Applying an Implicit Recommender System in the Preparation of Visits to Cultural Heritage Places

Pedro J. S. Cardoso[1,3(✉)], Pedro Guerreiro[3], Jânio Monteiro[2,3], and João M. F. Rodrigues[1,3]

[1] LARSyS, University of the Algarve, Faro, Portugal
{pcardoso,jrodrig}@ualg.pt
[2] INESC-ID, Lisboa, Portugal
jmmontei@ualg.pt
[3] ISE, University of the Algarve, Faro, Portugal
pmguerre@ualg.pt

Abstract. The visit to cultural places can be an enormous pleasure, where visitors are driven to see and explore historical or contemporary objects. The same journey can also be fastidious when the visitor is taken to lengthy walks, unwanted collections or the objects archive is too large. For many of us, a visit should be something limited in time, showing a variety of things of our own interest. Information systems and in particular mobile devices can play a fundamental role to get us close to that objective by collecting information used to feed recommender systems.

This paper studies the use of an implicit recommender system to propose which points of interest should be explored in cultural heritage places. In its present form, the recommender system, to be supported on data acquired by a mobile application, suggests artworks or artists supported on the user's history. Experimental results are presented, showing the effectiveness of the method.

Keywords: Implicit recommender system · Mobile application
Cultural heritage

1 Introduction

A pleasant journey on cultural heritage places can be influenced by many things. The visited objects and the route taken are two of those factors that impact the users' experience. In this sense, a visit is many times divided in three stages (Falk 2009; Falk and Dierking 2016): the before, the during and the after. The before starts when the user prepares its visit using available information, such as books, dedicated sites, or dedicated mobile applications. Interests are select and registered in order to properly take advantage of the visit. In this phase, the use of novel technology can introduce new challenges, such as, large collections are burdensome to browse, encompassing the peril of overwhelming the users with

M. Antona and C. Stephanidis (Eds.): UAHCI 2018, LNCS 10908, pp. 421–436, 2018.
https://doi.org/10.1007/978-3-319-92052-8_33

information. This overwhelming peril is also a part of the visit itself (during phase) as, on many cases, the collection are too vast to be properly explored during short visits. The after phase is used to get across any doubts raised during the tour, to further investigate the previously seen objects, or simply recall what was seen. As a consequence, the preparation of the visit is a fundamental step to an overall pleasant experience.

A solution to avoid the possibly time consuming task of using the previously stated sources (e.g., books and web sites) is to plan the visit using a recommender system. A recommender system is an information filter system which job is to forecast the user's preferences based on the its historical actions, options, and characterization, and also on the same aspects of other users with similar profiles (Aggarwal 2016; Hu et al. 2008; Jannach et al. 2010; Koren et al. 2009; Rao and Rao 2016; Ricci et al. 2015). In other words, the personalization of cultural heritage information requires a system that is able to model the users (e.g., interests, knowledge, age, context, genre), be aware of contextual aspects, select the most appropriate content, and deliver it in a suitable way (Cardoso et al. 2017). A system like this should also be able to respond to other potentially worth of attention aspects, such as accessibility enhancing and info-exclusion fighting. For instance, the inclusion of features that reflect the visitor's impairments along with its preferences would allow to do a recommendation according with their aspirations.

Depending on the available information and objectives, recommender systems are divided in several classes, e.g.: (a) content based recommenders use (a large number of) features associated to the items/objects, characterizing them, and then the system computes the probability of a user liking a certain item supported on the user's history and items characterizations; and (b) collaborative filtering recommenders where only items and users relations are used to make the predictions, not requiring any information about users or items. The second case uses some kind of rating which associates items and users, being in general divided as explicit or implicit collaborative filtering (Aggarwal 2016; Jannach et al. 2010; Ricci et al. 2015). The explicit case is characterized by the fact that the users provide classifications to the items (e.g., "like", score) which are then used to find users with similar tastes and make the suggestions. In the implicit case, the users do not classify the items, but the information system knows that the user interacted with the item (e.g., bought an item, clicked on a page, saw an artwork), allowing the recommender system to predict which item would be of the users preferences.

This paper introduces the use of an implicit recommender system to propose points of interest to be explored in cultural heritage places. The overall system uses data acquired by a mobile application to feed the recommender system in a non-intrusive way, i.e., suggestions of artworks or artists are supported on the mobile application usage. The recommendation method (adapted from the work of Hu et al. (2008)) uses a preference matrix decomposition in user factors and item factors matrices. The system has a running complexity which grows linearly with the size of the dataset. Some results are presented, supported on a built dataset, showing the effectiveness of the system. The main contribution to the state of the art is the application of the referred kind of recommendation methods in an architecture supported in a mobile system, dynamically applied to the visit of cultural heritage places.

The remaining document is structured as follows. Section 2 presents some preliminaries on Machine Learning with particular interest on recommender systems to cultural heritage places. Section 3 describes the data collecting methodology associated to the mobile application and the adopted recommending system. The last sections presents results from the conducted tests, conclusions and future works.

2 Preliminaries and Problem Formulation

2.1 Recommendation System

Although many times unnoticeable, most of us use recommendation systems on a daily basis. Information Technology (IT) systems use them to expose intelligence, making search engines, social media, e-stores, digital music services, etc. what we expected them to be. In short, recommender system algorithms are a class of information filtering system which have the job to predict the users' preferences, according with given or guessed profiles. Companies like Google, Facebook, Walmart, Amazon, etc., instead of methodically encoding the preferences of all its consumers, apply learning algorithms on their huge datasets and let them prophesy what customers want (Domingos 2015). As a consequence of this global evolution, many party agree that briefly most of the knowledge will be obtained and reside in computers, i.e., as stated by Alpaydin (2016), "data starts to drive the operation; it is not the programmers anymore but the data itself that defines what to do next". Other consequences are the entire industry building itself around these fields, along with emerging research and academic disciplines.

