Probabilistic Seeking Prediction in P2P VoD Systems

Weiwei Wang, Tianyin Xu, Yang Gao, and Sanglu Lu

State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, PRC
ww.wang.cs@gmail.com

Abstract. In P2P VoD streaming systems, user behavior modeling is critical to help optimise user experience as well as system throughput. However, it still remains a challenging task due to the dynamic characteristics of user viewing behavior. In this paper, we consider the problem of user seeking prediction which is to predict the user's next seeking position so that the system can proactively make response. We present a novel method for solving this problem. In our method, frequent sequential patterns mining is first performed to extract abstract states which are not overlapped and cover the whole video file altogether. After mapping the raw training dataset to state transitions according to the abstract states, we use a simpel probabilistic contingency table to build the prediction model. We design an experiment on the synthetic P2P VoD dataset. The results demonstrate the effectiveness of our method.

Keywords: User seeking prediction, State abstraction, Contingency table, P2P VoD systems, User behavior modeling, PrefixSpan.

1 Introduction

With the proliferation of emerging applications, including Internet TV, online video, and distance education, media streaming service over the Internet has become immensely popular and generated a large percentage of today's Internet traffic. Peer-to-peer (P2P) technology has been proved as a successful solution which can effectively alleviate server workload, save server bandwidth consumed and thus bring high system resilience and scalability[1,2,3,4,5]. In P2P media streaming systems, the users do not need to download the complete video files before playback which introduces long startup delay. Instead, "play-as-download" streaming service is provided to let the users watch videos while downloading. P2P live streaming, a typical media streaming service designed for all peers receiving streamed video at the same playback position, has been widely deployed to provide "play-as-download" service for a large number of users[4,5,6]. However, P2P video-on-demand (VoD) streaming is more difficult to design and deploy than P2P live streaming. Unlike live streaming, VoD systems allow users' interactive behaviors, i.e., users can seek forward or backward freely when watching video streams. If not handled properly, such seeking requests may lead to long response latency, which severely deteriorates users' viewing quality, e.g., playback freezing or even blackout.

To improve user viewing experience, user behavior understanding is critical. If a VoD system could detect or predict user seeking patterns, it could proactively make response to them in order to optimise media content delivery[7]. On the server side, the media

A. Nicholson and X. Li (Eds.): AI 2009, LNAI 5866, pp. 676–685, 2009.

server could use spare bandwidth to push out the appropriate media contents to a user before being requested. On the client side, a peer could prefetch media contents that are likely to be requested by upcoming seeking events. This can effectively reduce the response latency and maximize system throughput[8].

User behavior modeling has been already studied for a few years. Some researchers have studied single genres like sports videos[7] and education videos[9] while others have studied a range of video types[10,11]. Brampton et al.[7] analyzed the user interactivity characteristics for sports VoD systems and derived some statistical distributions for user behavior which we employ to generate synthetic P2P VoD dataset for experiments as currently the original viewer logs are not available to us. Both He et al.[12] and Huang et al.[13] performed association rule mining to learn user seeking patterns used to do prediction. Zheng et al.[14] analyzed the statistical pattern hidden in the VoD dataset and applied the optimal quantization theory to learn user seeking behavior. Vilas et al.[11] proposed a user behavior model on the observed dataset but no evaluation or utilization of that model was presented. A survey paper on probabilistic human behavior prediction models by Albrecht and Zukerman can be found in [15], which is a great material to get a thorough understanding of probabilistic approaches on human behavior modeling. However, user behavior modeling still remains a challenging task due to the dynamic characteristics of the viewing behavior which is always changing over time.

In this paper, we employ a simple probabilistic contingency table to solve the problem of user seeking prediction which is to predict the user's next seeking position so that the P2P VoD system can proactively make response. In the design of our method, frequent sequential patterns mining[16] is first performed to extract abstract states which are not overlapped and cover the whole video file altogether. After mapping the raw training dataset to state transitions according to the abstract states, we simple count the number of each seeking operation to build a transition model. We evaluate the prediction model on a synthetic P2P VoD dataset containing 4000 user viewing logs. The results demonstrate the effectiveness of our proposed method.

The rest of this paper is organized as follows. In Section 2, we state the user seeking prediction problem in P2P VoD systems. Then our method is presented in Section 3. Next in Section 4, our method is validated by the experiments on synthetic P2P VoD dataset. Section 5 concludes and discusses some future work.

2 Problem Statement

In this section, we present some related terminology and define the problem of user seeking prediction.

