Prefetching in Video-On-Demand Services Based on Recommender Systems

Dominik Schreiber <dominik.schreiber@stud.tu-darmstadt.de>

***Abstract*—Video-on-demand systems account for the majority of world-wide internet traffic. Therefore, efficient caching mechanisms and a well-understood model for impacts of recommender systems are vital.**

In this paper, scientific work is reviewed, discussed and compared, to examine a broad view on this topic.

It is found that today‘s recommender systems are implemented as Collaborative Filtering systems relying on neighborhood or latent factor models, which favor personalized recommendations over global “top k” lists.

However, studies of video-on-demand systems showed that video access rates are more than Zipf distributed. This could decrease with more personalized recommendations, but is still a good base for new caching mechanisms: a *“top 10%” cache* could serve 80% of all requests. Furthermore, it occurred that users “scan” through videos, watching only the beginning, until they find what they look for. This can be leveraged by *prefix caches*, that could serve 45% of requests with just the first 5% of every video cached. Lastly, it is shown that videos are often not watched again after an initial request, so *cache-on-first-hit* is likely to be ineffective.

I. Introduction

With more and more bandwidth available for internet usage, video-on-demand services grew heavily in the last years, and appear to be the uncontested number one internet traffic generator with 60 percent traffic share in 2013[1] and a compound annual growth rate of 29 percent, as the Cisco Visual Networking Index [Cis14] shows.

This huge amount of traffic needs to be handled and minimized to provide both an optimal user experience as well as minimal cost for the video-on-demand providers. A question arising is *“Which content should be prefetched in order to minimize the network traffic?”*. Current research focuses on “caching” – not deleting a requested file – instead of “prefetching” – requesting some files in advance – so this paper will go this way as well.

One approach to this is to look at recommender systems– *“you may also like these 32 videos”* – and how they impact the popularity of content. That particular problem will be studied in this paper, by reviewing, categorizing and discussing existing scientific work.

The remainder of this paper is organized as follows: Section II briefly explains video-on-demand services, probability distributions and studyable parameters. The following sections analyze the impact of recommender systems on special-purpose (i.e. educational services like the *eTeach* platform, covered in section III), general-purpose (i.e. *Netflix*, section IV) and user-generated content (i.e. *YouTube*, section V) video-on-demand services. A conclusion of the findings of the previous chapters is drawn in section VI, which ends this paper.

II. Background

To establish a common understanding of the topic for the rest of this paper, some history and concepts need to be introduced. These are in fact *video-on-demand services* themselves (II-A), *probability distributions* that could describe usage patterns and popularity of content (II-B), and *parameters of interest* that can be empirically studied and give insight through their probability distribution (II-C).

A. Video-on-demand services

Video-on-demand services originated in the mid 1990s and can be grouped into *general-purpose*, *special-purpose* and *user-generated-contend* services, that differ in the content being served and the usage patterns arising from this content.

1) general-purpose: Commercial video-on-demand services were pioneered by the *Cambridge Digital iTV Trial* in 1994 [RRD95], where content was streamed to 250 subscribers and a small amount of schools all around Cambridge.

The *Kingston interactive TV*, launched in 1999 by the U.K. telecommunications provider Kingston Communications, was the first video-on-demand service to span a larger amount of users all across the U.K. [ZMS11].

Today‘s big player in video-on-demand platforms, *Netflix*[2], started in 1999 as a digital DVD rental service, and was the first to put high emphasis on recommender systems in 2006 with the one million dollar *Netflix Prize* as an award for the first recommender system that beats their “CineMatch” algorithm by 10 percent [Kor08].

These more traditional services have in common that they provide professionally crafted content, like entertainment movies and TV shows, with a high quality and long length – i.e. in median 94 minutes for the european video-on-demand service *Lovefilm*[3] [CKR$^+$07].

2) special-purpose: Video-on-demand services for special content – i.e. only educational content for universities – like the *Berkeley Internet Broadcasting System* BIBS [RHPL01] from 1998 or the *eTeach* [FMSL02] from 2000 were the first to be investigated scientifically (see section III).

However, as they have a more narrow view, with only educational content and an homogenic audience of students,

[1] 17.455 petabyte per month internet video traffic from 29.071 petabyte per month in total

[2] netflix.com

[3] lovefilm.com, since february 2014 part of *Amazon Prime Instant Video* amazon.com/piv

the results from this research are hardly generalizable to other fields video-on-demand– i.e. students may show different viewing habits as they are "forced" to view lectures to get their grades.

3) user-generated content: The advent of Web 2.0 – dynamic data, virtual communities, user participation – led to another type of video-on-demand services: video-on-demand for user-generated content. Services like *YouTube*[4], *Vimeo*[5] or *Vine*[6] allow users to upload, share and discuss content.

This content is widespread, but shorter and usually of less quality than the professionally created content in traditional video-on-demand services. With 99 percent of these videos being shorter than 10 minutes [CKR$^+$07], there appears to be a change in user interest and usage patterns, which will be discussed in section V.

B. Probability distributions

When looking on the impact of recommender systems on user behavior, actually *changes in probability distributions* – i.e. user arrival rates, requests, upvotes – are studied. Beginning with an empiric distribution, one tries to find the theoretical model that fits best to this empiric data. Common theoretical models of interest are dividable into *exponentional distributions* (II-B1) and *power-law distributions* (II-B2). [AT84] covers them among others, providing a more holistic point of view. Figure 1 gives needed illustrations.

1) exponentional distributions: The probability density function of exponentional distributions is of the general form

$$p_\theta\left(X = x\right) = h\left(x\right)\ exp\left(\theta^\intercal \times T(x) - A(\theta)\right)$$

where

- x is an empirical value of the exponentional distributed random variable X,
- θ is the *natural parameter* of the distribution family
- $h(x)$ is the *underlying measure* used to create the distribution,
- $T(x)$ is a *sufficient statistic* of the distribution and
- $A(\theta)$ is the *log-partition function* that generates a cumulative of instances of X

With this general form, a large amount of distribution families can be described. Take for example the *Gaussian distribution*

$$p_{\mu,\sigma^2}\left(X = x\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\ exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right)$$

that fits into the general form of exponentional distributions by setting

- $\theta = \left\langle \frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2} \right\rangle$
- $h(x) = \frac{1}{2\pi}$
- $T(x) = \left\langle x, x^2 \right\rangle$
- $A(\theta) = \frac{\mu^2}{2\sigma^2} + \log\sigma$

Other exponentional distributions are for example *normal* and *log-normal* distributions, χ^2 or *Poisson* distributions, as well as the canonical *exponential* distribution, and many more.

