Research Article

Mining the IPTV Channel Change Event Stream to Discover Insight and Detect Ads

Matej Kren, Andrej Kos, and Urban Sedlar

Laboratory for Telecommunications, Faculty of Electrical Engineering, University of Ljubljana, Trzaska cesta 25, SI-1000 Ljubljana, Slovenia

Correspondence should be addressed to Matej Kren; matej.kren@ltfe.org

Received 24 December 2015; Accepted 23 February 2016

Academic Editor: Zoran Obradovic

Copyright © 2016 Matej Kren et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IPTV has been widely deployed throughout the world, bringing significant advantages to users in terms of the channel offering, video on demand, and interactive applications. One aspect that has been often neglected is the ability of precise and unobtrusive telemetry. TV set-top boxes that are deployed in modern IPTV systems can be thought of as capable sensor nodes that collect vast amounts of data, representing both the user activity and the quality of service delivered by the system itself. In this paper we focus on the user-generated events and analyze how the data stream of channel change events received from the entire IPTV network can be mined to obtain insight about the content. We demonstrate that it is possible to predict the occurrence of TV ads with high probability and show that the approach could be extended to model the user behavior and classify the viewership in multiple dimensions.

1. Introduction

Internet Protocol Television (IPTV) has become an essential part of modern triple-play offerings and is being widely deployed worldwide. In addition to its advantages in terms of enhanced and interactive content, and the possibility to include the long-tail content in the provider's offering, such systems also support effortless collection of data at multiple levels in the system [1, 2]. Some examples of this are the network quality metrics, image decoding problems, or viewer engagement information. One of the available data streams in such networks are the channel change events, which can be obtained in a multitude of ways: either reported by the STB diagnostics module, acquired through the IPTV middleware, or captured directly from the network elements of the core or access delivery network. Such data, despite its simplicity, hides a wealth of information about the user habits and about the content itself, since each channel change event is motivated by a combination of the viewer's habits and context.

In this paper we focus on analysis of the channel change data flow [3], obtained in a pseudoanonymized form from the IPTV operator, to try to infer the interestingness of

the broadcast content. Based on the hypothesis that TV ad segments represent undesired content to most viewers, we develop and validate an algorithm to detect ad occurrences based on the number of synchronous channel change events on the TV channel.

Most previous works on ad segment detection [4, 5] have based their strategies on studying the occurrence and patterns of audio silences and black frames as an indicator of ad boundaries. Another interesting approach is presented in [6], where application checks for logo presence in the video stream. Such approaches work reliably but are resource intensive and work with a different underlying set of assumptions (e.g., these ads are always accompanied by audio volume change and synthetic picture). As opposed to the existing mechanisms that primarily detect the content by its inherent features, our proposed approach focuses on the reaction of the viewers to the content. Thus, it is broader in scope and of more subjective nature and can be extended to modeling of TV ratings with fine granulation (individual scenes) in near-real-time by mining the data stream of viewers' channel change events.

Such crowdsourced ad detection can have many practical applications, both from the advertisers' and regulators' point

of view, as well as the viewer's. For instance, broadcasters and ad agencies could adjust the content timing and form to increase engagement. Regulators could use such system as an additional data source to monitor TV stations and whether they obey regulations regarding excessive broadcasting of commercials [6]. And finally, future over-the-top broadcasting systems could leverage a similar mechanism to flag and skip commercials or other undesired contents in video recordings or notify the viewer when the ad segment is over.

2. Materials and Methods

In this section we describe the data and the methods used for ad detection. Our research was conducted on pseudoanonymized channel change event data obtained from an ISP, a dataset of TV ad segment timestamps and durations, and additional synchronization information obtained by manually aligning both datasets to the live video feed.

