Recommending Multimedia Objects in Cultural Heritage Applications

Ilaria Bartolini[1], Vincenzo Moscato[2], Ruggero G. Pensa[3], Antonio Penta[3],
Antonio Picariello[2], Carlo Sansone[2], and Maria Luisa Sapino[3]

[1] University of Bologna, DISI
i.bartolini@unibo.it
[2] University of Naples "Federico II", DIETI
{vmoscato,antonio.picariello,carlo.sansone}@unina.it
[3] University of Torino, DI
{pensa,penta,mlsapino}@di.unito.it

Abstract. Italy's Cultural Heritage is the world's most diverse and rich patrimony and attracts millions of visitors every year to monuments, archaeological sites and museums. The valorization of cultural heritage represents nowadays one of the most important research challenges in the Italian scenario. In this paper, we present a general multimedia recommender system able to uniformly manage heterogeneous multimedia data and to provide context-aware recommendation techniques supporting intelligent multimedia services for the users. A specific application of our system within the cultural heritage domain is proposed by means of a real case study in the mobile environment related to an outdoor scenario, together with preliminary results on user's satisfaction.

1 Introduction

Italy's Cultural Heritage represents a worldwide resource of inestimable value, attracting millions of visitors every year to monuments, archaeological sites and museums. One of the most challenging and interesting research problems within such a scenario is surely the valorization of such heritage. Indeed, it should be important to provide a cultural environment with functionalities to represent the related knowledge derived from current digital sources describing cultural heritage, such as text descriptions, pictures, and videos, in order to allow a tourist visiting a site to enjoy multimedia stories in real time so as to enrich his/her cultural experience.

Our goal is to "extend" classical recommendation techniques (*content-based*, *collaborative filtering* and *hybrid* strategies [11,13] usually exploited for facilitating the browsing of web large data repositories) to support useful services (e.g. a multimedia touristic guide) that assist users visiting cultural environments (indoor museums, archeological sites, old town center), containing several *cultural Points Of Interest* - POIs - (e.g. paintings of museum rooms, buildings in ancient ruins or in an old town center, etc.) correlated with a large amount of multimedia data.

A. Petrosino, L. Maddalena, P. Pala (Eds.): ICIAP 2013 Workshops, LNCS 8158, pp. 257–267, 2013.

The recommendation strategy should be able to provide users with the more relevant information depending on the *context* [14] (i.e. user preferences, user location, observed objects, weather and environmental conditions, etc. as in *Context Aware Recommendation Systems* - CARS [15]) and eventually linked with other on-line touristic information and services, which are usually customized for indoor envronments without taking into account the context information [4].

The majority of approaches to recommendation in the multimedia realm generally exploits *high level* metadata - extracted in automatic or semi-automatic way from *low level* features - that are in different manners correlated and compared with user preferences. These approaches suffer from several drawbacks: (i) it is not always possible to extract in automatic and effective way useful high level information from multimedia features; (ii) for some kinds of multimedia data there is not a precise correlation between high and low level information; (iii) there is not always available explicit and useful information (*knowledge*) about user preferences and feedbacks; (iv) in the recommendation process sometimes it is useful to take into account features of the objects (*context*) that user is currently observing as content information.

Here, we propose a different approach which tries to avoid such drawbacks: (i) analyzing in a separate way low and high level information, i.e. both contribute to determine the utility of an object in the recommendation process; (ii) exploiting system logs to implicitly determine information about users and the related community, considering their browsing sessions as a sort of "ratings"; (iii) considering as relevant content for the recommendation the features of the object that a user is currently watching together with user preferences and other context information. In particular, we present a general multimedia recommender system able to uniformly manage heterogeneous multimedia data and to provide context-aware recommendation techniques supporting intelligent multimedia services useful for the users. In addition, we describe a real case study in the mobile environment, related to an outdoor scenario, together with some preliminary results on user's satisfaction.

The paper is organized as follows. Section 2 presents at a glance a functional overview of our recommender system. Section 3 describes the techniques used for multimedia data management, while Section 4 details the proposed recommendation strategy. Section 5 outlines the chosen case study with the related implementation details and preliminary experiments. Finally, Section 6 reports some conclusions.

