Predicting the Geographic Distribution of Videos Views from Tags

Adrien Luxey

▶ **To cite this version:**

Adrien Luxey. Predicting the Geographic Distribution of Videos Views from Tags. [Internship report] Ecole Normale Supérieure de Rennes; Université Rennes 1. 2016. ⟨hal-01651188⟩

HAL Id: hal-01651188

https://hal.science/hal-01651188v1

Submitted on 28 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Master research Internship

Internship Report

Predicting the Geographic Distribution of Videos Views from Tags

Domain: Social and Information Networks – Machine Learning

Author:
Adrien Luxey

Supervisor:
François Taïani
Davide Frey
ASAP – As Scalable As Possible

école normale supérieure

Abstract: User Generated Content (UGC) plays a major role in today's Web. People create and share a lot of multimedia content (like photos or videos), leading to heterogeneous and unpredictable distributions of the consumption of such content. To this end, ensuring an acceptable Quality of Service (QoS) to the end-user has become a major challenge for UGC sharing services. Content Delivery Networks (CDNs) provide replication servers that help bring the content closer to its public, through caching mechanisms and proactive placement of the content (that exploit *a priori* knowledge on the content to predict its viewing distribution). Some approaches study the sharing patterns of a video in social networks, or the trendiness of a video's topic by looking at mainstream media. We would like to propose a proactive placement technique that would only rely on data available to the UGC service, and particularly on the tags associated with content. During the forthcoming internship, we will study the prediction power of Youtube videos' tags on the geographic distribution of its views, how tags can be used for proactive placement in a CDN, and which machine learning techniques apply for this task. This paper will thus present the architectures of UGC systems, the existing placement mechanisms of content on a CDN, and self-contained solutions for proactive placement; including the prediction power of tags for the geographic distribution of the views of a Youtube video.

Contents

1 Introduction **1**

2 Background **1**
2.1 User Generated Content (UGC) systems 1
2.1.1 Overview of UGC challenges 2
2.1.2 Content Delivery Networks (CDNs) to the rescue 2
2.1.3 Locality patterns in UGC platforms 3
2.2 Content placement strategies 4
2.2.1 A reactive placement example: Facebook's photo caching 4

3 Presentation of the dataset **6**
3.1 Getting our hands on the tags 6
3.1.1 A definition of tags 6
3.1.2 Preprocessing tags 7
3.2 Dropping infrequent tags and videos without tag 8

4 Predicting videos per-country views from tags **8**
4.1 Proposed regression approaches 9
4.1.1 Using tags geographic popularity 9
4.1.2 Linear approach 9
4.1.3 Experimental results 10
4.2 Back to our assumptions 11
4.2.1 Foreword 11
4.2.2 Tags and videos views 12
4.2.3 Tags and geographic distribution of views 15

5 Predicting the geographic distribution of videos views from tags **18**
5.1 Our baseline: Delbruel's approach [4] 19
5.2 Clustering distributions and applying Bayes classification 20
5.2.1 Clustering 22
5.2.2 Matching new videos to a cluster 25
5.2.3 Experimental results 26
5.3 Nearest neighbours 28
5.3.1 Jaccard Index and the MinHash algorithm 28
5.3.2 Experimental results 30

6 Conclusion **31**

A K-means clusters of geographic distributions **32**

B Evaluation of K-means clusters as a function of k **33**

1 Introduction

User Generated Content (UGC), such as user-submitted video, has become one of the biggest sources of Internet traffic worldwide, making it more and more challenging to provide a good quality of service to an ever-growing number of users. UGC services rely on clusters of servers (Content Delivery Networks, or CDNs) to replicate and bring content closer to the end users. The strategies used to place content on these networks can be either *proactive* (based on *a priori* knowledge, such as the spoken language of a video) or *reactive* (such as caching, that reacts to content's demand). Proactive placement tries to infer the future geographic distribution of content consumption, along with its popularity, using various knowledge sources. Given these estimates, proactive placement mechanisms find the best servers to which each piece of content should be deployed. Considering that many of these strategies rely on third-party information (like social networks or online media) to infer geographic distribution and number of views, we would like to propose a self-contained proactive placement solution – where the UGC service's knowledge is enough to appropriately place content. Building upon previous experiments in our working group, we will, during the internship, study a Youtube dataset to find self-sufficient ways to predict the geographic distribution of videos views. Our primary interest will be on the predictive power of tags, which was already studied by Delbruel et al. [4]. After a thorough analysis of the videos' tags, they proposed a naive solution to predict the per-country amount of views of new videos. During our internship, our goal has been to determine how tags can be used to infer the amount and geographic distribution of videos views. On future work, we hope to strengthen our system, and finally use it to achieve proactive placement of the content in a UGC service, usng only the information available to it.

In the next Section, we will present some background on the topics we will work onto during the rest of this article. Then, Section 3 will present the dataset we will work on, and some of the preprocessing we did on it four our purposes. Then, Section 4 will present our attempts at predicting the per-country amount of views of newly uploaded content using only the tags it was uploaded with. We will see that tags do not convey enough information for us to predict an amount of views, but that can only be used to infer the geographic *distribution* of such views. Thus, in Section 5, we will propose several approaches to this end. FInally, Section 6 will conclude this internship report.

2 Background

In this Section, we will introduce some of the main concepts upon which our study is based. First of all, Section 2.1 will present User Generated Content (UGC) systems, the challenges they convey, and the characteristics we can exploit to overcome these challenges. We will then present some of the state of the art content placing techniques in Section 2.2.

2.1 User Generated Content (UGC) systems

User Generated Content (UGC) can be defined as "any form of *creative content*, developed and *willingly published* by an individual or a consortium on an online platform" [15]. There are three main overlapping models of UGC creation and distribution [15]. **(1) Creative content**: any type of multimedia production published by an individual on a sharing platform such as Twitter, Youtube, Flickr and citizen journalism sites. **(2) Small-scale tools**: minor additions / modifications to

existing software or hardware platforms, such as amateur smartphone apps or video-game mods. **(3) Collaborative content**: UGC produced collaboratively by formal or informal groups of user, like Open Source software (Linux, Apache, or any of the millions projects hosted on GitHub) and wikis like Wikipedia.

In the rest of this study, we will focus our attention on sharing platforms of *creative content*. Since its emergence around 2005 (Youtube's birth year, along with the first publications of user contributed content on mainstream news like CNN), creative content accounts for a major part of the Internet traffic, making it harder and harder to provide a good Quality of Service (QoS) to the end users on bandwidth-intensive content like photos and videos. In this Section, we will first show the challenges that UGC generates, before we briefly present Content Delivery Networks, that provide the network horsepower for intensive UGC usage. Finally, we will see locality patterns that arise in UGC platforms, that could be used to improve their QoS.

2.1.1 Overview of UGC challenges

As of the second semester of 2014, "Real-time entertainment" (which includes streamed multimedia services like Netflix, Youtube and Spotify) was accounting for 63.25% of the peak bandwidth consumption in North America (38.15% in Europe). Youtube alone consumed 13.25% of the peak traffic in North America, and 19.85% in Europe [21]. In 2014, North American users in the top 15 percentiles in terms of bandwidth consumption are clearly using streaming as their primary form of entertainment: they monthly consume 212 GB bandwidth on average, 72% being dedicated to streaming, with an average streaming time of 100 hours.

From an Internet Service Provider (ISP) point of view, such a tremendous amount of data transiting on their cables causes important optimization challenges. Their goal is to deliver content in the most straightforward manner from the UGC servers to the end-user. Guillemin et al. [6] investigated, in April 2012, the bandwidth consumption that Youtube was responsible for on Orange ISP's network, by installing probes on their French IP backbone. They measured that Youtube videos consumed 50TB of bandwidth in one week. They made an important point: 59% of these videos had been viewed more than twice.

Finally, from a UGC service perspective, the biggest challenge is to keep providing an impeccable Quality of Service (QoS), despite the steady growth of its user-base and the ever-increasing viewing quality they demand: since 2014, it is possible to see Youtube videos in 60 Frames Per Second (FPS) at a definition up to 4096×3072 px. To reduce latency, the content needs to be served to the consumer from the best available server in the UGC service's network (based on physical location, availability, bandwidth capacity and other metrics) [9].

2.1.2 Content Delivery Networks (CDNs) to the rescue

A Content Delivery Network (CDN) is an infrastructure of distributed servers deployed in multiple data centers, specially designed to ensure high availability and high performance on the applications or services it serves [16]. Through a combination of caching, load balancing, request routing and content serving, a CDN distributes content from its edge servers, located close to the end-users. The edge servers store the requested content on memory for some time if ever it was to be requested again soon (that is the definition of caching). A CDN also possesses storage nodes, that contain replicas of all the content, and provide edge servers with it. On the user's perspective, the CDN's

machinery is transparent, since it only aims at getting the various pieces of content they asked as fast as possible.

Some corporations, like Google or Facebook (whose architecture we will study in Section 2.2.1), administer dedicated CDNs for their applications. But most mid-scale companies whose services need a lot of bandwidth prefer calling for CDN providers, and avoid the pain of maintaining a complex network architecture. One of the world's leading CDN provider, Akamai, is alone responsible for 15 to 30% of the global Internet traffic, with more than 200,000 servers located in 1,400+ networks worldwide [1].

The technologies that operate in a traditional CDN already empower UGC providers with many optimizations to their system. Though, a lot of improvements are still to be done to lower the traffic congestion and improve the QoS, when we know that Youtube alone generates billions of views per day[2]. An important issue, pointed out by [9], is that, due to the DNS mechanisms that point users to a specific edge node, some users switch back and forth between servers, increasing the probability of cold cache misses.

Thus, improvements can still be made in intelligent placing and routing mechanisms. They should cache content to the most appropriate clusters, and point users to the edge servers that will be the most efficient for their content demand. We will study some content placement mechanisms in Section 2.2.

2.1.3 Locality patterns in UGC platforms

Many measurements, to which we will come back as we comment on articles, show patterns in content consumption that could be exploited to improve the QoS of UGC services. Mostly, three *locality* patterns clearly stand out:

- **Content locality**: Several pieces of content tend to be requested along one another [19, 11]. This is partly a side-effect of recommendation systems, that propose *related* content when a user is viewing something. It is also a consequence of users being interested in a particular topic, looking for content about it. An example is Youtube's related video mechanism, that encourages watching linked videos;

- **Geographic locality**: Content is often consumed in restrained geographic areas [2, 9, 19, 4, 11]. The first reason for this is the language barrier, such that only niche Brazilian users will be interested in French speaking videos, for example.

- **Temporal locality**: Most content knows one or several popularity instants, when it is more demanded than the rest of the time [6]. Indeed, a lot of content relates to events, and thus loose interest once the event is assimilated.

The aware reader will recognize here two of the concepts that underlie caching: spatial and temporal locality of requests. Indeed, Section 2.2.1 will demonstrate the power of caching in delivering content efficiently. Content locality, on the other hand, requires some knowledge on the structure of the content graph. Finally, topic modelling on news networks can provide insights on audience peaks of some content, for example.

[1] Source: `https://www.akamai.com/us/en/about/facts-figures.jsp`
[2] Source: `https://www.youtube.com/yt/press/en/statistics.html`

Overall, we see that UGC is a very dynamic source of traffic. It requires substantial effort, both from UGC providers and ISPs, to keep up with the increasing users demand for QoS, while this new communication mean keeps on getting more popular. We did not cover Peer to Peer (P2P) strategies on CDNs, though they constitute a promising approach to improve the reactivity of the network [8]. We will now present several content placement strategies on CDNs.

2.2 Content placement strategies

Given a network and servers and a community of users, assigning content to appropriate locations is crucial in order to optimize the network load, and thus the maintain a good user experience. We discussed the three main locality patterns that can be exploited to this end. But many dynamic factors also have to be considered, such as network congestion or content popularity bursts. Content placement is thus a complex problem, with very diverse solutions. We will here focus on *reactive placement* in the next Section. *Proactive placement using external knowledge*, that was covered in the previous bibliography, won't be covered here. Yet, other strategies exist. *Content prefetching strategies* [14] download to the user's cache the content that she will most certainly access soon, which can drastically increase the viewing experience. It has been widely used to prefetch multimedia advertisements. *P2P strategies* can also support CDNs, by making the users part of the data-serving. P2P has been successfully employed, alongside CDNs, in live streaming services like LiveSky [23], or for Video-on-Demand (VoD) applications [8, 10], such as Spotify, or the fully P2P and controversial Popcorn Time [3].

2.2.1 A reactive placement example: Facebook's photo caching

Maybe the best example of a successful reactive content placement mechanism is Facebook's photo caching strategy [9]. In this study, Huang et al. instrumented Facebook's photo serving stack at each of its layers (see Figure 1), with a focus on the US geographic distribution of traffic flow.

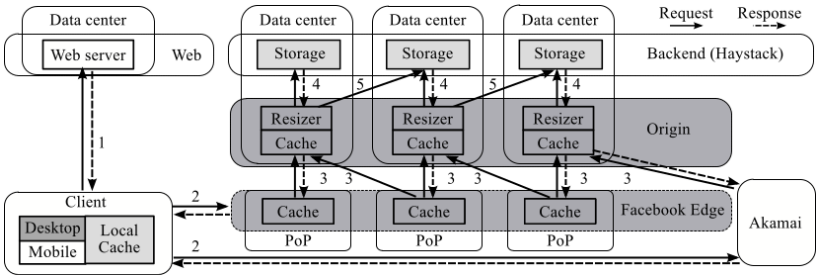


Figure 1: Facebook's photo serving stack

Facebook has three hierarchical layers of cache, that store pictures in original or resized format, so to fit the user's request according to their screen size. Closest to the end-user is their browser cache, that stores previously accessed content on disk, often using a Least Recently Used (LRU) replacement strategy. When a user requests an image from Facebook's frontend servers (step 1), their browser first looks at their own cache. If the request misses the browser cache, the user's browser sends a HTTP request to the Akamai CDN or one of Facebook's Edge servers, depending

[3]Official website: `https://popcorntime.sh/`

on the DNS (Domain Name System) resolution (step 2). Edge servers are a small amount of high-volume flash caches, located in Internet Points-of-Presence (PoP) that store resized pictures. There were nine Edge servers in the USA at the time of this study (2013), all using a First-In-First-Out (FIFO) cache replacement strategy. If an Edge server misses a photo, it sends a request to the Origin servers before adding it to its cache (step 3). The Origin cache servers are also high-volume caches, that store resized images. If a request misses their cache, they ask for the original picture to the Haystack servers, and the Origin takes care of the resizing, before caching the object in with a FIFO strategy (steps 4 and 5). Origin and Haystack servers are co-located in several data centers, meaning that Origin servers can often retrieve photos from nearby backend servers. Finally, the Haystack backend, that resides at the lowest level of the photo-serving stack, store every Facebook image in its original format.