2.1. Datasets. Two main datasets were used in our research. Firstly, a near-real-time pseudoanonymized event stream of all channel change events in the IPTV network was obtained from the ISP. The source of the data stream was diagnostic SNMP trap messages, sent both periodically and on every channel change event. This provided a continuous and time-varying data stream, ranging from 600 to 5000 events per minute. The data stream contained all channel changes of a set of 10,000 users, randomly sampled by the ISP. Two months' worth of data (from June 1, 2015, to August 1, 2015) was captured and stored into a key-value database (Apache Cassandra). The database scheme was optimized for efficient retrieval of data by time range and used the event timestamp as the key.

Next, a set of TV ad segment start and end times was obtained from the national TV broadcaster for a limited subset of the channels. We limited our analysis to the largest national TV channel, which captures a significant portion (16%) of the national viewership.

2.2. Method of Detection. Since both described datasets originate from two different systems (and two providers), there were no guarantees that the timing in both was in sync; thus, to be able to train our algorithm, the first step in data preparation was the synchronization of both datasets.

2.2.1. Dataset Synchronization. By plotting both datasets against time, a noticeable lag was evident (Figure 1). The figure shows the number of channel changes; the top x -axis represents time in minutes on June 20, 2015, from 23:22 to 23:52. Each minute contains six bars, each representing 10 seconds.

As we only had historical TV ad segment data, we could not check the alignment with TV footage manually. To calculate the offset of the ad segment data we identified the ad segments that repeat at known hours and calculated the offset to the IPTV live broadcast.

To synchronize the pseudoanonymized channel change data stream offset, we had to engineer an unlikely pattern of channel changes: starting early in the morning, when viewership was low, on a specific channel, we made a channel change event every minute on the minute mark, for 20 minutes. Finally, we identified the pattern in the captured dataset and were able to calculate the offset in the event timestamps.

Lastly, we have subtracted both offsets and we found a 70-second lag between the two datasets, which we used to adjust the smaller dataset (ad segment timestamps). Some inaccuracy on a scale of a couple of seconds is possible, but we have assessed that it does not affect the algorithm, which works on a larger time scale.

2.2.2. Data Cleaning and Preparation. The first step in data cleaning was removing all periodically reported events that did not carry channel change information (viewer remained on the same channel). Next, we discarded the events of all the channels, except the largest one, by viewership, which matched the ad segment data. The initial dataset size averaged 2 GB per day (for 60 days, from June 1, 2015, to August 1, 2015), which totals 120 GB. After filtering, the remaining dataset size was approximately 200 MB per day on average, totaling 12 GB of raw data throughout the studied timeframe.

Data preparation was done in the following way: by aggregating the data we obtained time series that represented a number of channel changes at a specific time of each day with 1 second granularity. The resulting time series data served as the first input to our algorithm. The time series is defined as a sequence of pairs $T = [(p_1, t_1), (p_2, t_2), \dots, (p_n, t_n)]$ ($t_1 < t_2 < \dots < t_n$), where each p_i is a data point in d -dimensional space, and each t_i is the timestamp at which p_i occurs [7]. In our case, a data point is a number of channel changes by all users on the investigated channel. Timestamps are 1 second apart, due to the fact that we also injected the timestamps with zero channel changes into the time series.

Both the channel change dataset and the time-adjusted ad segment dataset were split into 3 equal parts: the labeled dataset, the training dataset, and the validation dataset, each spanning 20 days. For each TV ad segment in the labeled dataset we used the start and end timestamps to generate subtitle series from the overall channel time series. We thus obtained several hundred time series representing a number of channel changes per second throughout the ad segment, relative to the start of the commercial. An example of a TV ad segment is represented in Figure 2. This data serves as the second input to the algorithm.

The algorithm depends on the time series comparison and we define similarity function of two time series. Given two time series T_1 and T_2 similarity function Dist calculates distance between the two time series, denoted by $\text{Dist}(T_1, T_2)$ [7]. As we examined the ad segment time series data we discovered that they are similar, but not in the sense of a classical Euclidean or Manhattan distance. For example, let us use sequences $p = [4, 5, 6, 3, 1, 1]$ and $q = [1, 13, 1, 3, 1, 1]$ that represent number of channel changes per second through an ad segment. The L_1 or Manhattan distance is defined

FIGURE 1: Channel changes in time with labeled ad segments. The interval in the chart represents 30 minutes (with each bar representing 10 seconds) on June 20, 2015, from 23:22 to 23:52. The height of the bars of either color represents the number of channel changes by all of the observed viewers on the channel; red color highlights the periods with labeled ads, as provided by the broadcasting company. A significant time lag is noted due to misaligned timestamps, which have to be adjusted before the training phase begins.

