Master's Thesis

Title

Analyzing Popularity Dynamics of YouTube Content and its Application to Content Cache Design

Supervisor

Professor Masayuki Murata

Author

Yuma Kitade

February 10th, 2015

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Analyzing Popularity Dynamics of YouTube Content and its Application to Content Cache Design

Yuma Kitade

Abstract

In recent years, UGC (User Generated Content) represented by YouTube has become popular service in the Internet. Video streaming delivery is severe for delay and necessary for data to arrive at constant timing. When delay increases by congestion of networks and servers, the quality of viewing that users sense deteriorates greatly. Therefore maintaining stable and good quality of viewing is important.

OTT (Over The Top) enterprises (YouTube, Skype, and so on) deliver content by using cache servers distributed in networks for reducing delay. It is necessary to control cache adequately for maintaining the stable quality of viewing, for example, we cache preferentially content which is expected that an effect to cache is high. Recently, many ISPs (Internet Service Provider) manage CDN (Content Delivery Network). Because the traffic of video streaming delivery is heavy, influence on networks is significant. Therefore, ISPs need to control cache adequately not only to improve quality of viewing but also to reduce traffic effectively in networks. Furthermore, Information-Centric Network (ICN, and one implementation of ICN is Content-Centric Network; CCN [1]) has been researched actively in recent years. CCN has been considered as a network architecture for the future because it is expected to reduce traffic greatly in networks. One of the functionalities of CCN is that it can implement routing without being aware of the location of nodes by routing using information about the content. The performance of CCN can be improved by creating the duplicates of content in any router along the delivery path, and by routing to the nearest node that has the desired content. To increase the advantages of CCN, efficient creation and placement of content caches is important. So far, cache has been

controlled based on past access pattern without distinguishing content. However, it is essentially desirable to control cache considering content popularity in the future.

Forecasting the future view count of each content is necessary for realizing it. Forecasting the popularity dynamics of UGC videos is more difficult than VoD (Video-on-Demand), video viewing service offered as commercial service by content providers, due to the incalculable number of videos and the diversity of content and popularity dynamics. Moreover, it is known that the view count of each video differs greatly. Therefore, there is a lot of research about the viewing trend and forecasting future view count by modeling the popularity dynamics of UGC. However, these prediction models can't be used to control cache, or executing them in networks is not realistic because control load is high.

In this paper, first, we analyze the viewing trend of YouTube from temporal and geographical viewpoint. Moreover, we propose a classifying method with k-means clustering which is often used as non-hierarchical cluster analysis to extract content of which a lot of audience are expected in the future easily from the pattern of early popularity dynamics. We show that we can get the rough trend of future popularity change in low control load. Furthermore, we apply a proposed classifying method to cache control, and show improving performance by a simulation.

Keywords

K-means Clustering
YouTube
Cache Control

Contents

List of Figures

1 Introduction

In recent years, UGC (User Generated Content) represented by YouTube [2] has become popular service in the Internet. Video streaming delivery is severe for delay and necessary for data to arrive at constant timing. When delay increases by congestion of networks and servers, the quality of viewing that users sense deteriorates greatly. Therefore maintaining stable and good quality of viewing is important.

OTT (Over The Top) enterprises (YouTube, Skype [3], and so on) deliver content by using cache servers distributed in networks for reducing delay. It is necessary to control cache adequately for maintaining the stable quality of viewing, for example, we cache preferentially content which is expected that an effect to cache is high. Recently, many ISPs (Internet Service Provider) manage CDN (Content Delivery Network). Because the traffic of video streaming delivery is heavy, influence on networks is significant. Therefore, ISPs need to control cache adequately not only to improve quality of viewing but also to reduce traffic effectively in networks. Furthermore, Information-Centric Network (ICN, and one implementation of ICN is Content-Centric Network; CCN [1]) has been researched actively in recent years. CCN has been considered as a network architecture for the future because it is expected to reduce traffic greatly in networks. One of the functionalities of CCN is that it can implement routing without being aware of the location of nodes by routing using information about the content. The performance of CCN can be improved by creating the duplicates of content in any router along the delivery path, and by routing to the nearest node that has the desired content. To increase the advantages of CCN, efficient creation and placement of content caches is important. So far, cache has been controlled based on past access pattern without distinguishing content. However, it is essentially desirable to control cache considering content popularity in the future.

Forecasting the future view count of each content is necessary for realizing it. Forecasting the popularity dynamics of UGC videos is more difficult than VoD (Video-on-Demand), video viewing service offered as commercial service by content providers, due to the incalculable number of videos and the diversity of content and popularity dynamics. Moreover, it is known that the view count of each video differs greatly. Therefore, the viewing trend of UGC is analyzed by a lot of research [4-8]. In [4], Cha et al. compared the viewing

trend of UGC with VoD by using the view count of plural days. In [5], Figueiredo et al. investigated the popularity dynamics of videos in popular ranking, videos which had been deleted by the infringement of copyright, and videos which had been selected by inputing random words in the search engine of YouTube. In [6], Broxton et al. analyzed the popularity dynamics pattern of YouTube videos which tend to be chosen in social methods (word of mouth, and so on).

Moreover, there is a lot of research in regard to forecasting future view count by modeling the popularity dynamics of UGC [9-17], as well. In [9], Szabo et al. paid their attention to have a linear correlation between early view count and view count thirty days later after uploaded in logarithmic graph, and described that future view count can be predicted by coordinating the parameters of a liner model in test set. In [10], Ghimire et al. modeled the popularity dynamics of content with Markov chain. In [11], Traverso et al. analyzed the access pattern of YouTube, and showed that conventional static Zipf model (IRM) can't consider the temporal change of demand frequency distribution. Traverso et al. divided content into six groups based on total view count and lifetime, and modeled in Poisson process rate changing respectively, and proposed a model (SNM) collected them. In [12], Gursun et al. analyzed the access pattern of YouTube, and showed that the daily view count change pattern of most content was classified roughly in accessed in long period and sporadically. Gursun et al. proposed a method forecasting future view count about each. In [13], Borghol et al. showed that the distribution of the number of access in the grain of one week of content randomly selected from YouTube differs in week of maximum view count, its past, and its future. Borghol et al. proposed a model based on the knowledge.

However, these prediction models can't be used to control cache, or executing them in networks is not realistic because a computational overhead is high. For example, in [9], it is necessary to calculate minutely a regression coefficient in a liner model between the time at which they want to know view count of content and the present time by using training set. In [12], it is necessary to store the number of days when it was viewed more than once in a year about each video. Moreover, because it is a method to predict view count in the year, it can't be used to predict future view count for cache control. In [13], it is necessary to find the week when view count about each video is maximum, and make

the distributions of past and future of the week by observing view count of all videos.

In this paper, first, we analyze the viewing trend of YouTube from temporal and geographical viewpoint. To be concrete, we get the knowledge by measurement analysis about the popularity dynamics of videos after uploaded, the difference of viewing trend between the weekday and the weekend, the difference of viewing trend in different days of each day of the week, the distribution of the day at which each video gains maximum view count, the trend about view count of videos passed long time after uploaded, the similarity of popular videos in each continent, language zone, and the entire world, and the viewing trend of in each country. Moreover, we propose a classifying method with k-means clustering which is often used as non-hierarchical cluster analysis to extract content of which a lot of audience are expected in the future easily from the pattern of early popularity dynamics. We show that we can get the rough trend of future popularity change in low control load. Furthermore, we apply a proposed classifying method to cache control, and show improving performance by a simulation.

2 Analysis of Viewing Trend in YouTube

In this section, we analyze the viewing trend of YouTube from temporal and geographical viewpoint by using the daily view count of YouTube videos and the lists of popular videos in each country. In addition, in this paper, we use the data of YouTube videos which we observed actually. We describe a method observing them and their details also.

2.1 Data Collection Method

We create the following three datasets of YouTube videos by using API offered by YouTube (YouTube Data API version 2.0) to analyze the temporal and geographical trend of YouTube videos and evaluate a proposed method.

- **recently uploaded** - Dataset for investigating the change of viewing trend after uploaded. This dataset consists of recently uploaded videos. Data acquisition period is from Sept. 6, 2014 to Feb. 4, 2015. The number of videos is 94,766.
- **popular videos in each country** - Dataset for investigating viewing trend in each country (total is 23 countries). This dataset consists of popular videos in each country. Data acquisition period is from Sept. 17, 2014 to Jan. 18, 2015. We describe the number of videos in each country in Subsection 2.3.
- **random** - Dataset for investigating the viewing trend of videos selected randomly. This dataset consists of random selected videos. Data acquisition period is from Oct. 20, 2014 to Jan. 14, 2015. The number of videos is 20,000.