This architecture is designed to minimize I/O, through the use of in-memory hash tables at every cache layer, leading to a single disk read per request. The study measured that 90.1% of the traffic was served by the different levels of cache (65.5% from the browser cache, 20% from the Edge, 4.6% from the Origin), while the remaining 9.9% misses were delivered by Haystack. This hit-rate shows the effectiveness of a cache-only strategy given a good infrastructure of delivery servers.

On the geographic distribution of their traffic, the authors make another interesting observation: the Edge server that delivers a picture does not correspond to the geographic location of the requesting user. Facebook's nine Edge servers were instrumented for data collection, in nine different US cities across the country, and their results show that each city is indeed served by the nine of them. The reason for this is that the best Edge server is computed by a weighted sum of the geographic distance, the latency, the current traffic and the traffic cost, not only the geographic distance. This observation first shows that geographic locality of the content cannot be the only factor determining its placement, since infrastructure-related dynamics can prove as important to the user's experience. On the other hand, a perverse effect of this Edge node choice is that some users might shift from Edge server to Edge server, increasing the odds of cold cache misses (the proportion of users served by two or more Edge servers being of 17.5% according to the authors). An envisaged solution for this issue is the use of collaborative cache at the Edge level, where they would communicate to one another the content they have in cache, and redirect users more intelligently.

Overall, this example on caching shows the strength of a reactive content placement mechanism. The hypothesis that underlies most caching strategies, according to which recently requested content is likely to be requested again soon, seems to be quite valid in UGC systems. We will now dive in proactive strategies, that look inside the content or other platforms, to infer knowledge on the futures views, and place it accordingly.

In this Section, we presented some of the state of the art placement techniques for content in UGC services. We saw that this content placement was crucial with the raising popularity of user generated content. We understood that the architecture of CDNs, with a worldwide infrastructure of caching systems, is a great help to user experience, and that they need information on the content to improve their efficiency even more. We will now go on presenting the dataset that we will work on, in the next Section.

3 Presentation of the dataset

Now that we covered the technical background needed to understand the work we achieved during this internship, we should briefly present the dataset we worked on.

It was gathered on Youtube by Kloudas et al. [11] in March 2011. The choice of Youtube was motivated by its situation as one of the leading UGC providers at this time. This platform generates a lots of traffic, and is the choice for many users that want to publish homemade clips, music videos, songs, or even podcasts (the proportion of users making podcasting their main activity has not stopped rising since).

The dataset comprises a total of 1,063,844 videos, that were sampled by taking the top 10 videos in 25 different countries. From these videos, the authors retrieved the related videos of each seed video recursively, leaving out videos with less than a thousand views, until they obtained the amount of videos they desired.

The information provided by the dataset include: the video id (a 11 character long alphanumeric string), its total number of views, its tags, and its Video Source Vector (VSV). This VSV represents, in a Google-defined format (Google's *Simple Encoding Format*), the amount of views a video has attracted in each of the 241 countries documented by Youtube. Its is fairly inaccurate, since every country is represented by an integer between 0 (no views in this country) to 61 (very popular in this country). Though, the algorithm that transforms the geographic distribution of videos views is not public. Delbruel et al. [4] managed to estimate the per-country amount of views by using several information sources. The primary source they used was Alexa, which is an authoritative source of Internet Traffic measurements (see `http://www.alexa.com`). In the remaining of our internship, we used the per-country views as estimated by Delbruel et al., keeping in mind that this distribution is a biased estimate.

A strong issue that arose is that Google changed their confidentiality policy at the end of 2011. The Video Source Vector, since then, is not available anymore. In other words, our experiment is not reproducible with more recent data. A question we might ask, then, is: does working on a five years old dataset provide valid results today? Well, our dataset represents a real-life snapshot of a UGC service's content at a certain point in time. The reality it describes is unique, because it is fixed in time and because it only represents Youtube's situation. Though, tagging behaviours have similar characteristics among time and platforms, and users from all over the world keep being attracted by various content depending on their topics. We argue that we can generalise these behaviours to any platform; but we will have to keep in mind which aspects of our work are time-specific (such as the trendiness of Justin Bieber in our dataset), platform-specific (such as the related videos system in Youtube, that generates a traffic flow specific to this platform), and where we made estimations (on the geographic distribution of views).

3.1 Getting our hands on the tags

Our interest in content's tags lead us to studying tagging behaviours, and which preprocessing should be applied to them before going any further.

3.1.1 A definition of tags

Gupta et al. [7] proposed a survey on social tagging techniques, in which they provided a thorough study of tags uses around the Web: from properties of tags streams, to the semantics of tags,

including applications of tags and problems associated with the usage of tags.

First and foremost, what are tags? Tags are a kind of metadata associated to content, providing additional information about it. They are written by users, often in form of keywords or terms. In comparison to fixed taxonomies like categories, tags can be very expressive and versatile. A fixed taxonomy is designed by a cataloguer at the creation of a sharing platform, leading to a rigid, centralised and subjective classification. Although often useful (like categories in a News site or music genres), fixed taxonomies cannot completely reflect the users point of view on their content. On the other hand, tags are proposed by users, without any predefined hierarchy or relationship between them. They are coined as a *folksonomy* (literally People – Classification – Management), that directly reflect users' language, vocabulary and needs.

Tags have been around the Web for a while. Their first use in social networks dates back to 2003, with their introduction on Delicious. Flickr soon followed, and tags became a major mean of content classification for its users since then. Twitter's hashtags popularity has even become a mainstream metrics of a topic's trendiness. Blogging platforms such as Wordpress or Blogger also offer the possibility to annotate their posts with tags. Finally, Youtube, among other multimedia sharing services, use tags as a way to add meta information on the content and facilitate indexation. Youtube, notably, emphasizes on the importance of tags for good indexation, as to motivate the users to take this publication step seriously, even though tags do not appear – anymore – on videos rendered pages.

Since tags are written by the uploaders on Youtube, a trivial inference is their spoken language, which is very correlated with the target audience geographic distribution. Besides, Gupta et al. [7] summarize several kinds of tags that provide information we could use to predict the geographic distribution of views: *content-based* tags, that describe the actual content; *context-based* ones, that provide information on the context of content (like the place and time it was created); purpose tags, that explain the aim of a piece of content, and so on. Given all these use cases, we can hope that tags indeed convey a good predicting power of the geographic distribution of the future views.

The authors dedicate a Section of their article to warn readers about tagging problems. Among these problems, the most commonly seen is *spamming*: some users typically pick a lot of popular tags that do not relate to their content, in order to trick recommendation systems and attract more views. This problem is recurrent in Youtube, where users can pick tags of a popular video in the hope that they will get as many views. This observation suggests that tags cannot be used to predict a video's number of views, but only their distribution; we will come back to that. Another common issue is about *ambiguities*: tags are context-free; yet, some homonyms carry very different meanings. The tag *Washington*, for example, can refer both to the capital of the United States (on the East Coast) and to the US state (on the West Coast). This kind of ambiguity could lead to misplaced videos, unless taken care of. Finally, the authors remind that users might not reach a global consensus over the appropriate tags set for a given content, leading in a unstable system.

After reading this article, we understand that tags can be of a great help as a folksonomy. Being the most versatile metadata available, it fits very well to uploaders intentions on their content. Though, their permissive policy is a danger, as prediction algorithms could easily be misleaded by inappropriate tags.

3.1.2 Preprocessing tags

Kloudas et al.'s dataset had several issues with its tags. A major one was that many non pure-ASCII characters were replaced by question marks. We thus have a lot of tags that are only successions of

question marks: '?????', or that contain question marks surrounded by alpha-numeric characters: 'ni?o' (which could be a poorly formatted version of the Spanish 'niño', but not only). Many tags also have useless leading or trailing characters. For example, 'music' and 'music␣' count as two separate tags. Finally, the case of the characters also leads to different tags: 'bieber, 'BIEBER' and 'Bieber' count as three tags.

The last issue was easily solved by lowering the case of every tag (using Python's implementation of lowercase function, that successfully transforms most UTF-8 characters (like 'Ñ'), which is not the case of most implementations (namely C++)).

Then, we chose to remove, from each tag, any leading and trailing character that was not not a number, or any kind of alphabetic character (counting accentuated characters as alphabetic ones). For instance, the tag '#élégant 42!' would have turned into 'élégant 42'. Of course, we removed the empty tags this transformation generated.

Our dataset originally contained 705,415 unique tags. After our first transformation, this number dropped to 590,825, thus removing 16% of them.

3.2 Dropping infrequent tags and videos without tag

The removal of tagless videos in the original dataset already made the video number drop from 1,063,844 to 590,897. Indeed, these videos will be useless for us, since we cannot infer anything from tags if we do not have any.

Then, after the tags preprocessing, we decided to remove from the dataset any tag that did not appear in at least 10 videos. We considered that below this amount of videos, a tag could not be used for prediction, because it accounted for not enough videos to be considered as anything but noise.

This led to an iterative process, where we first removed tags that did not appear in at least 10 videos, then videos that did not contain at least one tag, etc. From 590,897 videos and 590,825 tags, we ended up with a final, cleaned dataset containing 467,223 videos and 51,490 tags that appeared in at least 10 of them.

Now that we presented the dataset, the characteristics of tags, and the operations we did on our dataset to clean it up for our purpose, we will start presenting the work we did to try and predict the amount of views a video would attract, depending on the tags its uploaded gave it.

4 Predicting videos per-country views from tags

Several approaches are conceivable to predict per-country videos views from tags. We will cover two: the one proposed by Delbruel et al. [4], that uses the known tags *popularity* among countries (the tags per-country views) to infer new videos views; and a second one based on the general linear model, that considers the per-country views matrix as a linear combination of the videos' tags against a matrix of tags weights.

Both of these approaches attempt to predict, at the same time, the geographic distribution of videos views, and the number of views they will attract. Thus, are based on two assumptions: that tags carry enough information to infer global videos views; and that tags allow to predict videos' geographic distributions.

In this section, we will first present the regression techniques that we tried out to predict per-country videos views in Subsection 4.1; then we will get back to the assumptions these approaches made on the data in Subsection 4.2.

For now on, we will use the following notations: the dataset contains n videos viewed in p countries, comprising a total of m different tags. The matrix of videos views per country is called $\mathbf{CV} = \{cv_{i,k}\} \in \mathbb{R}_+^{n \times p}$ (for *Country Views*). We also compute the Bag of Words (BoW) representation of videos' tags $\mathbf{BoW} = \{bow_{i,j}\} \in \{0,1\}^{n \times m}$, such that $bow_{i,j} = 1$ when the video v_i contains tag t_j, and $bow_{i,j} = 0$ otherwise. Finally, for an individual video v, $\mathbf{bow}_v \in \{0,1\}^m$ is the BoW representation of its tags. In the remaining of this article, we will always use $i \in [0, n]$ as an index for the videos, $j \in [0, m]$ as an index for the tags, and $k \in [0, p]$ as the country's index.

4.1 Proposed regression approaches

4.1.1 Using tags geographic popularity

During previous work in our working group, Delbruel et al. [4] proposed an estimation technique of the per-country viewing vector of new videos using the geographic *popularity* vector $\mathbf{a}_{j,*} \in \mathbb{R}_+^p$ of already observed tags. For a given tag, this popularity vector is the average of the per-country viewing vectors of the videos it appears in.

The first step of the algorithm is to compute a matrix $\mathbf{A} \in \mathbb{R}_+^{m \times p}$ whose columns are the popularity vectors of any observed tag: $\mathbf{A} = \{\mathbf{a}_{j,*}\}_{0<j\leq m}$. Each element $a_{j,k}$ of $\mathbf{A}$ contains the average views, in country c_k, of every known video having the tag t_j:

$$a_{j,k} = \mathop{\mathbb{E}}_{i:t_j \in \text{tags}(v_i)} \Big(cv_{i,k} \Big) = \frac{\sum\limits_{i:t_j \in \text{tags}(v_i)} cv_{i,k}}{\big|\{v : t \in \text{tags}(v)\}\big|}$$

When a new video v is published, we first compute its BoW vector $\mathbf{bow}_v \in \{0,1\}^m$. Then, its estimated number of views $\hat{\mathbf{cv}}_v$ is the average of its tags' popularity vectors:

$$\hat{\mathbf{cv}}_v = \mathop{\mathbb{E}}_{j:t_j \in \text{tags}(v)} \Big(\mathbf{a}_{j,*} \Big) = \frac{\mathbf{bow}_v \times \mathbf{A}}{|\text{tags}(v)|} \in \mathbb{R}_+^p$$

4.1.2 Linear approach

The linear regression approach consists in finding the best weights matrix $\mathbf{A} = \{a_{j,k}\} \in \mathbb{R}^{m \times p}$ (plus a constant term per country: $\mathbf{c} \in \mathbb{R}^p$) according to the following equation:

$$\mathop{\arg\min}_{\mathbf{A},\mathbf{c}} \big\| \mathbf{X} \times \mathbf{A} + \mathbf{c} - \mathbf{Y} \big\|_2^2$$

In most of the literature and in our experiments, $\mathbf{A}$ is estimated by using the *ordinary least squares* method.

