program. The static features include the directors' name, writers' name and actors/actresses' name, country, language, categories, duration, premiere channel, premiere time and content description. These experiment results were computed using 10-fold cross. We split the dataset into 10 folds, where 9 are used as training set and one as test set and rotate the folds such that each fold is used for testing once.

## 4.2 Performance Metrics

A comprehensive and reasonable error analysis can effectively evaluate the performance of the prediction model. Commonly used metrics are divided into the absolute ones, i.e., MSE, RMSE (Root of MSE) Eq. (5) and MAE (Mean Absolute Error), and the relative ones, i.e., MRE (Mean Relative Error) and MRSE (Mean Relative Squared Error). When using the absolute metrics, researchers have to make a clear understanding of the numerical range of the prediction values. Relative metrics are useful to compare the efficiency of prediction algorithm across studies, except for actual values is zero. The quality of a numerical prediction can also be reported using the correlation coefficient or the coefficient of determination ($R^2$) Eq. (6). In order to compare the performance between existing methods to ours and avoid the zero-inflated problem, we choose RMSE and $R^2$ as performance metrics.

$$\mathrm{RMSE} = \sqrt{\frac{1}{|C|}\sum_{c \in C}\left(\hat{N}_c(t_i, t) - N_c(t_r)\right)^2} \tag{5}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{c \in C}\left(N_c(t_r) - \hat{N}_c(t_i, t)\right)^2}{\sum_{c \in C}\left(N_c(t_r) - \bar{N}_c(t_r)\right)^2} \tag{6}$$

## 4.3 Prediction Results

We use the scikit-learn [23], a Python machine learning package, to implement the required clustering and regression algorithms in this study. A few trials are performed to determine an appropriate value for the number of popularity trends (k) in our case study. This is achieved by running the DTW distance based k-medoids method with different values of k (ranging from 2 to 10). Figure 2 shows that as the value of k increases, the DTW distance value between different trends decreases significantly, reaching 8.3 at k = 4. However, the speed of DTW distance value decreasing is much slower when k is within [5, 10]. That means clustering program popularity into more trends than 4 will not improve the accuracy of the predicting model, but degrade the performance. Therefore, we decide to cluster the popularity evolution of Internet TV program into 4 categories.

Figure 3 shows the popularity trends discovered in our dataset. Each graph shows the number of views as function of time. We note that the 4 categories of trends produced by k-medoids based on DTW distance algorithm, are consist with similar shapes identified in other research [3, 11]. Although the 4 trends cannot match popularity evolution progress of all programs, the most prevalent trends are detected which can greatly improve the accuracy of our prediction models.

We measured the RMSE and $R^2$ to evaluate the prediction performance, and compared our prediction method with three other existing method: Szabo-Huberman (S-H) [5], Multivariate Linear [7] and MRBF model [11]. Table 3 shows the MRSE results produced by all 4 models, considering $t_i = 7$ and $t_r = 30$. Result for other values of $t_i$ and $t_r$ are quite similar. The overall MRSE reductions achieved with our model over the others, across all trends, are 20% for the datasets. The grains are especially large for trend c which popularity reaches a high peak and then declines sharply.

Figure 4 shows the coefficient of determination ($R^2$) values produced with different indication time by all 4 models. As the history data accumulates, $R^2$ values are close to 100%. The sooner reliable prediction results are made, the more profitable Internet TV service is. To get 95% $R^2$, other 3 models at least need collect popularity data of 12 days while our model needs only 9 days, which means our model can give a reliable result much earlier.