2 System Overview

Figure 1 describes at a glance a functional overview of the proposed system in terms of main components. The *Multimedia Data Management Engine* (MDME) is responsible for: (i) accessing by the *Data Indexing and Access* module to the media contents present in the different data sources (*Multimedia Data Repositories*), (ii) extracting by the *Feature Extraction* module from *Multimedia Data* high and low level features useful both for indexing and for the structured representation of the data itself (*Structural Description*).

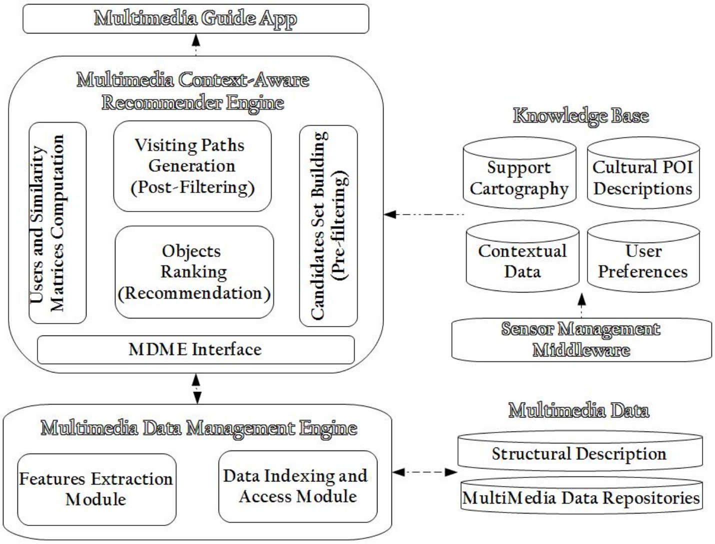
Multimedia Guide App
Multimedia Context-Aware Recommender Engine
Users and Similarity Matrices Computation
Visiting Paths Generation (Post-Filtering)
Objects Ranking (Recommendation)
Candidates Set Building (Pre-filtering)
MDME Interface
Knowledge Base
Support Cartography
Cultural POI Descriptions
Contextual Data
User Preferences
Sensor Management Middleware
Multimedia Data Management Engine
Features Extraction Module
Data Indexing and Access Module
Multimedia Data
Structural Description
MultiMedia Data Repositories

Fig. 1. System Overview

The *Sensor Management Middleware* is responsible for deriving, on the basis of information accessible via physical sensors (e.g. GPS, WSN), Web-services/API or wrapping techniques, the "knowledge" related to the context in which the user is located. In particular, the *Knowledge Base* is constituted by the *Contextual Data* (e.g. weather and environmental conditions registered for the considered place), *User Preferences* (explicitly or implicitly captured), *Cultural POI Descriptions* (in terms of multimedia information) and by a *Support Cartography* useful to geo-localize users and visualize their positions with respect to POIs.

The *MultiMedia Context-Aware Recommender Engine* provides a set of recommendation facilities for multi-dimensional and interactive browsing of multimedia data related to cultural POIs. In particular, on the basis of context information and user preferences the *Candidate Set Building* module selects a set of candidate objects for recommendation; successively, the *Objects Ranking* module performs a ranking of candidates exploiting a proper recommendation strategy (by *Users and Similarity Matrices Computation*). Eventually, the *Visit Paths Generation* module recommends to the users visit paths, providing all the support information (logistics, costs, services, etc..). Finally, each user device is equipped with a *Multimedia Guide App* that allows the fruition of multimedia contents (a questionnaire is submitted in order to capture his/her profile).

3 Management of Multimedia Data

Our data and retrieval models are inspired by the WINDSURF ones [5]. We have a database $\mathcal{O}$ of M multimedia objects, $\mathcal{O} = \{O^1, \ldots, O^M\}$, such as images, videos, and documents, where each objects O is composed of m_O *elements*, $O = \{o_1, \ldots, o_{m_O}\}$ representing, regions of an image, shots of a video, and parts of a document, respectively.

