Determining leaders and clusters in video consumption

Danny De Vleeschauwer, Chris Hawinkel
Bell labs
Alcatel-Lucent
Antwerp, Belgium
{danny.de_vleeschauwer, chris.hawinkel}@alcatel-lucent.com

Yannick Le Louédec
Orange Labs
Orange
Lannion, France
yannick.lelouedec@orange.com

***Abstract*—In this paper we analyze a trace of a deployed VoD (video on demand) system. Users issue requests for content items in an online VoD catalogue at given moments in time. Based on this information alone we identify communities of users that have similar content preferences, which we refer to as implicit social communities. We find that there is evidence for a limited number of groups of similar users. Next we also determine lead users, i.e., users that consume popular content items consistently before other users do. We show that such users can be identified in the considered data set. We also explain how these two pieces of information could be used to improve recommendation systems and content distribution networks.**

Keywords—social networks; lead users; user communities

I. INTRODUCTION

More and more multimedia content is shared over the Internet, either in closed systems or “over the top”. Much of this content is consumed in a social context, either via posting content items or links thereto in an OSN (online social networks) or because users simply share an interest in similar content. Social relationship of the first type is referred to as an explicit social graph, the latter as an implicit social graph. In this paper we concentrate on the latter. In particular we try to determine user communities with similar interests based on the user content consumption behavior.

Between users connected in some social way, within the whole community and sub-community thereof, certain dynamics exists. Some users within a community may be dominant in the sense that they consistently consume content items first, i.e., before other users do, and that their peers that are connected to them via an implicit or explicit social graph follow this behavior. In this paper we try to identify such leading users.

This paper is organized as follows. In the next section we discuss related work. In Section III we introduce the dataset that we used. Section IV is devoted to identifying lead users, while Section V determines implicit user communities. In Section VI we draw the main conclusions and discuss future research directions.

II. RELATED WORK

Some papers (e.g., [1], [5] and [11]) analyze content consumption as a whole without concentrating on the difference that may exist in communities. They usually conclude that content popularity has a long tail (e.g., Zipf) distribution. There are some papers that determine communities (i.e., user clusters) based on the content they consume [6] and how they are related in a social network [9]. The former concludes that communities based on past content consumptions are better in predicting future content consumption. Other papers exploit relations between users (based on their content consumption) to improve content dissemination. In [3] and [7] content interdependencies on YouTube are exploited to improve networking performance. While [3] designs a heuristic search technique that exploits the YouTube video graph to reduce the time for finding a video clip of interest in an overlay network, in [7] the authors use various centrality measures of the related video graph to enhance the hit ratio of a network cache. Social information is also used in the peer-to-peer systems described in [8] and [10] to enhance performance. Similarly, [4] shows how analyzing tweets in twitter can improve the dissemination of the content tweeted about. In [2] the user preferences are used for more efficient media delivery in online communities.

III. DATASET

In this paper we use the Orange VoD (Video on Demand) dataset, which is an anonymized trace of user requests for VoD movies: each request consists of a timestamp, a user identifier, a movie identifier and a city. Although there is information available for 12 cities, we only consider the requests made in one (i.e., the largest) city. There are 8054 active users in this city issuing requests for 30092 movies, many of which are only requested a few times: e.g., only 988 movies are requested at least 60 times and merely 2442 are requested at least 30 times.

We first perform some preprocessing on this dataset. Users are attributed an identifier in decreasing order of activity, so that user n has a total number of requests and $A_n \geq A_{n'}$ if $n<n'$. Likewise, content items are attributed an identifier in decreasing order of overall popularity, so that content item k has a total number P_k of requests and $P_k \geq P_{k'}$ if $k<k'$. Let N be

The research leading to these results has received funding from the European Union's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n° 318398, project eCOUSIN.

the total number of users (1≤n≤N) and K the total number of content items (1≤k≤K). The requests are ordered in increasing order of their timestamp. After this filtering step an ordered list of records of the type (timestamp, user_id, item_id) results, which forms the input to the algorithms described below.

IV. Determining Lead Users

First we try to identify lead users. It is intuitively obvious that tracking these lead users (if they exist) is beneficial for spotting early trends in content consumption. We reserve how caches and recommendation systems can exploit this idea for future research.