Terminology. Since the most basic user activity is the continuous viewing of a section of a video, a peer maintains such basic activity in a *user viewing record* (UVR) in playback time. The important parts of an UVR format is shown as follows.

(UID, MID, Start Position, Inter-Seek Duration, Jump Position)

where UID refers to the user's identifier while MID refers to the movie's identifier. The *inter-seek duration* is described as the number of segments contained in the section of

the video the user watched before seeking to a new position. The *start position* points to the first watched segment in the current inter-seek duration while the *jump position* points to the first segment in the next inter-seek duration.

In most cases, as soon as a user finishes recording one UVR, a new UVR is initialized to record the next inter-seek duration. A sequence of UVRs forms a *user viewing log* which represents a complete user viewing history called a *session*. For example, $\{(U1, V1, 1, 6, 73), (U1, V1, 73, 3, 4), (U1, V1, 4, 3, \text{End})\}$ depicts a session as follows: a user $U1$ first views the video $V1$ from the 1-st segment to the 6-th segment, then seeks forward to the 73-rd segment and views until the 75-th segment, and finally seeks backward to the 4-th segment, re-views for 3 segments and finishes playback.

User viewing logs can also be represented in *sequence format* as $\{s_0, s_1, \cdots, s_i, \cdots, s_{n-1}\}$, where s_i denotes that the user has viewed the s_i-th segment of the video. In this example, the corresponding sequence format is $\{1, 2, 3, 4, 5, 6, 73, 74, 75, 4, 5, 6\}$. Notice that the UVR format can be easily transferred into the sequential format which is used for mining frequent patterns in the state abstraction stage.

Problem Statement. Given a database of user viewing logs, the problem of *user seeking prediction* is to predict the next seeking position according to the user's viewing history in the current session.

Still use the example mentioned above. Given large volumes of user viewing logs of movie $V1$ and suppose the viewing history of user $U1$'s current session to movie $V1$ is $\{1, 2, 3, 4, 5\}$, we hope to predict $U1$'s next seeking position 73 and pre-fetch it in advance. As a result, when $U1$ finishes viewing segment 6 and requests to seek to segment 73, the client-side software can directly satisfy $U1$ with little response latency.

3 Learning User Seeking Behavior

To learn the seeking behavior, we first do frequent sequential pattern mining on the collected P2P VoD dataset and split the patterns into abstract states. Then we map the raw data to state transitions according to the abstract states. Finally, a prediction model is built using a simple contingency table.

3.1 State Abstraction

In typical P2P streaming systems, a video stream is divided into segments of uniform length and each segment contains 1 second video content [6,17]. Typical video stream on the Internet such as movies and sports videos take more than 1.5 hours (5400 seconds/segments) long. If we simply use "segment" as the unit to do learning, it would generate too many fine grained intermediates that bring difficulties for learning the prediction model. As a result, state abstraction is essential. In this problem, we extract the strongly associated segments into *abstract states* by distilling large volumes of user logs and then maps the raw user viewing logs into *state transitions*.

Mine Frequent Sequential Patterns. According to the measurements of real deployed media streaming systems [7,18,14], there are always some popular segments called

highlights which attracts far more viewing times than other segments. This indicates that users are much willing to watch some interesting scenes while skip boring scenes. Fig.1 is the segment popularity statistics in [7], which is collected from a real deployed 8200-second sports video, a football match between Argentina vs. Serbia and Montenegro in World Cup 2006. We can see that the match has about 10 highlights, either of which is a kick-off or a goal. Fig.2 is the segment popularity of the synthetic P2P VoD dataset generated according to the statistic distributions in [7]. We will explain in detail the generation process in section 4.1.

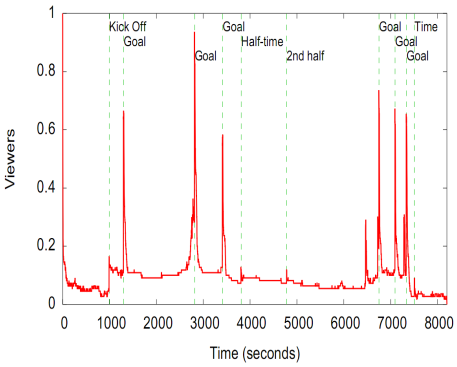

Fig. 1. Segment popularity of the dataset in [7]