Each element o is described by way of *low level features* F that represent, in an appropriate way, the content of o (e.g., the color distribution of image's regions). As for the retrieval model, given a query object $Q = \{Q_1, \ldots, Q_m\}$ composed of m elements, and an element distance function δ, that measures the dissimilarity of a given pair of elements (using their features), we want to determine the top-k objects in $\mathcal{O}$ that are the most similar with respect to Q. Similarity between objects is numerically assessed by way of a object distance function d_F that combines together the single element distances into an overall value. Consequently, object O^a is considered better than O^b for the query Q iff $d\left(Q, O^a\right) < d\left(Q, O^b\right)$ holds [6]. The computation of the object distance d_F is obtained by combining three basic ingredients: (i) the element distance δ, (ii) the set of constraints that specify how the component elements of the query Q have to be matched to the component elements of another (database) object O, and (iii) the aggregation function that combines distance values between matched elements into an overall object distance value (e.g., a simple average of distance values between matched elements).

The efficient resolution of queries over features is ensured by the *Data Indexing and Access* module which supports indices built on top of elements (e.g., image regions, and video shots) based on the M-tree metric index [8].

In particular, images are first segmented into regions, where pixels included in a single region o share the same visual content (i.e., color/texture) [5]. Image regions are then compared according to their visual features using Bhattacharyya distance metric δ; region scores are opportunely matched by solving a one-to-one matching problem, where each element of a document can be only matched to at most one element of the other document, and vice versa. Then a "biased" average d_F is used to aggregate distance values of matched elements. Videos are first segmented into shots [9]. Then, each shot o is represented by a single representative key frame (e.g., the first frame of the shot), so that shots can be compared by means of the above image similarity function d_F. Each keyframe is first segmented into visually coherent regions, then color/texture features are extracted and stored for each keyframe region [5]. Whole videos are compared by aggregating the similarities between shots (i.e., their representative keyframes). Documents are modeled as follows: each element o is a page contained in the document O and we want to discover whether a new document Q is similar to some existing documents in our database $\mathcal{O}$. Comparison between pages is performed by taking into account relevant contained keywords by using the vector space model [7], so that features extracted from each page include keywords using $tf \times idf$ values after stopping & stemming.

In order to enrich data representation, objects are annotated with high level (semantic) descriptors S (e.g., labels describing what a video shot is related to). Such descriptors are in the form of keywords (or tags) and are semi-automatically assigned to objects by means of a multimedia annotator that, starting from a training set of pre-annotated objects, is able to predict sets of "good" tags able to effectively characterize the content of new untagged objects [10].

We define the universe of tags T as the union of all possible semantic descriptors to be associated to objects in $\mathcal{O}$, and the membership relation $R \subseteq \mathcal{O} \times T$ that indicates that an object O has assigned a tag in T. The annotation process is modelled as a nearest neighbors (NN) problem on object elements and turn into a set of graph-based problems. First, we try to discover affinities between tags and an unlabelled object, which is done using a *Random Walk with Restart* algorithm on a graph that models current annotations as well as elements' similarities. Then, we compute pairwise tag correlations. Again, this relies on the analysis of links in a (second-order) graph. Finally, we combine the results of the two steps to derive a set of terms which are both semantically correlated each other and affine to the new object. This final step amounts to solve an instance of the *Maximum Weight Clique Problem* on a small graph [10]. Note that, while for objects of type image tags are directly associated to images, when annotating videos, we are able to predict tags not only for shots but even for videos, by opportunely propagate tags at the shot level to the video level [9].

Given a user-provided set of tags, as query semantic concepts, objects are selected by the query processor by applying a $co-occurrence$-based distance function d_S on T. The search provides the set of objects (i.e., images, videos/shots, documents) that share at least one tag with the input set.

Both low level features and semantic descriptors concur to determine the *multimedia relatedness* $d(O^i, O^j)$ among two objects. In details, if O^i and O^j are of the same type (e.g., we are comparing two images), we define their global distance as the average between the contribution given by low level features and the one provided by semantics, that is: $d(O^i, O^j) = (d_F(O^i, O^j) + d_S(O^i, O^j))/2$; on the other hand, if we are comparing objects of different type (e.g., a document with a video), their multimedia relatedness equals to their semantic distance only, i.e., $d(O^i, O^j) = d_S(O^i, O^j)$.