When a new video v is uploaded, we estimate its number of per-country views as the linear combination of $\mathbf{A}$'s weights with v's tags BoW $\mathbf{x}_v$:

$$\hat{\mathbf{cv}}_v = \mathbf{bow}_v \mathbf{A} + \mathbf{c} \in \mathbb{R}^p$$

We can see that Delbruel's approach is already more advanced than the linear model, since it takes the video's tags number into account when computing $\hat{\mathbf{cv}}_v$.

4.1.3 Experimental results

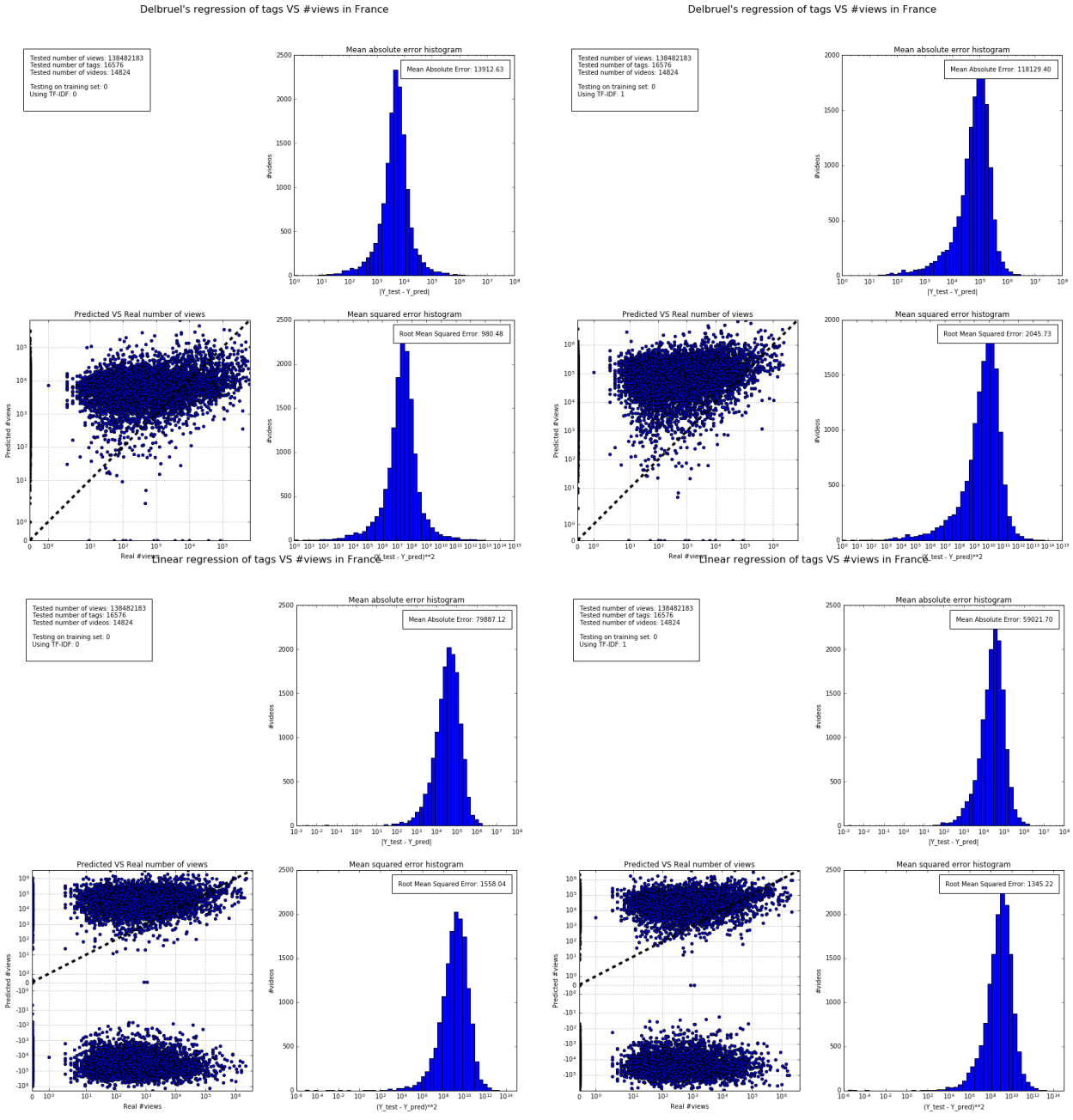


Figure 2: Results of both views prediction approaches in France

To test these approaches, we first randomly sub-sampled our dataset (leading to approximately 55,000 videos) to lower memory and CPU usage. Then, we split this subset into a training set $\mathcal{V}_{\text{train}}$ containing 70% of the videos, and a test set $\mathcal{V}_{\text{test}}$ containing the remaining 30%.

In both cases, we computed the weights matrix $\mathbf{A}$ using only the videos from $\mathcal{V}_{\text{train}}$, before computing views predictions on $\mathcal{V}_{\text{test}}$. This led, for each $v \in \mathcal{V}_{\text{test}}$, to a couple $(\mathbf{y}_v, \hat{\mathbf{y}}_v)$: the real

video's views and the predicted ones.

Figure 2 shows the results of both approaches at predicting views in France. The left hand scatter plot is composed of $(\mathbf{y}_v, \hat{\mathbf{y}}_v)$ points for each v in $\mathcal{V}_{\text{test}}$. The dotted black line is our objective: where each prediction $\hat{\mathbf{y}}_v$ equals $\mathbf{y}_v$. One the right hand histogram, we computed the Squared Error (SE) of the estimation against the real number of views as follows: $\mathbf{SE}(\mathbf{y}_v, \hat{\mathbf{y}}_v) = (\mathbf{y}_v - \hat{\mathbf{y}}_v)^2$. On top of this histogram, we included the Root Mean Squared Error (RMSE):

$$\mathbf{RMSE} = \sqrt{\frac{\sum\limits_{v \in \mathcal{V}_{\text{test}}} \left(\mathbf{y}_v - \hat{\mathbf{y}}_v\right)^2}{\left|\mathcal{V}_{\text{test}}\right|}}$$

First of all, we see that neither approach performs well: the scatter plot hardly follows the dotted line in both cases. That being said, Delbruel's approach gets a better RMSE than the linear regression. This correlates with our earlier comment that Delbruel's approach is more advanced than the linear model, since it normalises the weights by the video's number of tags. Finally, we see that the linear regression generates two clusters of points in the scatter plot, one having a negative predicted number of views. This comes from the unboundedness of the weights in $\mathbf{A}$: they can be negative. A variant of the linear model, called Non-Negative Least Squares (or NNLS) adds a constraint of positiveness to the parameters. Though, given the poorness of the obtained results, we rather chose to take a step back and ensure the validity of our hypotheses.

4.2 Back to our assumptions

As already stated, the previous approaches relied on the two following assumptions:

- $\mathcal{H}_A$: tags provide sufficient information to predict the amount of videos views;
- $\mathcal{H}_B$: tags provide sufficient information to predict the geographic distribution of videos views.

However, our previous attempts at predicting tags views per country were unsuccessful. We shall thus attempt to verify (or invalidate) our hypotheses on the dataset before going further. To do so, we will apply statistical hypothesis techniques, to assess the *statistical significance* of our assumptions.

4.2.1 Foreword

In statistical hypothesis testing, one has an assumption about a statistic in a sample of a population, and wants to access its validity on the whole population. First and foremost, because of the sampling, one will only be able to access degrees of certainty of an hypothesis on the population.

The test assumption, that suggests a relationship between the *test statistic* x and a control one x_0 (hypothetical, or measured from another dataset), is compared as an *alternative hypothesis* $\mathcal{H}_1$ to an idealized *null hypothesis* $\mathcal{H}_0$ that states no relationship between them. The test assumption is accepted when the difference between the quantities is unlikely to happen under the null hypothesis: the difference is deemed statistically significant according to a predefined threshold, the *significance level* α. In this case, the alternative hypothesis $\mathcal{H}_1$ is accepted, and $\mathcal{H}_0$ is rejected. On the other hand, if there is no statistically significant difference between the two statistics, the tests fails to reject the null hypothesis: given our sample, it is impossible to state a relationship between theses

quantities. This does not mean that the test hypothesis is false, but only that, given the current sample, this relationship is unlikely.

To assess the statistical significance of the relationship, a plethora of *statistical tests* exist, depending on the situation. They produce, by different means, the p-value $P(x|\mathcal{H}_0)$, that measures the probability of obtaining a result as extreme as x under the null hypothesis. If this probability falls under the significance level α, then the null hypothesis is rejected, and $\mathcal{H}_1$ is accepted. Otherwise, the test failed to reject $\mathcal{H}_0$.

4.2.2 Tags and videos views

Several observations make us doubt the validity of $\mathcal{H}_A$. Gupta et al. [7], for instance, conducted a survey on social tagging techniques. They classified several tags types and motivations. If content and context-based tags (such as 'music' or 'russia') could help predict viewing behaviours, they might not be a majority. Some users tag in order to attract attention (by using popular tags, may they not be related to their video), to express an opinion (like 'funny', the 5th most frequent tag in our dataset), or for their own attention. These kinds of tags do not seem correlated with a video's number of views; they even add noise that might severely damage the predicting power of the other tags. Another argument is that a plethora of other factors have an impact on a video's popularity. The most prominent factor would be its popularity on other media sources (like social networks, TV, or radio).

Defining the statistical hypotheses The first step of hypothesis testing is to understand what we are looking for. What we want to verify is whether the presence of a tag has an impact on a video's number of views. To do so, we will compare means from different samples. Since the viewing behaviours are very different in every country, we will want to apply our test in every country c_k separately, and for every tag t_j. Our control statistic x_0 will be the mean number of views for all videos in this country: μ_k. We will compare if to our test statistic x, the mean number of views, in country c_k, of all videos sharing tag t_j: $\mu_{j,k}$.

$$\mu_k = \mathop{\mathbb{E}}_{i:cv_{i,k}>0} cv_{i,k} \quad ; \quad \mu_{j,k} = \mathop{\mathbb{E}}_{\substack{i:t_j \in \text{tags}(v_i) \\ cv_{i,k}>0}} cv_{i,k}$$

Now, we need to propose two statistical hypotheses: the *null hypothesis* $\mathcal{H}_{A0}$, that states the absence of relation between tags and views, and the *alternative hypothesis* $\mathcal{H}_{A1}$: the one we are trying to prove. In other words:

- $\mathcal{H}_{A0}$: *The per-country mean of videos views is equal for the whole dataset and for videos sharing a tag*:
$$\mathcal{H}_{A0} : \mu_k = \mu_{j,k}$$
- $\mathcal{H}_{A1}$: *The per-country mean of videos views is different for the whole dataset and for videos sharing a tag*:
$$\mathcal{H}_{A1} : \mu_k \neq \mu_{j,k}$$

This formulation of our alternative hypothesis, where we consider that variations in either way of the control statistic are significant (e.g. $x_0 \neq x$), is called a *two-tailed test. One-tailed tests* consider that variations of x_0 are significant in only one way, e.g. $x_0 > x$ or $x_0 < x$.

Choosing the appropriate statistical test Depending on the task at hand, one has to choose between a plethora of statistical tests. In our case, we choose the well-known **Z-test**, which makes the assumption that the distribution of the test statistic (the mean number of views) is normally distributed under the null hypothesis. In other words, the Z-test assumes that, if the null hypothesis is true, then realisations of x will be normally distributed around their theoretical value x_0. This assumption relies on the central limit theorem, that states that most statistics are normally distributed around their theoretical value when the sample size is large enough.

Given the mean μ_k and standard deviation σ_k of the test statistic's normal distribution, the Z-test computes, for the j-th sample of size N, its empirical mean $\mu_{j,k}$ and empirical standard deviation $s_{j,k}$. This allows us to calculate the standard score $Z_{j,k}$, that represents the distance between μ_k and $\mu_{j,k}$ in σ_k units.

$$Z_{j,k} = \frac{\mu_{j,k} - \mu_k}{s_{j,k}} \text{ where } s_{j,k} = \frac{\sigma_k}{\sqrt{N}}$$

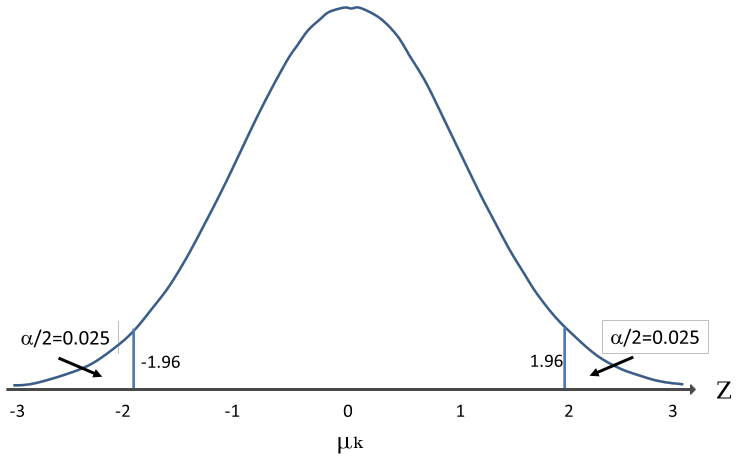


Figure 3: Rejection region of a two-tailed Z-test with $\alpha = 5\%$

Finally, we compute the probability of obtaining a result as extreme as $Z_{j,k}$ under the null hypothesis (the p-value), compare it to the *significance level* α (usually of 1 or 5%), and conclude on the validity of our hypothesis. As already stated, we compute the *two-tailed* p-value of $Z_{j,k}$: $p(Z_{j,k}) = 2P(|Z_{j,k}| \geq Z_{\alpha/2}|\mathcal{H}_{A0})$

We chose a significance level α of 5%, a common value in the literature. As depicted in Figure 3, this means that when $|Z_{j,k}| \geq 1.96$, $\mu_{j,k}$ will be considered significantly different from μ_k, thus rejecting the null hypothesis $\mathcal{H}_{A0}$ in favour of $\mathcal{H}_{A1}$.