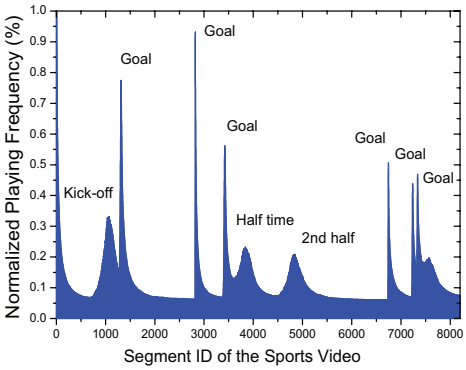

Fig. 2. Segment popularity of synthetic dataset

Directly cutting the 8200 seconds into equally length time series would not work well as it does not take the viewers' watching patterns into consideration. For example, if most viewers watched from 1000 to 1300 as a goal happened in that period, we should try to aggregate these seconds as a state or some contiguous states. From this point of view, we employ the frequent sequential pattern mining method ***Prefix**-projected **S**equential **pa**ttern* *mining (PrefixSpan)*[16] to mining frequent patterns from the dataset. The general ideal of *PrefixSpan* is to examine the prefix subsequences and project their corresponding postfix subsequences into projected databases. In each projected database, sequential patterns are grown by exploring only local frequent patterns. This method is considerably fast than the Apriori-based algorithms and ***Fre**quent pattern projected **S**equential **pa**ttern* *mining (FreeSpan)*[19]. We do some modification to the *PrefixSpan* method as we aim at finding the contiguous sequential patterns. For example, the original *PrefixSpan* will find patterns like $\langle 1, 2, 4, 5, 6, 7, 10\rangle$ which are not allowed in our result. We need patterns like $\langle 4, 5, 6, 7\rangle$ which are not only sequential but also contiguous. For this reason, we modified the *PrefixSpan* to generate only frequent sequential and contiguous patterns. A procedural form of *PrefixSpan* is given in Algorithm 1. We follow the code by Yasuo Tabei[20] in our implementation.

Split Sequential Patterns into States. As the patterns found are largely overlapped, e.g., $\langle 1, 2, 3, 4, 5, 6, 7\rangle$ and $\langle 5, 6, 7, 8, 9, 10, 11, 12\rangle$ may both exist in the mining result,

Input : A sequence database S, and the minimum support threshold min_sup
Output : The complete set of sequential patterns
Method : Call PrefixSpan($\emptyset$, 0, S)
Subroutine : PrefixSpan($\alpha, l, S|_\alpha$)
Parameters: α: asequential pattern; l: the length of α; $S|_\alpha$: the $\alpha -$ projected database, if $\alpha \neq \emptyset$; otherwise, the sequence database S

```
1 call S|α once, find the set of frequent items b such that: b can be assembled to the last
  element of α to form a sequential pattern; or ⟨b⟩ can be appended to α to form a
  sequential pattern;
2 foreach frequent item b and if b is contiguous after α do
3     Append it to α to form a sequential pattern α', and output α';
4 end
5 foreach α' do
6     construct α' − projected database S|α';
7     call PrefixSpan(α',l + 1,S|α');
8 end
```

Algorithm 1. Modified *PrefixSpan* for mining frequent and contiguous time series in P2P VoD systems

among which $\langle 5, 6, 7\rangle$ is the overlapping part. We need to split the patterns into intervals of different length which are not overlapped and remain contiguous. We design a simple splitting algorithm which scans over the sequential patterns and cuts them into intervals without overlapping, e.g., $\langle 1, 2, 3, 4, 5, 6, 7\rangle$ and $\langle 5, 6, 7, 8, 9, 10, 11, 12\rangle$ will be cut into intervals $[1, 7]$ and $[8, 12]$. For the intervals which do not exist in the mined sequential patterns, we take each of them as a separate interval. After that, we split the contiguous intervals into appropriate granularity intervals, called *states*, in order to fit the pre-fetching buffer. A too large interval makes no sense for pre-fetching because the buffer is size-limited. Here, we set two tunable parameters *min-state-len* and *max-state-len* which are the minimum and maximum state length allowed according to the condition of the client-side peer. The *min-state-len* avoids splitting an isolated segment as a state while the *max-state-len* is set as the pre-fetching buffer size in our experiment. Thus, the whole video stream can be represented by these abstract states.

Map Raw Dataset into State Transitions. With the abstract states generated from the above step, we can easily map user viewing logs into state transitions. The mapped results are in the following form:

$$\langle s,\ \ s'\rangle$$

where s is the state the current playback position is in while s' means the next state the viewer will seek into. For a single inter-seek duration, several contiguous state transitions may be generated as the duration may be very large for a single state.