4 Context-Aware Multimedia Recommendation Services

The basic idea behind our proposal is that when a user is near to a cultural POI, the related and personalized multimedia description is presented and the recommender system has to be able to: (i) determine a set of *candidate* objects for the recommendation, on the base of user needs and preferences (*pre-filtering stage*); (ii) automatically propose, when a user selects an object, other similar multimedia objects (*recommendation stage*); (iii) dinamically arranges the selected objects in apposite "visiting paths" considering other context information such as weather or environmental conditions (*post-filtering stage*).

Pre-filtering Stage. Each object subject to recommendation may be represented in different and heterogeneous feature spaces. For instance, the picture of a monument may be described by annotations concerning history of the monument, the materials it has been built with, low-level image features, experts' descriptions, visitors' descriptions and reviews, and so on. Each of these sets of features contributes to the characterization of the objects to different extents. Hence, it is important to consider congruently each type of descriptor during the recommendation process. The first step consists in clustering together "similar" objects, where the similarity should consider all (or subsets of) the different spaces of features. To this purpose, we employ high-order star-structured co-clustering techniques [12] to address the problem of heterogeneous data pre-filtering. In this context, the same set of objects is represented in different feature spaces. Such data represent objects of a certain type, connected to other types of data, the features, so that the overall data schema forms a star structure of inter-relationships. The co-clustering task consists in clustering simultaneously the set of objects and the set of values in the different feature spaces. In this way we obtain a partition of the objects influenced by each of the feature spaces and at the same time a partition of each feature space. The pre-filtering stage leverages the clustering results to select a set of candidate objects by using the user's profile, which is modeled as sets of descriptors in the same spaces as the objects' descriptors.

We now provide the formalization of our problem. Let $\mathcal{O} = O^1, \ldots, O^M$ be a set of M multimedia objects and $\mathcal{F} = \{F^1, \ldots, F^N\}$ a set of N feature spaces. A dataset can be viewed under the different views given by the different feature spaces F^k. Therefore, the view k is associated with each feature space F^k. Given a star-structured dataset $\mathcal{SD}$ over $\mathcal{O}$ and $\mathcal{F}$, the goal of the star-structured data co-clustering is to find a set of partitions $\mathcal{Y} = \{Y^1, \ldots, Y^N\}$ over the feature set $\mathcal{F} = \{F^1, \ldots, F^N\}$, and a partition $\mathcal{X}$ of the object set $\mathcal{O}$ by optimizing a certain objective function.

To solve the high-order star-structured co-clustering problem, several algorithms have been proposed based on different approaches. In this work, we adopt a parameter-less iterative algorithm that maximizes the Goodman-Kruskal τ, a statistical measure of association that automatically identifies a congruent number of high-quality co-clusters [12].

In our recommendation problem, a user is represented as a set of vectors $U = \{\boldsymbol{u}^1, \ldots, \boldsymbol{u}^N\}$ in the same N feature spaces describing the objects. Each vector $\boldsymbol{u}^k$ is updated each time the user visits (or re-visit) an object by considering the object features in each space at the instant of the visit. To provide a first candidate list of objects to be recommended, we measure the *cosine distance* of each user vectors associated to the k-th space, with the centroids of each object clusters in the k-th space. For each space, the most similar object cluster is chosen leading to N clusters $\{X_1^c, \ldots, X_N^c\}$ of candidate objects. Then, two different strategies can be adopted to provide the pre-filtered list of candidate objects $\mathcal{O}^c$: (i) *set-union strategy* - the objects belonging to the union of all clusters are retained, i.e., $\mathcal{O}^c = \bigcup_k X_k^c$; (ii) *threshold strategy* - the objects that appears in at least *ths* clusters ($ths \in \{1 \ldots N\}$) are retained.