2.1.1 Recently Uploaded

We investigate popularity dynamics after uploaded by using recently uploaded dataset. We keep collecting videos uploaded today and getting their daily view count to create the dataset. The processes of executing is the following. In addition, "videoset" is a set of videos of which we observe view count, and we suppose the present time to be H:m.

1. Request videos uploaded today (50 videos of maximum) to YouTube.
2. Add videos which "videoset" doesn't have among collected videos to "videoset", and their IDs and the dates on which they were uploaded to database.

3. Request the view count of videos which were uploaded in the present time (H:m) among videos in "videoset" to YouTube. Output them in a file.

4. Wait until H:$m + 1$, and return to 1.

2.1.2 Popular Videos in Each Country

Popular videos in each country dataset is dataset for investigating viewing trend in each country (total is 23 countries). We keep collecting popular videos uploaded today in each country and their daily view count to create the dataset. The processes to get daily view count of recently uploaded dataset perform to popular videos in each country.

2.1.3 Random

We create random dataset for investigating the viewing trend of videos selected randomly. There isn't API to randomly get videos in them of YouTube. Accordingly, we get randomly videos in the following method. In addition, "videoset" is a set of videos of which we observe view count.

1. Create random alphabetic string with three characters, and request one hundred videos to YouTube by using it as a search word.

2. Select randomly ten videos from videos got in 1.

3. Add videos which "videoset" doesn't have among selected videos to "videoset", and their IDs and the dates on which they were uploaded to database.

4. Go to 1 if the number of videos in "videoset" is lower than 20,000. In other case, finish the process.

Then, we acquire the daily view count of videos got in the method above mentioned. The processes of executing is the following. In addition, "videoset" is a set of the videos got in the method above mentioned, and we suppose that the present time is H:m.

1. Request the view count of videos which were uploaded in the present time (H:m) among videos in "videoset" to YouTube, and output them in a file.

2. Wait until H:$m + 1$, and return to 1.

2.2 Popularity Dynamics

First, we investigate popularity dynamics of videos after uploaded in YouTube. We investigate the view count of the s-th day after uploaded observed from Sept. 5, 2014 to Jan. 15, 2015 in recently uploaded dataset. Figure 1(a) is the CCDF (Complementary Cumulative Distribution Function) of view count of the s-th day after uploaded ($s = 1, 2, 3, 4, 5$), and Figure 1(b) is the CCDF of view count of the s-th day after uploaded ($s = 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 30$). In addition, the plots of these figures are randomly sampled with a probability of 0.001. From these figures, we observe that the difference of view count between two continuous days increases as s decreases. The difference is smaller as time passes after uploaded, and it is hardly seen twenty days later after uploaded. Moreover, it is clear that the graphs are gentle curve and their shapes are stable even though the elapsed day count changes. That is, videos just after uploaded obtain a lot of view count because audience is interested in them, but as time passes, interest of audience becomes weak. Therefore, view count decreases, and videos in which daily view count is zero increase.

Next, we investigate the difference of viewing trend between the weekday and the weekend. We analyze it by using daily view count observed from Oct. 29, 2014 (Wednesday) to Jan. 18, 2015 (Sunday) in random dataset. View count we can get is total view count of the entire world, and due to a time difference we can't gain simply view count in the weekday and the weekend. Because Japan is a county in which time zone is early, we use view count on Wednesday and Thursday to investigate viewing trend in the weekday, and view count on Sunday to investigate viewing trend in the weekend. Figure 2 shows the CCDF of view count in the weekday and the weekend. In addition, the number of samples on Wednesday, Thursday and Sunday is 60,059, 71,958 and 65,666 respectively, and the plots of the figures are randomly sampled with a probability of 0.01. This figure shows that the view count in the weekend is somewhat larger than in the weekday.

Moreover, we investigate the difference of viewing trend in different days of each day of the week by using three selected days of Wednesday, Thursday, and Sunday. Figure 3(a), 3(b), 3(c) show the comparison on Wednesday, Thursday and Sunday respectively. In addition, the plots of these figures are randomly sampled with a probability of 0.01.

These figures show that view count in different days of each day of the week don't change so much.

We investigate haw many days later after uploaded YouTube videos are most viewed. We show the CCDF of days in which videos gain maximum view count in Figure 4 by using daily view count observed from Sept. 5, 2014 to Jan. 15, 2015 in recently uploaded dataset. Due to a time difference, the observation time of data on the first day after upload (day 0) is short. Therefore, we exclude them. From this figure, it it clear that 90% of videos gain maximum view count a day later after uploaded, and about 97% of videos gain maximum view count within seven days after upload. Consequently, most videos gain maximum view count within a few days after uploaded.

Finally, we investigate how view count of videos passed long time after uploaded is maintained. Figure 5 is the CCDF of normalized view count, the value divided daily view count by the maximum view count in the observation period for each video, on $Y = 30$ or 60 days after uploaded. From this figure, the normalized view count 60 days after uploaded is fewer than 30 days after uploaded. From Figure 1(a), 1(b), view count decreases basically as time passes after uploaded, and from Figure 4, about 97% of videos gain maximum view count within seven days after uploaded. Consequently, The bigger Y is, the lower normalized view count is. When we pay our attention to the graph thirty days later after uploaded, the normalized view count of videos of about 90% is lower than 0.09, and the normalized view count of videos of about 99% is lower than 0.5. Therefore, view count of most videos thirty days after uploaded is much lower than those just after uploaded. If we can forecast a few videos maintained view count, it is expected that high performance content cache is realized. In this paper, we propose a method selecting content of which view count in the future is maintained based on the view count in early period by using k-means clustering. We describe the details and evaluate the method in Section 3.

2.3 Geographical Trend

We investigate the similarity of popular videos in each continent, language zone and the entire world. We use the lists of popular videos observed from Sept. 17, 2014 to Jan. 18, 2015 in popular videos in each country dataset. Abbreviations and the number of popular

videos we got are the following. Because the update frequency of popular videos list in each country differs, the number of observed videos is different. Accordingly, we compare a ratio of the number of agreement of popular videos.

- **Asia** - Hong Kong (HK, 923), India (IN, 1,947), Israel (IL, 834), Japan (JP, 7,962), Korea (KR, 3,345), Taiwan (TW, 2,264)
- **Europe** - the Czech Republic (CZ, 828), France (FR, 2,670), Germany (DE, 6,570), the United Kingdom (GB, 6,298), Netherlands (NL, 1,839), Ireland (IE, 788) , Italy (IT, 2,033), Poland (PL, 2,177), Russia (RU, 5,418), Spain (ES, 2,050), Sweden (SE, 904)
- **North America** - Canada (CA, 3,744), Mexico (MX, 4,794), the United States of America (US, 2,436)
- **South America** - Brazil (BR, 9,070)
- **Oceania** - Australia (AU, 1,425), New Zealand(NZ, 711)

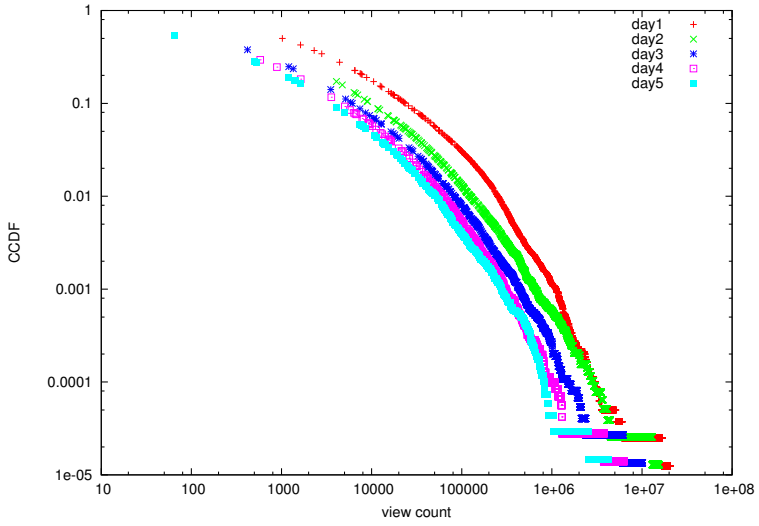
Figure 6 shows the ratio of the number of countries in which each video is listed in the popular-video list in each Asia, Europe, English zone and the entire world (23 countries in which we can get a list of popular videos in YouTube Data API version 2). For example, $x = 2$ of "English" shows the ratio of videos included in popular videos lists of two countries in English zone . In addition, English zone indicates Ireland, the United Kingdom, Canada, the United States of America, Australia and New Zealand. Asia and English zone have six countries, but Europe has eleven countries, and the whole world has twenty three countries. Therefore, the number of countries is different. We take the following processes in Europe and the entire world so that the number of countries is uniform.