Recommender systems are part of the Machine Learning (ML) research field. ML algorithms, supported in mathematical and computer science techniques, can be seen as engines that use large datasets to find patterns and correlations in order to build models, which will then be applied to new data in order to predict outcomes for it. In this sense, ML algorithms allow to save resources by automatically analyzing data, obtaining an expectable better overview of the available information and making more reasoned decisions. The algorithms are usually divided in some major classes, which include: supervised learning, unsupervised learning, semi-supervised learning, or reinforcement learning (Arulkumaran et al. 2017; Müller and Guido 2016). Supervised learning has the task of inferring models/functions from labeled data. i.e., data that has an input vector and desirable output value. Unsupervised learning is used to draw inferences from datasets of non labeled data. The goal of semi-supervised learning is to employ a large collection of unlabeled data jointly with a few labeled examples to implement a generalization. Finally, reinforcement learning supports agents decisions based on the notion of cumulative rewards.

In general, the ML workflow is divided in four large steps: (a) get (*enough*) data – collect data related to the problem; (b) clean, prepare, and manipulate data – converting the data into a form that computers can operate on (e.g., convert things to numerical data and categorize data); (c) define and train the selected model using test and validation data – depending on the type of problem being solved (e.g., regression, classification, clustering, or recommendations) build a mathematical model of the data; and (d) predict outcomes over new and unseen data – apply the trained model to unseen data to, depending on the problem at hand, predict values, classify or associate the data, recommend other objects/items, etc.

Recommender systems, of particular interest in this work, suffered from a sudden popularity after the Netflix Prize contest (Lohr 2009; Gomez-Uribe and Hunt 2015; Ricci et al. 2015), and are being divided in several categories (Jannach et al. 2010; Rao and Rao 2016), such as: (a) content based or (b) collaborative filtering. In the first case, features are associated to the items characterizing them. E.g., in the Pandora.com recommender system, songs were manually annotated by musicians with up to several hundred features such as instrumentation, influences, or instruments. Afterwards, the question is treated as a binary classification problem, where the probability of a user to like a certain song depends on its listening history and the songs previous characterizations. Items with greater probability of being liked are then suggested to the user. On the other hand, (b) collaborative filtering is supported only in the items and users relations, not requiring any information about the users or the items (Koren et al. 2009; Hu et al. 2008). In this case, some kind of rating is associated to the interaction between the items and the users, being in general divided as explicit or implicit. In the explicit case, users provide (or can be inferred) a classification to the items (e.g., a like indication, a score, or the number of times the item is "used" acts as a sign of affinity), and these values are used to suggest (or not) similar items to similar users. In the implicit case, the user does not classify the items, but somehow it is known the user's interactions with them. Cases of interaction are for instance buying the item or simply navigating to the item's web page. In this last case, navigating to the page represents a sign of preference but the opposite, not accessing the page, should not be seen as a dislike since many other factor can be the motive (e.g., lack of time, unaware of its existence).

2.2 Applications of Recommender Systems to Cultural Heritage Places

The user's preferences are being studied for some time in the preparation and visit of cultural heritage places. For instance, a routing system implementing a mobile museum tour guide, providing personalized tours tailored to the user's interests and position inside the museum, is offered by the Rijksmuseum Amsterdam (van Hage et al. 2010). The system includes tools for the interactive discovery of user's interests, semantic recommendations of artworks and art-related topics, and the (semi-)automatic generation of personalized museum tours. Benouaret and Lenne (2015), supported on the users' preferences and

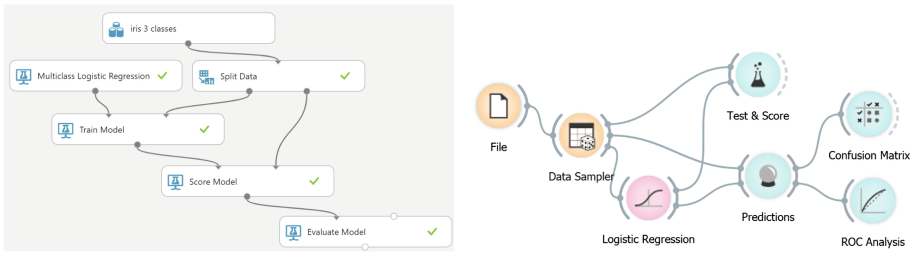


Fig. 1. ML GUI tools: Azure ML Studio (left) and Orange ML (right).

contexts, proposed a recommender system for mobile devices to build tours according to their preferences and constraints. Gavalas et al. (2014) presented a state-of-the-art in the field, proposing a classification of mobile tourism recommender systems and providing insights on their offered services. The CHESS (2017) project researches, implements and evaluates both the experiencing of personalized interactive stories for visitors of cultural sites and their authoring by the cultural content experts. Cardoso et al. (2017) proposed the combination association rules with the location of the user to design some modes to proceed with the visit, such as dynamic mode (the visit is constantly updated supported on the visitors actions), over-a-path mode (recommender can only suggest objects in a predetermined walk), free mode (recommender can suggest objects in any place of the museum), nearby mode (recommender uses rules to suggest objects which are near the present location), static mode (prepared visited) or the Surprise-Me mode (the visit is disclosed as the user walks through the museum). The Route Perfect (2017) application allows to easily plan a trip based on the traveler's preferences, budget and personal style. Several other works can be found in literature such as (Garcia et al. 2011; Verbert et al. 2012; Wang and Xiang 2012).

