A User Experience Model for Privacy and Context Aware Over-the-Top (OTT) TV Recommendations

Valentino Servizi
Technical University of Denmark
Kgs. Lyngby, Denmark
valse@dtu.dk

Sokol Kosta
Aalborg University Copenhagen
Copenhagen, Denmark
sok@cmi.aau.dk

Allan Hammershøj
Mediathand
Copenhagen, Denmark
allan@mediathand.com

Henning Olesen
Aalborg University Copenhagen
Copenhagen, Denmark
olesen@cmi.aau.dk

ABSTRACT

Conventional recommender systems provide personalized recommendations by collecting and retaining user data, relying on a centralized architecture. Hence, user privacy is undermined by the volume of information required to support the personalized experience. In this work, we propose a User Experience model which allows the privacy of a user to be preserved by means of a decentralized architecture, enabling the Service Provider to offer recommendations without the need of storing individual user data. We advance the current state of the art by: *i)* Proposing a model of User Experience (UEx) suitable for *Persona-based recommendations*; *ii)* Presenting a UEx collection model which enhances the user privacy towards the service provider while keeping the quality of her preferences predictions; and *iii)* Assessing the existence of the Persona profiles, which are needed for generating and addressing the recommendations. We perform several experiments using a real-world complete dataset from a medium-sized service provider, composed of more than 14,000 unique users and 33,000 content titles collected over a period of two years. We show that our architecture, in combination with our UEx model, achieves the same or better results, compared to state-of-the-art systems, in terms of rating prediction accuracy, without sacrificing user's privacy.

1 INTRODUCTION

In the early days, broadcast television was a *one-to-many* relationship. The signal traveled one-way from the content provider towards the consumer, and so did the contents. User privacy was at the highest possible level, since the user was anonymously and passively connected to the network.

With the introduction of an Internet-based "return path" for voting or rating of programs, initially in Digital Video Broadcast set-top-boxes, and later with Over-the-Top (OTT) TV [15], a completely different scenario has been set in terms of personalization and privacy. As IP-based services are taking over, almost any network today assigns an IP address to each node [15], and once the connection is established, the network allows bidirectional communication between each and every pair of nodes (e.g. between user and service provider).

By removing the constraint of one-way communication and assigning a unique address to each node of the network, service providers can easily collect detailed information from each node to build user profiles, and the level of personalization of the service theoretically has no limit [22]. In the best case scenario, users are only in control of the data collected actively, while information collected passively, such as user preferences and consumption patterns for example, seem to be out of their control. OTT TV services are defined as providers of the content that is usually associated with traditional broadcast television, but over the Internet. While this may sound suspiciously like IPTV (and they do share similar underlying technologies), it is in fact not the same thing: with IPTV customers pay the Internet Service Providers (ISPs) for the service, but with OTT TV the ISPs simply provide access to the Internet and, thereby, to the desired OTT TV service (which represents a different entity). Compared to broadcast TV, the common practice of OTT providers of gathering information about the content users consume raises the issue that personalization comes at the cost of privacy, causing concerns for the users, which are being exposed to over-disclosure of their personal data and viewing habits [26].

Many countries are pushing towards the concepts of *privacy by design* and *privacy by default*, in particular in the European Union (EU), driven by the General Data Protection Regulation (GDPR) law, which strongly highlights the concepts of *linkability* and *personal data* [10].

Inspired by these GDPR principles, and focusing on a solution that could ethically improve the "status quo", we here present the following contributions:
(1) We design a mathematical User Experience Model that allows a Service Provider to collect User Experience in a privacy-aware system that keeps personal information under the user's control within

her domain and a sanitized database within the Service Provider's domain.

(2) We validate the User Experience Model, which is used for user classification, by computing a type of rating based on weighted predictions with a many-fold comparative experiment, testing various similarity metrics and prediction algorithms.

(3) We perform extensive experiments using a real-world full dataset from a medium-sized service provider, composed of more than 14,000 unique users and 33,000 content titles collected over a period of two years. We use the Root Mean Squared Error (RMSE) to compare the performance of our solution with state of the art algorithms such as the FunkSVD, which is the winner of the popular Netflix competition, ItemItem and PearsMean [8]; We further apply the Lenskit tool [8], which includes the three algorithms, to our dataset as a benchmark for the algorithm we propose. Not only the Service Providers, but in particular the Content Providers (CP) have access to sensitive user data. The (CP) has access to the personal information of its paying customers; therefore, she can also link them to any k-anonymous data set maintained by the partnering Service Provider (SP), which the CP could access, probably by contract. In order to avoid this linkability hazard, we will prove that our solution raises the anonymity shield on the user consumption at the SP level and hence also at the CP level.

The results show that our User Experience Model achieves the same or better results, compared to state-of-the-art systems, while offering the potential for drastically increasing the user privacy. It enables Service Providers to accurately classify users' tastes by storing only a fraction of the users consumption data and thereby reducing drastically the linkability hazard. Moreover, we show how clustering techniques applied to the User Experience Model of an OTT TV user base leads to the description of distinct *Persona* profiles, defined as homogeneous groups of individuals whose consumption patterns and motivation can be represented as a set of statements derived from quantitative measures [6].

The concept of Persona is necessary for the anonymous user classification according to *Persona* profiles. Service personalization can then be made using the *Persona* profile to which a user belongs when accessing the service.

Our solution aims at assisting providers in fulfilling the obligations enforced by law. A privacy-enhanced design of the recommender system can provide a competitive advantage in at least two aspects for providers: *i*) Lower expected costs for implementing future system updates in order to comply with new regulatory restrictions; and *ii*) Lower risks for privacy litigations, such as the case settled by Netflix for $9 million in favor of its customers[1].