1. Select six countries randomly, and count how many their lists include each popular video of the six countries. Repeat this processes in ten times.
2. Sum up the number of popular videos in x countries ($x = 1, 2, 3, 4, 5, 6$) of each trial, and divide them by their total to calculate rates.

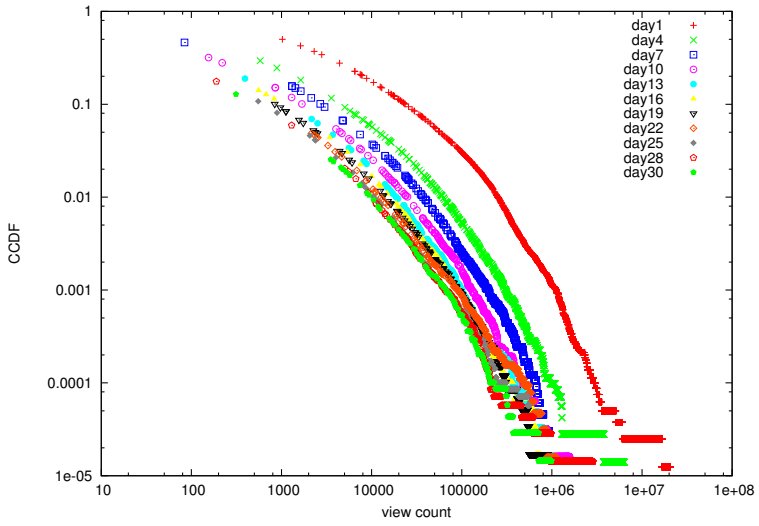
The figure shows that there are more common popular videos in language zone than in continent of Asia, Europe, and so on. Therefore, it indicates that viewed videos depend

on the language of the own country strongly. From this result, it is expected to perform more efficient to control caches cooperatedly in common language zone than in continent.

We investigate the average, median and standard deviation of view count of popular videos in each country on Jan. 15, 2015 to get the knowledge of viewing trend in each country. The number of samples differs in each country, and 565 of New Zealand is minimum. To make the number of samples uniform, we calculate average view count of each video from Sept. 17, 2014 to Jan. 15, 2015, and use superior 565 videos. In addition, view count of each videos is total in the entire world. Figure 7(a), 7(b), 7(c) show the comparison of average, median and standard deviation of view count of popular videos in each country respectively. From these figures, popular videos in countries in English zone tend to have high average and median of view count. Accordingly, as Figure 6 also indicates, this result indicates that popular videos in English zone, that is, popular English videos, are viewed in many countries. Moreover, the figure shows that the average view count in Asia is low, and that in Europe is higher than that in Asia. However, the median of view count doesn't differ so much between Asia and Europe.



(a) $s = 1, 2, 3, 4, 5$



(b) $s = 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 30$

Figure 1: CCDF of view count of s-th day after uploaded

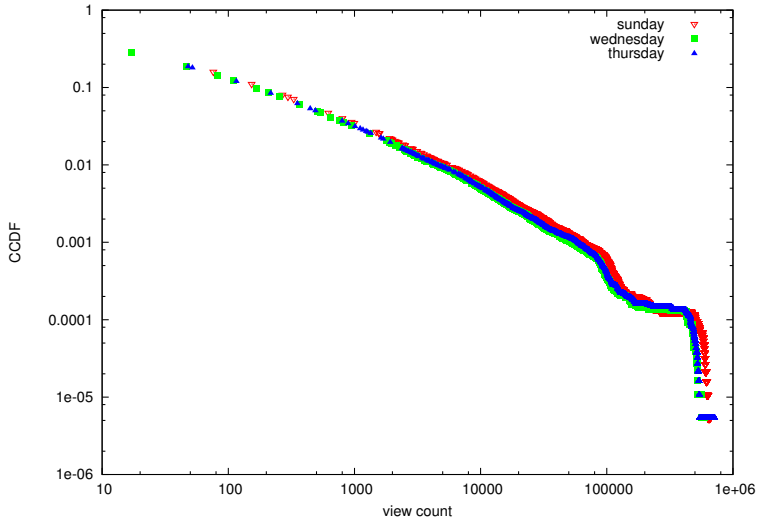
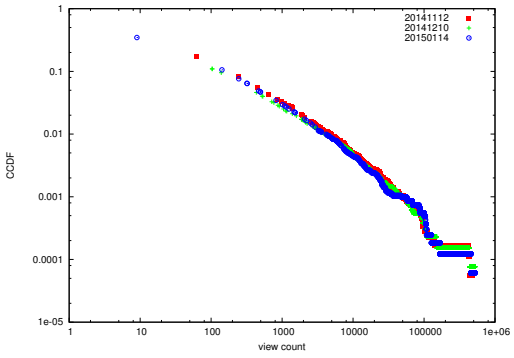
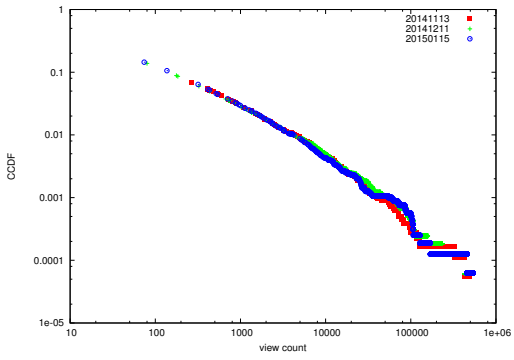


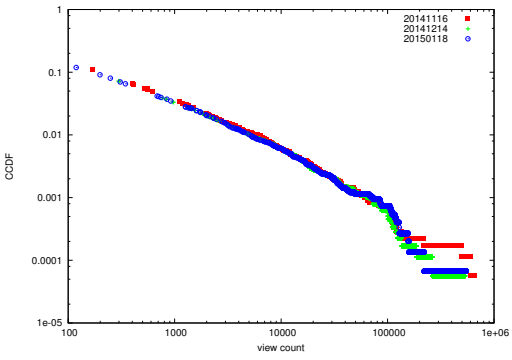
Figure 2: Difference of viewing trend between the weekday and the weekend



(a) on Wednesday



(b) on Thursday



(c) on Sunday

Figure 3: CCDF of view count in different days of each day of the week

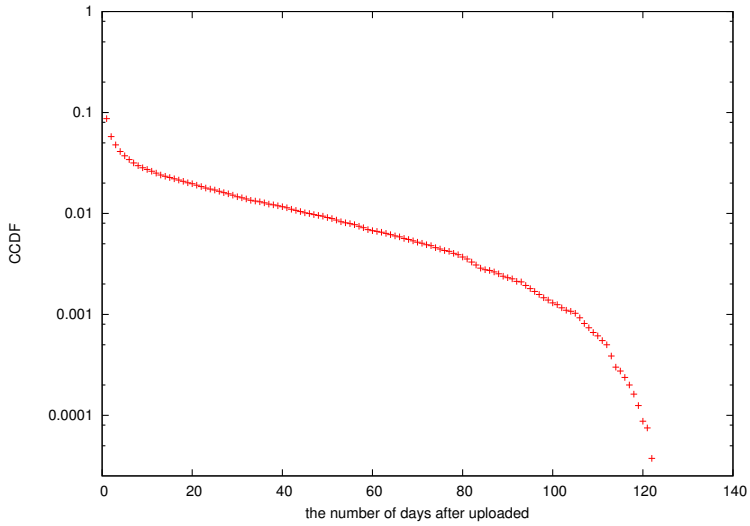


Figure 4: CCDF of day at which each video gains maximum view count

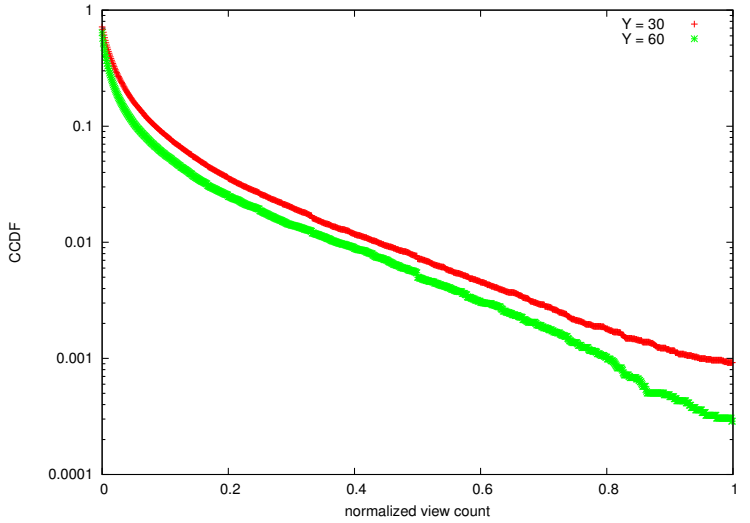


Figure 5: CCDF of normalized view count on Y days after uploaded ($Y = 30, 60$)