2) power-law distributions: When e is raised to the somehow xth power in exponentional distributions, x is raised to the $-\alpha$th power in power-law distributions, leading to a probability density function of the following form:

$$p(X = x) = C\ x^{h(x)}$$

which holds for $x > x_{min}$ where

- x is, again, an empirical value of the random variable X,
- x_{min} is a minimum value, which is needed to prevent an infinite area when $x \to 0$,
- C is a constant *scaling factor* that makes the total area 1, and
- $h(x)$ is a usually constant *growth rate* ($h(x) = -\alpha$), that creates a *power-law with exponentional cutoff* when depending on x (i.e. $h(x) = \alpha + \beta x$)

This general form can again be used to describe many known probability distributions. Common examples are the *Zipf distribution* ($C = \left(\sum_{n=1}^{N} n^{-1}\right)^{-1}$, $h(x) = -1$ with N the number of elements), the *Pareto distribution* ($C = \alpha\ x_{min}^{\alpha}$, $h(x) = 1 - \alpha$ with α the *tail index*), or the degree distribution in *scale-free networks*.

C. Parameters of interest

Knowing what probability distributions exist is not enough. One has to find *parameters* that can be studied in their probability distributions. This should lead to conclusions about the influence of recommender systems on video popularity. These parameters are, among others, *file access frequency* (II-C1) and *daily access rates* (II-C2).

1) File access frequency: Given a list of files, ordered by number of access requests (i.e. most requests to least requests), how often has any file been requested? If this parameter is for example Pareto distributed, it means that the highest-ranked 20 percent of files account for 80 percent of the file requests. Abnormalities in the file access frequency distribution could easily arise from a recommender system that enhances already higher-ranked files (or lower-ranked, respectively).

2) Daily access rates: While file access frequencies focus on a single point of time, daily access rates provide information about change in file access frequency (in their first derivation, mathematically) – which holds not only for daily, but also weekly, monthly or any time intervall access rates, just with changing degree of detail. Daily access rates can be seen as a list of file access frequencies ($AR_f = [\text{file-access-frequency}(f,d) \mid d \in \text{Days}]$). *Normalized* daily access rates would then pick the maximum daily access rate for every file, and show all daily access rates relative to this maximum (like $\hat{AR}_f = \left[\frac{x}{\max_{a \in AR_f} a} \mid x \in AR_f\right]$).

III. Special-purpose video-on-demand

The first video-on-demand services that have been scientifically studied were educational platforms of universities. Researchers had easy access to the data, and were able to gain deep insight into user behavior and recommender systems. But, as noted before, the findings may not be generalizable

[4] youtube.com

[5] vimeo.com

[6] vine.co

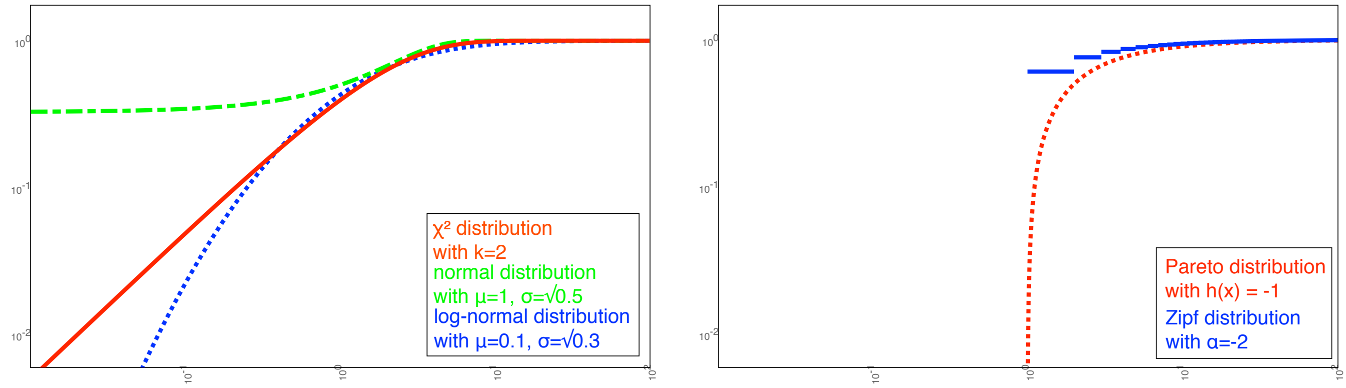

Figure 1. Cumulative density functions (CDFs) plotted for exponentional (left) and power-law (right) distributions; self-made

because of the special user population and the one-sided hosted content. Nonetheless, in this section, the results of this research are presented and discussed.

A. *mMOD*

In 1999, Acharya *et al.* found that there was no research on user access to video-on-demand services, so they studied the *mMOD* (**m**ulticast **M**edia **O**n **D**emand) system at Luleå University, Sweden, and published their findings in [ASP99]. Their focus was on user access patterns, file access frequencies and temporal/spatial locality of access requests.

The *mMOD* system was populated with videos averaging in 75 minutes, 121 MByte, and therefore a bitrate of 150kBit/s. As they described, videos were compressed using H.261 encoding and stored on the mMOD web server, from where users were able to receive them using the mMOD Controller, as well as Java applets.

They normalized the log data of the mMOD system from 29.08.1997 to 10.03.1998 by filtering out duplicate and irrelevant entries, and were able to extract useful parameters from it:

File access per day fell significantly during weekends and vacation, and rose enormously at the end. They conclude that, in the new term, more lectures used the system and most initial bugs had been fixed, what led to this rise.

From previous studies on content of the world-wide-web, they assumed that the *file access frequency* would be Zipf distributed – exactly with $p(X = x) = x^{-0.73}$ as they concluded from [Che94]. This was actually *not the case*, as higher-ranked titles were even more frequently accessed than the Zipf distribution predicted, the top 10 percent were responsible for 50 percent of all requests. Figure 2 shows their results.

64 percent of all *file accesses by machine* came from the internal campus network, with just the top ten percent of machines being responsible for 59 percent and the top twenty percent being responsible for 74 percent of all requests.

They found that the *request interarrival times* was 411 seconds in median, but did not further investigate it.

It appeared that only 55 percent of videos were actually watched until the end, with the other 45 percent, *partial accesses*, stopping during the first 5 percent of the watched video.

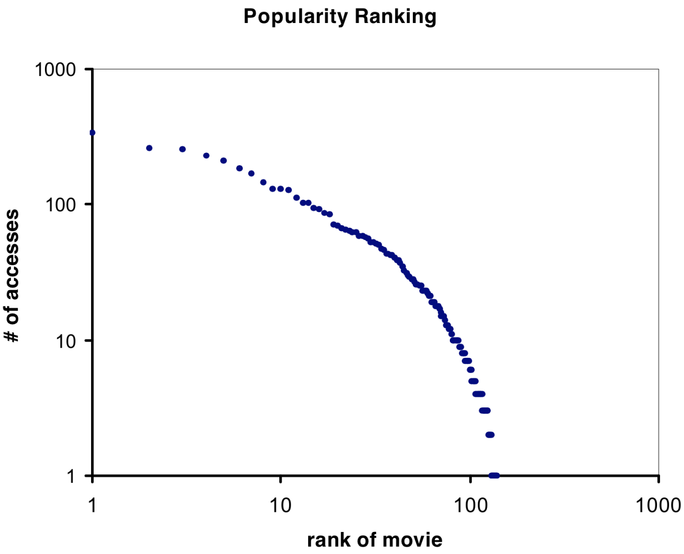

Figure 2. File access frequency in mMOD; from [ASP99]

Access patterns for content types seemed to differ. With the content of mMOD being merely educational, but also serving seven entertainment movies, Acharya *et al.* found that the educational content was accessed during small periods, mainly before homework assignment deadlines, while the entertainment content was evenly requested during the whole period.