2 RELATED WORK

In this section, we present the related work of the privacy and context-aware Recommender Systems, providing also the technological and regulatory background of the privacy problem which we define as follows:

> *How can we enable OTT TV providers to drastically enhance their users privacy by augmenting existing technologies, protocols as well as recommender systems?*

[1]Case No. 11-cv-00379 (U.S. District Court for the Northern District of California)

The intuition behind our solution is that a provider can satisfy the privacy of individual users by providing recommendations to groups of users with similar characteristics. The solution is based on the concept of *Persona* and *Locked Persona Profiles*. Succeeding in solving this challenge is not only beneficial for the users, but also convenient for the service providers. We provide a solution by breaking such a problem into the following research questions:

- How to turn the passive collection of consumption data identified by the ontology User <-> consumes <-> content, into a model of User Experience (UEx)?
- How to collect consumption patterns while avoiding the linkability to the related personal records?
- How to provide *Persona* profiles from such a model of User Experience?
- Do *Persona* profiles exist in the OTT TV context?

2.1 Background

Digital Rights Management (DRM) technology has been specified, designed, and implemented in order to fight piracy. As such, satisfying DRM sets a minimum level of possible privacy for the user. Yang *et al.* [28] describe an approach resting on the allocation of an anonymity UserID for authentication, which allows anonymous access to DRM protected contents. In [11], Intertrust supplies OTT TV Service Providers with a DRM Service named Express Play (EP), which relies on an architecture that allows anonymous access to the licenses necessary to decrypt the contents delivered via a Content Delivery Network (CDN), by use of bearer token technology. This demonstrates that it is possible for a user to access DRM-protected content while keeping her anonymity.

In order to provide a personalized recommendation to any anonymous user accessing DRM protected contents, the design approach theorized by Cooper *et al.* [5] seems extremely relevant. According to [6], there are three questions that the *Persona* approach attempts to address. With a slight adaptation to the goal of this project these can be stated as follows: *i*) What different sorts of people are using OTT TV services? *ii*) How do their needs and behaviors vary? *iii*) What ranges of behavior and contexts need to be explored? Adapting their example to the OTT TV service, we might identify: *i*) Frequency of access to the service; *ii*) Whether the user likes or dislikes the consumed contents; *iii*) The motivation for accessing the service, e.g. information, education, or entertainment.

In [16], Hussain *et al.* introduce the concept of *Locked Persona Profiles (LPP)*, which can be *"used to generate a set of unlinkable proofs of ownership"* while accessing a service. Locked Persona Profiles are perfectly compatible with both the need of protecting users' privacy and the need to protect operators from misuse threats such as piracy. Furthermore, Locked Persona Profiles would allow the user to access the service without being *linked*, thus giving the individual control over her personal information with no chance for the Service Provider (SP) to access this information without the user's knowledge or consent [16].

A Recommender System usually needs to collect some amount of personal information about the user in order to provide the recommendation. This includes attributes, preferences, contact lists, among others [23]. Context information is not a part of the user profile, but may also need to be protected, e.g. the user's current

location. Kim Cameron's first and second law of identity [20] emphasizes user control and minimal disclosure, and the OAuth 2.0 framework [17] can help to put the user in control. OAuth 2.0 allows the user to grant restricted access to her personal information towards a Service Provider and defines a protocol for securing application access to her protected resources, such as identity attributes, through Application Programming Interfaces (APIs) [17].

Several methods have been proposed in the past for reducing linkability – here defined as the possibility of discovering a relationship between a user's consumption records or between a user's consumption and her identity [16]. k-anonymity occurs when *"every tuple in the microdata table released is indistinguishably related to no fewer than k respondents"* [4]. This can be achieved by removing identifiable information from the dataset. Crowd Blending relies on storing records about a group of similar users as a unique entity [13]. Differential Privacy relies on storing similar consumptions of different users by the same perturbed record [12]. Zero-Knowledge relies on representing the consumption of the whole user population by storing only a sample [14]. This approach is particularly interesting for two reasons: It seems to be more effective than the other techniques mentioned, and the cluster analysis necessary to exploit possible Persona profiles on a Zero-Knowledge dataset could be carried out directly, because the sampling happens beforehand. Therefore, the stored data do not need further sampling.

2.2 User Experience

User Experience (UEx) is very important for the service providers. Considering the specificity of the OTT TV application field, the User Experience should be exploited to provide users with good recommendations about linear or Video on Demand (VoD) media contents. Therefore, the interest is not the User Experience for OTT TV service itself, but rather the User Experience about the content available in the Electronic Program Guide (EPG).

As such, service providers define several metrics related to User Experience:

Service Consumption. In order to have any experience with the media content, users need to consume the content [25].

Context of consumption. The context in which the interaction with the service happens contributes to the experience about the content consumed in such a context [25].

Subjective Perceived Value. A positive or negative experience results from the contribution of multiple drivers [21].

Usage Cycles. The experience might be determined by cycles of interactions [27], and the next cycle might be influenced by the quality of the content itself, based for example on the personal experience [24] or on the experience of other individuals [27].

User Behavior Frequency. In [18], based on a data set provided by BBC, evidence has been presented about each user repeated access to a small amount of items compared to the total amount of available items.

2.3 Privacy enhanced recommender systems

According to the review presented in [9], privacy concerns are mainly related to: (1) an attacker which correlates obfuscated data about user with data from other publicly-accessible databases in order to link users with the sensitive information, or (2) an attacker using partial information obtained e.g. by colluding with some users in the network, to attempt reverse engineering the entire dataset.

The main solutions described in the literature to preserve privacy are the following. (1) *"Privacy preserving approach based on peer to peer techniques using users' communities"* with recommendations generated on the client side without involving the server [9]. (2) *"Centralized recommender systems by adding uncertainty to the data by using a randomized perturbation"* [9], where such a perturbation could be achieved using e.g. differential privacy [12]. (3) *"Storing user's profiles on their own side and running the recommender system in distributed manner without relying on any server"* [9]. (4) Hybrid recommender system that uses secure two-party protocols and public key infrastructure. (5) *"Agent based middleware for private recommendations service"* [9] presenting good performance in terms of Mean Average Error but also tuning issues in order to get a good coverage on the largest part of the user's population. However, most of the solutions focused on protecting users from external security attacks rather than reducing their exposure towards the service provider.