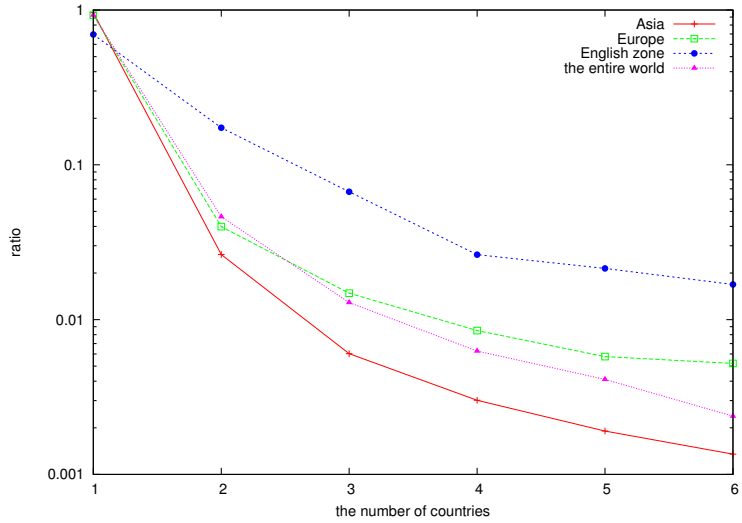
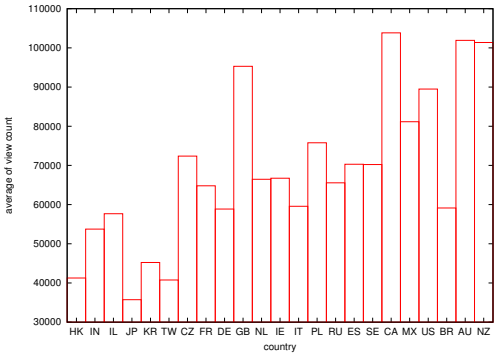
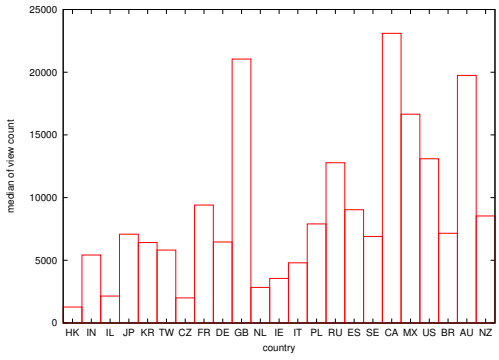


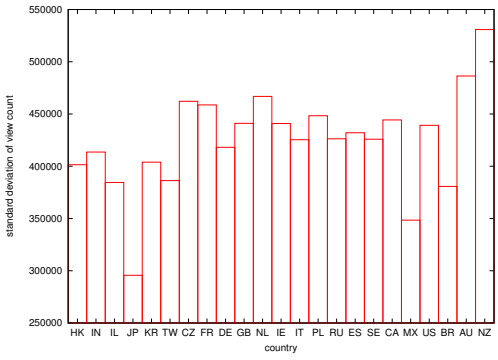
Figure 6: Ratio of the number of countries in which each video is listed in the popular-video list in each Asia, Europe, English zone and the entire world



(a) Average



(b) Median



(c) Standard deviation

Figure 7: Viewing trend of popular videos in each country

3 Classifying YouTube Videos based on Popularity Dynamics using K-means Clustering

In this paper, we propose to classify content with k-means clustering which is often used as non-hierarchical clustering analysis to easily extract content of which a lot of audience are expected in long time span from the pattern of early change of view count. In this section, we describe and evaluate the proposed classifying method for YouTube videos. We explain the outline of the proposed method in Subsection 3.1. We show that the proposed method can get general tendency of future view count change with a low control load in Subsection 3.2.

3.1 Outline of Proposed Classifying Method

We propose the method to extract content which is expected to maintain popularity in long time span from the change pattern of early popularity dynamics. To be concrete, we divide daily view count by the maximum view count in first x days about each video (the total number is v). From Figure 4, because most videos gain maximum view count within a few days after uploaded, we can get change pattern of early popularity in this process. Then we get the vectors of x dimension that have values of $0 \leq s \leq 1$. We classify videos with k-means clustering by using the vector of the view count in first x days of v videos.

It is widely known that clustering result of k-means clustering depends heavily on initial cluster. In this paper, we generate initial cluster with k-means++ [18]. K-means++ is the method to disperse centers of gravity as much as possible. First, we choose a member at random and make it the center of gravity of first cluster, then repeat to select a center of gravity of cluster with probability in proportion to square of distance to the nearest center of gravity of cluster about each member until k centers of gravity are selected.

They say that future view count and early view count have correlation in [9]. Therefore, it is predicted that videos which keep popular just after uploaded maintain view count in the future. We carry out the following processes to select the cluster that has many videos which keep popular just after uploaded.

1. Calculate $a_{i,k}$, the average of normalized view count among x days after uploaded,

values obtained by dividing daily view count by the maximum view count among x days after uploaded, for each video i when the cluster count is set to k.

2. Calculate A_k, the average of $a_{i,k}$ in each cluster.

3. Select the cluster which has maximum A_k as a representative of cluster k.

It is important to set up the appropriate value of k because the result of k-means clustering depends heavily on the number of clusters k. We investigate the relation between the number of clusters k and A_k in Subsection 3.2 and consider about a method to select the appropriate value of k. Moreover, we evaluate performance of the proposed method mentioned above with the view count of YouTube and confirm effectiveness in Subsection 3.2.

3.2 Numerical Result

In this subsection, we evaluate in simulation that a proposed method can extract videos which is expected to maintain the view count for long time period by a low computation overhead. We show the result of applying a proposed method to view count among x days after uploaded ($x = 5, 10, 20, 30$) on Jan. 15, 2015 with recently uploaded dataset from Sept. 6, 2014 to Jan. 15, 2015. We exclude videos of which there is a loss of view count among 30 days after upload because there is the case that we can't acquire view count in trouble of a server offering view count of YouTube. As a result, we use data of 23,934 videos. Then we supplement the remaining losses by calculating average.

Figure 8(a), 8(b) show the change of A_k and the number of members of the cluster with maximum A_k respectively when we change the number of clusters. From these figures, we can know that the value of A_k increases and the number of members of cluster with maximum A_k decreases by increasing k. Moreover the longer x is, the smaller A_k, and the number of member of cluster with maximum A_k. This is because view count decreases as days pass after uploaded as shown in Figure 1(a), 1(b) and the longer x is, the smaller A_k becomes. We can say that it is good to increase the number of clusters and increase the maximum A_k to improve the effect of extracting videos with maintaining popularity in long time period. However from about $k = 15$ the increase rate of A_k and the decrease

rate of the number of members in a cluster which has maximum A_k decrease by increasing k.