Results As already stated, we conducted the Z-test separately among each country c_k, for each tag t_j appearing in this country. We then plotted the Empirical Cumulative Distribution Function (ECDF) of the p-value $p(Z_{j,k})$ of each tag, grouped by country, which resulted in 241 ECDF plots. Figure 4 (a) shows the tags' p-values ECDF in France, Italy, Russia and United States. We see that 85% of the tags have a p-value above 70%. This means that the per-country average of views grouped by tag $\mu_{j,k}$ in these countries is far from being significantly different to the global per-country average of views μ_k. In Figure 4 (b), we computed the mean of the p-values ECDF for all

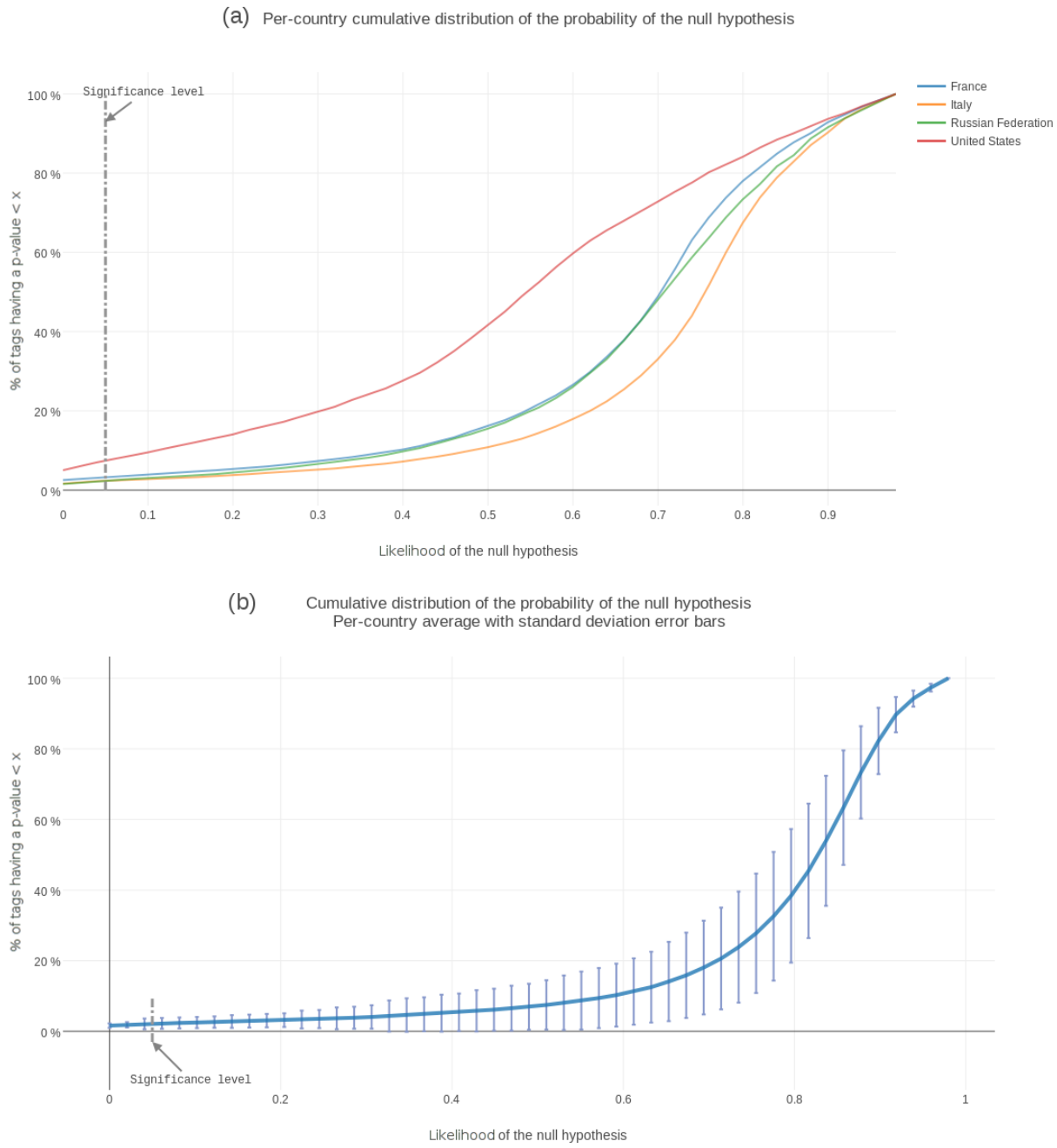


Figure 4: ECDF of the p-values of the Z-test determining the likeliness of obtaining the observed $\mu_{j,k}$ under the hypothesis $\mathcal{H}_{A0}$. The top plot shows the ECDF for several countries, while the bottom plot shows the average ECDF for all countries; error bars are the standard deviation among countries. Only $2.2\% \pm 1.5\%$ of the tags have a p-value below $\alpha = 5\%$ (7.5% in the United States). This means that only around 2% of the grouped-by-tag videos have a significantly different views mean than the per-country views mean, such that we cannot reject $\mathcal{H}_{A0}$.

countries, and added standard deviation error bars. This showed that only $2.2\% \pm 1.5\%$ of the tags had a significant impact on the per-country views mean.

A possible explanation is that, in most countries, the standard deviation of views σ_k is very high: it has a mean of 12,690 views among all countries. To be considered significantly different from μ_k, $\mu_{j,k}$ has to be twice lower or higher than the standard deviation in its country. Oddly, the United States is the country with the most tags' p-values below α (7% of them), whereas it also has the biggest $\sigma_k = 881,000$.

With such a low amount of tags rejecting the null hypothesis $\mathcal{H}_{A0}$, we can conclude that we failed to reject it. In other words, our dataset does not provide evidence that videos' tags are correlated with amount of views.

4.2.3 Tags and geographic distribution of views

We could not demonstrate that videos views could be predicted from tags, but we have reasons to believe that the prediction of geographic distribution of views should work better. Gupta et al. [7] argue that a major benefit of folksonomies (like tags) against other taxonomies, is that they express the real users' needs and language. Delbruel et al. [4] showed that tags were linked to geo-locations; they gave the example of 'favela', that was mostly appearing in videos viewed around Brazil, and 'bollywood', that was mostly watched around India.

For this experiment, we need to formally define the geographic distribution of videos views. From the vector $\mathbf{cv}_{i,*} \in \mathbb{R}_+^p$ of per-country views of any video v_i, we constructed its geographic distribution of views $\mathbf{cd}_{i,*} \in [0,1]^p$ (for *Country Distribution*) by normalising $\mathbf{cv}_{i,*}$ by its sum. We also computed the global distribution of views $\bar{\mathbf{cd}}$ as the average of all videos' geographic distributions.

$$\mathbf{cd}_{i,*} = \frac{\mathbf{cv}_{i,*}}{\sum\limits_{0 \leq k < p} cv_{i,k}} \text{ s.t. } \sum_{0 \leq k < p} cd_{i,k} = 1 \quad ; \quad \bar{\mathbf{cd}} = \mathop{\mathbb{E}}_{0 \leq i < n} \left(\mathbf{cd}_{i,*} \right)$$

Our interest is in the geographic distributions of tags views $\mathbf{cdt}_{j,*} \in [0,1]^p$ (for *Country Distribution of Tag*), that is the average – like the tags' popularity vector we already saw – of the geographic distribution of all the videos a tag t_j appears in:

$$\mathbf{cdt}_{j,*} = \mathop{\mathbb{E}}_{i : t_j \in \text{tags}(v_i)} \left(\mathbf{cd}_{i,*} \right)$$

Defining the statistical hypotheses To assess if tags have an influence on the geographic distribution of video views, we want to compare tags' geographic distributions of views $\mathbf{cdt}_{j,*}$ (the test statistic x) to the global distribution of views $\bar{\mathbf{cd}}$ (the control statistic x_0). Since we are comparing empirical distributions, we want to determine whether they are drawn from the same distribution or not. If they are not, since $\mathbf{cdt}_{j,*}$ is computed from the distribution of views of the videos having tag t_j, it will mean that tags have an influence on the geographic distribution of videos' views.

- $\mathcal{H}_{B0}$: *The geographic distribution of tags views and the global distribution of videos views are drawn from the same distribution*:

$$\mathcal{H}_{B0} : \bar{\mathbf{cd}} = \mathbf{cdt}_{j,*}$$

- $\mathcal{H}_{B1}$: *The geographic distribution of tags views and the global distribution of videos views are not drawn from the same distribution*:

$$\mathcal{H}_{B1} : \bar{\mathbf{cd}} \neq \mathbf{cdt}_{j,*}$$

Choose the statistical test The **Kolmogorov-Smirnov (KS) test** is a statistical test that compares cumulative distribution functions. As a one-sample test, it is used to quantify whether the test statistic is drawn from a reference probability distribution. We will use it in its two-sample form, that quantifies the probability that two empirical distribution samples are drawn from the same unknown reference distribution.

This test computes the KS-statistic $D_{n,n'}$ between two samples $F_{1,n}$ and $F_{2,n'}$ of size n and n' respectively. This statistic represents the maximum difference between their Empirical Cumulative Distribution Functions (ECDFs), as shown in Figure 5:

$$D_{n,n'} = \sup_t |F_{1,n}(t) - F_{2,n'}(t)|$$

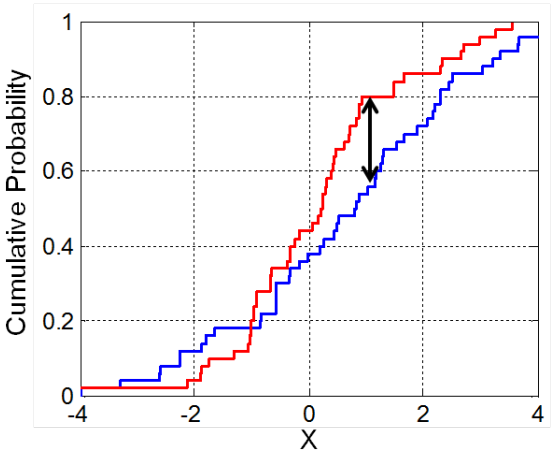


Figure 5: Illustration of the two-sample KS-statistic: the red and blue lines are the two compared ECDFs; the black arrow is the KS-statistic $D_{n,n'}$.

Given a significance level α, the critical value of the KS-statistic D_α is computed, such that the null hypothesis $\mathcal{H}_0$ is rejected if $D_{n,n'} > D_\alpha$; $\mathcal{H}_0$ is failed to reject otherwise. D_α uses a table of precomputed $c(\alpha)$ for each level of α:

$$D_\alpha = c(\alpha)\sqrt{\frac{n+n'}{nn'}}$$

Results The implementation of the two-sample KS-test we used, SciPy's `ks_2samp`, for Python, provided us with both the KS-statistic $D_{n,n'}$ and its conversion as a p-value, that we could directly compare to our significance level α of 5%.

As shown by Figure 6, 85% of the tags' p-values of geographic distributions fall below 5%. According to the KS-test, it means that most their distributions are very unlikely to come from the

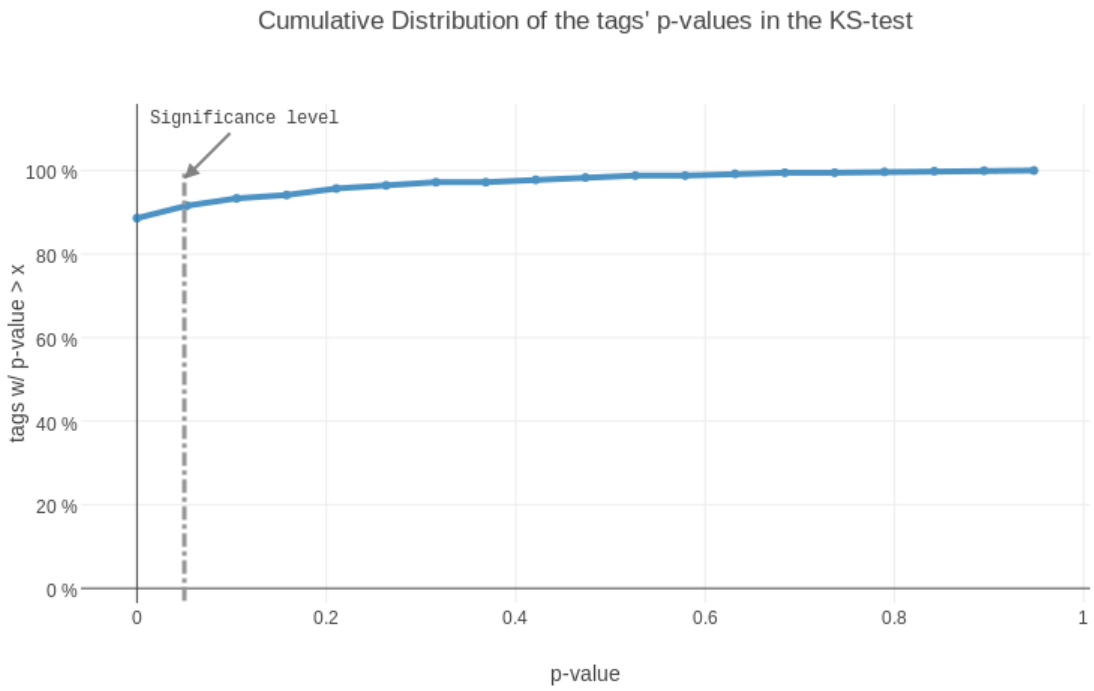


Figure 6: ECDF of the tags' p-values using the KS-test, comparing the global distribution of videos views, and the per-tag viewing distributions. 85% of the tags fall below the significance level $\alpha = 5\%$.

same distribution as the global geographic distribution of videos views. $\mathcal{H}_{B0}$ is rejected in favour of $\mathcal{H}_{B1}$: tags are considered as able to predict the geographic distribution of videos views.