3 OUR USER EXPERIENCE MODEL

The first step towards achieving the architecture described in the previous section is to understand *who* is consuming *what*, *when*, *where*, and whether *she liked* what has been consumed (or not). Our dataset consists of more than 700,000 ZAP events, representing the fact that a user taps on a content, 14,000 unique users, and 33,000 content titles collected over a period of two years by a medium-size OTT TV provider.

We define the User Experience Model (UEx) as a tensor:

$$UEx : C_1^{m_1} \otimes C_2^{m_2} \otimes \cdots \otimes C_k^{m_k} \otimes P^{m_p} \mapsto \mathbb{R}^{m_1+m_2+\cdots+m_k+m_p} \quad (1)$$

where each of the $C_x^{m_y}$ represents a vectorial space of the x^{th} context in which the user zaps into media contents, and P^{m_p} represents the vectorial space of the media contents that she zaps; $m_1, m_2, \cdots, m_k, m_p$ are the dimensions of each vectorial space.

In a traditional recommender system users may provide ratings, and Latent Semantic Analysis (LSA) [3] using the Term Frequency Inverse Document Frequency (TFIDF) matrix can be applied to the contents' descriptions. Instead of ratings, we propose to use a similar approach for the TV broadcasting scenario, where the representation of contexts and TV program weights for each user act as "ratings" in the User Experience model, and we introduce the concepts of Zap Frequency Inverse Context Frequency (ZFICF) and Zap Frequency Inverse Program Frequency (ZFIPF). The difference is that while ratings are collected actively, our User Experience is computed from the passive collection of the zaps into contents, which define the components of the User Experience. Further details are given in the following.

3.1 The User Experience as a vector

Linearizing the tensor defined in Eq. (1), we obtain the User Experience matrix presented in Eq. (2). The matrix consists of the Zap Frequency Inverse Context Frequency, one slice for each context, and the Zap Frequency Inverse Program Frequency (last slice). Each slice, except the last one, represents one of the k possible contexts of consumption, while the last slice represents the TV programs

$$\begin{pmatrix} UEx(1) \\ UEx(2) \\ \vdots \\ UEx(n) \end{pmatrix} = \left(\begin{array}{ccc|c|ccc|ccc} ZFICF_{1,1,1} & \cdots & ZFICF_{1,1,m_1} & \cdots & ZFICF_{1,k,1} & \cdots & ZFICF_{1,k,m_k} & ZFIPF_{1,1} & \cdots & ZFIPF_{1,m_p} \\ ZFICF_{2,1,1} & \cdots & ZFICF_{2,1,m_1} & \cdots & ZFICF_{2,k,1} & \cdots & ZFICF_{2,k,m_k} & ZFIPF_{2,1} & \cdots & ZFIPF_{2,m_p} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ ZFICF_{n,1,1} & \cdots & ZFICF_{n,1,m_1} & \cdots & ZFICF_{n,k,1} & \cdots & ZFICF_{n,k,m_k} & ZFIPF_{n,1} & \cdots & ZFIPF_{n,m_p} \end{array}\right) \quad (2)$$

Equation 2: Linearized representation of the tensor described in Eq. (1). It is composed of slices corresponding to each of the five features, i.e. the four contexts (i) Time of Day, (ii) Time of Week, (iii) Time of Month, and (iv) Time of Year, see Eq. (3), plus one (v) concerning the TV-Programs consumed by the user, see Eq. (4). Each row maps the i^{th} user experience UEx(i). Therefore, just by looking at the distances between the users represented, this model allows measuring the similarity between their experiences against the TV-Program consumed within the contexts.

consumed. Within each slice, except the last one, each column represents the m_k^{th} split of the k^{th} context, while each column of the last slice represents one of the TV titles (or programs).

Each row represents the experience of the i^{th} user in the time span of her interest. However, it could also represent the experience of the i^{th} component of a cluster (*Persona*) in the time span of Service Provider interest. Therefore, the experience of each user can be collected as a vector having the same representation of one row of the matrix. This vector would be the summary of the User Experience and could be easily collected by the Service Provider and be put under the user's control by using OAuth 2.0 on the zaps-to-contents. Besides, if the users mapped in this way are arranged in clusters, each cluster would represent homogeneous experiences. If such homogeneous groups exist, they allow for user classification as well as a *"set of statements"* derived from the measures, fitting to the definition of a Persona.

3.2 User Experience Components

In this subsection we elaborate on the meaning and the nature of the User Experience components, and discuss the concepts of *ZFICF* and *ZFIPF*.

3.2.1 Zap Frequency Inverse Context Frequency. The Zap Frequency Inverse Context Frequency relates to the first $m_1 + m_2 + \cdots + m_k$ components of the User Experience tensor defined in Eq. (1). We define the Zap Frequency Inverse Context Frequency (ZFICF) for user i on the context segment j of context k as:

$$ZFICF_{ijk} = \frac{Z_{ijk}}{Tot\ Z_{ik}} \cdot \log \frac{Tot\ C_k}{AC_{ijk}} \text{ , where:} \quad (3)$$

- i indicates the i^{th} user, where i = 1, ..., n;
- the k^{th} context can refer alternatively to Time of Day, Time of Week, Time of Month, Time of Year, or Device (D), i.e. k = 1, ..., 5.
- j indicates the j^{th} segment of the k^{th} context, j = 1, ..., m_k. The k^{th} context is divided into a total number of segments equal to $Tot\ C_k = m_k$. For example, Time of Day could be divided in 24 segments, thus $m_{TimeofDay} = 24$, one per hour; but it could also be divided in 2 segments as day and night, in such case $m_{TimeofDay} = 2$.