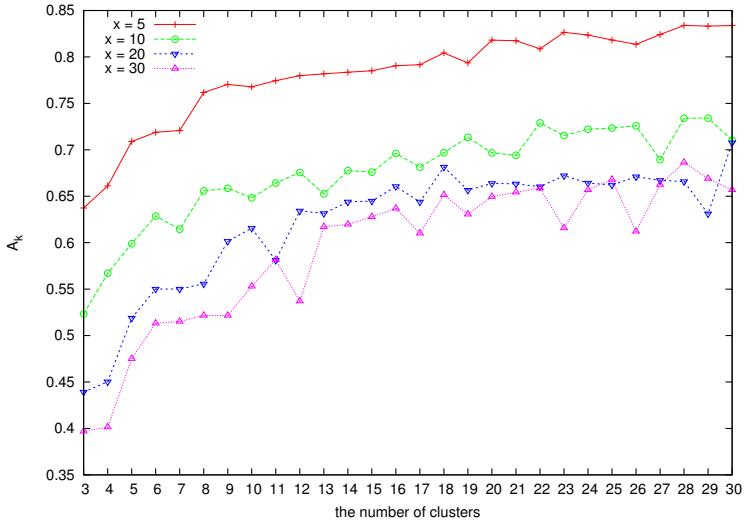
As shown in Figure 8(a), the larger the number of clusters is, we can extract a cluster which has more videos expected to maintain popularity from now on. However, it must be a method with a low calculation load to make use of method for controlling cache priority. Figure 9 shows the comparison of executive time when we change the number of clusters and the number of days used for clustering x. In addition, it shows the average of results when we change the seed of random number of k-means++ and try five times. We use a server of which the operating system is Mac OS X 10.9.5, the CPU is 2.93 GHz Intel Xeon X5670 processors, and the memory capacity is 96 GB. The program consists of storing the view count of each video, normalizing view count for clustering, excluding videos which nobody watches for a period of clustering x, setting up initial centers of gravity by k-means++, currying out k-means clustering, and writing clustering result on a file. The figure shows that the larger the number of clusters and the longer x is, the longer executive time becomes. It is a result of 23,934 videos. Thus, it is expected that it takes more time to apply a proposed method to cache priority control because it is necessary to make more videos clustered. Thus, it is desirable for this value to be small.

Figure 10(a), 10(b), 10(c), 10(d) show CDF (Cumulative Distribution Function) of result of normalizing view count 60 days later after uploaded by the maximum view count in a observation period of each member of clusters about the cluster of which A_k is maximum in each k when we change the number of clusters k. We assume the cross axle the days that passed from upload, and show the average of result of normalizing view count in each day by the maximum view count in a observation period about each member of the cluster of which A_k is maximum in Figure 11(a), 11(b), 11(c), 11(d). These figures show that it is practicable to abstract a cluster which includes many videos which can maintain popularity from now on as the number of cluster k becomes larger regardless of the number of days for clustering x. However, the improvement of performance is hardly seen from about $k = 15$. Moreover it is shown that we can abstract a cluster holding a lot of videos which can maintain popularity as we lengthen x.

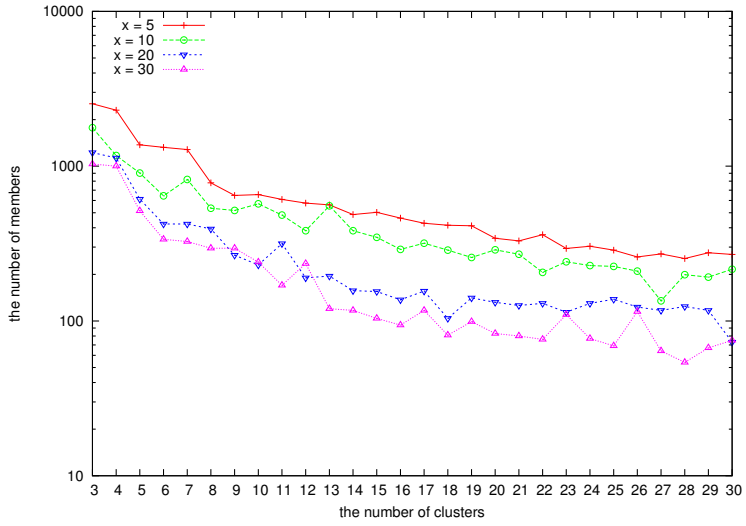
Figure 8(a) and 8(b) show that from about $k = 15$ the increase rate of A_k and the decrease rate of the number of members in a cluster which has maximum A_k decrease by

increasing k. From Figure 9, we can know that executive time is comparatively short in $k = 15$. Therefore, we can say that the desirable number of clusters is $k = 15$. Then, we analyze the clustering result of $k = 15$. In addition, here is the only result of $x = 30$. First, we analyze the trend of normalized view count. Figure 12(a) shows the comparison of centers of gravity in each cluster. Figure 12(b) shows the CDF of result of normalizing view count 60 days later after uploaded by the maximum view count in a observation period about each member of clusters. Figure 12(c) shows the average of result of normalizing view count in each day by maximum view count in a observation period about each member of clusters, and we assume the cross axle days that passed from upload. In addition, numbers in brackets of explanatory notes indicate the number of members in each cluster, and a cluster number is descending order of A_k. As these Figures show, a cluster which has maximum value of A_k, that is, contains a lot of videos that is expected to be maintain popularity in the future, contains a lot of them as expected.

Then, we analyze the trend of view count. Figure 13(a) shows the comparison of CDF of view count 60 days later after uploaded in each cluster. In addition, the plots of this figures are randomly sampled with a probability of 0.01 when the value of x-axis is larger than 1000. Figure 13(b) shows the change of average view count in each cluster, and cross axle shows days that passed from upload. From these figures, we can know that a cluster which has maximum A_k is expected not only high normalized view count but also a lot of view count in the future. Therefore, it is clear that a proposed method can abstract videos worth caching to be maintained popularity in the future.



(a) A_k



(b) the number of members of cluster of that A_k is maximum

Figure 8: Trend about A_k when the number of clusters is changed

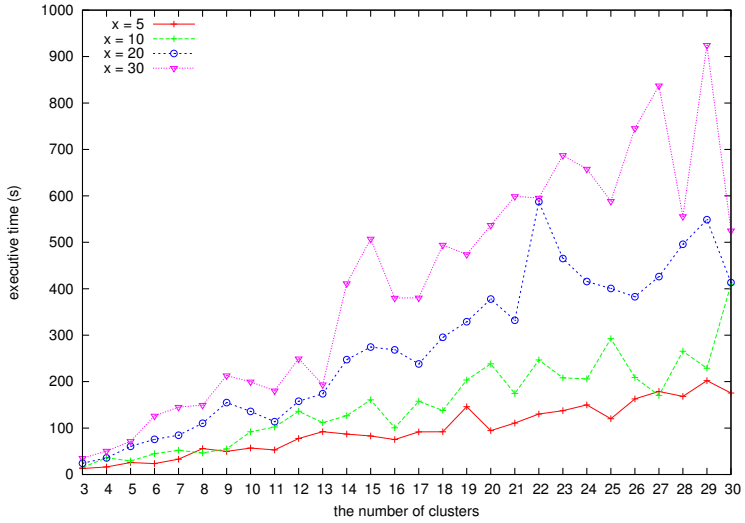
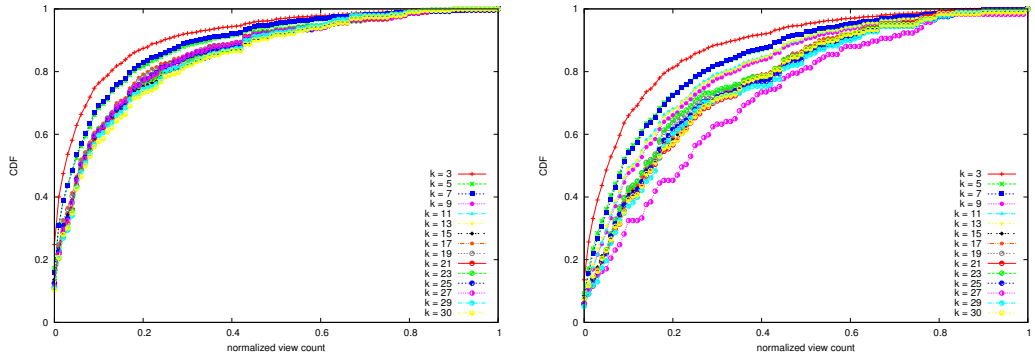
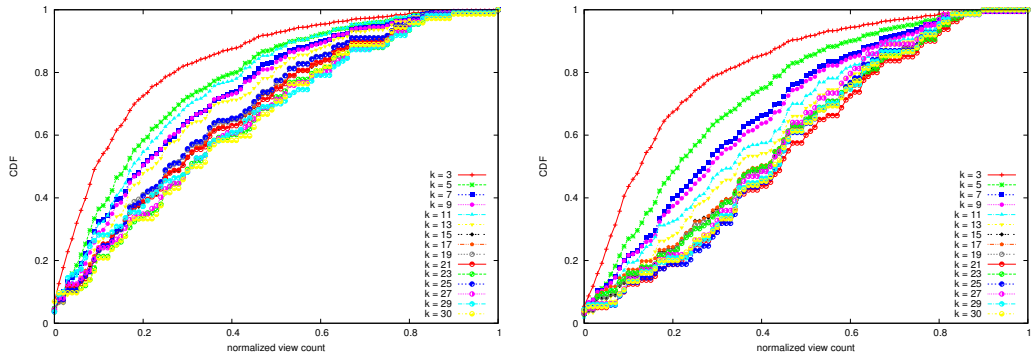