3.2 Probabilistic Seeking Model Building

We assume the user seek operation satisfies the Markov property, that is, the next seek position is dependent on the current position and independent on the previous positions

before the current position. With this assumption, we employ a simple contingency table to build a prediction model for predicting user behavior, that is the seeking operation.

Model Building using State Transitions. For the prediction task in this paper, we use a simple contingency table to represent the transition probability. The table is shown in Fig.3, in which s represents the current state in the mapped trainning data and s' represents the next state. By simply counting the number of seeking operation of each transition pair $\langle s, s' \rangle$, we can build this simple model in an efficient and incremental way.

s'	s	P(s'|s)
s_0	s_0	P_{00}
s_1	s_0	P_{01}
...	...	...
s_0	s_1	P_{10}
s_1	s_1	P_{11}
...	...	...

Fig. 3. A simple probabilistic contingency table for predicting user seeking behavior

After the training process, a model is built and can be used to do predictions. Given current state s, we can infer $P(s'|s)$ from the learnt transition table. According to this distribution, we can predict the next seeking, e.g. we can employ roulette wheel section or softmax selection. In our approach, we simply apply roulette wheel section strategy.

4 Performance Evaluation

In this section, we evaluate our method on the user seeking prediction problem. The data used here is the state transitions generated in the above steps.

4.1 Data Generation

In the experiment, we generated a synthetic P2P VoD dataset of user viewing logs according to the statistical distribution in [7]. The chosen video is the 8200-second football match described in Section 3.1. In [7], the segment popularity, the session length as well as the inter-seek duration follows some probability distribution, see Table 1.

For the generation, we modified the GISMO streaming generator [10] to produce 4000 user viewing logs in UVR format. We set most parameters of GISMO generator to the values in Table 1. Moreover, we modified the jump sub-routine in GISMO using a log-normal distribution to let users trend to jump around highlights. The segment popularity of the synthetic dataset is shown in Fig.2 which is similar with the real popularity statistics in [7]. So we believe our dataset can reflects the user behavior well.

Table 1. Metrics with their corresponding distribution used in data generation

Metric	Distribution	R-Square
Segment popularity	Log-normal, μ=0.016, σ=1.35	0.0941
Session length	Log-normal, μ=4.835, σ=1.704	0.127
Inter-seek duration	Log-normal, μ=1.4796, σ=2.2893	0.0358

4.2 Problem Analysis

The average number of seeking behavior is about 7 times in a whole session[14][7]. Most of the state transitions, $\langle s, s' \rangle$, are contiguous, that is, $s' = s + 1$ as shown in Table 2. These state transitions are useless for our prediction as the playback buffer has already done this job. For this reason, we skip all the contiguous state transitions in the training data, i.e., state transitions like $\langle s, s + 1 \rangle$ will be just skipped.

Table 2. Statistical data of the user behavior prediction problem

Description	Statistics
Forward seek ratio	99.28%
One step forward seek ratio	80.54%
Backward seek ratio	0.72%
One step backward seek ratio	31.9%

In our experiment, the streaming rate S of the video is set as 256 Kbps (most video stream over the Internet today is encoded in the 200-400 Kpbs range [2]). The total available downloading bandwidth of each peer is randomly distributed in $[1.5S, 5S]$. The length of the client-side buffer is 30Mbytes which can be easily accommodated in state-of-art personal computers, i.e., each peer can cache 120 segments. The client-side buffer is split into two parts: the playback buffer with 25Mbytes and the pre-fetching buffer with 5Mbytes. In each time slot, Peers download urgent segments in playback buffer in high priority of using bandwidth to guarantee continuous normal playback. If there is still residual bandwidth, peers pre-fetch the segments in the predictive states into the pre-fetching buffer for supporting user seeking behavior. Both of the two parts use LRU (Least Recently Used) as a default buffer replacement policy.