From least-recently-used stack analysis they found that the data served by mMOD shows high *temporal behavior*, as files are referenced in short periods of time again and again. Figure 3 shows this behavior.

Being the first research on user behavior in video-on-demand services, the study of Acharya *et al.* showed significant numbers for the use of an educational video-on-demand service. Especially their observation of content-type dependent access patterns emphasizes the point that studying special-purpose video-on-demand services alone will not be enough to provide an integral view on user behavior in video-on-demand

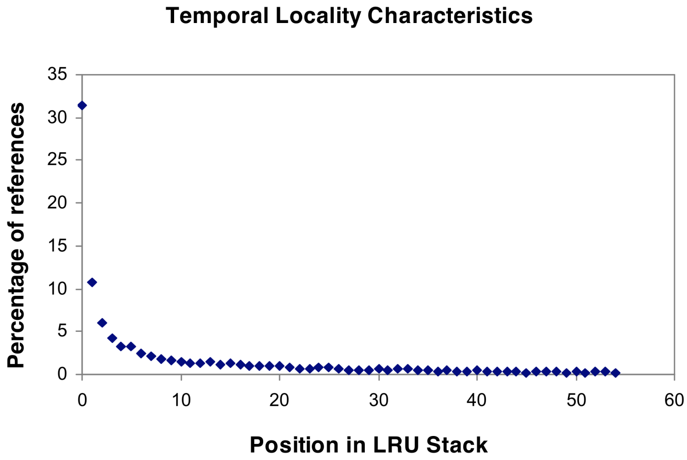

Figure 3. Temporal locality characteristics in mMOD; from [ASP99]

services – results may simply not be generalizable because the user behavior is so different in these services.

Some questions arise though, as the log cleanup prior to the analysis looks more than excessive. They removed a whole class of log entries because *"in practice, users rarely used this facility"* without providing any numbers on this *"rare use"*, and they removed consecutive requests on *"the assumption [. . .] that there were problems in getting the first request to run and that is why the users started another request for the same movie"* without confirming this assumption in any way. Had they been more thoroughly in these points, their results would probably be even more one-sided, as they removed duplicates, but this can not be confirmed from bare assumptions.

In a nutshell, Acharya *et al.* opened another field of research, provided it with initial data, and left enough questions open that others would join (as will be examined in the rest of this paper).

B. *eTeach & BIBS*

Almeida *et al.* deeply studied the video-on-demand systems *eTeach* – pure online courses for 280 enrolled computer science students, started in 2000 – and *BIBS* (the **B**erkeley **I**nternet **B**roadcasting **S**ystem) – multiple lecture recordings from and for the Berkeley campus – between 2001 and 2004 in a series of papers [AKV01], [AKEV01], [CCB$^+$04].

In [AKEV01] they analyzed log files from both systems, where the *eTeach* logs contain separate entries for every interactive request (i.e. play, pause, fast-forward) and the *BIBS* logs contain only one entry per client session. They studied *length of requested media*, *request interarrival times*, *media per session*, *file access frequency*, *segment access frequency*, *infrequently requested files* and the impact of *multicast strategies* for delivery. With curve-fitting, they tried to find the underlying distribution for this empirical data.

The *length of requested media* for the *eTeach* system seems to be mostly shorter than 5 minutes – this makes up about 30 percent of all requests – with an even distribution across all other 5 minutes intervals (5–10, 10–15, 15–20, 25–30, 30–35 minutes). For the *BIBS* system however, about 70 percent of all requests are for videos of 50–55 minutes length.

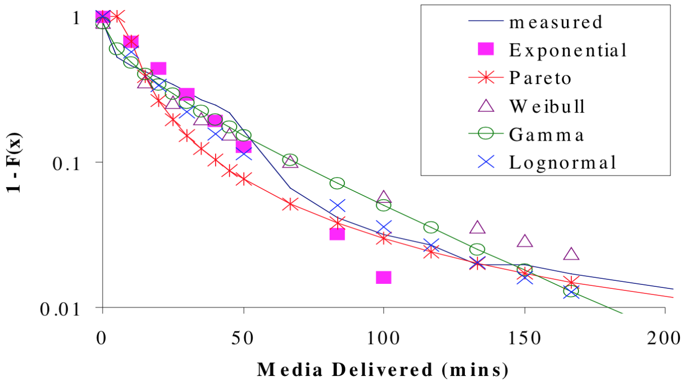

Figure 4. Media delivered per session for BIBS; from [AKEV01]

Request interarrival times were studied for periods of merely stationary request arrival rate for all files that had a significant number of requests in this period. By curve-fitting they found *BIBS* client session arrivals to be merely Poisson – exponentional, see II-B1 – distributed, while *eTeach* client session arrivals were heavy-tailed Pareto – power-law, see II-B2 – distributed.

Media per session is in contrast to *length of requested media* the amount that was actually delivered to the user (i.e. 34 minutes of a 55 minute lecture). This, observed on the *BIBS* servers, seems to be log-normal distributed (exponentional family) for videos with a length less than 5 minutes, but hybrid Gamma+Pareto distributed for videos of 50-55 minutes – Pareto modeling the tail of the distribution. Figure 4 shows their results – again gained by curve-fitting.

The *file access frequency* for both *eTeach* and *BIBS* seemed to be following *two* concatenated Zipf distributions – the frequency is nearly linear for the highest-ranked files (13 for BIBS), and for all lower-ranked files. Figure 5 shows $rank \to requests$ plots from [AKEV01]. This concatenation of Zipf distributions stands in contrast to previous studies that suggested a single Zipf distribution for various content from html/images on the web [BCF$^+$99], over offline DVD rental [DSS94], to streaming from multiple servers [CWVL01]. *While the authors fail to give a reason for this contrasting results, it is quite possible that some sort of recommender system– either automated as a "top ten" list or manually like a lecturer saying "you have to watch this to collect credit" – is responsible for this shift.*

Segment access frequency is especially important for caching strategies like *prefix caching*, where only the first segments of a file are cached. For *BIBS* and *eTeach* Almeida *et al.* find that all segments of high-ranked files are accessed equally often, but the first segments of lower-ranked files are accessed more frequently than their last segments. This means, prefix caching would work on lower-ranked files, but not on high-ranked ones, where segment access frequency can be considered constant.

They find that 70 percent of first-time requested videos are not requested again in the next 8 hours, what renders caching strategies like *cache on first hit* unuseable.