Figure 9: Executive time when the number of clusters and x are changed

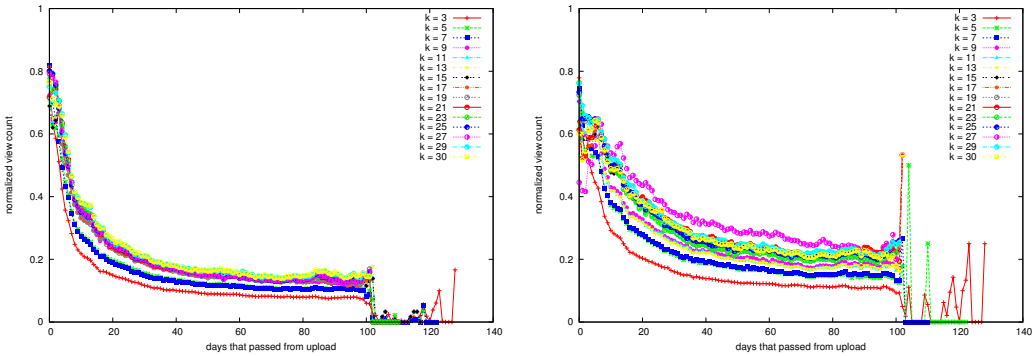


(a) $x = 5$ (b) $x = 10$

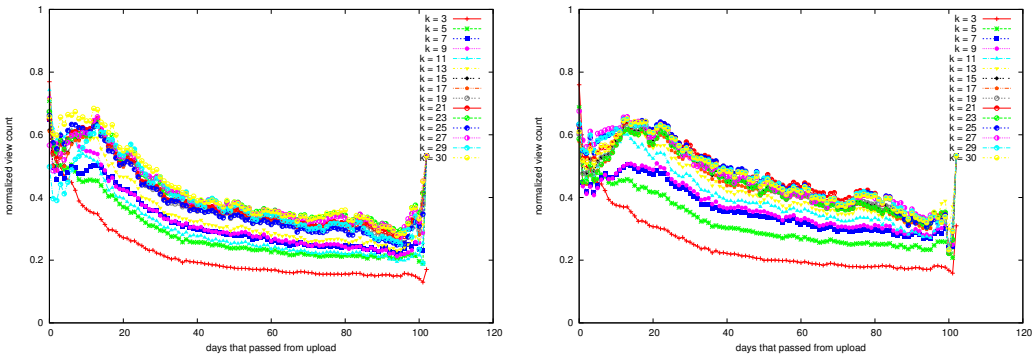


(c) $x = 20$ (d) $x = 30$

Figure 10: CDF of result of normalizing view count 60 days later after uploaded by the maximum view count in a observation period of each member of clusters about the cluster of which A_k is maximum

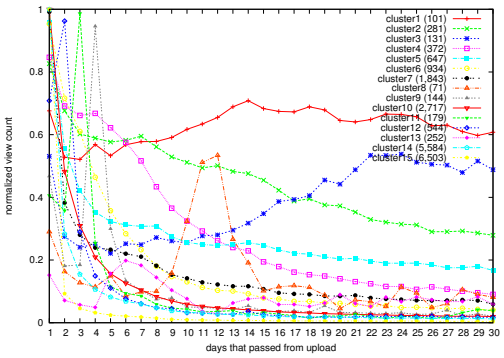


(a) $x = 5$ (b) $x = 10$

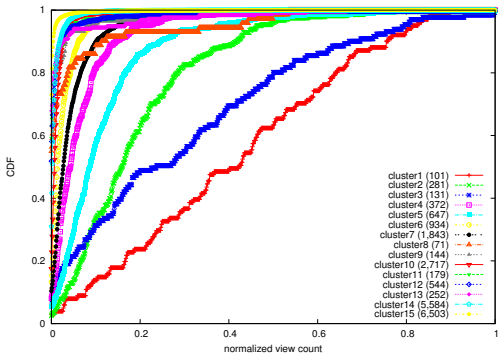


(c) $x = 20$ (d) $x = 30$

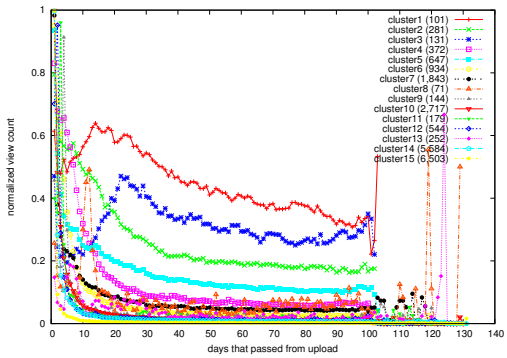
Figure 11: Average of result of normalizing view count in each day by the maximum view count in a observation period about each member of the cluster of which A_k is maximum



(a) centers of gravity

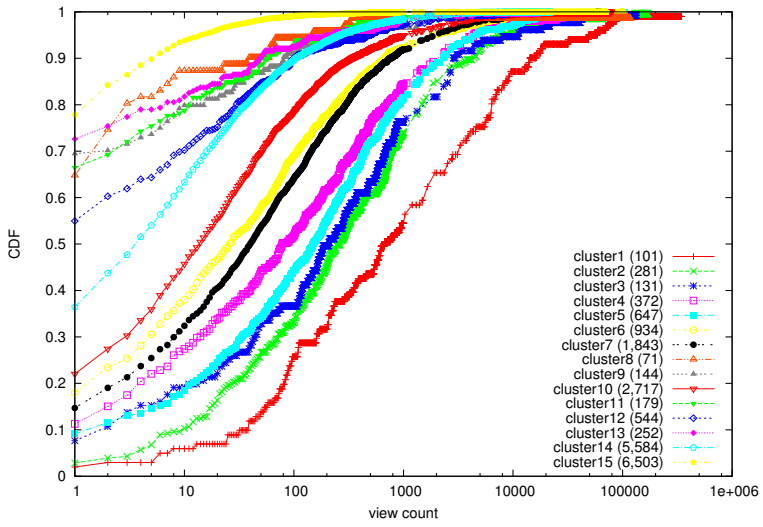


(b) CDF of result of normalizing view count 60 days later after uploaded by the maximum view count in a observation period about each member of clusters

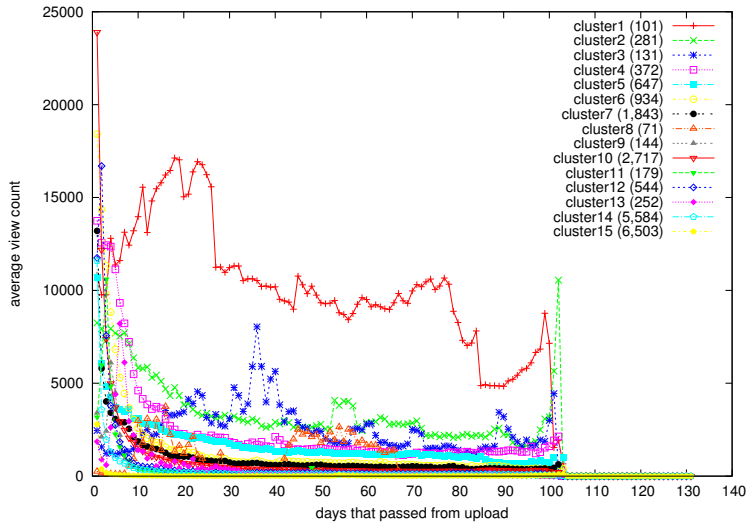


(c) average of result of normalizing view count in each day by maximum view count in a observation period about each member of clusters

Figure 12: Trend about normalized view count of clustering result ($k = 15, x = 30$)



(a) CDF of view count 60 days later after uploaded



(b) average view count

Figure 13: Trend about view count of clustering result ($k = 15, x = 30$)

4 Application to Cache Control

In this section, we describe a method to apply a classifying method proposed in Section 3 to cache control. Moreover, we evaluate efficiency by using observed data of YouTube videos.