FIGURE 2: Example of a TV ad segment (red). The interval in the chart represents 5 minutes (with each bar representing 10 seconds) on June 30, 2015, from 19:52 to 19:56.

as $L_1(p, q) = \sum_i |p_i - q_i| = 16$, which is relatively large, considering that the sum of all elements (channel changes) in sequence p is only 20. Ad segment sequence usually starts with a larger amount of the channel changes at the start of the segment. This is notable in the sequence q with 13 channel changes in the second element. Together in the first three elements of both sequences we have 15 channel changes. As a consequence of this, the similarity measure has to take into account a wider interval for comparison. We define

a new sequence that is a moving sum of the elements in a predefined window length, similar to the moving average of a sequence [8]. The moving sums of length three for previously defined sequences p and q are $p' = [15, 14, 10, 5]$ and $q' = [15, 17, 5, 5]$. If we calculate the Manhattan distance of p' and q' , we get 8, which is significantly less than 16. Our IPTV time series are collected with one-second granularity and we used a 30-second moving average.

2.2.3. Algorithm Training. The last phase builds a logistic regression model to predict whether TV ad was played in that interval or not. Logistic regression is mathematical modeling approach which can be used to describe the relationship of several independent variables to a dichotomous dependent variable, such as TV ad occurrence [9]. Logistic regression model with one independent variable can be expressed as follows:

$$p(y = 1 | x) = \frac{1}{1 + e^{-(b_0 + b_1 X)}}, \quad (1)$$

where x is the independent variable and $p(y = 1 | x)$ is an ad occurrence probability of an interval given the presence of independent variable x .

For building a logistic regression model we use IPTV time series data from the training dataset. We use one-parameter logistic regression with custom dependent variable that is calculated as suggested below.

We loop through IPTV time series in training dataset and for each 10-second interval we calculate distance from IPTV time series to all labeled commercials time series generated

FIGURE 3: ROC curve of the proposed model.

in the data preparation phase. For each 10-second interval we take the minimum distance. For each interval we check if an ad was actually played based on the shifted broadcaster ad segment timestamps. That leads to pairs of numbers (x, y) where x is minimum distance to some commercial time series and y is 0 or 1 depending on if an ad was played at that time or not. These pairs are used to determine the logistic regression model.

3. Testing and Results

The model was tested on the validation dataset, where we used our model to predict the ad segment and validate it against the broadcaster's ad segment data.

The Receiver Operating Characteristic (ROC) curve [10] is a widely used tool whose plot represents the compromise between the true positive and the false positive example classifications based on a continuous output along all its possible decision threshold values (the score). The closer the ROC curve is to the upper left corner (optimum point), the better the decision system is. Model fit can be assessed by the area under a relative operating characteristic curve (AUC—area under curve) procedure. The AUC value is between 0.5 and 1, where 1 indicates a perfect fit and 0.5 indicates a random fit [9]. The AUC value of our model is 0.88 for the current study, which is equivalent to an accuracy of 88%. The ROC curve of our model is plotted in Figure 3.

Results depend on users' viewing habits that change through the time of day and are strongly dependent on the type of the TV content as well. For example, the news broadcasts on the channels that we studied tend to have an ad segment at exactly the same time every day; regular viewers learn to anticipate when the ad segment will start and change the channel preemptively. Similarly, some TV shows have foreshadowing events before the ad segment, which trigger the viewers to change the channel even before the ad segment starts. Examples of our prediction model results during a movie (Figure 4) and during a news show (Figure 5) illustrate that point.