Finally, they simulate a *closest target* multicast delivery

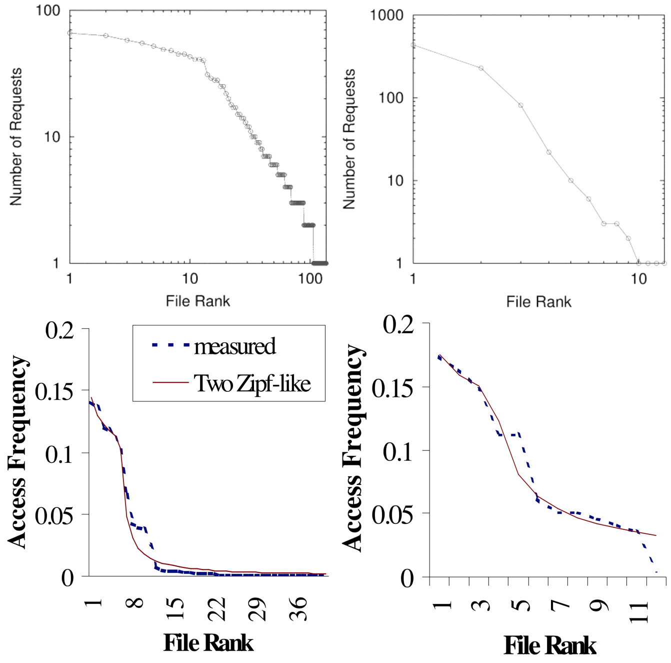

Figure 5. File access frequencies measured (top) for (left) BIBS and (right) eTeach, and modeled (bottom) by two Zipf distributions; from [AKEV01]

scheme [EVZ01] for *eTeach* videos, and are able to reduce the bandwidth requirements by 40–60 percent.

The research of Almeida *et al.* brought deep insight in the request and access distributions on educational servers and allowed to simulate workloads for media servers with high confidence in the data the simulation is based on.

Unfortunately this research fails to explain the data found, but just reports that it is found. Knowing that file access frequencies are "double"-Zipf distributed is good for simulation, but the explanation why this phenomenon exists would be of even more interest.

Also, the eTeach and BIBS systems were not described as detailed as needed. One does not know, for example, wether these systems use a recommender system or if they just show like random videos – which would be an implicit recommendation as well, just not well-engineered.

To sum it up, Almeida *et al.* laid a ground stone for future research, that will be discussed in the upcoming sections, gave great insight into the empirical data, but unfortunately lack in explanations.

IV. General-purpose video-on-demand

While special-purpose video-on-demand systems, as discussed in section III, target a homogenous but small audience, general-purpose video-on-demand systems are designed and used to serve content to a huge amount of clients. As described before, the results of early studies are hardly generalizable to this new field of use.

In the remainder of this section, the study of the chinese *PowerInfo* video-on-demand system by Yu *et al.* [YZZZ06] is described and discussed (IV-A). Furthermore, the recommendation algorithm that won the one million dollar Netflix Prize [Kor08] is analyzed (IV-B).

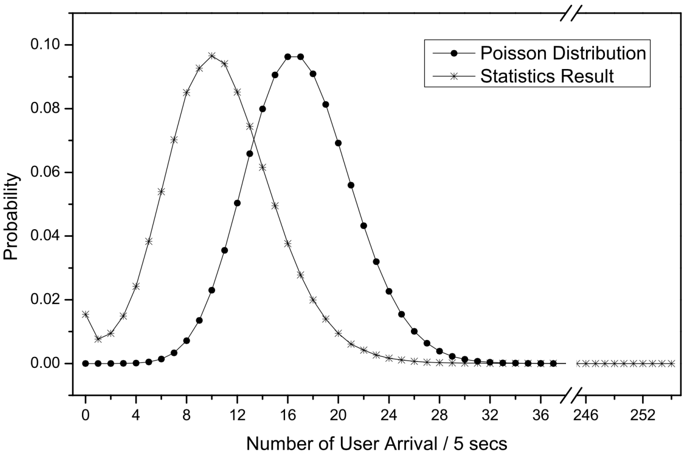

Figure 6. statistical user arrivals per 5 seconds in the PowerInfo system, with Poisson distribution with $\lambda = 15$; from [YZZZ06]

A. PowerInfo

In 2006, the research group of Yu *et al.* was able to study the chinese commercial video-on-demand system *PowerInfo*, that delivered video content to 1.5 million paying customers in 20 cities of China, and published their results in [YZZZ06]. They aggregated 21 million log entries of video requests from a period of 219 days including chinese national vacation – providing even insight into changing user behavior in special situations.

They studied *temporal access patterns*, *user arrival rates*, *session lengths*, the *popularity of content*, especially the impact that the PowerInfo *recommender system* had on video popularity, and the *temporal change in popularity*.

For *access patterns*, they found *daily* spikes in the midday break and in the evening and surges in the early morning and in the afternoon. *Weekly*, they observed fluctuating access rates in the first half of the week, and constant rising rates in the second half, peaking saturdays. As they were able to study *national vacations*, they saw that access rates peaked globally on the first day of vacation, but dropped below normal level afterwards.

In contrast to [ASP99], who observed one new user every five minutes (see III-A), Yu *et al.* had up to five users per second joining the PowerInfo system – this is most surely due to the large user base of twenty chinese cities versus one single campus of a swedish university. They curve-fitted user arrival rates with a *Poisson* distribution ($P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$), but found that this over-estimates user arrivals. Figure 6 shows this curve-fitting. They indeed propose a *modified Possion distribution* that fits their emirical data (figure 7):

$$P(x) = \frac{\lambda^{N-x} e^{-\lambda}}{(N-x)!}$$

where N is the maximum number of user arrivals per five seconds (from their data: $N = 27$), and λ is the variance of user arrivals (empirically: $\lambda = 17$).

Studying *session length*, they find a highly "impatient" audience, that closes 52.55% of all sessions in the first 10 minutes (37% in just 5min, over 70% before 20min – the

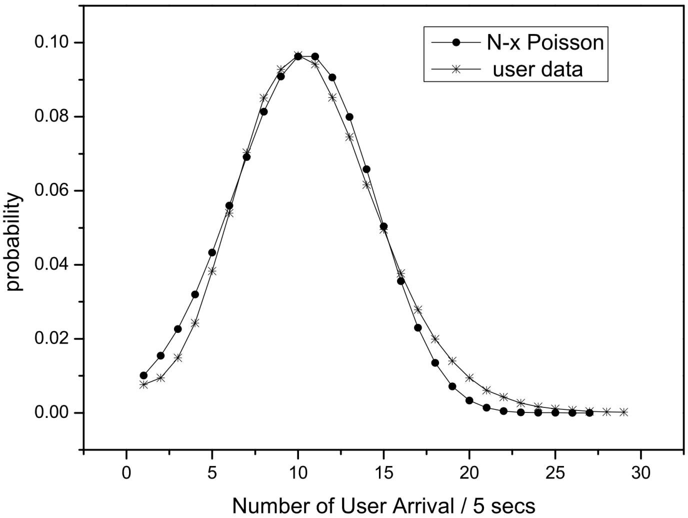

Figure 7. modified Poisson distribution with $N = 27$, $\lambda = 17$; from [YZZZ06]