4.1 Outline of Proposed Cache Control

We apply a classifying method proposed in Section 3 to cache control. Cache control is divided into (1) judgment if content which isn't stored is cached, and (2) judgment which content is replaced when cache is full. We apply a proposed classifying method to (1), and use LRU (Least Recently Used) in (2).

We describe detailed processes of (1). First, YouTube videos are divided into the following three set.

- **A** - a set of videos under x days after they were first requested.
- **B** - a set of videos which k-means clustering can apply to, over x days after they were first requested.
- **G** - a set of videos which preferentially are cached.

We execute a proposed cache control method in the boundary of days. First, we move videos in x days after they were first requested from **A** to **B**. Then, we classify videos in **B** by using k-means clustering, and set **A** and videos classified in the cluster with the maximum average of x centers of gravity in **G**. When content servers or routers with cache receive content which they don't store, they cache it if **G** include it, and cache it in a probability of 0.1 in other case.

4.2 Simulation Environment

We suppose that there is a router with cache between a content server and a client. Content requests are created by using recently uploaded dataset from Oct. 6, 2014 to Feb. 4, 2015. We exclude videos of which there is a loss of view count among 10 days after upload because there is the case that we can't acquire view count in trouble of a server offering

view count of YouTube. As a result, we use data of 18,501 videos. Then we supplement the remaining losses by calculating average. We suppose that view count in t-th day is X_t, and the view count of video m in t-th day is $X_{m,t}$. The one-hundredth of X_t requests is created in t day. The viewed video m is selected in a probability in proportion on $X_{m,t}$ in videos with $X_{m,t} > 0$. Cache size is set in 5% of total videos. We classify videos with k-means clustering on condition that the number of days used for clustering x is 10, and the number of clusters is 15. We compare a proposed cache control method with the case that a router always cache videos when it receives videos which aren't stored in it.

As performance metrics, we use the cache hit ratio of each day and the relative cache hit ratio of each video. The relative cache hit ratio of each video is the result of dividing the cache hit ratio of a proposed cache control method by it of a comparative method.

4.3 Numerical Result

Figure 14(a) shows daily cache hit ratio, and Figure 14(b) shows cache hit ratio from the beginning of simulation to a day x-axis indicates. From these figures, we can know that daily cache hit ratio decreases until about 25 days from the beginning of evaluation. Because the addition of videos stops in Oct. 27, 2014, the fall of cache hit ratio due to increase of videos stops and cache hit ratio become stable. A proposed cache control method improves cache hit ratio.

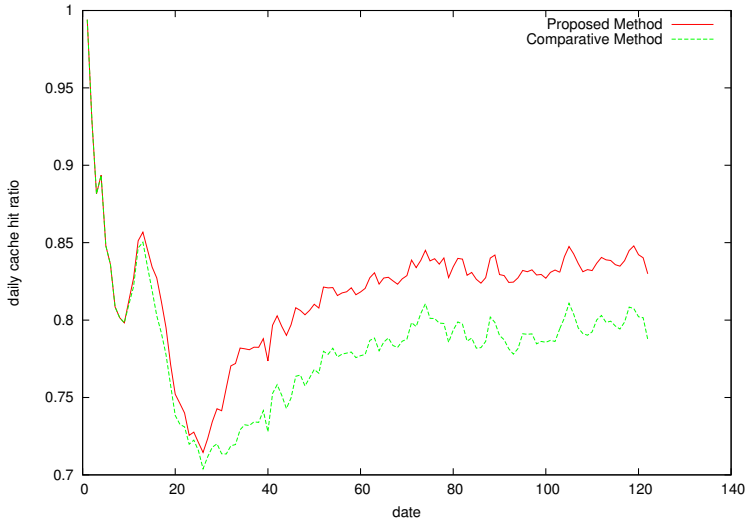
In Figure 15, the relative cache hit ratio of each video is shown. Figure 15(a) shows the case that x-axis indicates popularity rank based on average view count. Figure 15(b) shows the case that x-axis indicates popularity rank based on total view count. From these figures, a proposed cache control method maintains equivalent cache hit ratio in videos from 1st to 100-th in popularity rank, and improves cache hit ratio in videos from 100-th to 1000-th in popularity rank. Therefore, these result shows that a proposed cache control method can extract popular videos, and cache preferentially them.

Figure 16(a) shows the CCDF of cache hit ratio of each video. Figure 16(b) shows it in the case of extracting the part that cache hit ratio is high. These figures show that the number of videos in the part that cache hit ratio is high increases by a proposed cache control method. To be concrete, a proposed cache control method increases ratio of videos of which cache hit ratio is higher than 60% by about 10%.

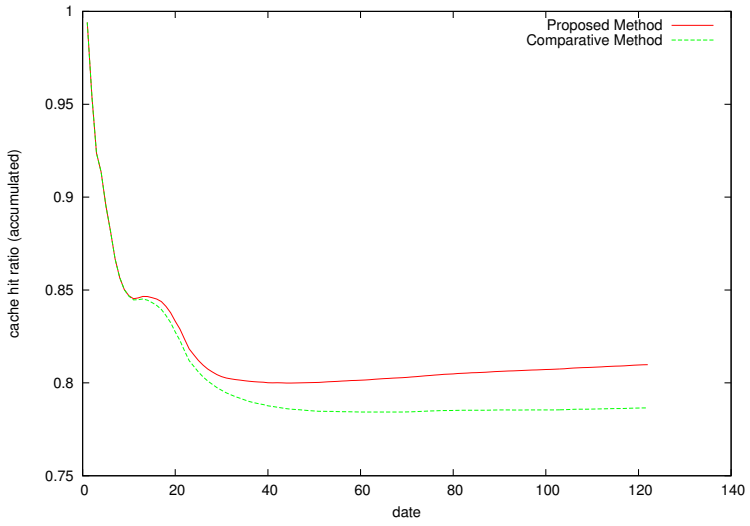
4.4 Other Application of Proposed Classifying Method

In Section 3, we proposed a method extracting content which is maintained popularity in long period from the change pattern of view count in early period. Then, we applied it to cache control. However, cache control is only one application example of a proposed classifying method. Because a proposed classifying method is a method to extract early content maintaining popularity in long period, there is a lot of application.

For example, we can apply a proposed classifying method to extract desirable website for on-line advertising. Advertisers desire that their advertising gathers the attention of a lot of people. They can extract website which a lot of people visit in long period by using a proposed classifying method with the number of visits of each website.