2.3 ML Tools

Nowadays, a ML user/programmer does not have to implement a large number of ML methods as they are included in several visual and programmable frameworks. For instance, Microsoft released the Azure ML Studio, defined as a powerfully simple browser-based, visual drag-and-drop authoring environment where no coding is necessary (Microsoft 2017). The studio includes hundreds of built-in packages and support for custom code, namely in R and Python. Furthermore, the Azure ML Studio can be used to deploy the created models into production as web service, callable from any device with any data source. The Orange is an open-source data visualization, machine learning and data mining toolkit (Orange 2017). With a visual programming front-end Orange allows exploratory data analysis and interactive data visualization, and can also be used as a Python library. Figure 1 shows an example for each of the applications. Other graphical user interfaces (GUI) ML tools exist such as WEKA (2017), MLJAR (2017) or Knime (2017).

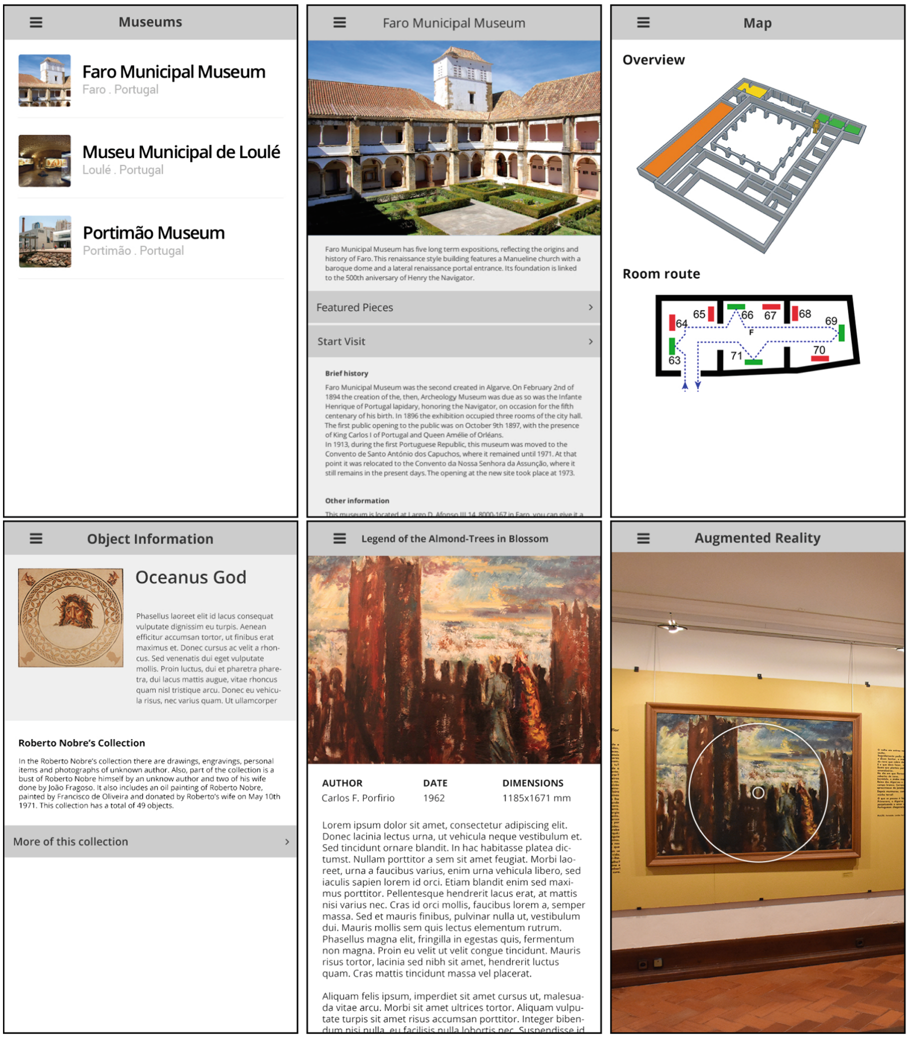
Many non visual libraries are also available such as TensorFlow (2018) – an open source software library for numerical computation using data flow graphs; VELES (2018) – a distributed platform, which provides machine learning and data processing services for a user; MLlib (Meng et al. 2016) – Spark's open-source distributed machine learning library; scikit-learn (Pedregosa et al. 2011) – provides tools for data mining and data analysis; Shogun-Toolbox (Sonnenburg et al. 2017) – open-source machine learning library that offers a wide range of efficient and unified machine learning methods; Torch (Collobert et al. 2011) – a scientific computing framework with wide support for machine learning algorithms using Graphics Processing Units (GPUs); or the MLPACK (Curtin et al. 2013) – a scalable machine learning library, written in C++, that aims to provide fast, extensible implementations of machine learning algorithms.

3 Implementation of the Recommender System

3.1 Data Collecting Process and Dataset

This work is part of the M5SAR (Mobile Five Senses Augmented Reality System for Museums) project. The M5SAR project aims at the development of an augmented reality system, which consists of a mobile application and a "gadget", to be incorporated in the mobile devices, in order to explore the 5 human senses (Rodrigues et al. 2017; Sardo et al. 2018). The system will constitute a guide in cultural, historical and museum events, complementing or replacing the traditional orientations given by tour guides, directional signs, or maps. Figure 2 shows some screens of the interface being developed: the initial screen (top-left) presents the list of available museums. After selecting the wanted museum, the user is taken to the museum's initial page (top-center) from where we can navigate to other pages such as maps (top-right), information about pieces (bottom-left and bottom center) or to the augmented reality page (bottom-right). The design and implementation of the application, besides the recommender system, is out of this paper's scope, but more details can be found in (Rodrigues et al. 2017).

In our case, the navigation through the mobile application pages can be seen as of interest by the object/item that can be collected to empower the recommender system. Given the stored and collected data, the recommender system can be implemented using a content based approach (as the majority of the object are characterized) or a collaborative filtering approach (as the interaction between the users and the items can be stored). In the later case, both implicit and explicit approaches are also available. The implicit approach is probably more suitable as it can be powered by simply storing the interactions with the items (e.g., information about pieces seen in the mobile app or the activation of the augmented reality over an item). Nevertheless, the explicit approach can be thought/implemented using for instance the time spent by the user in the augmented reality features associated to items/object or the time it takes to read an information page in the mobile app, i.e., a long time scrolling through the page probably means he is interested in the item, so it should be classified high, while a short time probably mean that the item is not of its interests.