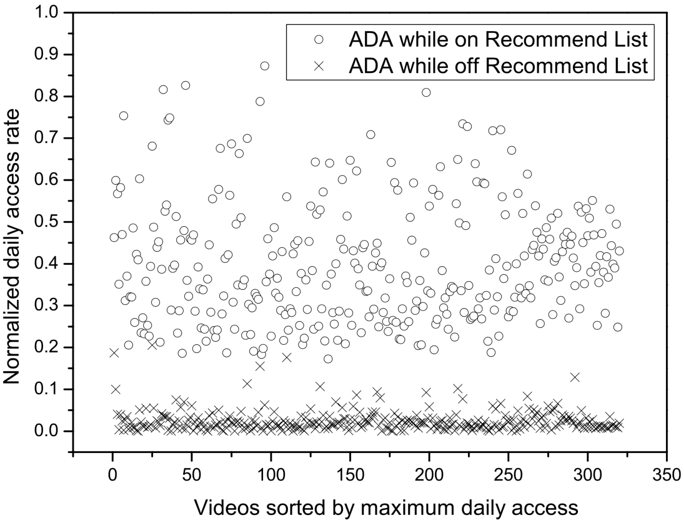

Figure 8. Recommender systems impacting daily accesses, normalized to maximum access rate; from [YZZZ06]

served content has usually a length of more than 60min), and conclude from the roughly Pareto distributed session lengths, that users tend to "scan" through videos watching only the first seconds/minutes until they find the video they are searching for. From this, they recommend caching mechanisms that cache only the first minutes of videos, and marketing mechanisms that offer the first minutes of videos for free. Moreover, they observe that lesser popular videos have a higher probability of being watched to the end – there is an inverse correlation between video popularity and session length. The observation of "scanning" users fits well with [ASP99], who found that even 45 percent of videos were stopped in their first 5 percent.

Yu *et al.* studied *video popularity* in detail:

First, they found a slightly more moderate distribution of videos by popularity than a canonical *Pareto distribution* ("80-20 rule") would suggest. They account the large library of PowerInfo for that, and see the need for larger caches than expected to achieve predicted cache hit rates.

Then, while [GDS$^+$03] suggested a *fetch-at-most-once* model as popularity distribution, where every video will be fetched not more than once (as the user has already seen it then), the PowerInfo videos seem to be *Zipf distributed* (with a "heavy tail of unpopular items").

Furthermore, *changes in user interest* in top 10, top 100 and top 200 videos were studied: top 100 and 200 were merely stable over some days, but highly diverged on single days, whereas the top 10 did not really change over a single day, but were totally different over multiple days. Therefore they suggest a two-level caching mechanism with a small and fast "top 10"-cache that updates daily – this cache could then serve 80-90% of most frequently requested videos – and a larger but slower "top 100"-cache for the rest.

To learn about the impact of the PowerInfo recommender system, Yu *et al.* picked representative single videos for a case study, and followed them through their lifecycle. Huge drops in user interest were found once the videos left the "most popular" list, also correlation of interest rise with external events (a legacy video regained popularity as a remake was released). They also normalized the *daily access rate* of every video with the videos *maximum access rate* (the maximum of this videos daily access rates), and found that videos on the "recommended" list performs between 20 and 90 percent of its maximum access rate, while videos that are not on a list perform on less than 5 percent (see figure 8). While the PowerInfo "recommended" list computed on a daily basis, the system also offers a "top 15" list that is computed on a monthly basis – a video has to be popular for over a month to be considered a "top 15" candidate. The videos on this list seemed to be less impacted when dropped from the list, and the conclusion is that the videos have been popular before being put on this list, so the list was not vital to the popularity in the first place.

The results presented by Yu *et al.* are well-stubstantiated by the empirical data they studied. They are the first to give insight into a 1.5 million user video-on-demand system. In contrast to Acharya *et al.* (III-A) and Almeida *et al.* (III-B), they do not only provide analyzed data in a sink or swim manner, but also try to explain their findings and propose mechanisms (i.e. for caching videos, marketing, and simulating user load).

One must note, that their research has aged since 2006, technology has advanced, and new technology may render their results invalid for today‘s systems. For example, while users were connected with "broadband" 512kB/s to the PowerInfo system, today‘s users have 3.3mB/s[7] at their hands [Ben14] – this may or may not lead to different user behavior in any way. But, as research was based on the work of Yu *et al.*, their work is still quite impacting.

B. BellKor‘s Pragmatic Chaos

To gain more insight in the mechanisms of recommender systems, one of the most elaborate recommender systems, the

[7] on *global* average, 8.7mB/s average for the U.S., 7.3mB/s for Germany

algorithm that won the "Netflix Prize", to beat Netflix own "CineMatch" by at least 10 percent, will be described here. The team that designed the winning algorithm, *BellKor‘s Pragmatic Chaos* from the american telecommunication provider *AT&T*, published the theoretical foundations for their work in [Kor08].

In general, the task of a recommender system is to recommend *items* (indices i and j in the following) to users (indices u and v). They are fed with *ratings* r_{ui} (user u rates item i) and estimate unknown ratings $\hat{r}_{uj}$ (estimated rating of user u for item j) to return the K items with the highest estimated rating for user u.

Like a vast majority of recommendation algorithms, this algorithm is based on *Collaborative Filtering* [GNOT92], what basically means to use only the previous behavior of a user to generate personalized recommendations.

There are two ways to utilize the user‘s history, namely *neighborhood models* and *latent factor models*, and the proposed algorithm is the first to combine both approaches to a single algorithm.

Neighborhood models try to rate the relation between user and item. First models looked at the users "neighborhood": they used ratings of similar users as estimates ("people who bought this also bought these 37 items"). Later models placed items in a "neighborhood" and collected ratings of the single user for similar items, which had the nice side-effect of yielding easy-to-understand explanations for recommendations ("you bought season 1, you may also like season 2..5"). To formalize this, one needs a "similarity measure": a *correlation coefficient* between items:

$$s_{ij} = \frac{n_{ij}}{n_{ij} + \lambda_1} \rho_{ij}$$

where n_{ij} is the amount of users that explicitly rated i and j, ρ_{ij} is the Pearson correlation coefficient ($\rho_{ij} = \frac{cov(i,j)}{\sigma_i \sigma_j}$) between i and j, and $\lambda_1 = 100$ is empirically found. Then the estimated rating can be defined as

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in S^k(i;u)} s_{ij}(r_{uj} - b_{uj})}{\sum_{j \in S^k(i;u) s_{ij}}}$$

where b_{ui} is a baseline for this rating, and $S^k(i;u)$ are the k nearest neighbors for u to i.