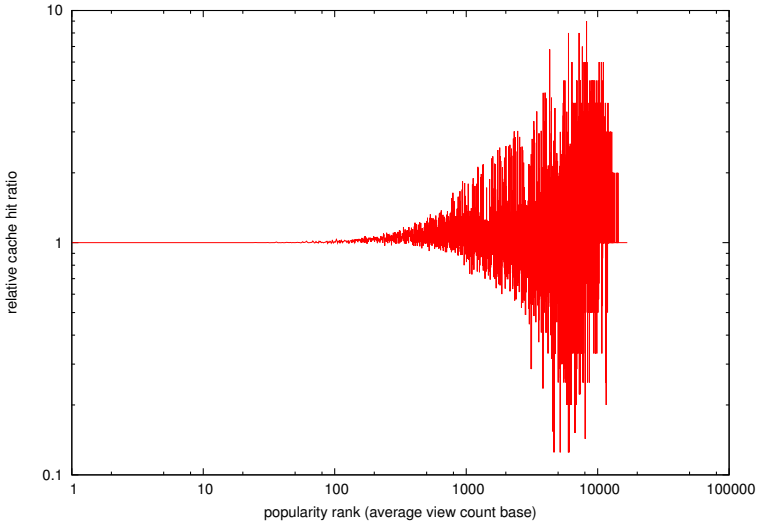


(a) daily

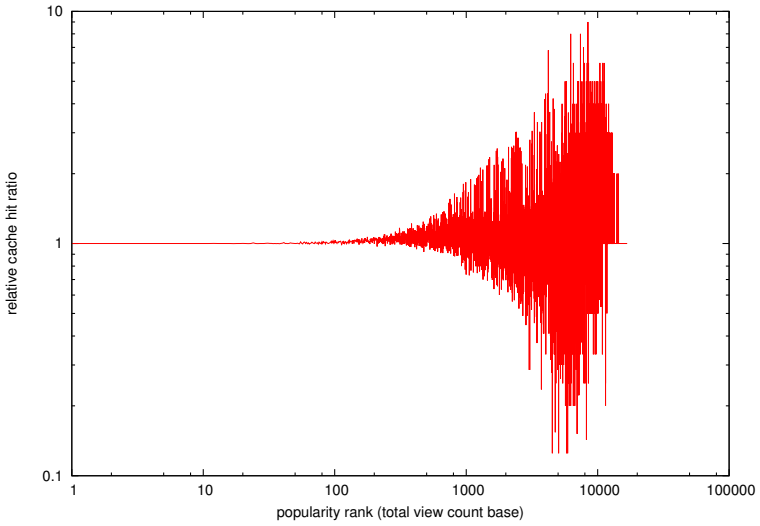


(b) accumulated

Figure 14: Cache hit ratio

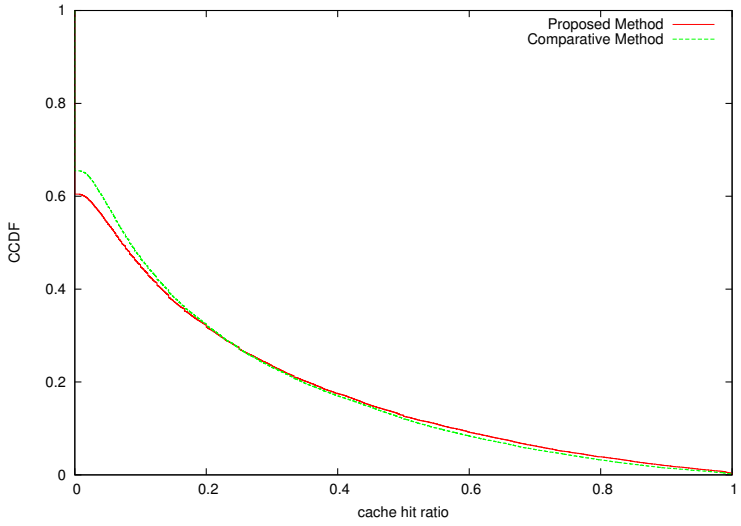


(a) average view count

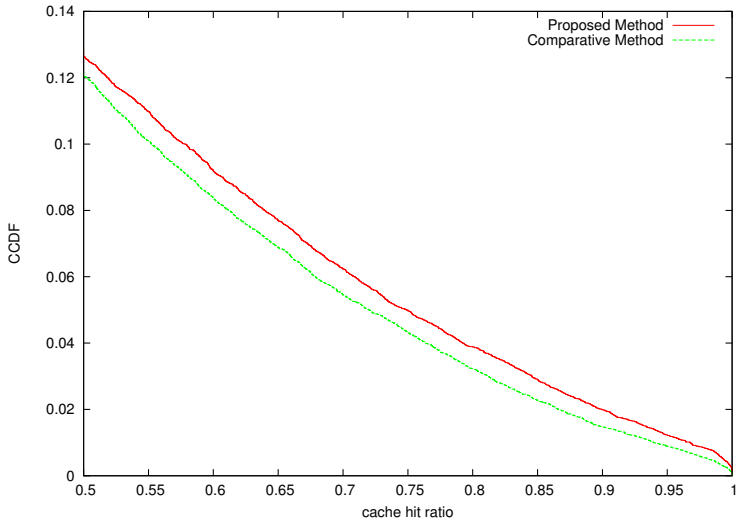


(b) total view count

Figure 15: Relative cache hit ratio of each video



(a) the whole



(b) the part that cache hit ratio is high

Figure 16: CCDF of cache hit ratio of each video

5 Conclusion

In this paper, first, we analyzed the viewing trend of YouTube from temporal and geographical viewpoint and got the knowledge about them. Moreover, we proposed a classifying method with k-means clustering which was often used as non-hierarchical cluster analysis to extract content of which a lot of audience were expected in the future easily from the pattern of early popularity dynamics. We showed that the method could extract content maintaining popularity in long time span in a low computation overhead. Furthermore, we applied a proposed classifying method to cache control, and showed that a proposed cache control method could extract popular videos, and cache preferentially them by a simulation.

As future work, we investigate optimum parameters in cache control. and as we described in Subsection 4.4, we apply a proposed classifying method to others.

Acknowledgements

This paper would not be accomplished without many people's advice and support, and we experienced the exciting and instructive research life thanks to them. Foremost, I would like to express the deepest appreciation to my supervisor, Professor Masayuki Murata of Osaka University, for his accurate advice, guidance and continuous encouragement. I would like to express my gratitude to Dr. Noriaki Kamiyama of NTT Network Technology Laboratories. This paper would not be accomplished without his exact advice and kind support. I am deeply grateful to Professor Shingo Ata of Osaka City University, for his elaborated guidance and indication getting to the point.

Moreover, I would like to offer my special thanks to Associate Professor Shin'ichi Arakawa, Assistant Professor Yuichi Ohsita, and Assistant Professor Daichi Kominami of Osaka University, for their precise comments and encouragement.

Furthermore, I would like to thank Mr. Atsushi Ooka. Discussion with him provided me a lot of knowledge and awareness. I want to thank Ms. Kazama, for her cordial support.

Finally, I'm grateful to my friends and colleagues in the Department of Information Networking, Graduate School of Information Science and Technology of Osaka University for their support.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of ACM CoNEXT 2009*, pp. 1–12, Dec. 2009.

[2] "YouTube." https://www.youtube.com/.

[3] "Skype." http://www.skype.com/.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 1357–1370, Oct. 2009.

[5] F. Figueiredo, F. Benevenuto, and J. M. Almeida, "The tube over time: characterizing popularity growth of YouTube videos," in *proceedings of fourth ACM international conference on Web search and data mining (WSDM 2011)*, pp. 745–754, Feb. 2011.

[6] T. Broxton, Y. Interian, J. Vaver, and M. Wattenhofer, "Catching a viral video," *Intelligent Information Systems*, vol. 40, pp. 241–259, Apr. 2013.

[7] J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani, "Characterizing and modeling the dynamics of online popularity," *Physical Review*, vol. 105, p. 158701, Oct. 2010.

[8] F. Olmos, B. Kauffmann, A. Simonian, and Y. Carlinet, "Catalog dynamics: Impact of content publishing and perishing on the performance of a lru cache," in *Proceedings of 26th International Teletraffic Congress (ITC)*, pp. 1–9, Sept. 2014.

[9] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Communications of the ACM*, vol. 53, pp. 80–88, Aug. 2010.

[10] J. Ghimire, M. Mani, and N. Crespi, "Modeling content hotness dynamics in networks," in *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2010)*, pp. 428–431, July 2010.

[11] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today's content caching: why it matters and how to model it," *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 5–12, Oct. 2013.

[12] G. Gursun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *Proceedings of IEEE INFOCOM Mini-Conference 2011*, pp. 16–20, Apr. 2011.

[13] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti, "Characterizing and modelling popularity of user-generated videos," *Performance Evaluation*, vol. 68, pp. 1037–1055, Nov. 2011.

[14] G. Chatzopoulou, C. Sheng, and M. Faloutsos, "A first step towards understanding popularity in YouTube," in *Proceedings of INFOCOM IEEE Conference on Computer Communications Workshops*, pp. 1–6, Mar. 2010.

[15] K. Lerman and T. Hogg, "Using a model of social dynamics to predict popularity of news," in *Proceedings of the 19th international conference on World wide web*, pp. 621–630, Apr. 2010.

[16] J. G. Lee, S. Moon, and K. Salamatian, "An approach to model and predict the popularity of online contents with explanatory factors," in *Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 623–630, Aug. 2010.

[17] D. A. Soysa, D. G. Chen, O. C. Au, and A. Bermak, "Predicting youtube content popularity via facebook data: A network spread model for optimizing multimedia delivery," in *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 214–221, Apr. 2013.

[18] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of ACM-SIAM symposium on Discrete algorithms (SODA 2007)*, pp. 1027–1035, Jan. 2007.