In order to find the lead users we first perform an additional filtering operation on the ordered list of requests. We only retain a record if its associated content item is consumed for the first time, i.e., has the smallest timestamp. Remark that in this new ordered list (resulting after this filtering operation), each content item only appears once. We refer to such requests as "firsts". We count the number F_n of *firsts* associated to each user. Fig. 1 shows the distribution of *firsts* for the considered VoD dataset, for the 5000 most active users (the remainder of the users did not generate any *firsts*).

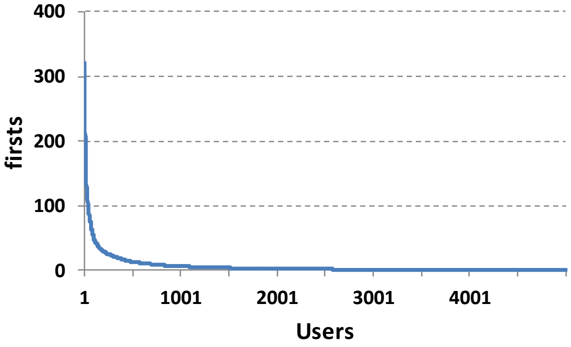


Fig. 1. *Firsts* distribution for the considered VoD dataset

If the choices of the users were completely random, more active users would have more *firsts*. For this (random) case, it can be proven (under the assumption that user requests follow Poisson processes and that the number of content items is large enough) that F_n is proportional to A_n. In general, within a large, real user population we observe considerable differences in the ratio F_n/A_n due to the differences in user behavior and interests. Fig. 2 illustrates this for the considered VoD dataset. We notice that a large fraction of the users does not generate any *firsts*, whereas a smaller fraction generates only *firsts*.

We also observe significant differences between users in the impact of their *firsts*: certain users request many content items that are not viewed by any other user. We refer to such requests as 'singles'. Another category of users generates almost no *singles*: their *firsts* are gradually picked up by other users, consuming the same content item at a later time. In order to refine the characterization of the users, we also count the number S_n of *singles* associated to each user. Next, we analyze the considered VoD dataset to assess the relation between *firsts* and *singles*. In Fig. 3 we show the number of *firsts* for the first 200 users shown in Fig. 1 (ranked by decreasing number of *firsts*), together with the corresponding number of *singles* within these *firsts*.

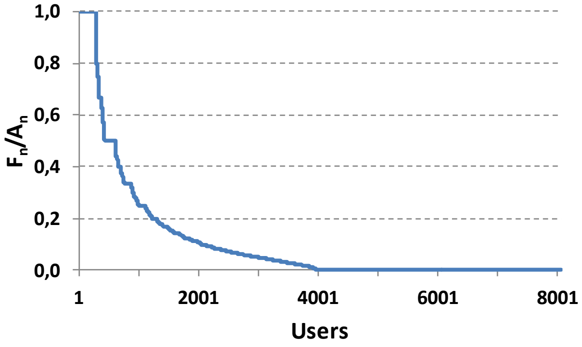


Fig. 2. F_n/A_n distribution for considered VoD dataset

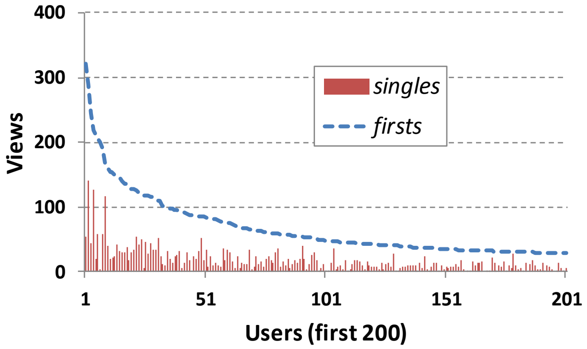


Fig. 3. *Firsts* and *singles* for first 200 users for the considered VoD dataset

We notice some overall correlation between the numbers of *firsts* and *singles*, but also that there are strong variations. Users generating the highest number of *firsts* but also a high number of *singles* are not systematically followed by many other users to view the same content, whereas other users with a relatively high number of *firsts* AND a low number of *singles* do generate many follow-up views by other users. We define as a 'follow-up' request for a content item any request coming after the first request for that content item. We denote U_n as the number of *follow-ups* resulting from *firsts* generated by user n. Fig. 4 shows the number of *follow-ups* in relation to the corresponding *firsts*, and this for the first 200 users (ranked again by decreasing number of *firsts*).