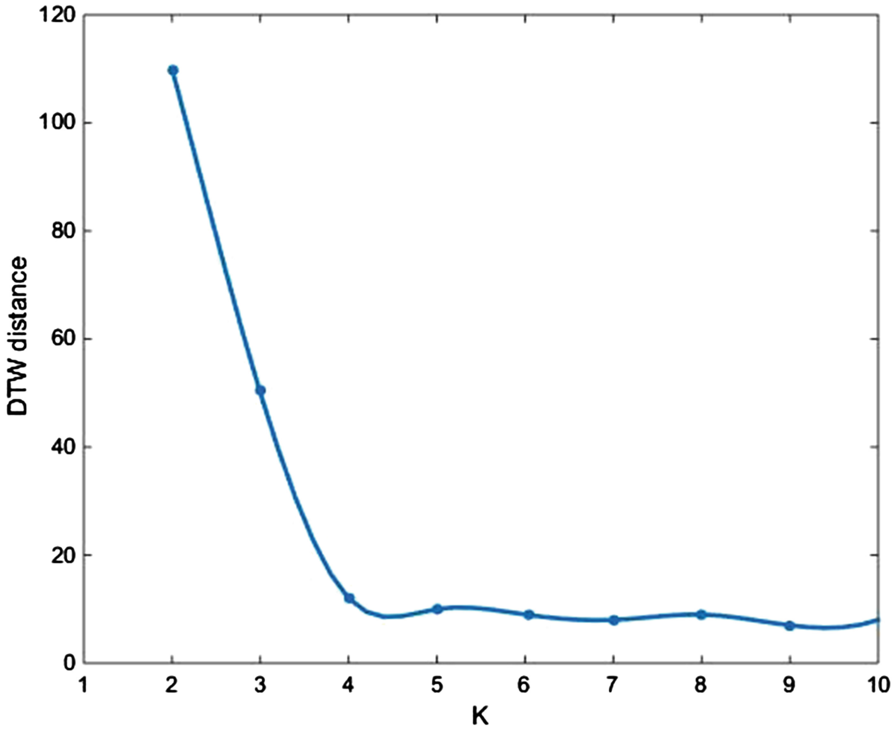


**Fig. 2.** The DTW distance between trends with different values of $k$

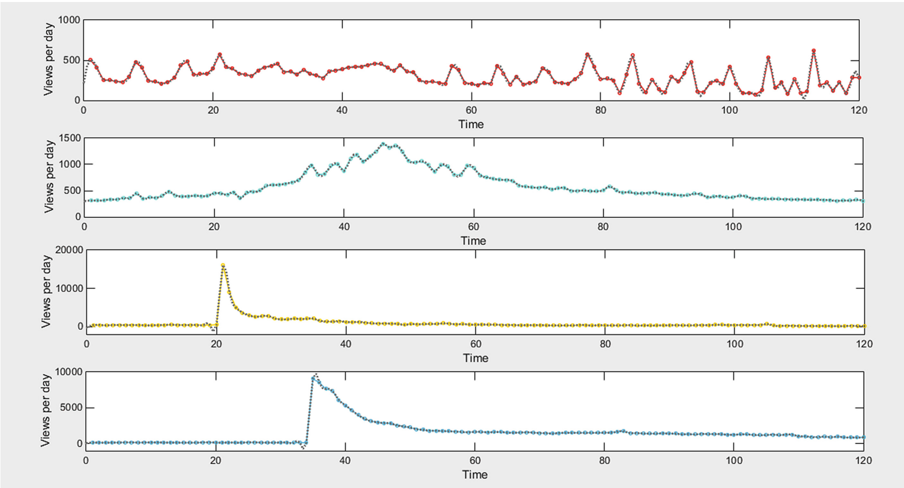


**Fig. 3.** Prevalent popularity trends of Internet TV programs

**Table 3.** Prediction RMSE for S-H, ML, MRBF and our models for programs with different popularity trends.

| Popularity trend | Numbers of programs | S-H | ML | MRBF | Our model |
|---|---|---|---|---|---|
| Trend a | 1554 | 0.4588 ± 0.0283 | 0.1402 ± 0.0274 | 0.1268 ± 0.0319 | 0.1039 ± 0.0263 |
| Trend b | 80528 | 0.2086 ± 0.0040 | 0.1788 ± 0.0041 | 0.1713 ± 0.0040 | 0.1576 ± 0.0040 |
| Trend c | 6636 | 0.1921 ± 0.0124 | 0.1707 ± 0.0283 | 0.1490 ± 0.0199 | 0.1302 ± 0.0172 |
| Trend d | 24143 | 0.3351 ± 0.0099 | 0.2929 ± 0.0120 | 0.2641 ± 0.0111 | 0.2223 ± 0.0104 |
| Overall | 112861 | 0.2382 ± 0.0038 | 0.2022 ± 0.0043 | 0.1892 ± 0.0032 | 0.1677 ± 0.0028 |

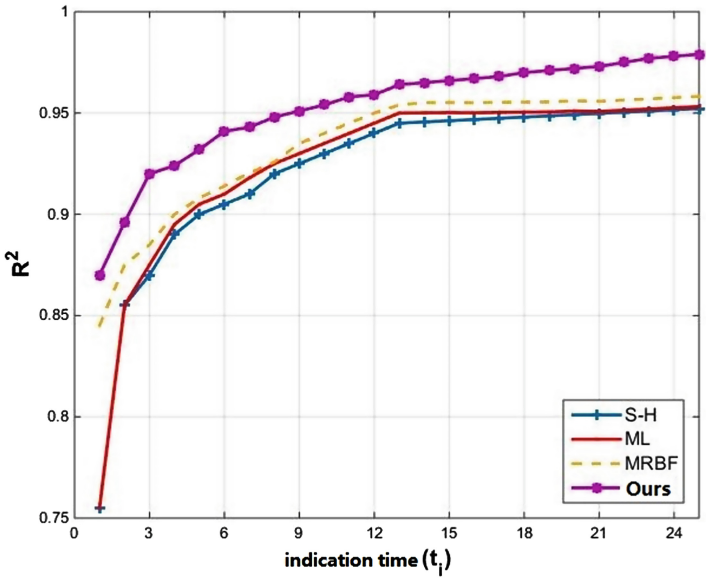


**Fig. 4.** Comparison of coefficient of determination as function of indication time

## 5 Conclusions

In this paper, we have analyzed massive user behavior data and presented our improved method to predict the popularity for Internet TV programs. To the extent of our knowledge, we are first work to tackle the problem of prediction of programs popularity in Internet TV platform. We applied a dynamic time warping (DTW) distance based k-medoids algorithm to group programs with similar popularity into 4 evolution trends, which has the ability to capture the inherent heterogeneity of program popularity. Moreover, we built trend-specific prediction models using random forests regression, which have an overall higher predictive performance than a single model trained from the entire data set.

