Context-Driven Tour Planning Service: An Approach Based on Synthetic Coordinates Recommendation

Alexey Kashevnik[1, 2], Sergei Mikhailov[1, 3], Harris Papadakis[4], Paraskevi Fragopoulou[4]
[1] SPIIRAS, St. Petersburg, Russia
[2] Petrozavodsk State University, Petrozavodsk, Russia
[3] ITMO University, Saint Petersburg, Russian Federation
[4] Technological Educational Institute of Crete, Heraklion, Crete, Greece
{alexey, sergei.mikhailov}@iias.spb.su
adanar@ie.teicrete.gr, fragopou@ics.forth.gr

***Abstract*—The paper presents a hybrid context/model-based tour planning service aimed at recommendation generation by providing the tourists the sequence of attractions that are more interesting for him/her based on previous activity with the service. The service is developed based on SCoR recommender system that is aimed at recommendation generation based on calculating the synthetic coordinate between tourists of the service in according with their ratings. SCoR is a model-based collaborative filtering algorithm, constructing a model based on the user's personal ratings as well as exploiting collaborative information from the ratings of the rest of the users. One of the main advantages of SCoR's model is its ability to incorporate additional training information (new ratings) without having to perform the training process from the beginning. The prototype has been implemented for Android-based smartphone and has been evaluated for St. Petersburg city. For the evaluation the attraction database has been formed that includes attraction location information from OpenStreeMaps platform, location description and media from Wikipedia, and ratings from Google Place.**

I. Introduction

With the penetration of Internet access all over the globe, consumers' choices have multiplied exponentially. Even though this allows for a multitude of choices and a wider variety of selection, it has become increasingly difficult to match consumers preferences with the most appropriate products, especially given the diversity of needs between them. Recommender systems [1], [2] try to amend this situation by analyzing consumer preferences and trying to predict the preference of a user for any single item.

The problem of content recommendation can be described as follows. Given a set U of users, a set I of items and a set R of rankings (evaluations) of users for items, we need to estimate (predict) the ratings for a user-item pair which is not in R.

One of the well known application of recommender systems is their use in consumer sites with a vast range of products, in order to provide consumers with targeted information about products that might interest them. Another application can be found in designing marketing strategies, where recommender systems are used to predict the popularity of products. Recommender systems are also used to provide users with recommendations for other entities than consumer items, such as web pages.

A promising application area for Recommender Systems is that of Tourism and Travel (see [3], [4], [5]). The use of such techniques in this field presents itself with several opportunities as well as challenges. The obvious advantage stems from the fact that, most of the times, tourists visit a previously unknown place. A recommender system can provide visitors with filtered information about places to visit and/or events to attend, tailored to their personalized needs and preferences. In addition, studies have shown that people have difficulty in expressing their preferences and needs [6]. A tool which can provide them with personalized recommendations will make planning a trip much easier as well as enjoyable. Recent study [7] have also shown that the majority of travelers rely on the web and other digital tools (social media, search engines etc.) to research and decide on their potential destinations. In addition most users make frequent use of applications in their mobile phone, thus making it easier for them to adopt and use a mobile Recommender System. Finally, especially in the case of planning a trip, Recommender Systems are more beneficial to the user that in other cases. Planning a vacation usually involves combining a large number of diverse entities (accommodation, points of interest, events, travel routes) as compared for searching for a specific type of product to buy for instance. Furthermore, from the point of view of operators and service providers, Recommender Systems for Tourism can boost tourist flows and revenues by providing an easier and more pleasurable experience to visitors. By recommending the appropriate suggestions to the visitors, chances of him/her of purchasing a service or product or attending some event are increased.

The paper presents an approach to context-driven tour planning service that is based on the SCoR recommendation system aimed at recommendation generation based on information about tourists, attractions, and ratings that tourists provide for attractions. The described prototype of the service is based on recent research conducted by authors [8], [9] earlier.

The paper is structured as follows. Section II describes the related work in the area of recommendation generation for tourists. The SCoR recommendation system is described in section III. Section IV presents the developed tourist decision support system. The system evaluation is presented in Section V. Main results are summarized in the Conclusion.

II. Related Work

As mentioned, Recommender Systems [10] purpose is to direct the user to a small subset of items out of a large pool, which might interest him/her the most. As such, they can be defined as a subclass of information filtering systems with the aim of predicting the preference of a user for an unknown (to him/her) item. A wide range of techniques and approaches have been proposed over the years in the effort to accurately predict the preferences of users, leading to a diversity of efforts to tackle the aforementioned problem. In this Section, we shall provide a brief but needed categorization of the various approaches as well as a brief overview on the most recent advances in the use of recommender systems in the field of Tourism and Travel.