Museums

Faro Municipal Museum
Faro . Portugal

Museu Municipal de Loulé
Loulé . Portugal

Portimão Museum
Portimão . Portugal

Faro Municipal Museum

Faro Municipal Museum has five long term expositions, reflecting the origins and history of Faro. This renaissance style building features a Manueline church with a baroque dome and a lateral renaissance portal entrance. Its foundation is linked to the 500th aniversary of Henry the Navigator.

Featured Pieces ›

Start Visit ›

Brief history

Faro Municipal Museum was the second created in Algarve. On February 2nd of 1894 the creation of the, then, Archeology Museum was due as so was the Infante Henrique of Portugal lapidary, honoring the Navigator, on occasion for the fifth centenary of his birth. In 1896 the exhibition occupied three rooms of the city hall. The first public opening to the public was on October 9th 1897, with the presence of King Carlos I of Portugal and Queen Amélie of Orléans.
In 1913, during the first Portuguese Republic, this museum was moved to the Convento de Santo António dos Capuchos, where it remained until 1971. At that point it was relocated to the Convento da Nossa Senhora da Assunção, where it still remains in the present days. The opening at the new site took place at 1973.

Other information

This museum is located at Largo D. Afonso III 14, 8000-167 in Faro, you can give it a

Map

Overview

Room route

Object Information

Oceanus God

Phasellus laoreet elit id lacus consequat vulputate dignissim eu turpis. Aenean efficitur accumsan tortor, ut finibus erat maximus et. Donec cursus ac velit a rhoncus. Sed venenatis dui eget vulputate mollis. Proin luctus, dui et pharetra pharetra, dui lacus mattis augue, vitae rhoncus quam nisl tristique arcu. Donec eu vehicula risus, nec varius quam. Ut ullamcorper

Roberto Nobre's Collection

In the Roberto Nobre's collection there are drawings, engravings, personal items and photographs of unknown author. Also, part of the collection is a bust of Roberto Nobre himself by an unknow author and two of his wife done by João Fragoso. It also includes an oil painting of Roberto Nobre, painted by Francisco de Oliveira and donated by Roberto's wife on May 10th 1971. This collection has a total of 49 objects.

More of this collection ›

Legend of the Almond-Trees in Blossom

AUTHOR	DATE	DIMENSIONS
Carlos F. Porfirio	1962	1185x1671 mm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec lacinia lectus urna, ut vehicula neque vestibulum et. Sed tincidunt ornare blandit. In hac habitasse platea dictumst. Nullam porttitor a sem sit amet feugiat. Morbi laoreet, urna a faucibus varius, enim urna vehicula libero, sed iaculis sapien lorem id orci. Etiam blandit enim sed maximus porttitor. Pellentesque hendrerit lacus erat, at mattis nisi varius nec. Cras id orci mollis, faucibus lorem a, semper massa. Sed et mauris finibus, pulvinar nulla ut, vestibulum dui. Mauris mollis sem quis lectus elementum rutrum. Phasellus magna elit, fringilla in egestas quis, fermentum non magna. Proin eu velit ut velit congue tincidunt. Mauris risus tortor, lacinia sed nibh sit amet, hendrerit luctus quam. Cras mattis tincidunt massa vel placerat.

Aliquam felis ipsum, imperdiet sit amet cursus ut, malesuada vitae arcu. Morbi sit amet ultrices tortor. Aliquam vulputate turpis sit amet risus accumsan porttitor. Integer bibendum nisi nulla, eu facilisis nulla lobortis nec. Suspendisse id

Augmented Reality

Fig. 2. Top to bottom, left to right: list of available museums, specific museum information, museum map with a route calculated, example of a card piece, image recognition (AR), and information about the detected piece.

The M5SAR's recommender system collected data is stored in a relational database, with a partial sketch presented in the Enhanced Entity–Relationship (EER) model shown in Fig. 3. Some characteristics of the stored date include: basic user information is stored in the `app` (user) table; An `artwork` can be seen by many `apps`; An `app` can see many `artworks`; An `artwork` can have many `artists` authoring it, as they can author many `artworks`. The proposed structure allow for proper SQL queries to the database to return the pieces and artists seen by the users (apps).

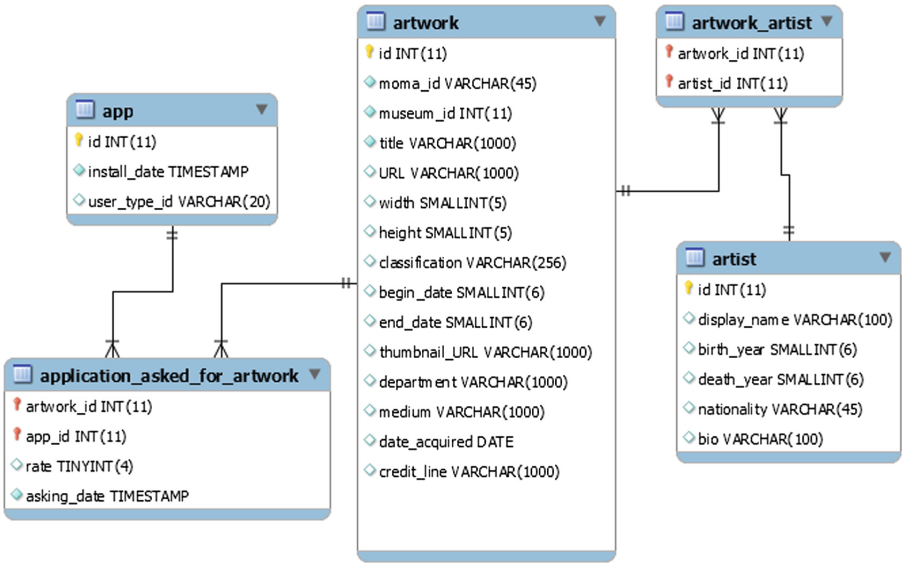


Fig. 3. EER model for the recommender system database.