The first factor of Eq. (3) is the Zap Frequency (ZF) recorded for the user i in segment j of context, where Z_{ijk} represents the amount of zaps recorded during the time span of interest, and $Tot\ Z_{ik} = \sum_{j=1}^{m_k} Z_{ijk}$ is the total amount of zaps recorded for user i in context k during the (same) time span.

The second factor is the Inverse Context Frequency (ICF) recorded for user i in context k, where $Tot\ C_k = m_k$, as already mentioned, is the total number of segments composing the context k; and $AC_{ik} = card\left(\left\{j \mid Z_{ijk} \neq 0,\ \forall j \in [1, m_k]\right\}\right)$ indicates the number of segments within the context k, in which the i^{th} user zapped at least once.

3.2.2 Zap Frequency Inverse Program Frequency. The Zap Frequency Inverse Program Frequency (ZFIPF) relates to the last m_p components of the User Experience tensor defined in Eq. 1. It is similarly defined by the following equation:

$$ZFIPF_{ij} = \frac{Z_{ij}}{Tot\ Z_i} \cdot \log \frac{Tot\ P}{AP_i} \text{ , where:} \quad (4)$$

- i denotes the i^{th} user, where i = 1, ..., n;
- the j^{th} element indicates the j^{th} TV program, where j = 1, ..., m_p. The total amount of TV programs available on the EPG is $Tot\ P = m_p$.

The first factor of Eq. (4) is the Zap Frequency (ZF) for the i^{th} user on the j^{th} program, where Z_{ij} represents the amount of zaps recorded during the time span of interest for user i on the TV program j, and $Tot\ Z_i = \sum_{j=1}^{m_p} Z_{ij}$ is the total amount of zaps of user i within the (same) time span.

The second factor represents the Inverse Program Frequency (IPF_i) for user i in the time span of her interest, where $Tot\ P$ is the total amount of TV programs available on the EPG, and $AP_i = card\left(\left\{j \mid Z_{ij} \neq 0,\ \forall j \in \left[1, m_p\right]\right\}\right)$ indicates the number of programs for which user i zapped at least once within the (same) timespan.

It is important to note that the ZFIPF over a large interval of time relies on the grouping of the items, especially in environment where, for example, TV series or news programs might recur continuously. If the grouping is not done properly, then the measure will be affected by a large error.

4 EVALUATION

In this section, we provide several experiments that prove the existence of Persona profiles based on User Experience. They can be effectively computed using a Zero-Knowledge Consumption Data Base (DB), which we define as a collection of data sampled on-line from the user consumption feedback in such a way that only a

portion having size below 10% of the consumption data is retained and stored in the DB in order to fulfill the definition provided for Zero-Knowledge in [14].

As the results of these experiments, we show and validate that: *i)* User Experience is effective for classifying a user according to such Persona profiles; *ii)* Zap Frequency Inverse Program Frequency predictions computed by looking for similar users from a sample of the dataset are reasonably accurate;. Finally, we provide an example of Persona profile, obtained by clustering a sample of the dataset, according to the definition stated in the introduction.

We use a real-world dataset, provided by a Media Company, which provides both Video On Demand and Linear TV on behalf of a Content Provider[2]. The data can be represented in a minimalistic and generic way by the following: <*contractId, contentId, deviceId, deviceType, zap_timestamp*>, where: *contractId* is the ID assigned to the contract by the Content Provider, e.g. at the start of subscription, in order to authenticate the user, referring to one or possibly more users active with the same contract; *contentId* is the ID assigned to the media content by the Content Provider, e.g. the TV Network; *deviceId* is the ID of the user's device authenticated while accessing the content, this is also used in combination with the *contractId* to identify a single user (*userId*); *deviceType* identifies e.g. Mobile, PC or TV; and *zap_timestamp* is the timestamp determining when the user switches the TV channel (it is an attribute of the ontology user - zaps into - content).

At this stage, for the purpose of organizing users in homogeneous groups, it is necessary selecting data about any user consuming contents alone and filter out data about any user consuming contents together with other individuals, because while the first group represents a clean signal we want to analyze, the second is affected by noise. Therefore, since it seems reasonable to consider mobile and tablet devices as personal and separate them from PCs and TVs, which are considered social devices. Distinguishing these devices based on the screen size[3], from the full dataset we restrict the data only to mobile devices and only keep the following information: <*userId, contentId, contentTitle, zap_timestamp*>. The resulting dataset is composed of 743975 Zaps[4], 14518 Users, and 33357 Content titles consumed over 2 years.

4.1 Persona classification and rating prediction

The purpose of this experiment is to verify the consistency of the User Experience modeled in Eq. 1 by predicting 90% of the users' preferences employing only 10% of them for the computation. Succeeding in such a challenging purpose will demonstrate that Service Providers could serve recommendations by maintaining only 10% of the data currently retained about the users consumption. The experiment consists in two steps: *i)* Testing the consistency of our solution by predicting the Zap Frequency Inverse Program Frequency (ZFIPF) and measuring the RMSE. We use 90% of the dataset to find target users and 10% for computing the prediction. *ii)* Testing the consistency of the User Experience Model by applying three popular algorithms implemented within the last updated version of Lenskit [8], to the ZFIPF computed on the whole same dataset.

4.1.1 Nearest Neighbor Mean, User Experience classification after SVD using Cosine and Chebyshev metrics (setup and execution). In order to compute the User Experience vector for each user from the zaps recorded in the dataset, in particular for computing $ZFICF_{ijk}$, we have set up each context, where: *i)* Time of Day context is divided in 4 segments: Morning, Noon, Evening, Night; *ii)* Time of Week context is divided in 3 segments: Week Days, Saturday and Sunday; *iii)* Time of Month context is divided in 2 segments: Far From Pay Check and Close To Pay Check (which has been considered at the end of the month); *iv)* Time of Year context is divided in 4 segments: Spring, Summer, Autumn, Winter.