The first strategy is suitable when user's vectors are associated to very small clusters (e.g., because the user likes very uncommon objects). In any other situation, the second strategy is the most appropriate. As a final step, objects already visited/liked/browsed by the user are filtered out. Notice that, thanks to this approach, users are not described by set of objects, but by sets of features that characterize the objects they visit, like or browse.

Recommendation Stage. In this stage we use a technique that combines several features of multimedia objects (low-level and semantics), eventual past behaviour of individual users and overall behaviour of the whole "community" of users to [2]. Our basic idea is to assume that when an object O_i is chosen after an object O_j in the same browsing session, this event means that O_i "is voting" for O_j. Similarly, the fact that an object O_i is very similar in terms of multimedia features to O_j can also be interpreted as O_j "recommending" O_i (and viceversa). Thus, we model a browsing system for the set of candidate objects $\mathcal{O}^c$ as a labeled graph (G, l), where $G = (\mathcal{O}^c, E)$ is a directed graph and $l : E \rightarrow \{pattern, sim\} \times R^+$ is a function that associates each edge in $E \subseteq \mathcal{O}^c \times \mathcal{O}^c$ with a pair (t, w), where t is the type of the edge which can assume two enumerative values (*pattern* and *similarity*) and w is the weight of the edge. We list two different cases: (i) a *pattern label* for an edge (O_j, O_i) denotes the fact that an object O_i was accessed immediately after an object O_j and, in this case, the weight w^i_j is the number of times O_i was accessed immediately after O_j ; (ii) a *similarity label* for an edge (O_j, O_i) denotes the fact that an object O_i is similar to O_j and, in this case, the weight w^i_j is the similarity between the two objects. Thus, a link from O_j to O_i indicates that part of the importance of O_j is transferred to O_i .

Such an importance is then measured by means the introduction of a recommendation grade $\rho(O)$, and in [1], it has been shown as the ranking vector $R = [\rho(O_1) \dots \rho(O_n)]^T$ of all the objects can be computed as the solution to the equation $R = C \cdot R$, where $C = \{w^i_j\}$ is an ad-hoc matrix that defines how the importance of each object is transferred to other objects. Such a matrix can be seen as a linear combination of a *local browsing matrix* $A_l = \{a^l_{ij}\}$ for each user u_l, a *global browsing matrix* $A = \{a_{ij}\}$ and a *multimedia similarity matrix* $B = \{b_{ij}\}$ such that $b_{ij} = \frac{1-d(O^i,O^j)_{ij}}{\Gamma}$ if $1 - d(O^i, O^j)_{ij} \geq \tau \ \forall i \neq j$, 0 otherwise ($\tau$ is a threshold and Γ is a normalization factors which guarantees that $\sum_i b_{ij} = 1$, see [1] for more details).

The final step is to compute customized rankings for each individual user. In this case, we can then rewrite previous equation considering the ranking for each user as $R_l = C \cdot R_l$, where R_l is the vector of preference grades, customized for a user u_l. We note that solving the discussed equation corresponds to find the stationary vector of C, i.e., the eigenvector with eigenvalue equal to 1. In [1], it has been demonstrated that C, under certain assumptions and transformations, is a real square matrix having positive elements, with a unique largest real eigenvalue and the corresponding eigenvector has strictly positive components. In such conditions, the equation can be solved using the *Power Method* algorithm.

Post-filtering Stage. Finally, the list of suggested items, which are selected as interesting by users, is organized in apposite visiting paths (considering the distances from user location): they are not fixed and are arranged on the base of weather and environmental situations. The recommendation grades of objects, which come from certain cultural POIs with a certain number of persons or with particular values of temperature or humidity, are penalized and such objects could be excluded from recommendation.

5 A Case Study

We consider as real case study the archeological site of *Paestum*, one of the major Graeco-Roman cities in the South of Italy. The main cultural attractions are represented by a set of ancient buildings; in particular, three main temples of Doric style (i.e. the first Temple of Hera, also called Basilica, the second Temple of Hera, also known as Temple of Neptune, and the Temple of Athena), the Roman Forum with several ruins, and the amphitheater, all surrounding by the remains of the city's walls. In addition, there is a museum near the ancient city containing many evidences of the graeco-roman life (e.g. amphorae, paintings and other objects). Thus, all the cited buildings will constitute cultural Points of Interest for our case study.