Here we observe that the overall correlation between the *firsts* and the *follow-ups* is limited, and that there are strong variations in the number of *follow-ups.*

We define now a "lead user" as a user that, by initiating many *firsts*, generates a high number of *follow-ups,* and at the same time generates a low number of *singles.* In practice, this implies that, when a lead user generates a *first,* there is a high chance that the related content item will be requested again, and probably by a considerable number of other users.

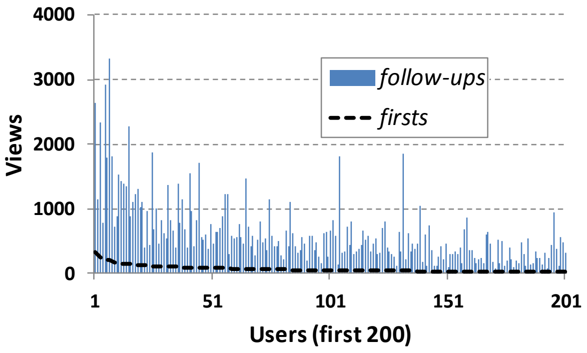


Fig. 4. *Firsts* and *follow-ups* for first 200 users for the considered VoD dataset

In line with this definition, we consider now U_n/S_n as a metric to identify lead users (where zero values for S_n are converted to 1). We apply the metric to all the users of the considered VoD dataset, and then rank them according to this metric. For comparison, we add the metric value for the total amount of requests for this city, or $U_{\mathrm{TOTAL}}/S_{\mathrm{TOTAL}}$, whereby U_{TOTAL} corresponds to the total requests minus the *firsts,* and S_{TOTAL} corresponds to the total amount of *singles*. As an example these values for the first 10 users (ranked by decreasing $U_n\backslash S_n$) together with ($U_{\mathrm{TOTAL}}\backslash S_{\mathrm{TOTAL}}$) are shown in Fig. 5.

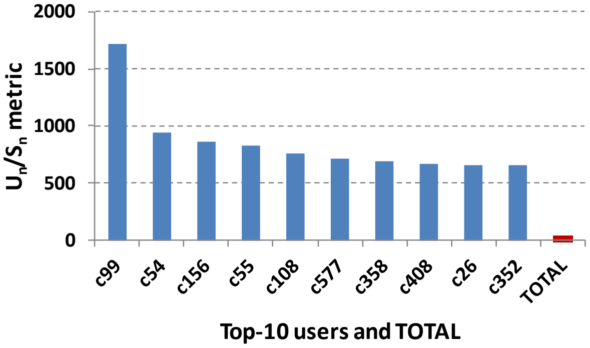


Fig. 5. Top-10 for lead user metric for the considered VoD dataset

The metric graph shows clearly that the top users score considerably higher than the average user: the number of *singles* generated per *follow-up* delivery is considerably lower than for average users (represented by the TOTAL). Future research on alternative metrics to identify lead users based on the S_n, F_n and U_n parameters will target specific objectives, such as maximizing the efficiency of caching strategies.

V. Determining Clusters

Next we cluster users in the considered VoD dataset in communities based on the consumption patterns of the users. Since a community harbors users with similar consumption patterns, the popularity distributions of the movies associated with each community differ. How the knowledge that a user belongs to a certain community together with the monitoring of the popularity distributions associated with each community can be exploited by caching and recommendation systems, is a topic for future research.

To identify the clusters we construct a matrix R of dimensions NxK, where entry $R[n,k]$=1 indicates that user n requested content item k at least once, and $R[n,k]$=0 otherwise. This matrix can be easily constructed by running through the list of requests. This matrix is sparse. For the considered VoD dataset the matrix R only contains 1.3% nonzero entries if only the first (i.e., most popular) 1000 movies are considered and N=8054 (and R is even sparser if more, i.e., less popular, movies are considered).