Moreover, in order to avoid the Cold Start Problem (CSP) related to Users [23] while generating User Experience from the dataset, the Minimum Zap Threshold (MZT) for collecting both the Model User Sample (MUS) and the Target User Sample (TUS) has been set to 25 zaps after a number of preliminary experiments. As for the Cold Start Problem related to Contents [23], for collecting the Target Content Sample (TCS) we set the MZT to 1000 zaps. After applying Singular Value Decomposition (SVD) [3], for User classification we look for max 4 Nearest Neighbors (KNN) comparing the results obtained by using Chebyshev and Cosine metrics.

The User is represented by her User Experience Vector which is a row of the matrix in Eq. 2. Thus, we classify each User belonging to the Target Users Sample represented as in Eq. 2 against each of the relevant users belonging to Model Users Sample represented as in Eq. 2. Each random sample counts 10% of the population from which it has been extracted [29]. To make predictions with target a relevant amount of users having $ZFIPF \neq 0$ at least for one element of the Target Contents Sample (TCS). Therefore, when collecting Target Users Sample make sure to avoid any overlapping with Model Users Sample (MUS) adopted for computing the predictions. The experiment has been repeated 600 times (600-fold) and each of the times every sample (TCS, TUS, MUS) has been randomly collected again from the whole dataset, first splitting the MUS from the rest of the data.

The user classification and the ZFIPF prediction have been accomplished by the following steps. *i*) Compute the User Experience vector for each user of both the Target Users Sample and the Model User Sample. *ii*) Store the Zap Frequency Inverse Program Frequency (ZFIPF) regarding each media content within the TCS from the User Experience Vector (UEV) of each user belonging to the Target Users Sample in a new array (NA) and then set to zero the ZFIPF within the UEV origin. *iii*) For each user belonging to Target Users Sample, compute the K-Nearest Neighbor belonging to Model Users Sample having the relevant Zap Frequency Inverse Program Frequency not-null, using all the mentioned metrics. *iv*) For each metric, compute a Zap Frequency Inverse Program Frequency as mean of the values available from KNN. *v*) Compute the Root Mean Squared Error (RMSE) of each prediction from the NA. The random prediction has been generated using the range $ZFIPF \in [0, max(\text{NA})]$ instead of $ZFIPF \in [0, \infty]$, therefore it is more accurate.

4.1.2 Lenskit Experiment. Lenskit is a very popular and effective tool developed by the Grouplens from the Minnesota University [8].

[2]Data provided by Mediathand, company based in Denmark, which operates the OTT TV service on behalf of Glenten.
[3]We assume that the chance of a user watching some content together with others is directly proportional to the screen size.
[4]Event recorded when a user taps into a content.

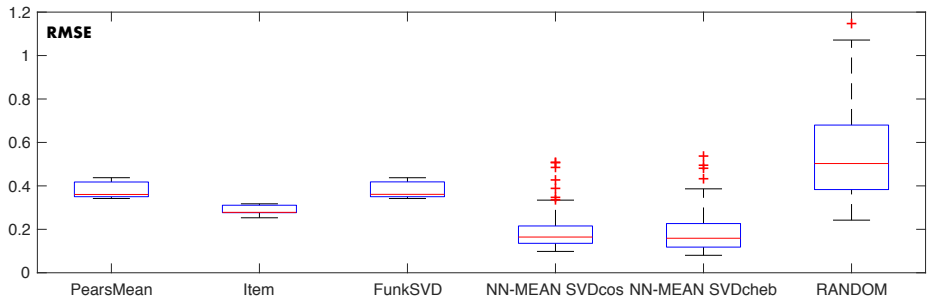

Figure 1: Boxplot of the RMSE resulting from (i) Lenskit 5-fold experiment - PearsMean, ItemItem, FunkSVD - on the Zap Frequency Inverse Program Frequency calculated using the whole dataset as $ZFIPF : TV\ Programs^{10^4} \mapsto \mathbb{R}^{\sim 10^4}$ and (ii) the 600-fold experiment (see Sec. 4.1.1) run using the whole dataset to predict ZFIPF as the Mean of the Target User Nearest Neighbors (NN-MEAN) against a random sample having 10% of the dataset size; the neighbors are based on Cosine or Chebyshev after SVD of the User Experience Model as $UEx : Time\ of\ Day^4 \otimes Time\ of\ Week^2 \otimes Time\ of\ Month^2 \otimes Time\ of\ Year^4 \otimes TV\ Programs^{10^4} \mapsto \mathbb{R}^{\sim 10^4}$.

It encapsulates several recommender systems algorithms such as ItemItem, PearsMean and FunkSVD. In particular, these have been taken into account for this experiment. The experiment was run on the ZFIPF computed using the whole dataset and provided in the same format of the Movielens ratings. The setup of the experiment is pretty straightforward and involves the selection of the ratings domain, which we set in the range $ZFIPF \in [0, max(\text{NA})]$. The precision has been set as maximum resulting from ZFIPF. Through some preliminary experiment we found the parameters ensuring the lowest RMSE. For example, the amount of features for FunkSVD, which is set to 40 by default, performs very poorly, while at 10 seems to achieve the best performance. Lenskit completes a 5-fold validation on the whole dataset. The RMSE results of the experiment are around 0, 4 and are presented in Figure 1.

4.1.3 Experiment Outcome (Figure 1). We aim at proving the consistency of the User Experience Model by comparing RMSE resulting from the experiment run with Lensikit against the results harvested using our solution. From the cross-validation based on the same dataset, it is possible to notice as a first result that the model predictions outperform the random assignment and are aligned to the state of the art algorithms provided by Lenskit. Moreover, from these results we can notice that the similarity metrics used for classification do not influence the RMSE, they all yield closely the same results. However, we chose Chebyshev, because e.g. compared to Cosine it could identify clusters where users are closer to a normal distribution.