Users visiting ruins could be happy of having a useful multimedia guide able to describe the main cultural attractions (POIs) and to suggest automatically visiting paths containing multimedia objects of interest. In particular when a user is approaching to a cultural POI (e.g. Temple of Neptune), the related multimedia description is delivered on the user's mobile device. Successively, the recommendation services determine first the list of possible interesting objects (images of other Temples and of Roman Forum) in according to users' preferences (the user prefers to see only images) and then compute a visiting path, shown on a map. The paths have to take into account the current context (in terms of actual position – obtained by GPS – and the selected multimedia data), and to consider the weather and environmental conditions and the previous paths of other users, thus enhancing the visiting experience. Once acquired such kind of information, the path can dynamically change in the case of crowded or unfit to use areas (e.g. too high temperature/humidity or a closed area). Eventually, the visiting paths could be enriched with other touristic POIs (e.g. restaurants, hotels, etc.). A graphic user interface gives the detailed view of the suggested path on an proper cartography, reporting a preview of cultural POIs.

In the following, we report some implementation details concerning the developed prototype for Paestum ruins.

The Multimedia Data is constituted by a collection (managed by *PostegreSQL* DBMS) of about 10,000 images and texts coming from several multimedia repositories (e.g. Flickr, Panoramio, Facebook, Wikipedia, etc.) and related to all the main attractions of Paestum. We associated to each object a set of metadata and spatial information (managed by *PostGIS* spatial database extension), in according to the CIDOC-CRM model [1]. All the data are managed by the Multimedia Data Management Engine that is based on the *Windsurf* library[2]. The Sensor Management Middleware collects and manage sensors' messages from

[1] http://www.cidoc-crm.org/

[2] http://www-db.deis.unibo.it/Windsurf/

Table 1. Comparison between our system and no facilities

	Experts		Medium Exp.		Not Experts	
TLX factor	**With rec.**	**Without**	**With rec.**	**Without**	**With rec.**	**Without**
Mental demand	25.2	27.1	33.1	33.8	35	41
Physical demand	25	31	28	35	31.5	44
Temporal demand	27	32.5	29	35	30	35
Effort	24.8	31.2	33	42	35	50
Perfomances	71	68	71.8	71.2	73.5	73.1
Frustation	24	34	25.1	31.6	25	32

users' mobile devices. By means of apposite *JAVA* libraries and exploiting *GPS* facilities, it is able to capture user location and some environmental parameters (number of a persons in a given area). The Knowledge Base, realized using the *Sesame* Repository and *JENA* libraries, allows to map the observed context instances in the RDF linked open data format. From the other hand, the Multimedia Context-Aware Recommender Engine exploits proper *JAVA* libraries (developed for the systems presented in [2] and integrated with co-clustering libraries) to accomplish its tasks. Eventually for the support cartography, we decided to use *Google Maps*. Finally, a user can interact with our system using – at the moment – an *Android* Multimedia Guide App. The presentation logic is based on apposite widgets. The client requests are elaborated by *JAVA Servlets* and the results are sent to the client in form of XML data.

Wedesigned and carried out several preliminary experiments to investigate how helpful the recommendations offered by our system are, demonstrating that the introduction of such techniques can improve the tourists' experience.

In the training phase, we have chosen 10 users among graduate students that used for 2 days the system without recommendation facilities to capture their browsing sessions (GPS sensor of mobile devices were used to locate user positions) during their visit to build a consistent matrix A for the described collection. We then asked a different group of about 10 people (this group consisted of 5 not-expert users on graeco-roman art, 3 medium expert users and 2 expert users) to complete several visits (3 visits per user) of different complexity within the Paestum ruins and without any recommendation facility. After this test, we asked them to browse once again the same collection with the assistance of our recommender system and complete other tasks of the same complexity. In a similar manner, in a second time we asked another group of 10 people to browse the same collection first with the assistance of our recommender system completing other different tasks and then without any help.