Latent factor models, unlike neighborhood models, try to compare users and items directly in a *latent factor space*. A factor could (in the video domain) for example be "video type" (thriller vs drama, quite obvious), or "depth of character development" (somehow obscure), or something completely uninterpretable – it is automatically inferred from the data. In math, each user is represented by a user-factors vector $p_u \in \mathbb{R}^f$ and each item with a item-factors vector $q_i \in \mathbb{R}^f$. Then, the estimated rating is

$$\hat{r}_{ui} = b_{ui} + p_u^\intercal q_i$$

and the task of the algorithm is to estimate features, which can be done by solving

$$\min_{p_*, q_*, b_*} \sum_{(u,i) \in \mathcal{K}} (r_{ui} - \mu - b_u - b_i - p_u^\intercal q_i) + \lambda_2 (||p_u||^2 + ||q_i||^2 + b_u^2 + b_i^2)$$

where $\mathcal{K}$ is the set of all explicit ratings, μ is the global average over all ratings, b_u and b_i are user/item biases to the average, and λ_2 is a constant that increases the penality for overfitting.

BellKor‘s Pragmatic Chaos invented an algorithm that combines both neighborhood and latent factor models that excels at detecting localized relationships, where neighborhood models shine, as well as in estimating an overall structure – what latent factor models are good at. The proposed algorithm does as well include both explicit feedback – when a user gives a 1–5 star rating – and implicit feedback, which is in this case the fact that a user rated a video in any way, as a brilliant source for implicit feedback, a users browsing history, was not available in the netflix data. The algorithm is essentially

$$\begin{aligned}\hat{r}_{ui} = \mu + b_u + b_i + q_i^\intercal \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_i \right) \\ + |R^k(i;u)|^{-\frac{1}{2}} \sum_{j \in R^k(i;u)} (r_{uj} - b_{uj}) \, w_{ij} \\ + |N^k(i;u)|^{-\frac{1}{2}} \sum_{j \in R^k(i;u)} c_{ij}\end{aligned}$$

where (in addendum to the definitions before)

- $N(u)$ are the items implicitly rated by u,
- $R(u)$ are the items explicitly rated by u,
- $N^k(i;u) = N(u) \cap S^k(i)$,
- $R^k(i;u) = R(u) \cap S^k(i)$,
- y_i is an additonal factor vector for items,
- w_{ij} are global weights (so, no user-specific correlations), and
- c_{ij} are offsets to baseline estimates, depending on the implicit preference to j

All factors were optimized in an iterative process, using a test set of Netflix data, that converged after 30 iterations, and the so trained recommender system was tested against a validation set, and performed more than 10 percent better on the gold standard set than the original Netflix recommender algorithm.

While it might seems a bit dry to talk about the theoretical foundation of a recommender system, this knowledge is more than useful in answering the original question, "how do recommender systems influence video popularity". At least for Netflix, by far the largest competitor in commercial video-on-demand services, we can draw some conclusions from this mathematical theory:

First, because of the item neighborhood model, the users personal interest is heavily responsible for his recommendations. In contrast to a global "top N" list, where N videos are recommended to *all* users, now each user sees more or less videos no one else sees. It is therefore not as much as a goal for a video to make it to the "recommended" list, as it will not effect its popularity by that high amount – only a single user may see it. This should highly decrease the impact of recommender systems on video popularity.

Second, because of the latent factor model, global factors come still into play. Whatever global factors that are – recall: they are automatically learned – they impact the recommendations of all users. So, like search-engine-optimization (SEO) [Dav06] is common for websites to rank high in search engine results, it could become common for videos in video-on-demand service to somehow "latent-factor-optimize" to be recommended to more users. But, as the factors are learned automatically, and may even change from time to time, this "latent-factor-optimization" could become metaphysical or even purely random.

Finally, as implicit ratings made a significant difference, even in the simplest "rated"/"not rated" form, the already immense user data collection could even increase, as more and more implicit rating mechanisms are found – from obvious ratings like search query texts, to obscure ratings like the time the user stayed at a video after she watched it in total ("Maybe she still thinks about it? Maybe she likes it?"). This is a privacy concern, and should be discussed in detail, but it is beyond the bounds of this work to do so.

V. User-generated content video-on-demand

Web 2.0, with its dynamic content and participating users, and an increased bandwith available to these users, led to video-on-demand systems that, instead of serving professionally-produced content, focus on user-generated content so that users can upload, view, share and discuss videos and the system just provides a platform for that.

The allover changed content might cause changes in user behavior, so this type of video-on-demand system has been studied on its own. Results of this studies, of Cha *et al.* in 2009, are presented in V-A. To better understand recommendations in this field, the recommender system used in YouTube is described in V-B.

A. YouTube & Daum

Cha *et al.* crawled two big user-generated content video-on-demand providers, YouTube and Daum Videos[8], studied their crawling results, and published the findings in [CKR$^+$07], [CKR$^+$09].

In [CKR$^+$09] they first tried to tell commercial and user-generated content video-on-demand services apart. Therefore they used information from *IMDb*[9], *Netflix*, *Lovefilm* and *Yahoo! Movies*[10] for examples of commercial videos.

For *content creation*, they found that the *content production rate* differs for very active users: while the most active film director of commercial videos created about 100 movies in his lifetime, there are users that create more than 1000 videos in a few years in user-generated content services. As they studied *content length*, they found a reason for that: commercial videos average at 94 minutes, but user-generated content videos in median only account for 30 seconds (commercials) to 203 seconds (music videos). Furthermore *it occured, that video popularity did not correlate with video length*. Then, they saw that 50 percent of video uploads are performed between 8PM and 2AM, the rest evenly distributed.

On *content consumption*, it was revealed that popularity is more widespread on YouTube than on commercial platforms and, as shown in figure 9, that single videos overall don‘t have high view counts as for example videos from the Yahoo! platform. For both commercial and user-generated content they found an overall low user participation, but no difference between *rating behavior* for popular and unpopular videos. Lastly, although YouTube encourages embedding videos in different platforms, it was shown that only 3 percent of all views come from embedded videos, but that popular videos are more likely to be embedded and linked to.

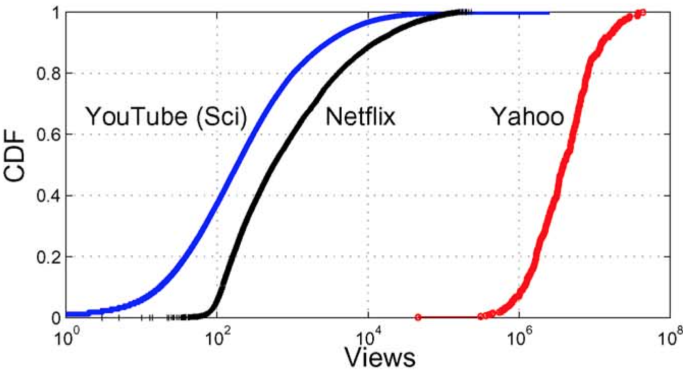

Figure 9. CDF of video popularity, sorted by views, for scientific YouTube videos, Netflix, and Yahoo!; from [CKR$^+$09]

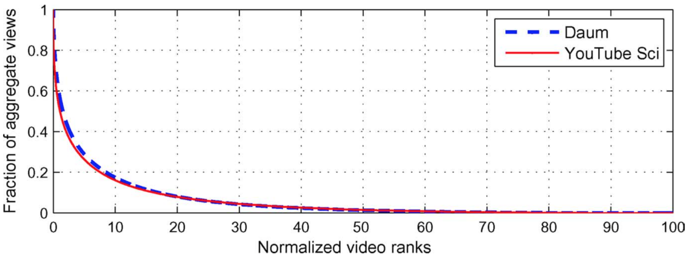

Figure 10. Video popularity of YouTube and Daum Videos, plotted against fraction of views; from [CKR$^+$09]

After finding the differences between user-generated content and commercial services, Cha *et al.* searched for the probability distribution underlying the observed user-generated content video popularities.