4.3 Experimental Results

We use our data generation method to produce 4000 user viewing logs. The threshold value used in *PrefixSpan* is $1/10$ of the population, that is 400, and *min-state-len* is set to 5 while *max-state-len* is 20. After preprocessing, we split the whole data into a training dataset and a validation dataset with a split ratio of 0.7. We run all the experiments 10 times and average the results. Fig.4 shows the learnt user behavior model, in which each color represent a transition probability $P(s'|s)$ for a specific state s. We can see that our learnt transition model seems very similar to the segment popularity in Fig.2,

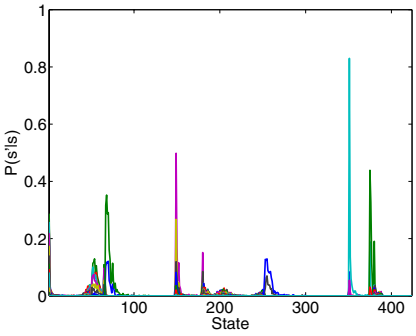

Fig. 4. Learnt user behavior model

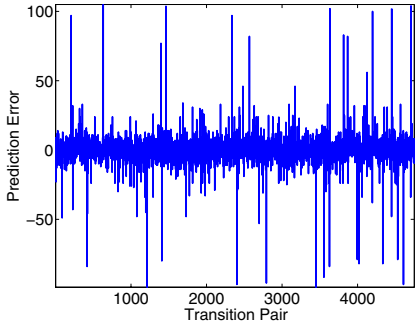

Fig. 5. Prediction curve

which should not be surprising as we can expect users are more willing to seeking to the highlights and this also validates our method. The prediction error curve of the validation set is shown in Fig.5, in which, the x-coordinate is the transition pair and the y-coordinate represents the prediction error between the predictive next state and the actual next state in the validation data.

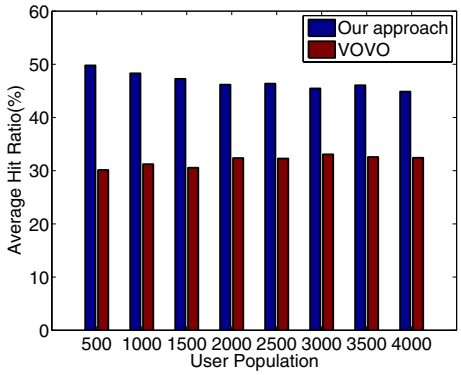

Fig. 6. Comparison of VOVO and our approach

We apply our learnt model to do prediction-based per-fetching and evaluate the pre-fetching performance in terms of hit ratio which is calculated using (1). When a user imposes a seeking request on a peer, the peer checks its local buffer (both the playback buffer and the pre-fetch buffer). If its local buffer contains segments in the requested state, the seeking is considered as a hit event and the peer can continue playback without jitter. Otherwise, it is a miss event and the peer must try to search and download the requested segment from other peers, which leads to long latency.

$$Average\ hit\ ratio = \frac{Total\ number\ of\ hit\ events}{Total\ number\ of\ seeking\ requests} \quad . \tag{1}$$

The system settings are described in Sect. 4.2. When entering the system, each peer is assigned a unique user viewing log and thus its playback procedure is determined. We further assume all the segments are available for all the peers. Each peer just predicts the next seeking state and sequentially pre-fetches segments in the predictive state into its pre-fetch buffer according to its download bandwidth. The results shows that the average hit ratio is about 45% as shown in Fig. 6, more accurate than VOVO approach proposed in [12]. In our approach, the minimum support threshold is $1/10$ of each different population. Moreover, for fair comparison, the prediction range of VOVO is set to 20 segments, which equals to the *max-state-len* in our approach, i.e., as long as the VOVO's predictive result is within the interval $[R - 10, R + 10]$ (R is the real seeking segment), we consider it as a hit event. Notice that we do not consider any collaboration between peers in this paper. Thus, we can prospect a much higher hit ratio with the help of peer collaboration from which peers can exchange their buffer contents with neighbors and are more likely to find appropriate contents. Consequently, the results demonstrate the effectiveness of our approach.

5 Conclusion and Future Work

In this paper, we propose a new method on the user seeking prediction problem and get good results. In data preprocessing, we extract abstract states from the raw user viewing logs through frequent sequential pattern mining. Then we employ a simple contingency table to build a state transition model. State abstraction as a step of preprocessing plays an important role in our solution. Furthermore, the learnt user seeking model can be used to do pre-fetching suggestions, that is we can mark the highlights beside the video and offer suggestions for pre-fetching before the occurrence of seeking operations.

However, the accuracy is still not very satisfactory and much improvement could be done in the future research. We intend to introduce time series analysis approach into this problem to release our Markov property assumption of the state transitions. Besides, model transfer or transfer learning is also a very important research for our future work so as to use the current available learnt model to build new model for new videos instead of starting from scratch. Finally, our code and synthetic dataset are publicly available at my homepage `http://cs.nju.edu.cn/rl/people/weiweiwang` to all researchers who are interested in this novel problem.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant No. 60775046, 90718031, 60721002 and 60903025; the National Basic Research Program of China (973) under Grant No. 2009CB320705; the Natural Science Foundation of Jiangsu Province under Grant No. SBK200921645.