Since the data collected so far is not large enough to properly train the recommender system, we decided to populate the database using the Museum of Modern Art dataset (MoMA) (Robot 2016) and the retail market basket dataset provided in (Brijs et al. 1999). Each row of the retail market basket dataset was considered as a user and an one-to-one correspondence was made between the items in the basket and the artworks. On the other hand, the MoMA's data was used to populate the artworks and artists tables. A total of 89952 distinct baskets/apps/users were considered, corresponding to the interaction with 16470 distinct artworks and 16147 artists.

3.2 Estimating Preferences

The implemented recommender system adapts the work by Hu et al. (2008). In general, let us assume that $u, v, \ldots$ represent users, $i, j, \ldots$ are items, r_{ui} represents the input data associating user u to item i, m is the number of users, and n is the number of items. For example, in our case, r_{ui} represents the number of times user u saw artwork i or (in alternative) the number of times the same user u saw artworks from artist i. As already mentioned, other rating could also be thought of, as for instance the time the user took opened an artwork object page in the mobile application (e.g., Fig. 2 bottom-left) or the percentage of the artwork's/artist's page scrolled. If there is no action between user u and item i then r_{ui} will be set to 0, with a "cloudy" meaning (e.g., the user does not like the item, might be unaware of the existence of the item, or unable to see it due to some reason). On the other hand, the fact that a user interacts with some item does not necessarily mean the user likes it, for instance, the interaction could be unintentional or someone else might be using the user's mobile device.

Returning to the formulation of the model, $R = [r_{ui}]$ can be seen as a matrix, where each row represents the preferences of a user and each column the preferences/interactions over an item. Most likely, R is a sparse matrix as most users did not interact with the majority of the items.

The proposed model, induced by singular value decomposition of the user/item matrix, associates to each user u a user-factor vector x_u and to each item i a item-factor vector y_i, both belonging to $\mathbb{R}^{\mathrm{nf}}$ where nf is the number of latent factors. The inner product between x_u and y_i, $\hat{r}_{ui} = x_u^T y_i$, predicts the user u preferences associated to item i. In this sense, two (dense) matrices can be though: matrix $U = [x_u] \in \mathbb{R}^{\mathrm{m}\times\mathrm{nf}}$ where each row is a vector associated to a users and $V = [y_i]^T \in \mathbb{R}^{\mathrm{nf}\times\mathrm{n}}$ where each column is a vector associated to an item. Multiplying the two feature matrices should approximate the original matrix, i.e., $R \approx U \times V$. The goal is to find, for each user and each item, vectors x_u and y_i, i.e., find the matrices U and V that best approximate $R \approx U \times V$.

To compute U and V, Hu et al. (2008) defined the minimization problem

$$\min_{x_*, y_*} \sum_{u,i} c_{ui} \left(p_{ui} - x_u^T y_i\right)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right), \tag{1}$$

where p_{ui} is equal to 1 if $r_{ui} > 0$ and 0 otherwise, $c_{ui} = 1 + \alpha r_{ui}$, and α, λ are algorithm parameters. Variable p_{ui} indicates user's u estimated preference relative to item i, where the term "estimated" translates the "cloudy" meaning of the r_{ui} values. In this sense, c_{ui} was introduce as a measure of confidence relative to the observation of the relation between u and i. Equation (1) last part, $\lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2\right)$, regularizes the model, avoiding its over-fitting.

The minimization problem is affected by the fact that the number of users (m) and items (n) are in general large, easily reaching millions of non-linear parcels to be taken into account, and probably making more traditional optimization techniques suffer from low performance. To tackle this situation, Hu et al. (2008) observed that if either the user-factors or item-factors were constant, the objective function becomes quadratic and therefore "swiftly" globally optimizable. Therefore, the author proposed an alternating least squares optimization process (Zhou et al. 2008), where the iterations take turns between computing user-factors and item-factors, guarantying the lowering of the cost function in each step. Important is the fact that the process can be run in parallel, solving one feature vector at a time. Further details can be found in the original work from Hu et al. (2008), namely the running computational complexity which is is linear with the size of the input.

4 Tests and Validation

Traditional machine learning algorithms test and validate the trained models using a set of data previously unseen to them. Many times, this is done by randomly splitting the initial dataset into a training set and a test/validation set. In the collaborative filtering case this is not possible, since each user must be present in the final user-factor matrix in order to make preferences predictions. A solution is to mask a percentage of the user/item entries from the model during the training phase and then, during the testing phase, verify which items initially masked were suggest. Other testing and validation solutions might be thought on other scenarios, such as dividing the observations in training and validation by periods of time, if the users and item space does not change. We adopted the first solution, mask some user/item entries, by setting $r_{ui} = 0$ with probability p_{mask} where $r_{ui} > 0$. The result was a masked observation matrix R', i.e., $R' = [r'_{ui}]$ where

$$r'_{ui} = \begin{cases} 0 & \text{if } r_{ui} = 0 \text{ or } rand() < p_{mask} \\ r_{ui} & \text{otherwise} \end{cases},$$

and $rand() \in [0, 1]$ is a randomly generated value. The optimization process described in Sect. 3.2 was then applied to R' in order to compute the users (U') and items (V') factors matrices. Given that the product of line u of U' with column i of V' should return an approximation to r'_{ui}, the computation of k recommendations for user corresponding to line u, is done by multiplying line u of U' by V' and returning the indexes of the k largest values in the resulting vector.