4.2 Persona profiles from Zero-Knowledge DB

This experiment aims to finding distinct Persona Profiles. Using mostly the same set up of the previous experiments, therefore, a sample of users having 10% of the whole population's size, this experiment should take into account the following challenges: *i*) Which clustering algorithm is the best fit for finding distinct groups of similar users? *ii*) How to choose the amount of clusters in order to avoid over-fitting?

To choose between k-means [1] and hierarchical clustering [19], we perform a qualitative analysis of the Model Users Sample plotted after Singular Value Decomposition, choosing the first and the third main eigenvectors of the eigenbasis as latent dimensions (see Figure 2). From these results, we conclude that the second algorithm seems to be the right choice, since Shape, Density, and Size of the potential clusters seem irregular [1]. About the estimation of the amount of clusters, in order to avoid over-fitting it is possible to rely on a selection of algorithms for cross-validation, such as Calinski-Harabasz (CalHar) [2] and Davies-Bouldin (DavBou) [7]. In particular, these two methods have been used to restrict the interval of probable amount clusters from $[1, 60]$ to $[2, 30]$ and the latter interval has been used for setting up the final experiments carried out using Davies-Bouldin.

4.2.1 MUS clusters estimation. It is a starting point in the description of Persona profiles according to the definition provided in the Sec. 2.1. Moreover, thanks to this experiment, we converge towards the most probable amount of clusters in order to provide the top list of contents for each cluster as well as Time of Day, Time of Week, Time of Month and Time of Year, as features to characterize Persona profiles from the Model Users Sample which here is representing the Zero-Knowledge Consumption DB . The results of Algorithm 1 are presented in Table 1.

4.2.2 Experiment Conclusion. As we can notice from Fig 2, some clusters could be merged or discarded towards the best compromise. Perhaps a thorough cluster analysis would reduce the amount of relevant clusters to the minimum algorithmic estimation. Nevertheless, since the sampling applied to obtain the Model Users Sample is simulating the Zero-Knowledge Consumption DB, irrelevant clusters might represent important seeds for new clusters and new Pesona profiles could grow from there. Therefore, we considered relevant keeping clusters below 2% of the total population as seeds for potential new Persona profiles.

5 CONCLUSIONS AND FUTURE DIRECTIONS

Our analysis and experiments have shed light upon significant additions to the solution of the problem as formulated and the following statements are an attempt of summarizing these contributions: (a) It is possible to provide Persona-based recommendations for OTT TV, because apparently homogeneous group of users exist. (b) Based

Table 1: Persona profiles. They are described as a set of statements, one for each of the features. The statement about Contents has been formulated looking at those having the highest $Mean_{ZFIPF}$ and the lowest $StDev_{ZFIPF}$ within the cluster. $Statement_{feature:TV-Programs}$, $Statement_{feature:ToD}$, $Statement_{feature:ToW}$, $Statement_{feature:ToM}$, $Statement_{feature:ToY}$

Name	Size	**Persona profile**
Clust. 6	59%	Addicted to News $_{Content}$. Mainly active during the Evening$_{ToD}$, Mostly during the week, never Saturday$_{ToW}$ and slightly more during Summer$_{ToY}$.
Clust. 2	17,6%	News is first, then fitness$_{Content}$. Mainly active during Evening (less at Night and Noon)$_{ToD}$, Mostly during week, never Saturday$_{ToW}$. slightly more far from Pay Check$_{ToM}$, Summer and Autumn$_{ToY}$
Clust. 9	5,9%	Passionate about Sport News, Talent Show and Cars, Interested in Crime, some times cartoons (perhaps for intertaining her children)$_{Content}$. Mainly active during the Evening (never at Night)$_{ToD}$Mostly during the week, some times in the weekend $_{ToW}$, far from Pay Check$_{ToM}$and during Winter$_{ToY}$
Clust. 1	4,5%	Passionate about Cycling and Interested in Football$_{Content}$. Mainly active during Morning and Evening (some times at Night)$_{ToD}$, Far from Paycheck$_{ToM}$and Summer$_{ToY}$.
Clust. 7	4%	Passionate about Cooking and Drama, Interested in News and Crime$_{Content}$. Mainly active during Morning and Evening (never at Noon)$_{ToD}$. Mostly during the week, never Saturday$_{ToW}$, Far from Pay Check$_{ToM}$and Autumn$_{ToY}$
Clust. 8	2,25%	Passionate about Fitnes and Interested in Sport News and News$_{Content}$. Mainly active at Noon, never at Night$_{ToD}$. Mostly Active in Winter, Never during Spring.$_{ToY}$.
Clust. 3, 4, 5, 10, 12, 13	< 1, 8%	Seed Clusters

Algorithm 1: It samples from the dataset and it computes the User Experience matrix (see Eq. 2). Then, after dimensionality reduction using Singular Value Decomposition, it estimates the amount of clusters and it detects the clusters using hierarchical clustering. Finally, for each cluster it returns the measures on the features necessary for providing the Persona profiles.