FIGURE 4: Graph with TV ad segment prediction on commercial block during a movie. The interval in the chart represents 5 minutes (with each bar representing 10 seconds) on July 22, 2015, from 23:27 to 23:32. The beginning of the segment is determined correctly (with precision of <10 seconds), while the end of the segment is overestimated.

The reason this error occurs is that we only take into account number of channel changes and usually ad segment starts with sudden increase in channel changes as seen in Figures 1 and 2. It turns out that when users anticipate when ad segment will start this sudden increase occurs earlier than expected. News shows are just one example, where users tend to predict the ad segment start. This also occurs in regular shows with a “trailer” before the ad segment.

4. Conclusion and Future Work

In this paper we have presented a model that detects TV ads based on synchronous channel change events. We trained the model based on the logistic regression using the labeled ads provided by the broadcaster. After the training, the model performed with high reliability (88%). This also confirms the hypothesis that a high percentage of the unwanted content is in fact advertising.

Algorithm depends on time series similarity measure that is defined in a manner that fits the ad properties reasonably well. Considering that the number of ad sequence time series representatives is a constant and this time series length is small, the computational complexity of the algorithm is $O(n)$, where n is the length of investigated period in seconds.

In the future work we will continue to focus on crowd-sourced data mining of the IPTV network. Further work also needs to be done on detailed classification of the undesired segments, for example, dividing such content into ads and

FIGURE 5: Graph with ad-prediction on an ad segment during a news show where our prediction model predicts the segment too soon. The interval in the chart represents 5 minutes (with each bar representing 10 seconds) on July 22, 2015, from 19:33 to 19:39. In the case of a regular commercial break, the anticipation and channel switching before the segment starts triggers a false detection.

other types of content, which would provide an automated TV ratings system with relatively high temporal resolution and would have many practical applications. The second field of future work will be focused on user modeling based on the viewing habits. By clustering the viewers by their content voting preferences, a finer viewer model could be developed that would offer personalization and content discovery options to aid the viewer.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank companies Telekom Slovenije and RTV Slovenija for providing the datasets and all collaboration that made this research possible. The work was supported by the Ministry of Education, Science, and Sport of Slovenia and the Slovenian Research Agency.

References

- [1] U. Sedlar, M. Volk, J. Sterle, A. Kos, and R. Sernec, "Contextualized monitoring and root cause discovery in IPTV systems using data visualization," *IEEE Network*, vol. 26, no. 6, pp. 40–46, 2012.

- [2] A. Kos, S. Tomažič, J. Salom, N. Trifunovic, M. Valero, and V. Milutinovic, "New benchmarking methodology and programming model for big data processing," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 271752, 7 pages, 2015.
- [3] A. Kos, V. Ranković, and S. Tomažič, "Sorting networks on Maxeler dataflow supercomputing systems," *Advances in Computers*, vol. 96, pp. 139–186, 2015.
- [4] H.-G. Kim, H.-S. Cho, and J. Y. Kim, "TV advertisement search based on audio peak-pair hashing in real environments," in *Proceedings of the 5th International Conference on IT Convergence and Security (ICITCS '15)*, pp. 1–4, IEEE, Kuala Lumpur, Malaysia, August 2015.
- [5] D. A. Sadlier, S. Marlow, N. O'Connor, and N. Murphy, "Automatic TV advertisement detection from MPEG bitstream," *Pattern Recognition*, vol. 35, no. 12, pp. 2719–2726, 2002.
- [6] A. Albiol, M. J. C. Fullà, A. Albiol, and L. Torres, "Detection of TV commercials," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, pp. 541–544, Montreal, Canada, May 2004.
- [7] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.
- [8] W. W. S. Wei, *Time Series Analysis*, Addison-Wesley, 1994.
- [9] H. Zhang, X. Han, and S. Dai, "Fire occurrence probability mapping of northeast china with binary logistic regression model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 1, pp. 121–127, 2013.
- [10] W. J. Conover, *Practical Nonparametric Statistics*, 1980.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>