We performed an extensive experimental evaluation of our method, comparing it with 3 representative methods. Our method outperforms with gain in accuracy of at least 20%, and can give reliable prediction result much faster.

In future, we plan to apply our method to the infrastructure of Internet TV platform, and try to build cache replacement strategy which can proactively adapt to evolution of programs popularity.

## References


1. Cisco Visual Networking Index: Forecast and Methodology. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.htm
2. Tencent Video. http://v.qq.com/rank/detail/2_-1_-1_-1_1_1.html
3. Figueiredo, F., Almeida, J.M., Gonçalves, M.A., Benevenuto, F.: TrendLearner: early prediction of popularity trends of user generated content. Inf. Sci. **349–350**, 172–187 (2016)
4. Kaltenbrunner, A., Gómez, V., López, V.: Description and prediction of slashdot activity. In: 2007 Latin American Web Conference (LA-WEB 2007), pp. 57–66 (2007)
5. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM **53**, 80–88 (2010)
6. Lee, J.G., Moon, S., Salamatian, K.: Modeling and predicting the popularity of online contents with Cox proportional hazard regression model. Neurocomputing **76**, 134–145 (2012)
7. Tatar, A., Leguay, J., Antoniadis, P., Limbourg, A., Amorim, M.D., Fdida, S.: Predicting the popularity of online articles based on user comments. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics, pp. 1–8. ACM, Sogndal (2011)
8. Kim, S.D., Kim, S.H., Cho, H.G.: Predicting the virtual temperature of web-blog articles as a measurement tool for online popularity. In: 2011 IEEE 11th International Conference on Computer and Information Technology, pp. 449–454 (2011)
9. Jamali, S., Rangwala, H.: Digging digg: comment mining, popularity prediction, and social network analysis. In: 2009 International Conference on Web Information Systems and Mining, pp. 32–38 (2009)
10. Wang, K., Shao, Y., Shu, L., Zhu, C., Zhang, Y.: Mobile big data fault-tolerant processing for ehealth networks. IEEE Netw. **30**, 36–42 (2016)
11. Pinto, H., Almeida, J.M., Gon, M.A.: Using early view patterns to predict the popularity of youtube videos. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 365–374. ACM, Rome (2013)
12. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput. **14**, 2531–2560 (2002)
13. Gürsun, G., Crovella, M., Matta, I.: Describing and forecasting video access patterns. In: 2011 Proceedings IEEE INFOCOM, pp. 16–20 (2011)
14. Wang, K., Shao, Y., Shu, L., Han, G., Zhu, C.: LDPA: a local data processing architecture in ambient assisted living communications. IEEE Commun. Mag. **53**, 56–63 (2015)
15. Crane, R., Sornette, D.: Robust dynamic classes revealed by measuring the response function of a social system. Proc. Nat. Acad. Sci. **105**, 15649–15653 (2008)
16. Ahmed, M., Spagna, S., Huici, F., Niccolini, S.: A peek into the future: predicting the evolution of popularity in user generated content. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 607–616. ACM, Rome (2013)

17. Wang, K., Lu, H., Shu, L., Rodrigues, J.J.P.C.: A context-aware system architecture for leak point detection in the large-scale petrochemical industry. IEEE Commun. Mag. **52**, 62–69 (2014)
18. Sun, W., Xiang, L., Liu, X., Zhao, D.: An improved k-medoids clustering algorithm based on a grid cell graph realized by the P system. In: Zu, Q., Hu, B. (eds.) HCC 2016. LNCS, vol. 9567, pp. 365–374. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31854-7_33
19. Mueen, A., Keogh, E.: Extracting optimal performance from dynamic time warping. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2129–2130. ACM, San Francisco (2016)
20. Wang, K., Mi, J., Xu, C., Zhu, Q., Shu, L., Deng, D.J.: Real-time load reduction in multimedia big data for mobile internet. ACM Trans. Multimedia Comput. Commun. Appl. **12**, 1–20 (2016)
21. Hannan, M.A., Ali, J.A., Mohamed, A., Uddin, M.N.: A random forests regression based space vector PWM inverter controller for the induction motor drive. IEEE Trans. Ind. Electron. **64**, 2689–2699 (2017)
22. Wang, Y., Feng, D., Li, D., Chen, X., Zhao, Y., Niu, X.: A mobile recommendation system based on logistic regression and Gradient Boosting Decision Trees. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1896–1902 (2016)
23. Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., Mueller, A.: Scikit-learn: machine learning without learning the machinery. GetMobile Mob. Comput. Commun. **19**, 29–33 (2015)