Each row of this matrix R is a K-dimensional vector associated with user n, which characterizes this user's consumption pattern. It is on this cloud of N vectors (one per user) in this vector space that we determine clusters. Even though, we only take the 1000 first most popular movies into account, this value for K is still large. Therefore, we first determine the main axes of this cloud of points via an SVD (singular value decomposition) after subtracting the average vector (averaged over all users) from each vector. Since the 51st singular value is smaller than 20% of the first we only retain the 50 main directions. We verified that this choice is not critical.

On this set of 50-dimensional vectors we perform a K-means clustering. First we investigate the impact of the number J of clusters. Therefore, we first consider the sum of squared distances of the points to their cluster center. It is easy to see that this total sum of squared differences, which we refer to as the total variability, can be written as

$$\sum_{n-1}^{N}(x_n-\mu)^2=\sum_{j-1}^{J}\sum_{n\in C_j}(x_n-\mu_j)^2+\sum_{j-1}^{J}N_j\cdot(\mu_j-\mu)^2 \qquad (1)$$

where x_n are the points to be clustered, μ is the center of the cloud of points (i.e., the origin in our case as we subtracted the average vector), μ_j are the cluster centers, C_j the j-th cluster and N_j the number of points in the j-th cluster. Eq. (1) shows that the total variability in the cloud of points (i.e., the left hand side of the equation) is equal to the sum of the intra-cluster variabilities (i.e., the first term of the right hand side) and the inter-cluster variability (i.e., the second term of the right hand side). Of clouds with clearly pronounced clusters we would expect that the intra-cluster variabilities would be small (i.e., we would expect that all points would be fairly concentrated around the center of the cluster), so that the total variability in the original cloud is more or less captured in the inter-cluster variability. In other words, if the ratio of the inter-cluster variability to the total variability is close to 1 we expect that the original cloud clusters well in a few centers. However, we do not want too many (close) clusters either. In order to keep the number of clusters under control, the minimum distance between cluster centers should not to be too small. Based on this, we define the clusterability index as

$$\frac{\sum_{j-1}^{J}N_j\cdot(\mu_j-\mu)^2}{\sum_{n-1}^{N}(x_n-\mu)^2}\cdot\min_{i\neq j}\left\{\|\mu_i-\mu_j\|\right\} \qquad (2)$$

Fig. 6 shows how this clusterability index changes as the number of clusters increases. Notice the log-scale on the *X*-axis. The value of *J* where this clusterability index peaks is a good choice for the number of clusters: *J*=2 has the largest clusterability index and *J*=5 is a local peak. The error bars in the figure stem from the fact that the K-means algorithm starts from random cluster centers and iterates towards stable cluster centers. We performed 500 experiments for each *J* and calculated the average and the standard deviation of the clusterability index. If the clustering is good we expect K-means to converge always to the same stable centers. This is the case for a small number of clusters (*J*<10). For a large number of clusters this is not the case as witnessed by the large standard deviation.

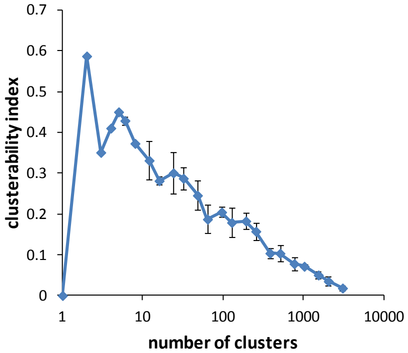
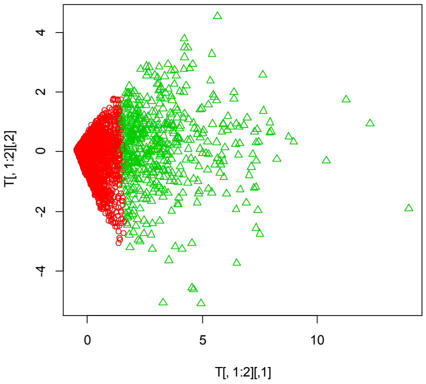