function DetectPersonaProfiles;
Input : $Dataset$, $MZT = 25$, $SamplingCoefficient = 0.1$, $Mean_{Min} = 0.05$, $StDev_{Max} = 1$, $SimilarityMetric = Chebyshev$, $ClustersEstimationMethod = DavBou$, $ClusteringTech = HierarchicalClustering$;
Output: $Persona(ClusterSize, TOPContentsList, mean(ZFICF_{ToD}), mean(ZFICF_{ToW}), mean(ZFICF_{ToM}), mean(ZFICF_{ToY}))$
MUS = Sampling(MZT, $SamplingCoefficient$, $Dataset$);
Zaps = CollectZaps(Dataset, MUS);
$ZFICF_{ijk}$ = zficf(Zaps);
$ZFIPF_{ip}$ = zfipf(Zaps);
User Experience = $HorizontalConcatenate(ZFICF_{ijk}, ZFIPF_{ip})$;
[U,V,S] = SingularValueDecomposition(User Experience);
ClustersAmount = ClustersEstimationMethod(U, SimilarityMetric);
Clusters = ClusteringTech(U, ClustersAmount, SimilarityMetric);
for *Cluster* ∈ *Clusters* **do**
  return $ClusterSize = \frac{100 * countElement(Cluster)}{countElement(MUS)}$, $TopContentsList = \{Contents \mid mean(ZFIPF_{Content}) > Mean_{Min} \ \& \ StDev(ZFIPF_{Content}) < StDev_{Max}\}$, $mean(ZFICF_{ToD})$, $mean(ZFICF_{ToW})$, $mean(ZFICF_{ToM})$, $mean(ZFICF_{ToY})$;
end

on the ontology represented by the dataset, it is possible to define a model of User Experience that can be collected and updated by the Service Provider in samples (as e.g. the Model Users Sample) which no longer would be linkable to users. Therefore, it would be possible for the Service Provider to collect a privacy-aware Zero-Knowledge Consumption DB. In particular: *i)* We defined a mathematical representation of User Experience suitable for the specific application field, which has been implemented and tested thoroughly with dimensionality reduction and hierarchical clustering.

ii) We showed that, in this application field, Personalization-based on Persona-profiles is possible, homogeneous groups of users exist, and familiar clustering techniques can distinguish between users with different interests even when close to each other, such as those preferring sports Vs. fitness.

iii) We demonstrate that the quality of prediction of a popular Recommender Systems such as FunkSVD, where the user privacy represents a huge issue, is comparable with our predictions, where we potentially can keep the privacy at ZK-level since we need only 10% of the user data constantly stored on the service provider side and this would make extremely difficult linking any user to such data set.

5.1 Future Work

Further work should also be prioritized towards the following focus areas. *i)* **Data sampling**. It is a very critical component of both Recommender System and Privacy enhanced architecture. More sophisticated and effective options, such as the smart sampling presented by Google [29], could be investigated. *ii)* **Recommender System**. We will extend the current work by designing and implementing an effective Persona based Recommender System architecture. Indeed, the recommender system evaluation through the computation of "precision" and "recall", starting from the results presented in this project is quite a straightforward application from the list of top programs of each user on a Target Users Sample and the lists of top programs derived from each Persona profile detected within a MUS. However, the filtering criteria, necessary to improve the recommendation, represents a critical work, the progress of which should be measured looking e.g. at diversity and/or serendipity. *iii)* **Persona Management**. Persona life-cycle [16] in the User Domain differs from the Service Provider domain. For example, from the SP's perspective, Persona may exist and cease to exist at any time. Nonetheless, when they cease to exist they could also *resurrect* to serve another user. Persona may appear as seeds within the working User Experience sample and grow, then mature and become dominant in some context and it could become strategic as solution of cold start problems concerning new users. From the user's perspective, alternative Persona could be used to access the same service in different contexts and always get the best recommendation "Privacy and Ethically Enhanced".

ACKNOWLEDGMENTS

The authors would like to acknowledge the help provided by Hossain Ahmad and Naoufal Medouri.

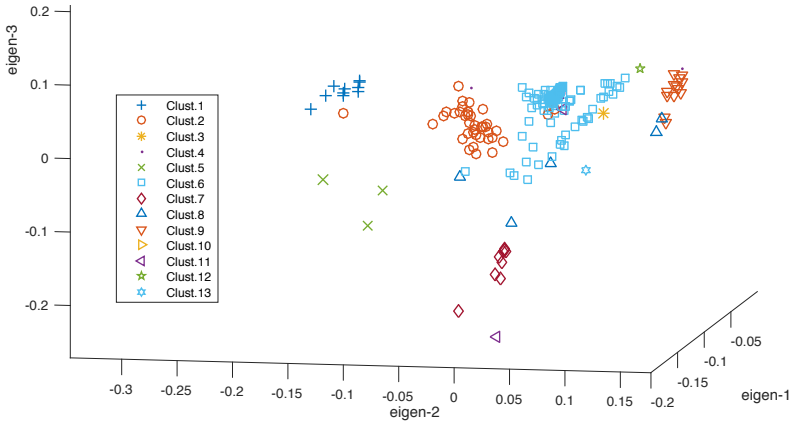

Figure 2: The User Experience computed on a random sample having size 10% of the full dataset. After dimensionality reduction via Singular Value Decomposition, each user has been plotted according to the three main eigenvectors, hierarchical clustering and Chebyshev.

REFERENCES

[1] David Arthur and Sergei Vassilvitskii. 2007. K-Means++: the Advantages of Careful Seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* 8 (2007), 1027–1025. DOI: http://dx.doi.org/10.1145/1283383.1283494

[2] T Caliński and J Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* 3, 1 (1974), 1–27. DOI: http://dx.doi.org/10.1080/03610917408548446

[3] Michal Campr and Karel Ježek. 2015. *Comparing Semantic Models for Evaluating Automatic Document Summarization.* Springer International Publishing, Cham, 252–260. DOI: http://dx.doi.org/10.1007/978-3-319-24033-6_29

[4] V Ciriani, S De Capitani di Vimercati, S Foresti, and P Samarati. 2007. *K-Anonymity.* Springer US. 36 pages. http://spdp.di.unimi.it/papers/k-Anonymity.pdf

[5] Alan Cooper. 2004. *The Inmates Are Running the Asylum.* Sams Publishing. 288 pages. DOI: http://dx.doi.org/10.1007/978-3-322-99786-9{_}1