Moreover, we would like to thank Yinghuan Shi, Yongyan Cui and Liangdong Shi for their helpful comments.

References

1. Sripanidkulchai, K., Maggs, B., Zhang, H.: The feasible of supporting large-scale live streaming applications with dynamic application end-points. In: Proc. of the ACM SIGCOMM 2004 (August 2004)
2. Huang, C., Li, J., Ross, K.W.: Can internet video-on-demand profitable? In: Proc. of ACM SIGCOMM 2007 (August 2007)
3. Huang, Y., Fu, T.Z.J., Chiu, D.M., Liu, J.C.S., Huang, C.: Challenges, design and analysis of a large-scale p2p-vod system. In: Proc. of ACM SIGCOMM 2008 (August 2008)
4. PPLive: A free p2p internet tv software, `http://www.pplive.com/`(September 2007)
5. Joost: A website to watch videos, music, tv, movies and more over the internet, `http://www.joost.com/` (June 2006)
6. Zhang, X., Liu, J., Li, B., Yum, T.S.P.: CoolStreaming/DoNet: A data-driven overlay network for efficient live media streaming. In: Proc. of IEEE INFOCOM 2005 (March 2005)
7. Brampton, A., MacQuire, A., Rai, I.A., Race, N.J.P., Mathy, L., Fry, M.: Characterising user interactivity for sports video-on-demand. In: Proc. of ACM NOSSDAV 2007 (April 2007)
8. Annapureddy, S., Guha, S., Gkantisdis, C., Gunawardena, D., Rodriguez, P.: Is high-quality vod feasible using p2p swarming? In: Proc. of ACM WWW 2007 (May 2007)
9. Costa, C., Cunha, I., Borges, A., Ramos, C., Rocha, M., Almeida, J., Riberio-Neto, B.: Analyzing client interactivity in streaming media. In: Proc. of ACM WWW 2004 (May 2004)
10. Jin, S., Bestavros, A.: GISMO: A generator of internet streaming media objects and workloads. In: Proc. of ACM SIGMETRICS 2001 (June 2001)
11. Vilas, M., Paneda, X.G., Garcia, R., Melendi, D., Garcia, V.G.: User behaviour analysis of a video-on-demand service with a wide variety of subjects and lengths. In: Proc. of IEEE EUROMICRO-SEAA 2005 (August 2005)
12. He, Y., Liu, Y.: VOVO: VCR-oriented video-on-demand in large-scale peer-to-peer networks. IEEE Transactions on Parallel and Distributed Systems 20(4), 528–539 (2009)
13. Huang, C.M., Hsu, T.H.: A user-aware prefetching mechanism for video streaming. World Wide Web: Internet and Web Information Systems 6(4), 353–374 (2003)
14. Zheng, C., Shen, G., Li, S.: Distributed prefetching scheme for random seek support in peer-to-peer streaming applications. In: Proc. of the ACM P2PMMS 2005 (November 2005)
15. Zukerman, I., Albrecht, D.W.: Predictive statistical models for user modeling. In: User Modeling and User-adapted Interaction, pp. 5–18. Kluwer Academic Publishers, Dordrecht (2001)
16. Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: Mining sequential patterns by pattern-growth: The PrefixSpan approach. IEEE Transactions on Knowledge and Data Engineering 16(10), 1–17 (2004)
17. Xu, T., Chen, J., Li, W., Lu, S., Guo, Y., Hamdi, M.: Supporting VCR-like operations in derivative tree-based P2P streaming systems. In: Proc. of IEEE ICC 2009 (June 2009)
18. Yu, H., Zheng, D., Zhao, B.Y., Zheng, W.: Understanding user behavior in large-scale video-on-demand systems. SIGOPS Oper. Syst. Rev. 40(4), 333–344 (2006)
19. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.C.: Freespan: frequent pattern-projected sequential pattern mining. In: Proc. of the ACM SIGKDD 2000 (August 1999)
20. Tabei, Y.: PrefixSpan: An implementation of prefixspan (prefix-projected sequential pattern mining) (December 2008), `http://www.cb.k.u-tokyo.ac.jp/asailab/tabei/prefixspan/prefixspan.html`