Taking a glance at the tags with the highest p-values (whose distributions are the closest to the global distribution of views), we encountered tags that were already identified by Delbruel et al. [4] as the most viewed tags, such as 'hip', 'hop', 'pop', or 'funny'. On the other side, tags with the lowest p-values (whose distributions were the farthest from the global one) were often non-English ('katastrofa': Czech, Polish, and Serb for *catastrophe*) – thus more tied to geographic areas; but we also found 'fifa' (one of the tags with the most geographic entropy [4]), or even 'justin bieber' (which was, back in 2011, one of the most trending tags on Youtube). Overall, it is hard to conclude on the reasons that make a tag's geographic distribution of views farther from the global one.

In this Section, we attempted to predict per-country views of videos using regression techniques; that is, we tried to predict both the amount of views of incoming videos, and their geographic distributions, using their tags. Given our poor results, we took time to test our hypotheses with statistical tools, and found that, if tags provide sufficient information to predict the geographic distribution of videos views, they cannot be used, with our dataset, to infer their amount of views.

This is a very bad result for the application we wanted to experiment with this dataset: proactive placement of videos on a Content Delivery Network using only the information available to the User Generated Content service. Indeed, if proactive placement needs to determine the geographic areas

where content will be consumed, it is also crucial to know if a piece of content is expected to get hundreds or millions of views: it tells us the amount of replications we need to make of it. Fortunately, other studies, such as Wang et al.'s [22], have shown that previous information on the number of videos shares could be used to predict their future popularity. Other approaches of time series forecasting, such as regression or neural forecasting, could also be envisaged for that matter. Since past views or content shares are supposed to be available to a UCG provider, we still consider that predicting both number and geographic distribution of content's views, using only the UGC service's data, is feasible. Our dataset, unfortunately, does not contain enough information to predict the future amount of views.

Given these conclusions, we decided to focus the rest of our study on the prediction of the geographic distribution of videos views from tags, by using the same dataset. This will be the topic of the next Section.

5 Predicting the geographic distribution of videos views from tags

Attempting to predict the geographic distribution of incoming videos' views raises many scientific questions:

- How can we use tags to match videos to existing viewing patterns?
- What use can we make of known videos' distributions of views to infer knowledge on new ones'?
- How can we evaluate the prediction of a geographic distribution?
- How can we compare approaches?

Predicting these distributions partly falls into the machine learning process of *density estimation*. Indeed, *kernel density estimation* is a well documented problem [20]: it consists in fitting a kernel function's parameters (such as a uniform or normal distribution) to available samples of distributions drawn from an unknown density function. Alas, in our situation, we want to predict densities in a categorical space of countries, which is incompatible with the use of kernels. This leaves us with a fistful of other directions to solve our problem.

In this Section, we will propose several approaches that were tried during this internship, compare their results, and explain their strengths and weaknesses in regard of the several challenges we face. First off, we will present, in Section 5.1, the previous proposals that were made by Delbruel et al. [4] in their previous work, which will be our baseline. We will move on, in Section 5.2, presenting another approach we studied: clustering existing distributions before applying Bayes classification on new videos. We will there find that clustering distributions – without a ground-truth – is quite hard. This is why we will present, in Section 5.3, another approach that avoids the clustering step: nearest-neighbours search.

In this whole Section, we will use the same notations as in Section 4: $\mathbf{BoW} \in \{0,1\}^{n \times m}$ represents the Bag of Words (BoW) of all n videos. As so, $\mathbf{bow}_{i,*} \in \{0,1\}^m$ accounts for which tags, among the m in the dataset, are present in video v_i. However, we will not make any more use of $\mathbf{CV} \in \mathbb{R}_+^{n \times p}$, the matrix of all videos views among the p countries. Instead of that, we will use the matrix $\mathbf{CD} \in [0,1]^{n \times p}$ (for *Country Distribution*), such that $\mathbf{cd}_{i,*}$ represents the empirical geographic distribution of video v_i's views. As already covered in Section 4.2.3, $\mathbf{cd}_{i,*}$ is obtained by normalizing $\mathbf{cv}_{i,*}$, the per-country views of video v_i, by its sum.

5.1 Our baseline: Delbruel's approach [4]

In this Section, we will present the approach, proposed by Delbruel et al. [4], to achieve prediction of the geographic distribution of videos' views. We will then reproduce their experiment, and compare our results with theirs.

Before they employed tags geographic *popularity* (or per-country views) for proactive placement – such as already covered in Section 4.1.1, Delbruel et al. thoroughly studied tags' geographic *distribution* of views to ensure the validity of their placement technique.

The approach The authors first split their dataset in two equal parts: a training set $\mathcal{V}_{\text{train}}$ and a testing set $\mathcal{V}_{\text{test}}$. Then, they computed, for each tag in the training set, the geographic distribution of its views $\mathbf{cdt}_{j,*} \in [0,1]^p$, the same way we did in Section 4.2.3:

$$\mathbf{cdt}_{j,*} = \mathop{\mathbb{E}}_{\substack{i:t_j \in \text{tags}(v_i) \\ v_i \in \mathcal{V}_{\text{train}}}} \left(\mathbf{cd}_{i,*} \right)$$

Then, they used this matrix $\mathbf{CDT} = \{\mathbf{cdt}_{i,*}\}_{v_i \in \mathcal{V}_{\text{train}}}$ to estimate new videos' viewing distribution. For a new video $v \in \mathcal{V}_{\text{test}}$, its estimated distribution of views is the average of the known tags it contains:

$$\hat{\mathbf{cd}}_v = \mathop{\mathbb{E}}_{j:t_j \in \text{tags}(v)} \left(\mathbf{cdt}_{j,*} \right)$$

We did try other aggregation methods than the average, both when creating the **CDT** matrix and computing the estimation of viewing distribution $\hat{\mathbf{cd}}_v$: namely the maximum and the median. However, they both performed worse than the average. We thus decided to stick with averaging distributions, even though we still feel that this aggregation method could be improved (by weighting videos and tags depending on their amount of views, or entropy of their views, for example).

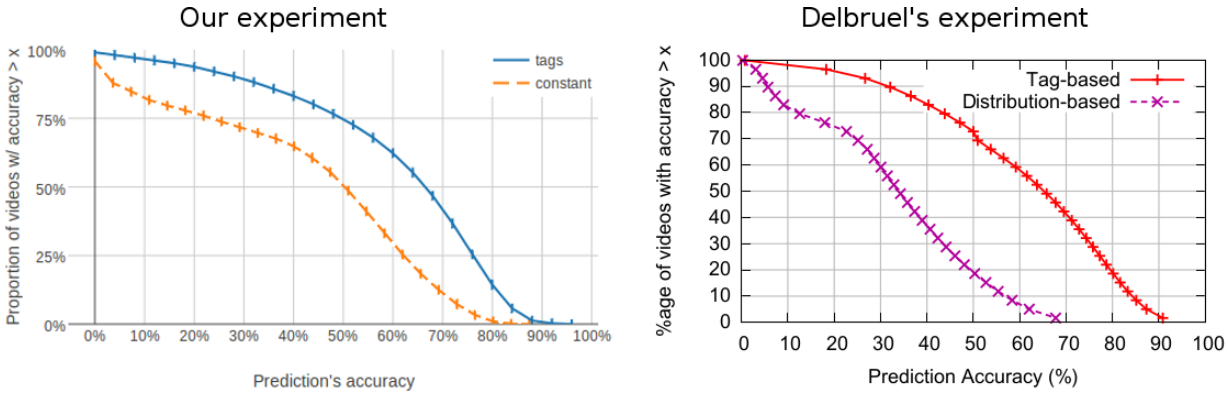
Baseline They compared this approach against a baseline: the average distribution of global Youtube views, that they estimated from data by Alexa Internet Inc. (for the 40 countries producing the most Youtube traffic) and the International Telecommunication Union (for the remainder). We did not make such an estimation, but we do have the global distribution of views of our dataset, $\bar{\mathbf{cd}}$, that was, again, proposed in Section 4.2.3. Given that $\bar{\mathbf{cd}}$ is computed from the data directly, we are likely to have lesser error between the ground-truth and this prediction as the authors did. However, their use of an external source of information to compute their baseline is laudable, even though it probably added even more inaccuracy due to estimation errors.

Evaluation metric Now, given an estimate $\hat{\mathbf{cd}}_v$ of a video v's geographic distribution of views, one needs to compare it to the ground-truth $\mathbf{cd}_v$. Delbruel et al. proposed an accuracy metric, that we will call $acc(v)$, based on the Manhattan (or taxicab, cityblock, L_1...) distance:

$$acc(v) = 1 - \frac{1}{2} \times \|\hat{\mathbf{cd}}_v - \mathbf{cd}_v\|_1 \quad \text{s.t.} \quad \|\mathbf{a} - \mathbf{b}\|_1 = \sum_i |a_i - b_i| \tag{1}$$

We found this metric to be suitable as an accuracy or similarity metric. Indeed, the L_1 distance precisely describes the proportion of misplaced views between the predicted distribution and the

ground-truth, meaning that $acc(v)$ gives a ratio of correctly placed views. We will thus keep using this metric along our experiments.



Statistics for our results

	mean	*median*
Tag-based prediction	64.6%	70.5%
Distribution-based prediction	46.9%	53.9%

Figure 7: Top-left: Cumulative Distribution Function (CDF) of the accuracy of the tag-based and constant distribution-based prediction of video densities of views. Top-right: Same experiment as conducted by Delbruel et al. [4] Bottom: Statistics of our experiment.

Results With the full dataset, we reproduced the experiment that Delbruel et al. performed. Using a constant distribution as the prediction, they obtained a mean accuracy of 32.9%, with a median of 33.9%. Using their tags distribution technique, they obtained a mean accuracy of 61.3% and a median of 65.9%. The comparison of our results is shown in Figure 7. A one can see, our results were a little better for the tag-based prediction, which can be explained by our better preprocessing of the dataset. For the distribution-based prediction, we score 14% better than the authors in terms of mean, which is quite higher. As already explained, we believe this increase is due to our computation of the constant distribution: the mean distribution from the data, when Delbruel et al. estimated this constant using external information sources. Apart from these differences, the shapes of the CDFs look similar. We can thus go on using these two approaches as baselines to compare our proposed approaches with them.

5.2 Clustering distributions and applying Bayes classification

In our study of the Youtube dataset, we soon sensed the presence of common viewing patterns among videos. Figure 8 shows the similarity matrix (in terms of the accuracy $acc(v)$ defined in Equation 1) of videos distributions as a colormap, on a random subsample of one fortieth of the dataset (indeed, the full similarity matrix would have consumed 2.6Tb of RAM, which was prohibitive). This Figure gives first insights of the presence of common viewing patterns: many rows are mostly white (meaning this video's distribution is very dissimilar to other ones) except for a few columns they cross (to which they are similar). On the other hand, some rows are mostly black, which means these videos' distributions are not well differentiable from others. Still,

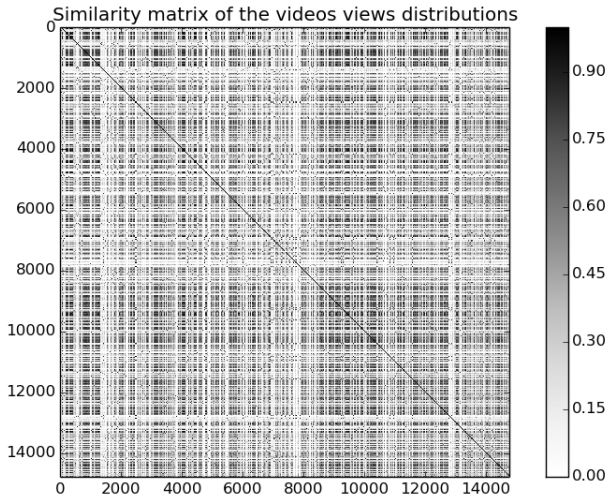


Figure 8: Similarity matrix of the distributions of a random subsample of the Youtube dataset, using the similarity metric proposed in Equation 1.

we felt a good potential for clustering in these distributions, and envisaged a prediction scheme for geographic distributions of videos' views based on it: by creating clusters of videos grouped by similarity of their geographic distributions, it should be possible to study the most used tags among a certain viewing behaviour. When an incoming video would be presented, we could match it to a certain cluster according to its tags, and thus infer its distribution as being the same as the cluster's mean distribution of views.

The process of this prediction would be the following:

- Split our dataset in two parts: the training set $\mathcal{V}_{\text{train}}$, and the testing set $\mathcal{V}_{\text{test}}$, according to a certain *training set size* γ (usually between 50 and 80% – we used 70%);
- On $\mathcal{V}_{\text{train}}$, apply a clustering algorithm on the videos viewing distributions matrix $\mathbf{CD}^{\mathcal{V}_{\text{train}}}$, leading to K clusters $\{\mathcal{C}_l\}_{0\leq l<K}$, each with a mean distribution of views $\mathbf{cd}_{\mathcal{C}_l} \in [0,1]^p$;
- Count tags among each cluster $\mathcal{C}_l$, leading to a matrix $\mathbf{BoW}_{\mathcal{C}} = \{\mathbf{bow}_{\mathcal{C}_l}\} \in \mathbb{N}^{K\times m}$, such that $\mathbf{bow}_{\mathcal{C}_l,j}$ represents the number of times that tag t_j appears in $\mathcal{C}_l$;
- When a new video $v \in \mathcal{V}_{\text{test}}$ is presented, apply a classification algorithm to compare its BoW of tags $\mathbf{bow}_v$ to the rows in $\mathbf{BoW}_{\mathcal{C}}$, and find the cluster $\mathcal{C}_{l_v}$ it best fits in;
- The estimate of video v's geographic distribution of views is the cluster $\mathcal{C}_{l_v}$'s mean distribution: $\mathbf{cd}_{\mathcal{C}_{l_v}}$.

As one may see, this technique is twofold: it contains a clustering part, that we will cover in Section 5.2.1 and a classification part, that will be presented in Section 5.2.2.

5.2.1 Clustering

As a preliminary experiment, we started out by applying the k-means clustering algorithm [12] to our full dataset, with the videos geographic distribution of views **CD** as input. K-means is one of the most famous and straightforward clustering algorithm, with only one main parameter: k, the number of clusters. It falls into the category of *partition-based* clustering algorithms. We empirically set k to 300, which is slightly higher than the number of countries (241), in order to obtain at least more than country-specific clusters.