[6] Alan Cooper, Robert Reimann, and David Cronin. 2007. *About Face 3: The essentials of interaction design.* Vol. 3. Wiley Publishing. 610 pages. DOI: http://dx.doi.org/10.1057/palgrave.ivs.9500066

[7] D L Davies and D W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* 1, 2 (1979), 224–227. DOI: http://dx.doi.org/10.1109/TPAMI.1979.4766909

[8] Michael D. Ekstrand, Michael Ludwig, Joseph A. Konstan, and John T. Riedl. 2011. Rethinking the recommender research ecosystem. *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11* (2011), 133. DOI: http://dx.doi.org/10.1145/2043932.2043958

[9] Ahmed M. Elmisery and Dmitri Botvich. 2011. An agent based middleware for privacy aware recommender systems in IPTV networks. *Smart Innovation, Systems and Technologies* 10 SIST (2011), 821–832. DOI: http://dx.doi.org/10.1007/978-3-642-22194-1_81

[10] European Union. 2016. Regulation 2016/679 of the European Parliament and the Council of the European Union. *Official Journal of the European Union* (2016). Retrieved 2016-12-17 from http://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A32016R0679

[11] ExpressPlay. 2017. Key Storage - ExpressPlay. (2017). Retrieved 2017-04-26 from https://www.expressplay.com/developer/key-storage/

[12] Arik Friedman, Shlomo Berkovsky, and Mohamed Ali Kaafar. 2016. A differential privacy framework for matrix factorization recommender systems. *User Modeling and User-Adapted Interaction* 26, 5 (12 2016), 425–458. DOI: http://dx.doi.org/10.1007/s11257-016-9177-7

[13] Johannes Gehrke, Michael Hay, Edward Lui, and Rafael Pass. 2012. Crowd-Blending Privacy. (2012), 479–496.

[14] Johannes Gehrke, Edward Lui, and Rafael Pass. 2011. Towards privacy for social networks: A zero-knowledge based definition of privacy. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6597 LNCS (2011), 432–449. DOI: http://dx.doi.org/10.1007/978-3-642-19571-6{_}26

[15] Ilsa Godlovitch, Bas Kotterink, J. Scott Marcus, Pieter Nooren, Jop Esmeijer, and Arnold Roosendaal. 2015. *Over-The-Top players (OTTs): Market dynamics and policy challenges.* European Union. 137 pages. DOI: http://dx.doi.org/10.2861/706687

[16] M. Hussain and D. B. Skillicorn. 2011. Mitigating the linkability problem in anonymous reputation management. *Journal of Internet Services and Applications* 2, 1 (2011), 47–65. DOI: http://dx.doi.org/10.1007/s13174-011-0020-4

[17] B. Michael Jones and Dick Hardt. 2012. The OAuth 2.0 Authorization Framework. (2012), 1–76. Retrieved 2016-12-17 from https://tools.ietf.org/html/rfc6750

[18] Dmytro Karamshuk, Nishanth Sastry, Mustafa Al-Bassam, Andrew Secker, and Jigna Chandaria. 2016. Take-away TV: Recharging Work commutes with predictive preloading of catch-Up TV content. *IEEE Journal on Selected Areas in Communications* 34, 8 (2016), 2091–2101. DOI: http://dx.doi.org/10.1109/JSAC.2016.2577298

[19] Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2, 1 (2012), 86–97. DOI: http://dx.doi.org/10.1002/widm.53

[20] Arun Nanda, Andre Durand, Bill Barnes, Carl Ellison, Caspar Bowden, Craig Burton, James Governor, Jamie Lewis, John Shewchuk, Luke Razzell, Marc Canter, Mark Wahl, Mike Jones, Phil Becker, Radovan Janocek, Ravi Pandya, Robert Scoble, and Scott C Lem. 2005. The Laws of Identity. (2005), 13.

[21] Jakob Nielsen and Don Norman. 2015. The Definition of User Experience. (2015). Retrieved 2016-12-17 from http://www.nngroup.com/about-user-experience-definition

[22] Henning Olesen, Josef Noll, and Marlo Hoffman. 2009. User profiles, personalization and privacy. *Outlook, Wireless World Research Forum* 3 (2009), 1–38.

[23] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender systems handbook.* Springer. 1003 pages.

[24] Alistair Sutcliffe. 2009. *Designing for User Engagement: Aesthetic and Attractive User Interfaces.* Vol. 2. Morgan and Claypool. 1–55 pages. DOI: http://dx.doi.org/10.2200/S00210ED1V01Y200910HCI005

[25] David Sward and Gavin Macarthur. 2007. Making User Experience a Business Strategy. *Towards a UX Manifesto* 2, September 2007 (2007), 35–42. DOI: http://dx.doi.org/10.1183/09031936.00022308

[26] The European Commission. 2011. SPECIAL EUROBAROMETER 359 Attitudes on Data Protection and Electronic Identity in the European Union. (2011), 330. Retrieved 2016-12-17 from http://ec.europa.eu/public_opinion/index_en.htm

[27] Julie R Williamson and Stephen Brewster. 2012. A performative perspective on UX. *Communications in Mobile Computing* 1, 1 (2012), 3. DOI: http://dx.doi.org/10.1186/2192-1121-1-3

[28] Jen Ho Yang, Chih Cheng Hsueh, and Chung Hsuan Sun. 2010. An efficient and flexible authentication scheme with user anonymity for Digital Right Management. *Proceedings - 4th International Conference on Genetic and Evolutionary Computing, ICGEC 2010* (2010), 630–633. DOI: http://dx.doi.org/10.1109/ICGEC.2010.161

[29] Celal Ziftci and Ben Greenberg. 2015. GTAC 2015: Statistical Data Sampling. (2015). Retrieved 2016-12-15 from https://www.youtube.com/watch?v=cXi1Jo5V7UM