They observed an even more one-sided distribution than a *Pareto distribution* would suggest: just the top 10 percent (not 20 as the canonical 80-20 rule would state) of videos are responsible for 80 percent of all views (see figure 10). This stands in contrast to [YZZZ06] (see IV-A), where the Pareto distribution was seen as too strict. Since the latter studied a traditional video-on-demand system and the former studied user-generated content video-on-demand, this is seen as the first observed difference between the two. While surprising, this one-sided distribution *may be caused by recommender systems* that heavily favor already popular videos. Furthermore, they suggest efficient caching mechanisms that focus on making the top 10 percent fast, as this would serve 80 percent of requests.

Then they studied the distribution of *popular content* and found the videos distributed with a *power-law distribution with exponentional cutoff* ($h(x)$ not constant), which they found by curve-fitting as shown in figure 11. They blame the aging effect of videos and recommendations that show only a subset

[8] a popular user-generated content video-on-demand provider in Korea
[9] imdb.com
[10] yahoo.com/movies

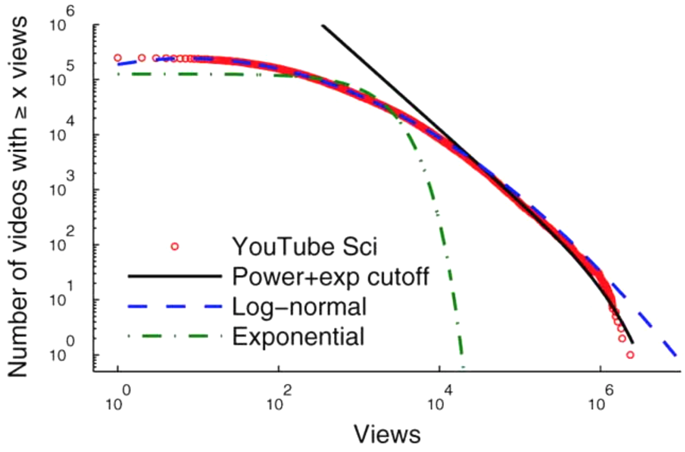

Figure 11. Probability distributions curve-fitted to the scientific YouTube data; from [CKR+09]

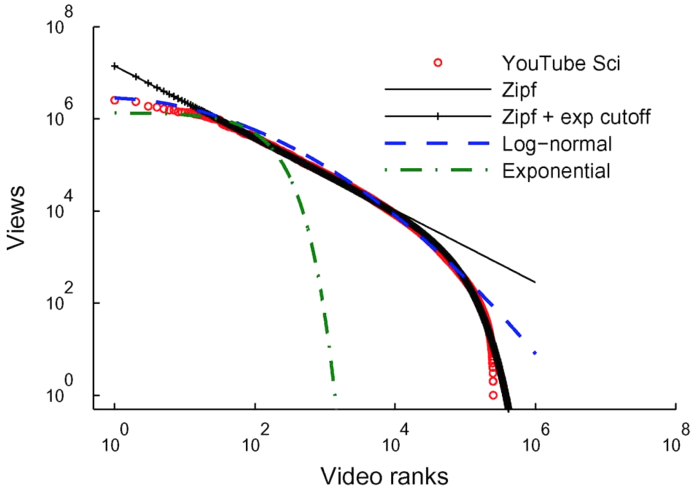

Figure 12. Probability distributions curve-fitted to the tail of unpopular YouTube science videos; from [CKR+09]

of videos for this distribution.

Even more interestingly, they studied the distribution of *unpopular content* and curve-fitted a *Zipf distribution with exponentional cutoff* as best option (see figure 12). They suspect the YouTube recommender system to be the bottleneck that causes the long tail of unpopular content – as users may never even see that this content exists.

Furthermore, they studied *video popularity changes over time*. It occurred that most-viewed videos were recently uploaded but lost popularity rapidly, in favor of fewer older videos. Looking at *daily access rates* they found drops at certain point (i.e. one month old, one year old) that could be caused by a video classifier in the recommendation algorithm that classifies differently after a certain amount of time. An interesting result is also, that video popularity is almost fully determined the first day after uploading, and that video popularity on the days 3+ correlates with a correlation coefficent around 0.9 with the popularity before – so, video popularity *can be predicted* knowing its popularity in the first two days. This could be used to create *intelligent caches* that cache only videos that are predicted to be popular (even if they are not that popular at the moment).

They then observed that the list of *most popular videos* changes rapidly over time: in average 12 percent of the videos stay the same from one day to the next, while 88 percent change. Also, they find that most videos gain 30 percent of their lifetime views on the first day, what makes this first day even more important.

Finally they looked at *content aliasing* – the same content uploaded multiple times – and *illegal uploads* and find that from 216 tested videos 184 had aliases (making 85% "alias rate") that, in many cases, make up for 100 times more views than the original video. Illegal uploads, namely videos that have been deleted because of copyright infringement, were found only in 0.02% of all videos (0.4% were deleted, from them only 5% because of copyright infringement).

Cha *et al.* provided deep analysis of current "real-life" data and served explanations for occuring phenomenons. They suggested probability distributions for video popularity, that can be used to design caches and adjust recommendation algorithms. Altough no effort was made to really change an existing user-generated content system or come up with a really new one that takes their results into account, this work is essential for future designs of such systems.

B. *The YouTube recommender system*

To provide a final understanding of recommender systems in user-generated content video-on-demand systems, the recommender system at YouTube is examined. Davidson *et al.*, the designers of the algorithm, gave some insight into it in [DLL+10].

Similar to [Kor08] (see IV-B) they use the users history for recommendations, but have more high-quality explicit and indirect ratings at their hands, like watched/favorited videos or the browsing/searching history.