Fig. 6. Clusterability index as a function of the number of clusters for the considered VoD dataset



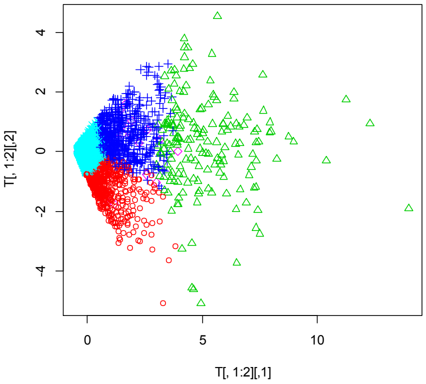


Fig. 7. Two (top) and five (bottom) clusters identified in the considered VoD dataset

Fig. 7 illustrates the clusters found in the considered VoD dataset with two (*J*=2) and five clusters (*J*=5). This figure shows the first two components of the SVD where points belonging to the same cluster have the same color and symbol. It is a projection of the 50-dimensional vector space with clustered points on the plane made up of the two main axes.

VI. CONCLUSIONS

In this paper we have analyzed a VoD dataset, which consists of records revealing which user consumed which content item at which moment in time. We have shown that based on this information alone (i.e., without other information how the users are socially related) we can identify clusters (i.e., user communities) and lead users (users who consistently consume soon-to-be-popular content items first).

In our future work we will continue our investigation in two directions. First we will determine if communities and lead users determined based on social graphs are similar to the ones we identified in this paper: which of the two pieces of information (i.e., the social graph or the similarity in consumption patterns) is most powerful to determine the communities and leaders and to which extent both pieces of information can reinforce each other. Second we will assess if the extracted information, i.e., lead users and communities, can be beneficially exploited to improve content dissemination networks, in particular, caching and recommendation systems.

REFERENCES

[1] H. Abrahamsson and M. Nordmark, “Program popularity and viewer behaviour in a large TV-on-demand system”, In Proc. of the conf. on Internet measurements, pp. 199-210, New York (NY): ACM, 2012.

[2] J. Chakareski and P. Frossard, “Context-adaptive information flow allocation and media delivery in online social networks,” IEEE J. Selected Topics in Signal Processing, vol. 4, no. 4, pp. 732–745, Aug. 2010.

[3] X. Cheng and J. Liu, “Nettube: Exploring social networks for peer-to-peer short video sharing,” in Proc. conf. on Computer Communications (INFOCOM). Rio de Janeiro, Brazil: IEEE, Apr. 2009, pp. 1152–1160.

[4] X. Cheng and J. Liu, “Tweeting videos: coordinate live streaming and storage sharing,” in Proc. Int’l Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV). Amsterdam, The Netherlands: ACM, Jun. 2010, pp. 15–20.

[5] C. Griwodz, et. al., “Long-term movie popularity models in video-on-demand systems: or the life of an on-demand movie”, In Proc. of the 5th int’l conf. on Multimedia, pp. 349-357, New York, NY, ACM: 1997.

[6] X. Han, et. al., “Community Similarity Degree: Finding Similarity to Improve Recommendations in On-line Social Networks”, Social Network Analysis and Mining journal, in press.

[7] V. Kulkarni and M. Devetsikiotis, “Communication timescales, structure and popularity: Using social network metrics for Youtube-like multimedia content distribution,” in Proc. Int’l Conf. Communications. Cape Town, South Africa: IEEE, May 2010.

[8] K. C.-J. Lin, et. al., “Socionet: A social-based multimedia access system for unstructured P2P networks,” IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 7, pp. 1027– 1041, Jul. 2010.

[9] J. McAuley and J. Leskovec. “Learning to Discover Social Circles in Ego Networks”, NIPS, 2012.

[10] J. A. Pouwelse, et. al., “Tribler: A social-based peer-to-peer system,” Concurrency and Computation: Practice & Experience, vol. 20, no. 2, pp. 127–138, Feb. 2008.

[11] H. Yu, et. al., “Understanding user behavior in large-scale video-on-demand systems”, SIGOPS Oper. Syst. Rev. 40, 4 (April 2006).