Table 1 presents an example of a recommendation given by the method for a certain user. The table contains 3 columns: *Observed* – presenting the "real" observation associated with the user; *Masked* – the observations used to train the model, i.e., after randomly masking/hiding observations; and *Recommended* – the list of suggestions returned by the algorithm. In the example, 3 artists were hidden in the mask phase (Bernard Tschumi, Giovanni Guerrini, and Mario Romano). From those hidden artists, two were recommended by the method (Giovanni Guerrini and Mario Romano) and one belonging to the original data (Bernard Tschumi) was not recommended. In this case 10 suggestions were taken from the method and therefore there are 8 other artists which were not in the original data (*Observed*).

A systematic set of tests was made by varying parameters according with the values in Table 2. Furthermore, the tests were supported on 100.000 observations from 17827 users/apps, corresponding to 994 artworks and 263 artists. The sparsity of the observation matrices R (i.e., the observation matrix before applying the mask) was of 99.5% for the user vs. artwork and 98.3% for the user vs. artist.

Considering the probability of masking observations equal to 10% ($p_{mask} = 0.1$), Table 3 shows the 10 best recommender results, ordered by the mean (μ_{sug}) (standard deviation, σ, in parentheses) of correctly suggested artworks ratio, between the masked ones, when fixing the number of items to recommend to $n_p = 10$ (Table 3(a)) and $n_p = 30$ (Table 3(b)). Suggesting 10 artworks, the method correctly suggests approximately 40% of the masked ones, rising to approximately 50% when 30 artworks were suggested. About the parameters,

Table 1. Example of recommendations made to an user: column *Observed* lists the artist seen by the app ("user"), column *Masked* shows the set of artists after applying the mask (training phase), and column *Recommended* the artists that are suggested by the recommender system.

Observed	Masked	Recommended
Bernard Tschumi	–	–
Otto Schönthal	Otto Schönthal	–
Alessandro Mendini	Alessandro Mendini	–
Ernesto Bruno La Padula	Ernesto Bruno La Padula	–
Giovanni Guerrini	–	Giovanni Guerrini
Mario Romano	–	Mario Romano
Richard Meier	Richard Meier	–
–	–	Fumihiko Maki
–	–	Frank O. Gehry
–	–	Robert A. M. Stern
–	–	Alison Sky
–	–	Michelle Stone
–	–	Toyo Ito
–	–	Massimo Vignelli
–	–	David Jacob

Table 2. Parameter variation

Parameter	Observation
$\alpha \in \{10^3, 25000, 50000, 10^5\}$	Equation (1)
$\lambda \in \{0.01, 0.1\}$	Model regularization, Eq. (1)
$nf \in \{100, 200, 400\}$	Number of latent factors, Sect. 3.2
$N_{it} \in \{50, 200\}$	Number of iterations (alternating least squares), Sect. 3.2
$N_p \in \{10, 30\}$	Number of preferences returned
$p_{mask} \in \{0.1, 0.25\}$	Probability of masking an element

although not completely conclusive from the presented results, a good set of values seems to be $\alpha = 25000$, $nf = 400$, and $N_{it} \in \{50, 200\}$. When the probability of masking is increased to 25% ($p_{mask} = 0.25$) the percentages of correctly suggested artwork decreases around 2% (Table 4). About the parameter values, $\alpha = 25000$, $nf = 400$, and $N_{it} = 50$ also seem a good set, from the ones tested.

Similar tests were made considering artist instead of artworks. Tables 5 and 6 summarizes the 10 best solutions (ordered by the mean of correctly suggested ratio) considering $p_{mask} = 0.1$ and $p_{mask} = 0.25$, respectively. In this case, the mean of the correct suggested ratio was slightly higher with values around

Table 3. Ordered by mean value of correctly suggested artworks, the best 10 recommender results considering $p_{mask} = 0.1$.

(a) $n_p = 10$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	200	50	0.10	0.397 (0.5)
25000	400	50	0.01	0.397 (0.5)
1000	100	50	0.01	0.396 (0.5)
25000	400	50	0.10	0.396 (0.5)
25000	200	200	0.01	0.394 (0.5)
25000	400	200	0.01	0.391 (0.5)
25000	200	50	0.01	0.391 (0.5)
25000	200	200	0.10	0.390 (0.5)
25000	400	200	0.10	0.389 (0.5)
50000	400	200	0.01	0.389 (0.5)

(b) $n_p = 30$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	400	50	0.10	0.479 (0.5)
25000	400	50	0.01	0.478 (0.5)
25000	200	50	0.01	0.478 (0.5)
25000	200	50	0.10	0.477 (0.5)
50000	400	200	0.10	0.471 (0.5)
100000	400	200	0.01	0.470 (0.5)
50000	400	200	0.01	0.470 (0.5)
25000	400	200	0.01	0.470 (0.5)
25000	400	200	0.10	0.470 (0.5)
50000	200	200	0.10	0.469 (0.5)

Table 4. Ordered by mean value of correctly suggested artworks, the best 10 recommender results considering $p_{mask} = 0.25$.

(a) $n_p = 10$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	400	50	0.10	0.375 (0.4)
25000	400	50	0.01	0.373 (0.4)
25000	200	50	0.10	0.372 (0.4)
25000	200	50	0.01	0.371 (0.4)
25000	200	200	0.01	0.370 (0.4)
25000	200	200	0.10	0.366 (0.4)
50000	400	50	0.01	0.364 (0.4)
25000	400	200	0.10	0.364 (0.4)
50000	400	50	0.10	0.363 (0.4)
25000	100	50	0.10	0.362 (0.4)

(b) $n_p = 30$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	400	50	0.01	0.454 (0.4)
25000	200	50	0.10	0.45 (0.4)
25000	400	50	0.10	0.45 (0.4)
25000	200	50	0.01	0.446 (0.4)
25000	100	50	0.01	0.444 (0.4)
50000	400	50	0.10	0.440 (0.4)
25000	100	50	0.10	0.440 (0.4)
50000	400	50	0.01	0.437 (0.4)
50000	200	200	0.01	0.433 (0.4)
50000	200	50	0.10	0.433 (0.4)

65% and 80% when approximately 10% of the observations were masked and the number of suggested artists were $n_p = 10$ and $n_p = 30$, respectively. When the probability of masking observation rises to 25%, as expected, as lower information is given, the referred mean value decreases to 58% when $n_p = 10$ and 74% when $n_p = 30$. About the parameters, considering the tested ones, again $\alpha = 25000$ and $nf = 400$ seem good choices. About the number of iteration, $N_{it} = 200$ seems a reasonable choice as it ensures some of the best solution in all the cases.