Given a dataset of n vectors $(\mathbf{x}_1, ..., \mathbf{x}_n)$, the k-means algorithm aims at finding the best set of k partitions (or clusters) $\mathbf{S} = \{S_1, ..., S_k\}$ to minimise the intra-cluster variance (the sum of the euclidean distance of each element of partition S_i to its center μ_i). In other words:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

K-means is known to have several limitations: due to its random initialization of the clusters' centers $\boldsymbol{\mu}_i$, it does not yield the same result after each run (we will thus study a particular draw of clusters, not an optimal partitioning); it can fall into local minima, or suboptimal clusters, because it only aims at reducing the intra-cluster variance, but does not account for inter-cluster variance; it is not robust to outliers, which can lead to big clusters of unrelated data; finally, k-means assumes that clusters are convex, which is likely to be false in our high-dimension distribution space.

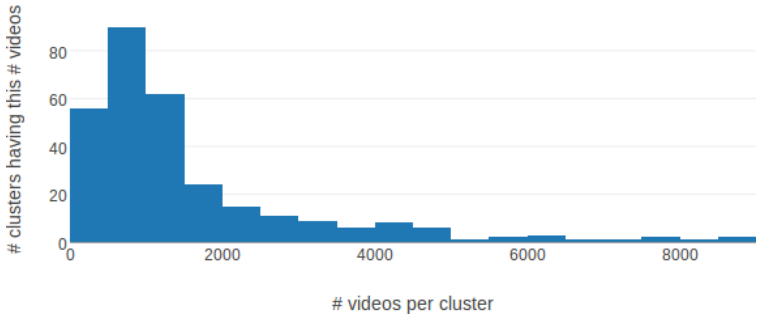


Figure 9: Histogram of the number of videos per cluster

We displayed, in Appendix A, some information on six of the three hundred clusters we computed using k-means. For each cluster $\mathcal{C}_l$, we gave the number of videos it contains (to be compared with Figure 9 that reads the distribution of videos per cluster), a world map showing its mean geographic distribution of views $\mathbf{cd}_{\mathcal{C}_l}$ (such that darker countries have the most views), and a cloud of its most frequent tags in $\mathbf{bow}_{\mathcal{C}_l}$ (the bigger the tag, the more frequent it is in the cluster). We recognised three main types of clusters, and displayed one cluster type per column in Appendix A. They are the following:

- *Country-specific clusters*, such as cluster #3 (that has 88% of its views in Brazil) and #18 (with 91% of its views in France). As one can see, in both cases, the originating country's name appears as one of the most frequent tag ('brazil,', 'brasil', 'france'), along with words in their spoken tongue ('humour', 'chanson'...) city names ('marseille'), and celebrities ('ronaldinho', 'booba'...). This observation comforts us in our assumption that tags are correlated with

viewing behaviours, and origins of the video's consumers. Some of these clusters comprise several countries with the same spoken language (such as France, Morocco, Algeria and Tunisia, or Spain, Mexico, Argentina and Colombia);

- *Topic-specific clusters*, such as cluster #8 (whose frequent tags clearly show a focus on the video game *Call of Duty*) and cluster #34 (which is dedicated to soccer, with a focus on Hispanic clubs and celebrities). We did not anticipate this type of cluster, but their appearance is interesting: it means that some topics have a specific distribution of their audience, may it be scattered among many countries. To give the detail: in cluster #8, the USA is responsible for 72% of the views, the UK 14%, while Canada, Germany and Australia lie between 1 and 4%, the other countries falling below 1%. The USA accounts for 23% of cluster #34's views, Spain 11%, Mexico 5%, and 15 other countries lie between 1 and 4%. As one may see, seeing such complex distributions of views arise as one cluster, with such specific tags, was hard to expect. It is reminiscent of the content and geographic localities that we presented as characteristics of UGC services in Section 2.1.3;

- *Misc clusters*, such as clusters #21 and #242. These clusters tend to be large, to have the same most frequent tags as the whole dataset (such as 'the', 'funny', 'music', 'video'...), and a similar geographic distribution as the whole dataset ($\bar{\mathbf{cd}}$) with a major US component (that accounts for 28% of the global distribution of views in our dataset). These clusters are legion, but hard to interpret, since they look alike a lot. We envisage two types of misc clusters: well clustered popular American videos (given the presence of a lot of American celebrities in the tags), showing small differences that we could not foresee (as for the topic-specific clusters); and poorly clustered videos, that are not so similar, but exhibit the same characteristics as the whole dataset once grouped together, as we could expect from a random sampling.

Of course, we need to keep in mind the limitations of this first clustering: we picked an empirical number of clusters, which may cause a lot of damage to k-means' results, since outliers can severely perturb clusters. Besides, k-means, its euclidean norm, and its assumption on the convexity of clusters, might be too simplistic for the structure of our data. Overall, we conclude that clustering is a valid approach to estimate the geographic distribution of new videos, as we see that clusters containing specific tags arise, such that it should be possible to match new videos to precise viewing behaviours using the tags.

Evaluating clustering results We now want to assess the quality of a clustering result, in order to calibrate the best parameters for an algorithm, or even to compare the performance of several algorithms. Many clustering evaluation metrics exist [17], that can be split in two categories: external evaluation, and internal evaluation metrics. External metrics use information that was not available to the clustering algorithm: a ground-truth such as the real cluster label of each datum. In our case, we do not have such ground-truth, since we do not know in advance which cluster a video belongs to. Thus, we will have to use internal metrics to evaluate the quality of our predictions.

These metrics usually assign the best score to the clustering with the highest similarity inside clusters, and the lowest similarity between clusters. The Silhouette coefficient [18] goes even further: it assigns to each clustered datum a measure of how similar it is to its own cluster (cohesion) compared the other clusters (separation). Once averaged, it provides a refined evaluation metric

for the whole clustering. Alas, Silhouette requires precomputing a dense matrix of element-wise similarity, or calculating a prohibitively large amount of distance computations. In other words, the Silhouette coefficient is not scalable to our 600'000 elements dataset.

After a study of the remaining metrics available, we decided to focus our attention on two of them: the Dunn Index [5], and the Davies-Boulding Index [3]. Given k clusters, they both only make use of the same information: the intra-cluster distance vector $\Sigma = \{\sigma_i\} \in \mathbb{R}_+^k$ and the inter-cluster distance matrix $D = \{d_{i,j}\} \in \mathbb{R}_+^{k \times k}$. We computed these distances using the Manhattan norm, as proposed in Section 5.1:

$$\sigma_i = \frac{1}{2} \sum_{\mathbf{x} \in S_i} \sum_{\substack{\mathbf{y} \in S_i \\ \mathbf{x} \neq \mathbf{y}}} \|\mathbf{y} - \mathbf{x}\|_1 \quad ; \quad d_{i,j} = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_1 \text{ where } \boldsymbol{\mu}_i = \frac{\sum_{\mathbf{x} \in S_i} \mathbf{x}}{|S_i|}$$

From these measures, the indexes are computed as follows:

- The Dunn Index $DI(\mathbf{S})$ consists in a ratio of the minimal inter-cluster distance over the maximal intra-cluster distance:

  $$DI(\mathbf{S}) = \frac{\min\limits_{1 \leq i < j \leq k} d_{i,j}}{\max\limits_{1 \leq i \leq k} \sigma_i}$$

  As such, *a higher Dunn Index indicates a better clustering.* A problem with this formulation is that poorly behaved clusters will greatly influence the Dunn Index because of the max term at the denominator, even if all the other clusters are tightly packed;

- The Davies-Bouldin Index $DBI(\mathbf{S})$ computes the average of the biggest $R_{i,j}$ among clusters. $R_{i,j}$ is a measure of dissimilarity among clusters, such that $R_{i,j}$ is smaller when S_i and S_j are well separated, and higher when they look alike. The Davies-Bouldin Index thus reads the average maximum pairwise dissimilarity among clusters:

  $$DBI(\mathbf{S}) = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} R_{i,j} \text{ s.t. } R_{i,j} = \frac{\sigma_i + \sigma_j}{d_{i,j}}$$

  As such, *a lower Davies-Bouldin Index indicates a better clustering.*

Both these metrics have some flaws: because of the inter and intra-cluster distances measures, they assume convexity of the clusters, as k-means does; and they are data and algorithm-dependant. This means that we cannot use them to compare results among different clustering algorithms. Indeed, because different algorithms will perform very different partitions of the data, these scalar metrics will not be pertinent to understand the differences between the algorithms' results. Nevertheless, since k-means does also assume convexity of the clusters it proposes, we considered using these indexes to calibrate k, the number of clusters. This result could be generalised, as a general result on the ideal amount of clusters on our dataset.

Appendix B shows the indexes values after running several iterations of k-means on our full dataset, as a function of k. The top plot reads the results of varying k from 180 to 480 with an increment of 20, with 50 different k-means results per value of k. The bottom plot reads similar

results with k varying from 100 to 200, with a step of 10, this time with only 25 iterations per value of k. The values represent the mean indexes among the iterations, while the error bars represent the standard deviation between iterations for this value of k. As one can see, it is hard to figure out a best k from these indexes. The high standard deviations at every point are a sign of the huge variability of k-means results. Nevertheless, the top plot seems to indicate that 180 is a very good value for k, since DI is highest (0.044) and DBI lowest (2.4) at this point, both with a very small standard deviation (resp. 0.0016 and 0.022). But, after redoing the experiment with new values for k, as shown on the bottom plot, we obtained way less optimistic results (with a DI of 0.037 and a DBI of 2.44), with far worse variances (of resp. 0.015 and 0.036), that did not stand out from the other values of k. We blame luck for the encouraging results of the top plot, and conclude – after other unsuccessful attempts at the same experiment – that Dunn and Davies-Bouldin Indexes do not provide sufficient information for us to determine an optimal number of clusters in this k-means scenario. We suggest that several iterations of k-means, as already stated, yield very different and hardly comparable clusters.

We envisaged other solutions, such as the DBSCAN clustering algorithm, which is a *density-based* clustering technique. Though, it also has its set of parameters to calibrate, that are not as easily understood as the number of clusters in k-means. In the end, other clustering algorithms, when they scaled to our dataset, did not perform better than k-means. We preferred to keep k-means, with an empirical value of k of 300, that proved to work, and went on to classifying new videos to their closest cluster in terms of tags.

5.2.2 Matching new videos to a cluster

Matching a new video to a cluster using tags is a problem of document classification, a well-defined problem in computer sciences. We decided to use Naive Bayesian inference [13] for that matter, which is based on the Bayes rule:

$$P(\mathcal{C}_l|\mathbf{X}) = \frac{P(\mathcal{C}_l)P(\mathbf{X}|\mathcal{C}_l)}{P(\mathbf{X})}$$

where $\mathbf{X}$ represents the data we want to test, $\mathcal{C}_l$ is an hypothesis (in our case, the cluster label of X). The probabilities in the equation are: $P(\mathcal{C}_l|\mathbf{X})$, the *posterior probability*, the probability of class $\mathcal{C}_l$ given $\mathbf{X}$; $P(\mathcal{C}_l)$ is the *prior probability* (shortened as *prior*), the probability of class $\mathcal{C}_l$; $P(\mathbf{X}|\mathcal{C}_l)$ is the *likelihood*, the probability of class $\mathcal{C}_l$ given a fixed $\mathbf{X}$, or the compatibility of an input with a fixed hypothesis; finally, $P(\mathbf{X})$ is the *model evidence*, the probability of the input in the studied world.

In Bayesian learning, we study the probability of having a fixed input $\mathbb{X}$ belonging to a certain class $\mathcal{C}_l$, which is the varying parameter. As such, we do not need to compute the model evidence, that will stay constant when $\mathcal{C}_l$ changes. Instead, we only compute the terms at the numerator, that are proportional to the posterior probability for a given piece of data.

In our case, $\mathbf{X}$ is the bag of words of tags of the video v we wish to match to a cluster: $\mathbf{bow}_v = \{bow_{v\,j}\}_{0 \leq j < m}$ for m the number of tags in our full dataset. *Naive* Bayes makes the assumption that every feature in $\mathbf{X}$ is independent from one another; in our case, the tags. This is a strong assumption, which is most often false (as it is certainly in our dataset: some tags are very correlated, and appear together most of the time). Though, Naive Bayesian inference still provides very good results, and is widely used in the literature. With these simplifications, we can rewrite the posterior probability as follows:

$$P(\mathcal{C}_l|\mathbf{bow}_v) \propto P(\mathcal{C}_l)P(\mathbf{bow}_v|\mathcal{C}_l) = P(\mathcal{C}_l) \prod_{0 \leq j < m} P(bow_{v\,j}|\mathcal{C}_l)$$

With this simplifications, we can compute all the right terms from our data: $P(\mathcal{C}_l)$, the prior, could be computed as the proportion of videos inside cluster $\mathcal{C}_l$, but we prefer a uniform probability, since the biggest clusters are not the most accurate with our k-means clustering; $P(bow_{v\,j}|\mathcal{C}_l)$, the likelihood of having tag t_j in cluster $\mathcal{C}_l$, is in direct relation to the number of times t_j appears in $\mathcal{C}_l$. We compute the likelihood using additive smoothing, adding a parameter α, in order to avoid null values in the likelihoods (otherwise, the absence of a new video's tag in a cluster would make the posterior probability for this cluster fall to zero). α can have a maximum value of 1 (which is called Laplace smoothing), down to 0 (no smoothing). The likelihood of tag t_j in cluster $\mathcal{C}_l$, given the bag of words of this cluster's tags $\mathbf{bow}_{\mathcal{C}_l}$, is thus computed as follows:

$$P(t_j|\mathcal{C}_l) = \frac{\alpha + bow_{\mathcal{C}_l,j}}{m\alpha + \sum\limits_{0 \leq j < m} bow_{\mathcal{C}_l,j}}$$