We have subdivided browsing tasks in the following four broad categories: (i) ***Low Complexity*** tasks (T_1) - explore at least 5 POIs related to ancient buildings; (ii) ***Medium Complexity*** tasks (T_2) - explore at least 10 POIs related to graeco-roman *temples* or *amphitheaters* or *Roman forum buildings*; (iii) ***High Complexity*** tasks (T_3) - explore at least 15 POIs related to *Roman forum buildings* or *amphitheaters* or *city walls' gates*.

The strategy we used to evaluate the results of this experiment is based on NASA TLX (*Task Load Index factor*). To this aim, we then asked the users to express their opinion about the advantage of our system to provide an effective user experience in completing the assigned visitng tasks, with respect to use a simple museum map and a touristic guide. We also considered situations (that we have simulated during experimentation) in which an area is closed.

Thus, we obtained the average results scores for each of three categories of users reported in Table 1 (the lower the TLX score — in the range $[0 - 100]$ — the better the user satisfaction).

Note that not-expert users find our system more effective than the other users' category in every sub-scale, because they consider very helpful the provided suggestions. Instead, for expert and medium expert users' opinion, our system outperforms a classical touristic guide in every sub-scale except for *mental demand and performances*: this happens because an expert user considers sometimes not useful the automatic suggestions just because they know what they are looking for.

6 Conclusions

In this paper we proposed a novel recommender platform in the Cultural Heritage domain. We realized a system for Paestum ruins, providing to tourists personalized visiting paths. Then we investigated the effectiveness of the proposed approach in the considered scenario, based on the browsing effectiveness and users satisfaction. Experimental results showed that our approach is quite promising and encourages further research.

References

1. Albanese, M., d'Acierno, A., Moscato, V., Persia, F., Picariello, A.: Modeling recommendation as a social choice problem. In: Proc. of ACM RecSys 2010, Barcelona, Spain, pp. 329–332. ACM (2010)
2. Albanese, M., d'Acierno, A., Moscato, V., Persia, F., Picariello, A.: A multimedia semantic recommender system for cultural heritage applications. In: Proc. of ICSC 2011, Palo Alto, CA, USA, pp. 403–410. IEEE (2011)
3. Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. Human Mental Workload 1, 139–183 (1988)
4. Kabassei, K.: Personalisation systems for cultural tourism. In: Multimedia Services in Intelligent Environments, Springer, pp. 101–111. Springer (2013)
5. Bartolini, I., Ciaccia, P., Patella, M.: Query processing issues in region-based image databases. Knowl. Inf. Syst. 25(2), 389–420 (2010)
6. Ilyas, I.F., Beskales, G., Soliman, M.A.: A survey of top-k query processing techniques in relational database systems. ACM Comput. Surv. 40(4) (2008)
7. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley, Reading (1989)
8. Ciaccia, P., Patella, M., Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces. In: Proc. of VLDB 1997, Athens, Greece, pp. 426–435 (1997)
9. Bartolini, I., Patella, M., Romani, C.: SHIATSU: tagging and retrieving videos without worries. Multimed. Tools Appl. 63(2), 357–385 (2013)

10. Bartolini, I., Ciaccia, P.: *imagination*: Exploiting link analysis for accurate image annotation. In: Boujemaa, N., Detyniecki, M., Nürnberger, A. (eds.) AMR 2007. LNCS, vol. 4918, pp. 32–44. Springer, Heidelberg (2008)
11. Bartolini, I., Zhang, Z., Papadias, D.: Collaborative filtering with personalized skylines. IEEE Trans. Knowl. Data Eng. 23(2), 190–203 (2011)
12. Ienco, D., Robardet, C., Pensa, R.G., Meo, R.: Parameter-less co-clustering for star-structured heterogeneous data. Data Min. Knowl. Discov. 26(2), 217–254 (2013)
13. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender Systems Handbook. Springer (2011)
14. Dourish, P.: What we talk about when we talk about context. Personal and Ubiquitous Computing 8(1), 19–30 (2004)
15. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proc. of ACM RecSys 2010, Barcelona, Spain, pp. 79–86. ACM (2010)