We should observe that, the higher percentage of correctly suggested items associated with the artist (compared to artworks) seems to be related with the reduction of items space (994 artworks vs. 263 artists). Another question is the standard deviation value which is high. In this case, we should observe that

Table 5. Ordered by mean value of correctly suggested artists, the best 10 recommender results considering $p_{mask} = 0.1$.

(a) $n_p = 10$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	400	200	0.10	0.643 (0.4)
25000	200	200	0.01	0.634 (0.5)
25000	400	200	0.01	0.632 (0.5)
25000	200	200	0.10	0.632 (0.5)
50000	400	200	0.01	0.624 (0.5)
50000	400	200	0.10	0.623 (0.5)
50000	200	200	0.10	0.618 (0.5)
25000	100	200	0.01	0.616 (0.5)
25000	100	200	0.10	0.614 (0.5)
50000	200	200	0.01	0.608 (0.5)

(b) $n_p = 30$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	100	50	0.01	0.803 (0.4)
25000	200	50	0.10	0.798 (0.4)
25000	400	50	0.10	0.797 (0.4)
25000	400	50	0.01	0.797 (0.4)
25000	200	50	0.01	0.796 (0.4)
25000	100	50	0.10	0.795 (0.4)
50000	400	200	0.01	0.788 (0.4)
50000	400	50	0.10	0.784 (0.4)
100000	200	200	0.01	0.784 (0.4)
50000	400	200	0.10	0.784 (0.4)

Table 6. Ordered by mean value of correctly suggested artists, the best 10 recommender results considering $p_{mask} = 0.25$.

(a) $n_p = 10$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	400	200	0.01	0.575 (0.4)
25000	400	200	0.10	0.569 (0.4)
50000	400	200	0.01	0.563 (0.4)
25000	200	200	0.10	0.554 (0.4)
25000	200	200	0.01	0.553 (0.4)
50000	400	200	0.10	0.548 (0.4)
50000	100	200	0.01	0.535 (0.4)
25000	400	50	0.01	0.533 (0.4)
100000	400	200	0.01	0.533 (0.4)
25000	100	200	0.01	0.531 (0.4)

(b) $n_p = 30$

α	nf	N_{it}	λ	$\mu_{sug}(\sigma)$
25000	400	50	0.01	0.737 (0.4)
25000	100	50	0.01	0.736 (0.4)
25000	400	50	0.10	0.732 (0.4)
25000	200	50	0.01	0.723 (0.4)
25000	100	50	0.10	0.723 (0.4)
50000	400	50	0.01	0.719 (0.4)
25000	200	50	0.10	0.715 (0.4)
50000	400	200	0.01	0.711 (0.4)
50000	100	50	0.01	0.707 (0.4)
50000	100	200	0.01	0.703 (0.4)

each user has associated a relatively low number of observations, which results in a great difference in the ratio of correctly suggested items value, e.g., if for a certain user only 1 observation was masked then suggesting it would give a correct suggestion ratio of 100% while not suggesting it would give a 0% correct suggestion ratio.

5 Conclusions

The way people experience cultural heritage is taking advantage of the latest technologies to improve the users experiences. Traditional visits, where everyone, despite their interests or limitations, have to follow a predetermined route are being replaced by a more interactive and dynamic approach. Furthermore, the overwhelming of the visitors is a risk, as many times the number of items is very large, the time available to explore them is limited, or they are not of the user's interest.

This paper propose the use of a recommender system to help in the preparation and following visit of cultural heritage places. The recommender empowers the use of a mobile application, by suggesting which items (e.g., museum objects, interesting buildings, or buyable objects) the user should take into consideration, according with its profile and historical actions. The used method belongs to the class of implicit collaborative filtering, because users do not classify the items, i.e., the system is only aware of the interaction between the users and the items. The data acquisition is to be supported in the use of a mobile application, part of the M5SAR project.

At the moment of writing, the recommender system runs on a server, although efforts are being made to transpose it to mobile devices. Also in study is the use of a mixed (implicit/explicit) collaborative filtering asking the users to score the artwork and/or observing their interest in some of the artworks (e.g., by measuring the time they spend observing the items). Another objective is to integrate the system with a route optimization method, capable of optimizing the walks through cultural heritage places supported in several objectives (e.g., users' preferences, items diversity, walked distance) and constraints (e.g., related with the users mobility limitation).

Acknowledgments. This work was supported by the Portuguese Foundation for Science and Technology (FCT), project LARSyS (UID/EEA/50009/2013), CIAC, and project M5SAR I&DT n. 3322 financed by CRESC ALGARVE2020, PORTUGAL2020 and FEDER. We also thank Faro Municipal Museum and the M5SAR project leader, SPIC - Creative Solutions [www.spic.pt].