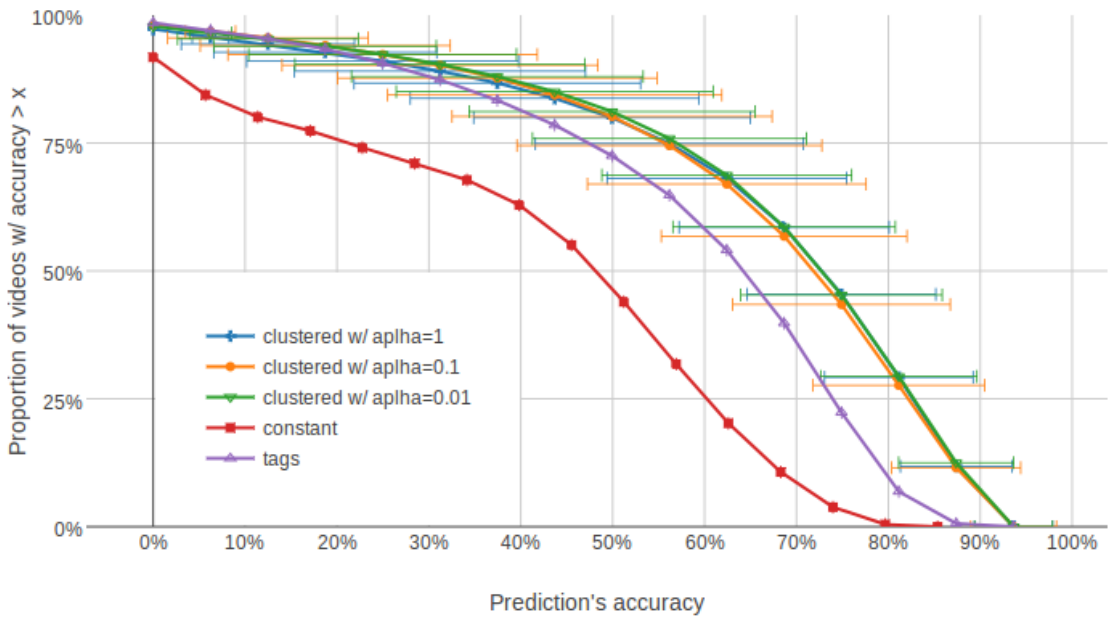
5.2.3 Experimental results

Given the pipeline that we just presented, we made several experiments using several configurations of our algorithms. For each experiment, we split our dataset in two: a training set $\mathcal{V}_{\text{train}}$ (containing 70% of the dataset, or 327,057 videos) and a testing set $\mathcal{V}_{\text{test}}$ (containing the remaining 30%, or 140,166 videos). First, k-means was applied to the training set, generating $k = 300$ clusters of videos (although other values were tried). For each cluster $\mathcal{C}_l$, the mean geographic distribution of its videos $\mathbf{cd}_{\mathcal{C}_l}$ was computed, along with its bag of words $\mathbf{bow}_{\mathcal{C}_l}$, the sum of its videos respective bag of words. Then, Bayes was trained: we gave to each cluster $\mathcal{C}_l$'s prior $P(\mathcal{C}_l)$ the same uniform probability of $1/k$, and each tag t_j's likelihood $P(t_j|\mathcal{C}_l)$ was computed using $\mathbf{bow}_{\mathcal{C}_l}$, as explained previously. Finally, in the prediction phase, when a new video v was presented, we computed its bag of words of tags $\mathbf{bow}_v$, and applied the Bayesian classifier to find the cluster $\mathcal{C}_l$ that had the biggest posterior probability $P(\mathcal{C}_l|\mathbf{bow}_v)$, and gave this cluster's mean distribution as v's geographic distribution of views: $\hat{\mathbf{cv}}_v = \mathbf{cd}_{\mathcal{C}_l}$.

Once all the videos from $\mathcal{V}_{\text{test}}$ had been predicted, we obtained a vector of predicted distributions $\hat{\mathbf{CV}}_{\mathcal{V}_{\text{test}}}$ that we compared against the ground-truth $\mathbf{CV}_{\mathcal{V}_{\text{test}}}$ to obtain an accuracy vector $\mathbf{acc}_{\mathcal{V}_{\text{test}}} = \{acc(v)\}_{v \in \mathcal{V}_{\text{test}}}$.

Finally, since the k-means clustering brings in a lot of variability, we made 10 experiment iterations per configuration of our algorithm, with the same training and testing sets. This allowed us, per video, to obtain their mean accuracy $\mu_{acc(v)}$ and accuracy's standard deviation $\sigma_{acc(v)}$, that we compared by plotting the Cumulative Distribution Functions (CDFs) of a configuration's videos accuracies, with errors bars representing the mean standard deviation of videos having an accuracy of x.

Figure 10 reads the results of our predictions with three different values of α: 1, 0.1, and 0.01, along with our baselines (the constant prediction and the prediction using tags distributions seen in Section 5.1). We see that the three plots have better results than our baselines, which shows that clustering distributions is a valid approach to predict future videos' distributions of views. We also see that our three predictions are graphically very close to one another. The statistics point the same fact: differences between the three approaches in terms of means and medians are minimal.



Statistics for our results

Approach	mean acc.	median acc.	mean std. dev.
Clustering with $\alpha = 1$	0.717	0.792	0.091
Clustering with $\alpha = 0.1$	0.725	0.791	0.095
Clustering with $\alpha = 0.01$	0.718	0.783	0.107
Constant distribution	0.469	0.539	n.a.
Tags distribution	0.646	0.706	n.a.

Figure 10: On top: CDF of the accuracy of geographic distribution of videos' views prediction using different configuration of our clustering & classification algorithm, over 10 iterations. The error bars represent the standard deviation of videos having an accuracy of x. Bottom: statistics of our experiment.

However, the standard deviation of the results is inversely proportional to α. We understand that a bigger value of α allows to reduce the variability of the predictions, partly caused by the poorness of the clustering.

We informally tried two other modifications of our algorithm, but lacked time to run several iterations of the experiment as we did with the α values. We report our observations here:

- We considered changing the number of clusters returned by k-means, between 200 and 400. Surprisingly, the accuracy remained unchanged with an α value of 1, and did not vary much with other values either. This is a sign of a number of redundant clusters, and a good indicator of the Bayes classification's efficiency;

- When computing the closest clusters to a new video v, we made an average of the top 3 or 5 clusters' distributions, weighted by the values of $P(\mathcal{C}_l|\mathbf{bow}_v)$ returned by the Bayes classification. The results were, in every case, worse than using only the best cluster's distribution as the prediction of v's distribution of views. We believe that the first cluster is accurate

enough to achieve a good prediction, and that averaging with other clusters only adds noise. Indeed, we just saw that the clusters were already redundant, so this averaging is most likely useless.

In the end, we are satisfied with the results of our approach, that show much better results than the tags distribution approach. The clustering, though, is a complicated problem to solve in our 241 space of distributions. We believe the first step to improving this would be to preprocess the distributions, through principal component analysis (PCA) or feature aggregation (that groups together similar countries), for instance. We could then use other clustering techniques, more complex and less fast than k-means, on an input space with a reduced dimensionality. Another problem that often arises with clustering is the dynamic updating of the clusters. Indeed, the clusters that were found in this 2011 dataset are probably very different from the ones we would find if our dataset was brand new – the topic-specific clusters, mostly, would probably be very different, or at least be updated with new trending topics with a particular distribution of views.

5.3 Nearest neighbours

Given the difficulties we encountered with our clustering approach, we proposed another method, based on the *k-nearest-neighbours* (kNN) algorithm. It consists in finding, for any new video, its k nearest videos in the training set, in terms of tags. Then, the predicted distribution of a new video is computed by aggregating the geographic distributions of views of its nearest neighbours.

This approach has the advantage to get rid of the clustering issue, and is thus more adaptable to the fast update rate of an UGC service's database. Instead, the problem lies mostly in the finding of nearest neighbours of newly uploaded videos in a huge database of existing videos (which is an indexing problem), and in defining suitable aggregation techniques of the distributions. Also, since we will directly use the tags to match videos together, preprocessing and filtering tags would be more important here. Indeed, some tags add more information than others, when they do not only add noise (like 'the', which is too widely used to be useful, against 'brasil', that gives information on the language and on the location of the video).

We only had little time to spend on this approach, so we will only focus on the indexing part, with the use of the MinHash algorithm, that gives an approximation of the Jaccard Index, a similarity metric specifically intended to compare sets of data like our tags vectors.

5.3.1 Jaccard Index and the MinHash algorithm

To find videos with similar tags is a problem of defining the right metric in a high-dimension binary space: the bag of words of the videos' tags. An accurate solution for that matter is the Jaccard Index, that is used to measure the similarity of two sample sets A and B as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \in [0,1]$$

In our case, the intersection $\cap$ between two videos is the number of tags they share, while their union $\cup$ is the sum of their tags number minus their intersection. Two videos having exactly the same tags will thus have a Jaccard Index of 1, while two videos that do not have any tag in common will have a Jaccard Index of 0.

Since a lot of videos do not have single tag in common, an $n \times n$ matrix of pairwise Jaccard similarity between videos could fit in memory in our case, because it would be very sparse (containing

mostly 0 values). Though, the processing time needed to compute the Jaccard similarity of a new video to all other existing videos in the dataset remains prohibitive. A faster approximation of the Jaccard similarity is the MinHash algorithm [1], that avoids explicitly computing the intersection and union between the two sets.

It consists in choosing a number n_{MH} of hash functions h. We define $h_{min}(S)$ to be the minimal number of S with respect to h – that is, the minimal value of $h(x)$ for x in S. The interesting result is that the probability that $h_{min}(A) = h_{min}(B)$ is equal to the probability of $J(A, B)$:

$$P(h_{min}(A) = h_{min}(B)) = J(A, B)$$

Now, we have defined a family of hash functions $\mathcal{H} = \{h_l\}_{0 \leq l < n_{MH}}$. We compute, for each video v, a signature vector $\mathbf{mh}_v \in \mathbb{N}^{n_{MH}}$ that contains the min hash value of each hash function with regard to the indices of the positive values in the BoW of v: $\mathbf{bow}_v \in [0, 1]^m$:

$$\mathbf{Ids}_v = \{j \ : \ (\mathbf{bow}_{v,j} = 1)\} \ : \ \mathbf{mh}_v = \{h_{min}(\mathbf{Ids}_v)\}_{h \in \mathcal{H}}\}$$

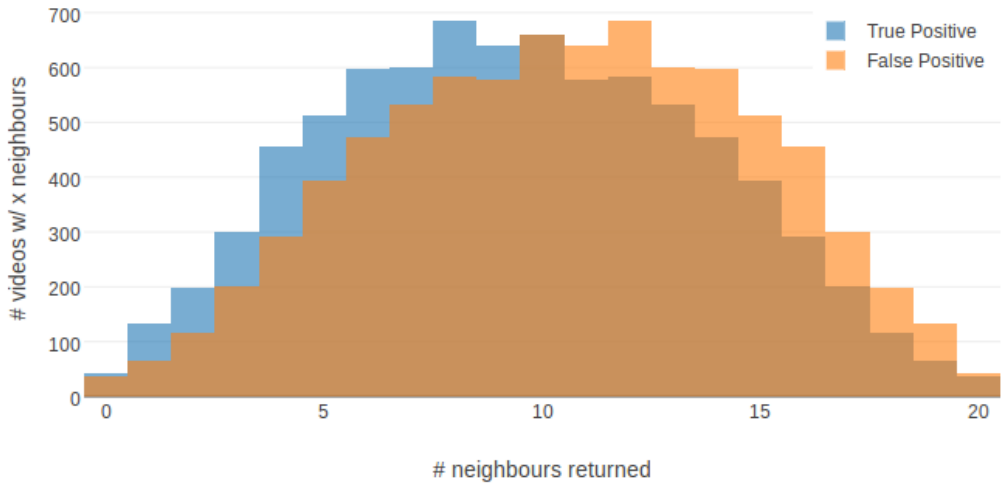


Figure 11: Histogram of the True Positive and False Positive neighbours returned by MinHash with regard to the Jaccard similarity, for an ensemble of 20 nearest neighbours requested for each video in the dataset.

Testing MinHash's efficiency Before employing MinHash as a similarity metric, we experimented the results it output, compared to the full computation of the Jaccard similarity matrix. We took the whole dataset, computed the full Jaccard matrix of pairwise similarities (thus of size $[0, 1]^{n \times n}$), and the MinHash signature vector $\mathbf{mh}_v$ for each video in the dataset. Then, we picked the 20 closest videos of every video using Jaccard and MinHash, and counted the number of True Positive (similar videos output by MinHash and Jaccard), and the number of False Positive (videos that MinHash considered the closest, that were not in the set returned by Jaccard). An histogram of the results is given in Figure 11. The mean of True positive results was 9.44, its median number 9.00; the mean of False Positives was 10.6, and its median 11.0:

As we see, the results are not very encouraging: more than half of the 20 neighbours returned by MinHash are False Positives. We will still attempt to predict videos' distributions of views using this approach, knowing this indexing part has a lot for room for improvement.

5.3.2 Experimental results

Once again, we performed our experiments using the same pipeline as already proposed: we split our dataset in two (with the same ratio of 70% of videos in the training set $\mathcal{V}_{\text{train}}$), and compute the MinHash signatures of all the videos in $\mathcal{V}_{\text{train}}$. Once a new video v is presented, we compute its MinHash signature, and use it to retrieve the closest neighbours of v in the trainset. Finally, we compute the average of the neiret neighbours' viewing distributions, and set it as v's predicted distribution of views.

Figure 12 shows the results, in the same fashion as in Figure 10 obtained for 1, 5, 10 and 20 nearest neighbours returned, as opposed to our two baselines and our previous clustering scheme with $\alpha = 1$.