In the terminology of Collaborative Filtering algorithms, they rely solely on user neighborhood models (unlike the combined item neighborhood + latent factor model of BellKor's Pragmatic Chaos). Starting at a *seed set* S of videos, they create recommendations using "daily co-visitation counts" c_{ij} to expand this seed set iteratively:

Let $r_{ij} = \frac{c_{ij}}{n(c_i,c_j)}$ be the *relatedness score* (with $n(c_i,c_j)$ a normalization, in the simplest case $n(c_i,c_j) = c_i c_j$) and R_i be the top related videos to the seed video i, the final recommendations are computed as

$$C_1(S) = \bigcup_{v_i \in S} R_i$$

$$C_n(S) = \bigcup_{v_i \in C_{n-1}(S)} R_i$$

$$C_{\text{final}}(S) = \left(\bigcup_{i=0}^{N} C_i(S) \right) \setminus S$$

This means, $C_{\text{final}}(S)$ contains videos reachable in N steps from the seed videos S (that come from the users history). The steps from seed videos to recommendation videos are tracked, to explain recommendations to the user ("you watched season 1, here is season 2").

Recommendations are ranked after generation, based on *video quality* (determined by number of views, ratings, comments, favorites and shares), *user specifity* (determined on base of the seed video that caused the recommendation), and *diversification* – as the system should recommend new and surprising videos as well.

In a side note, they describe that "recommendations account for about 60% of all video clicks from the home page", what indicates an immense impact of recommender systems.

While this work is a pure description of the algorithm, it allows to conclude important aspects. Just like [Kor08], the recommendations rely heavily on users personal interest and are absolute different from user to user. This should decrease the impact on the popularity of single videos by a large amount, as it is hard for them to reach a large user base, if they are not reachable from nearly every video that could be in a seed set. This as well would make caching a harder topic, as it is hard to predict which videos are reachable from a large set of seed videos. However, one must keep in mind that this personal "recommended list" is not the only way to find videos, and there might still be a "global top k" list, that could account for the findings in [CKR+09] that made 10 percent videos responsible for 80 percent views.

VI. Conclusion

In this work, the impact of recommender systems on video popularity in special-purpose/educational, general-purpose/commercial and user-generated content video-on-demand systems was studied by reviewing and discussing existing scientific work. It was furthermore examined how this could influence caching mechanisms.

Research suggests "top k"-caches in favor of cache-on-first-hit strategies, as video access rates are more than Zipf distributed. Furthermore, economic caches should cache prefixes, like the first 5 percent, only, as users tend to "scan" through videos and not watch them in total.

Recommender systems of Netflix and YouTube have been described as Collaborative Filtering systems, where Netflix relies on both item neighborhood and latent factor models, and YouTube simply uses a user neighborhood model.

Daily access rate distributions showed drops at distinct points, for what feature classifiers in recommender systems are blamed. Even more, videos on a global "recommended" list, performed 4 to 18 times better in access rates than videos not on these lists. However, personalized recommender systems may decrease this effect in the future.

Since technology evolves rapidly and recommender systems are an active field of research, it might be interesting to analyze current data and compare it to the already collected (but aged, the newest study, [CKR+09], is 5 years old by the time of writing) data. This is left open to further research.

References

[AKEV01] Jussara M. Almeida, Jeffrey Krueger, Derek L. Eager, and Mary K. Vernon. Analysis of Educational Media Server Workloads. In *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '01, pages 21–30. ACM, 2001.

[AKV01] Jussara M. Almeida, Jeffrey Krueger, and Mary K. Vernon. Characterization of User Access to Streaming Media Files. In *Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '01, pages 340–341. ACM, 2001.

[ASP99] Soam Acharya, Brian C Smith, and Peter Parnes. Characterizing User Access to Videos on the World Wide Web. In *Electronic Imaging*, pages 130–141. International Society for Optics and Photonics, 1999.

[AT84] Alfredo H-S Ang and Wilson H. Tang. *Probability Concepts in Engineering Planning and Design*. J. Wiley & Sons, 1984.

[BCF+99] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134. IEEE, 1999.

[Ben14] David Benson. The State of the Internet, 2nd Quarter, 2013 Report. Technical report, Akamai, 2014.

[CCB+04] Cristiano P. Costa, Italo S. Cunha, Alex Borges, Claudiney V. Ramos, Marcus M. Rocha, Jussara M. Almeida, and Berthier Ribeiro-Neto. Analyzing Client Interactivity in Streaming Media. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 534–543. ACM, 2004.

[Che94] Ann Louise Chervenak. *Tertiary Storage: An Evaluation of New Applications*. PhD thesis, University of California at Berkeley, 1994.

[Cis14] Cisco White Paper. Cisco Visual Networking Index: Forecast and Methodology, 2013–2018. Technical report, Cisco, 2014.

[CKR+07] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 1–14. ACM, 2007.

[CKR+09] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. Analyzing the Video Popularity Characteristics of Large-scale User Generated Content Systems. *IEEE/ACM Trans. Netw.*, 17(5):1357–1370, October 2009.

[CWVL01] Maureen Chesire, Alec Wolman, Geoffrey M Voelker, and Henry M Levy. Measurement and Analysis of a Streaming Media Workload. In *USITS*, volume 1, pages 1–1, 2001.

[Dav06] Harold Davis. *Search Engine Optimization*. O'Reilly Media, Inc., 2006.

[DLL+10] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The YouTube Video Recommendation System. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 293–296. ACM, 2010.

[DSS94] Asit Dan, Dinkar Sitaram, and Perwez Shahabuddin. Scheduling Policies for an On-Demand Video Server with Batching. In *Proceedings of the second ACM international conference on Multimedia*, pages 15–23. ACM, 1994.

[EVZ01] Derek Eager, Mary Vernon, and John Zahorjan. Minimizing Bandwidth Requirements for On-Demand Data Delivery. *Knowledge and Data Engineering, IEEE Transactions on*, 13(5):742–757, 2001.

[FMSL02] Julie Foertsch, Gregory Moses, John Strikwerda, and Mike Litzkow. Reversing the Lecture/Homework Paradigm Using eTEACH® Web-based Streaming Video Software. *Journal of Engineering Education*, 91(3):267–274, 2002.

[GDS+03] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-peer File-sharing Workload. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, pages 314–329. ACM, 2003.

[GNOT92] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, December 1992.

[Kor08] Yehuda Koren. Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[RHPL01] Lawrence A. Rowe, Diane Harley, Peter Pletcher, and Shannon Lawrence. BIBS: A Lecture Webcasting System. *Center for Studies in Higher Education*, 2001.

[RRD95] J.E. Redford, K.S. Ruttle, and T.M. Dobson. Video over ATM: experience from the Cambridge Interactive TV Trial. In *Image Processing, 1995. Proceedings., International Conference on*, volume 1, pages 1–4 vol.1, Oct 1995.

[YZZZ06] Hongliang Yu, Dongdong Zheng, Ben Y Zhao, and Weimin Zheng. Understanding User Behavior in Large-Scale Video-on-Demand Systems. In *ACM SIGOPS Operating Systems Review*, volume 40, pages 333–344. ACM, 2006.

[ZMS11] Sherali Zeadally, Hassnaa Moustafa, and Farhan Siddiqui. Internet protocol television (IPTV): architecture, trends, and challenges. *Systems Journal, IEEE*, 5(4):518–527, 2011.