References

Aggarwal, C.C.: Recommender Systems. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29659-3

Alpaydin, E.: Machine Learning: The New AI. The MIT press, Cambridge (2016)

Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: a brief survey. IEEE Sig. Process. Mag. **34**(6), 26–38 (2017)

Benouaret, I., Lenne, D.: Combining semantic and collaborative recommendations to generate personalized museum tours. In: Morzy, T., Valduriez, P., Bellatreche, L. (eds.) ADBIS 2015. CCIS, vol. 539, pp. 477–487. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23201-0_48

Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Using association rules for product assortment decisions: a case study. In: Knowledge Discovery and Data Mining, pp. 254–260 (1999)

Cardoso, P.J.S., Rodrigues, J.M.F., Pereira, J.A.R., Sardo, J.D.P.: An object visit recommender supported in multiple visitors and museums. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2017. LNCS, vol. 10277, pp. 301–312. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58706-6_24

CHESS: CHESS - cultural heritage experiences through socio-personal interactions and storytelling (2017). http://www.chessexperience.eu/

Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: A matlab-like environment for machine learning. In: BigLearn, NIPS Workshop (2011)
Curtin, R.R., Cline, J.R., Slagle, N.P., March, W.B., Ram, P., Mehta, N.A., Gray, A.G.: MLPACK: a scalable C++ machine learning library. J. Mach. Learn. Res. **14**, 801–805 (2013)
Domingos, P.: The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. Basic Books, New York (2015)
Falk, J.H.: Identity and the Museum Visitor Experience. Left Coast Press Inc, Walnut Creek (2009)
Falk, J.H., Dierking, L.D.: The Museum Experience Revisited. Left Coast Press Inc, Walnut Creek (2016)
Garcia, I., Sebastia, L., Onaindia, E.: On the design of individual and group recommender systems for tourism. Expert Syst. Appl. **38**(6), 7683–7692 (2011)
Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G.: Mobile recommender systems in tourism. J. Netw. Comput. Appl. **39**, 319–333 (2014)
Gomez-Uribe, C.A., Hunt, N.: The netflix recommender system: algorithms, business value, and innovation. ACM Trans. Manage. Inf. Syst. **6**(4), 13:1–13:19 (2015)
Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 263–272. IEEE (2008)
Jannach, D., Felfernig, A., Zanker, M., Friedrich, G.: Recommender Systems. Cambridge University Pr (2010)
Knime: Knime (2017). https://www.knime.com/. Accessed 14th Dec 2017
Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **8**, 30–37 (2009)
Lohr, S. (2009). Netflix awards $1 million prize and starts a new contest. New York Times, 21
Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M.J., Zadeh, R., Zaharia, M., Talwalkar, A.: MLlib: machine learning in apache spark. J. Mach. Learn. Res. **17**(1), 1235–1241 (2016)
Microsoft: Azure ML Studio (2017). https://studio.azureml.net. Accessed 14th Dec 2017
MLJAR: MLJAR (2017). https://mljar.com/. Accessed 14th Dec 2017
Müller, A.C., Guido, S.: Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, Sebastopol (2016)
Orange: Orange (2017). https://orange.biolab.si/. Accessed 14th Dec 2017
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
Rao, R., Rao, M.: A survey on recommender system. Int. J. Comput. Sci. Inf. Secur. **14**(5), 265–271 (2016)
Ricci, F., Rokach, L., Shapira, B. (eds.): Recommender Systems Handbook. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6
Robot, O.D.: Moma collection - automatic monthly update (2016)
Rodrigues, J.M.F., Pereira, J.A.R., Sardo, J.D.P., de Freitas, M.A.G., Cardoso, P.J.S., Gomes, M., Bica, P.: Adaptive card design UI implementation for an augmented reality museum application. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2017. LNCS, vol. 10277, pp. 433–443. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58706-6_35

Route Perfect: Route perfect (2017). https://www.routeperfect.com/. Accessed 29th June 2016

Sardo, J.D.P., Semião, J., Monteiro, J.M., Pereira, J.A.R., de Freitas, M.A.G., Esteves, E., Rodrigues, J.M.F.: Portable device for touch, taste and smell sensations in augmented reality experiences. In: Mortal, A., Aníbal, J., Monteiro, J., Sequeira, C., Semião, J., Moreira da Silva, M., Oliveira, M. (eds.) INCREaSE 2017, pp. 305–320. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-70272-8_26

Sonnenburg, S., Strathmann, H., Lisitsyn, S., Gal, V., García, F.J.I., Lin, W., De, S., Zhang, C., Frx, T., Andreev, E., JonasBehr, S., Mazumdar, P., Widmer, C., Zora, P.D., Toni, G.D., Mahindre, S., Kislay, A., Hughes, K., Votyakov, R., Khalednasr, Sharma, S., Novik, A., Panda, A., Anagnostopoulos, E., Pang, L., Binder, A., Serialhex, Esser, B.: Shogun-toolbox/shogun: Shogun 6.1.0 (2017)

TensorFlow: TensorFlow (2018). https://www.tensorflow.org. Accessed 14th Jan 2018

van Hage, W.R., Stash, N., Wang, Y., Aroyo, L.: Finding your way through the Rijksmuseum with an adaptive mobile museum guide. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 46–59. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13486-9_4

VELES: VELES (2018). https://velesnet.ml. Accessed 14th Jan 2018

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., Duval, E.: Context-aware recommender systems for learning: a survey and future challenges. IEEE Trans. Learn. Technol. **5**(4), 318–335 (2012)

Wang, D., Xiang, Z.: The new landscape of travel: a comprehensive analysis of smartphone apps. In: Fuchs, M., Ricci, F., Cantoni, L. (eds.) Information and Communication Technologies in Tourism 2012. Springer, Vienna (2012). https://doi.org/10.1007/978-3-7091-1142-0_27

WEKA: WEKA: Waikato Environment for Knowledge Analysis (2017).https://www.cs.waikato.ac.nz/ml/weka/. Accessed 14th Dec 2017

Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-scale parallel collaborative filtering for the netflix prize. In: Fleischer, R., Xu, J. (eds.) AAIM 2008. LNCS, vol. 5034, pp. 337–348. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68880-8_32