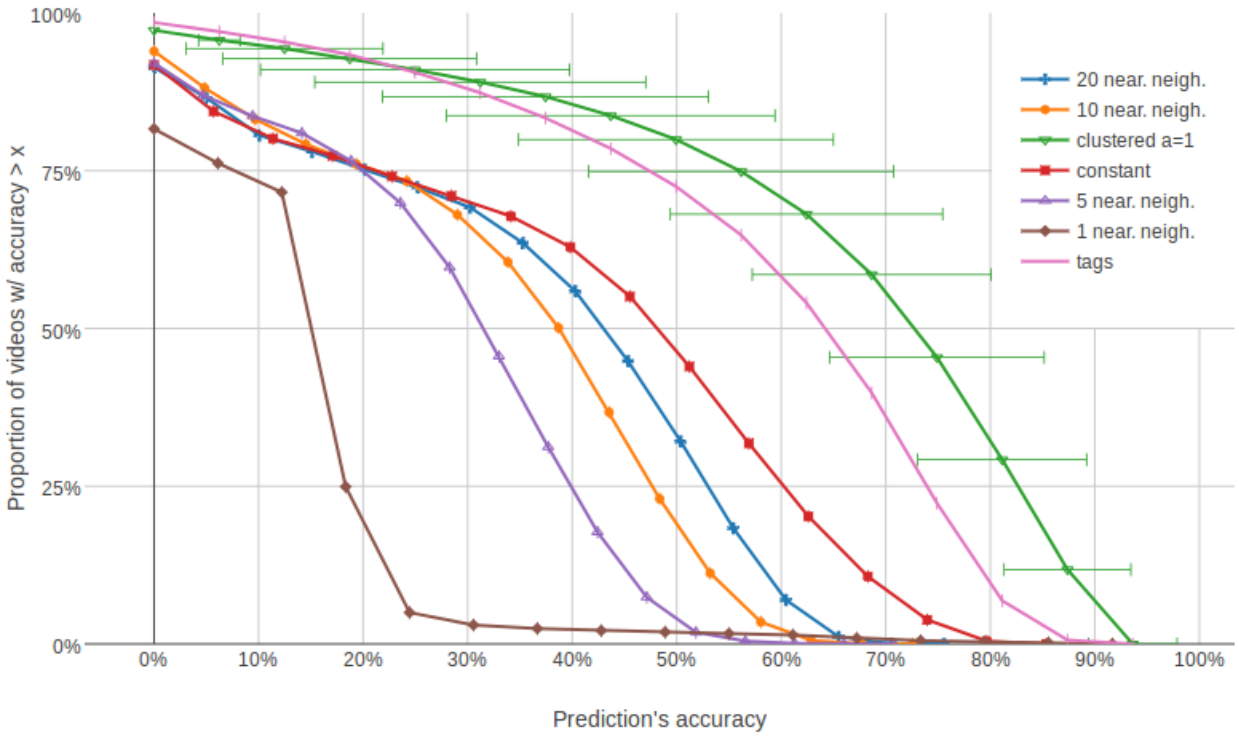


Figure 12: CDF of the accuracy of geographic distribution of videos' views prediction using a different number of nearest neighbours, with the two baselines and the result of clustered prediction with $\alpha = 1$.

As one may see, the results are, in all cases, worse than accuracy with the constant prediction, which is very poor. The best results obtained are with 20 nearest neighbours, with a mean accuracy of 0.416, a median accuracy of 0.482, and a mean standard deviation of 7.1×10^{-7}.

We believe that kNN is a valid approach, but it would require a lot of tweaking and optimizing to give acceptable results. We saw that MinHash, as is, is already not giving acceptable nearest

neighbours. Maybe does it need some improvement, or maybe MinHash cannot give acceptable results with such sparse vectors as our BoWs. The tags also really need to be preprocessed and filtered, here. We think of TF-IDF (Term Frequency-Inverse Document Frequency – we would mostly benefit the IDF term), that is a famous weighting technique of bag of words (or n-grams) used in text classification; we should also remove too frequent or geographically scattered tags (such as 'the') using metrics we could extract from the geographic locality of the tags in our dataset. On our clustering approach, the separation of the clustering and classification part accounts for such preprocessing. Indeed, the clustering is made on the distributions, where the noise inherent to tags is not a problem. Then the Bayes classifier matches videos to clusters using the likelihoods, that give more importance to infrequent tags, that we only find in a few clusters. Though, Bayes would surely benefit such filtering, but the poor performances of the clustering are far more problematic at this point of our work.

On the other hand, we are glad to see that the kNN approach has a far more deterministic behaviour than our clustering approach, when we read this negligible standard deviation it scores. Again, this is due to the poor results on the clustering part of our previous approach, which motivated our attempt at using kNN in the first place. There would also be some work to be done on the aggregation of the geographic distributions of the nearest neighbours returned by kNN. Our averaging is quite poor, and could benefit from weighting, for example.

For future work, we will have to choose between improving the clustering, or finding ways to enhance our kNN results; both issues being fairly hard to overcome.

6 Conclusion

During this internship, we thoroughly studied the predictive power of tags, in a UGC perspective, to infer the geographic distribution of views of newly uploaded videos.

Our first attempt at directly predicting the amount of views that had to be expected for a new upload, using only tags, proved to be impossible on our dataset. We believe this result can be generalised to other UGC scerarii. Indeed, the popularity of a video is far more influenced by factors that are not explained by the tags: amount of shares on other media (including radio, TV, word of mouth, Internet, and so on), marketing strategies, trendiness of the topic some content conveys, etc. Although this conclusion is pessimistic for the application we wanted to make with tags (using them in a proactive placement strategy), we believe it is an important result.

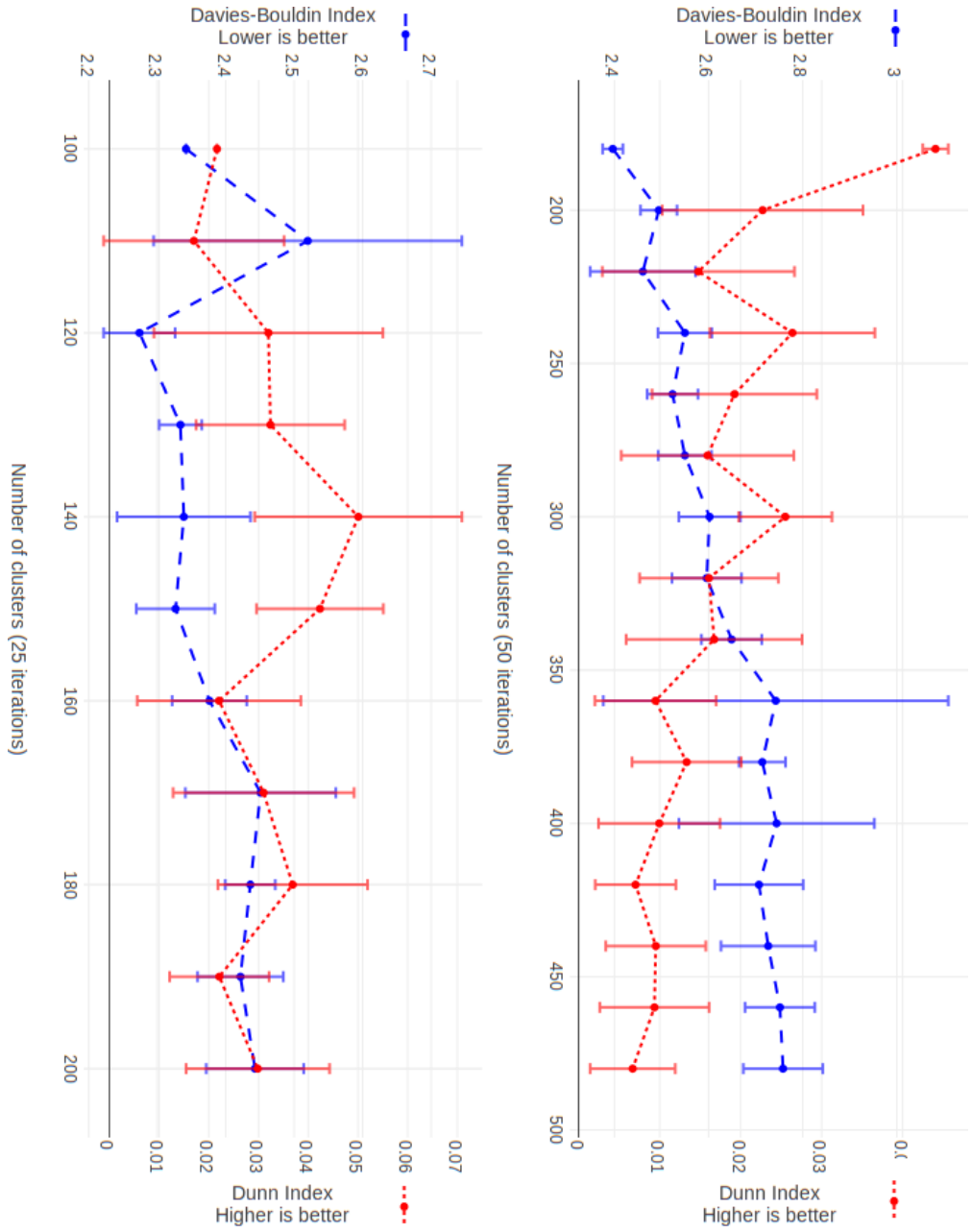
What more, it did not stop us on using tags to predict the geographic distribution of content's consumption. We showed that this was doable using statistical hypothesis testing, and proposed several approaches to achieve this goal. Although there are still many paths for improvement, we are faithful in the power of tags to infer where content will be mostly consumed, notably due to the language information tags convey. Whatsoever, other techniques could allow us to infer the amount of views of new uploads, which, combined with the geographic distribution we can infer from tags, will allow us to achieve successful proactive placement of the content on a CDN, using only the information available to UGC services providers.

A K-means clusters of geographic distributions

The following Figure shows four of the three hundred clusters that we computed by feeding the k-means clustering algorithm with the geographic distributions of all videos (**CG**). The world map shows the mean geographic distribution of videos views among a cluster (darker countries have a higher density of views), while the cloud of words reads the most frequent tags in this cluster (the bigger the tag, the most frequent it is).

B Evaluation of K-means clusters as a function of k

In this Figure, we show the Dunn Index and Davies-Bouldin Index scores of k-means clustering as a function of k, the number of clusters. The top plot shows the averaged scores over 50 k-means iterations for k varying from 180 to 480 with a step of 20. The bottom plot shows averaged scores over 25 iterations of the algorithm, with k varying from 100 to 200, with a step of 10.



References

[1] A. Z. Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29, Jun 1997.

[2] Anders Brodersen, Salvatore Scellato, and Mirjam Wattenhofer. Youtube around the world: Geographic popularity of videos. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 241–250, New York, NY, USA, 2012. ACM.

[3] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, April 1979.

[4] Stephane Delbruel, Davide Frey, and François Taïani. Exploring the Geography of Tags in Youtube Views. Research Report RT-0461, IRISA, Inria Rennes ; INRIA, April 2015.

[5] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[6] F. Guillemin, B. Kauffmann, S. Moteau, and A. Simonian. Experimental analysis of caching efficiency for youtube traffic in an isp network. In *25th International Teletraffic Congress (ITC 25)*, 2013.

[7] Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. Survey on social tagging techniques. *SIGKDD Explor. Newsl.*, 12(1):58–72, November 2010.

[8] Cheng Huang, Angela Wang, Jin Li, and Keith W Ross. Understanding hybrid cdn-p2p: why limelight needs its own red swoosh. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 75–80. ACM, 2008.

[9] Qi Huang, Ken Birman, Robbert van Renesse, Wyatt Lloyd, Sanjeev Kumar, and Harry C. Li. An analysis of facebook photo caching. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, SOSP '13, pages 167–181, New York, NY, USA, 2013. ACM.

[10] Yan Huang, Tom ZJ Fu, Dah-Ming Chiu, John Lui, and Cheng Huang. Challenges, design and analysis of a large-scale p2p-vod system. In *ACM SIGCOMM computer communication review*, volume 38, pages 375–388. ACM, 2008.

[11] Kévin Huguenin, Anne-Marie Kermarrec, Konstantinos Kloudas, and François Taïani. Content and Geographical Locality in User-Generated Content Sharing Systems. In *22nd SIGMM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pages 77–82, Toronto, ON, Canada, June 2012.

[12] M. A. Wong J. A. Hartigan. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[13] Liangxiao Jiang, Dianhong Wang, Zhihua Cai, and Xuesong Yan. *Advanced Data Mining and Applications: Third International Conference, ADMA 2007 Harbin, China, August 6-8, 2007. Proceedings*, chapter Survey of Improving Naive Bayes for Classification, pages 134–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[14] Samamon Khemmarat, Renjie Zhou, Dilip Kumar Krishnappa, Lixin Gao, and Michael Zink. Watching user generated videos with prefetching. *Signal Processing: Image Communication*, 27(4):343–359, 2012.

[15] Pamela McKenzie, Jacquelyn Burkell, Lola Wong, Caroline Whippey, Samuel Trosow, and Michael McNally. User-generated online content 1: Overview, current state and context. *First Monday*, 17(6), 2012.

[16] Al-Mukaddim Khan Pathan and Rajkumar Buyya. A taxonomy and survey of content delivery networks. *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report*, 2007.

[17] Lior Rokach. *Data Mining and Knowledge Discovery Handbook*, chapter A survey of Clustering Algorithms, pages 269–298. Springer US, Boston, MA, 2010.

[18] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[19] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Jon Crowcroft. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proceedings of the 20th international conference on World wide web*, pages 457–466. ACM, 2011.

[20] Simon J. Sheather. Density estimation. *Statist. Sci.*, 19(4):588–597, 11 2004.

[21] Sandvine Incorporated ULC. Global internet phenomena report - 1h & 2h 2014.

[22] Zhi Wang, Lifeng Sun, Chuan Wu, and Shiqiang Yang. Enhancing internet-scale video service deployment using microblog-based prediction. *Parallel and Distributed Systems, IEEE Transactions on*, 26(3):775–785, 2015.

[23] Hao Yin, Xuening Liu, Tongyu Zhan, Vyas Sekar, Feng Qiu, Chuang Lin, Hui Zhang, and Bo Li. Livesky: Enhancing cdn with p2p. *ACM Trans. Multimedia Comput. Commun. Appl.*, 6(3):16:1–16:19, August 2010.