A. Collaborative Filtering

Collaborative Filtering approaches provide user recommendations by "studying" the preference patterns of all available users. Collaborative Filtering approaches usually only require an existing dataset of user-item preferences (ratings), which is used to deduce future user preferences. Initial approaches using this method employed a ratings-based similarity function and are called Memory-based Collaborative Filtering methods [11]. Since then however, the majority of the literature is comprised by the so called Model-based Collaborative Filtering methods [12]. These approaches employ Machine-Learning technique in order to "study" the known preferences and construct a model, which in turn can provide the necessary recommendations.

B. Content-based

Content-based recommender systems [13] require the existence of item meta-data (i.e. descriptive attributes/features) that accompany each item. These meta-data provide additional information (other than just simple user-item ratings) on the nature of each item which is exploited by recommendation systems in order to provide more accurate predictions. In such systems, meta-data are often used either in item-to-item similarity functions and/or in order to automatically construct a user preference profile. Their main draw-back is that the required meta-data are not always available or can be easily mined from existing raw data.

C. Context-based

According to [14], "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application (including the user and the application themselves)". Context can thus be any situational information, usually related to the user, such as time of day, weather conditions, mood, location and so forth. These approaches [15] take this type of information into account as well as possible additional information usually related to other types of recommender systems. Context-based Recommender Systems are especially suitable and popular approaches for a Tourism-related recommender system.

In the rest of this Section we shall provide some recent examples of uses of recommender systems in the Tourism and Travel sector. For additional analysis of Tourism and Travel related recommender systems we refer the interested reader to [9].

Tourec [16], [17] is a fully functional, mobile-based application for tourist trips containing suggested points of interest (POIs) in an iterative form. It is a hybrid Context/Content-based system where POIs are automatically retrieved by FourSquare and suggested to the user. The user first specifies his/her preferences for various types of POIs, creating a preference profile. Then he/she specifies an origin, a destination and the preferred duration of the trip. Subsequently, the system compiles a list of POI recommendations and combines them in order to create the optimal route which meets user's specifications. This route is presented to the user through the mobile application's graphical interface.

The authors in [18] aim at studying the correlation between the features of tourist destinations and the seven-factor model of tourist preferences [19]. Datasets with extensive features per destination were used and linear regression analysis was employed which showed the significant relation between the features and the factors of the model. In addition, feature-based clustering on the available destinations was performed, which resulted in their classification into six categories, which contributes to a more generalized solution of mapping without the need of prior information.

SigTur/E-Destination [20] is another hybrid approach that employs a number of content-based as well as collaborative filtering-based techniques to provide personalized recommendations of touristic activities. Users access the service through a web-based interface and specify their preferences of content types. Content is described through a detailed ontology which captures the nature of the destinations as well as their suitability for various types of activities. Several methods are used to generate recommendations for a particular user. Knowledge-based techniques are used to provide content-based recommendations. In addition, collaborative filtering recommendations are also produced by employing a wide range of different similarity functions. All recommendations are combined based on each predicted rating and its confidence level, to present the user with the final recommendations. In addition the user is able to rate the proposed destinations/activities and thus automatically update his/her preference profile.

The work described in [21] was designed as an content-based upgrade to the recommendation engine of wOndary (https://wondary.com), a platform where independent travelers can plan their trips. An automatic method was employed, aggregating information both from Google Places types (https://developers.google.com/places/web-service/supported_types) and FourSquare (https://developer.foursquare.com/docs/resources/categories) in order to assign each possible destination some type derived from a pre-designed ontology and thus generate the required content information for each destination. User preferences are also automatically derived from his/her existing travels, stored on the wOndary site.

Finally, a recent, interesting notion that has emerged in the context of Tourism-related Recommender Systems is that of group recommendations. Classic recommender systems' purpose is to provide personalized recommendation to each of its users. It is common practice however for tourists to travel in groups (tour-operated groups, friends, family, etc). In recent years, several approaches have emerged in order to address the issue of group recommendations. A popular approach [22] is the aggregation of individual recommendations using aggregation strategies such as *Average, Average without Misery*, and *Most Pleasure*. Other approaches such as [23] employ group consensus strategies.

III. The SCoR algorithm

The basis for providing the necessary Recommendations is SCoR, a novel personalized recommendation algorithm. We refer the reader to [24] for a detailed description of the algorithm. The algorithm uses a Model-based Collaborating Filtering approach, which is dependent on a known set of user-to-item preferences, in order to train a preference prediction model. Thus, a number of known preferences of each user for some items must be already known, each one provided in the form of a triplet (u, i, r), where r is a scalar rating of user u for item i.

In the core of our proposal lies the spring metaphor which was inspired the Vivaldi synthetic network coordinate algorithm [25]. Thus, the basis of the SCoR algorithm is a Synthetic Euclidean Coordinates system, which randomly assigns an initial position in an N-dimensional space to each element in the user U and the item I sets. The algorithm iteratively updates the positions of all elements until, for every known rating (u, i, r), the Euclidean distance between user u and item i corresponds to the rating value r. The positions are updated using (1), as follows:

$$p(x) = p(y) + \delta \cdot (dd(x, y) - d(x, y)) \cdot \frac{p(x) - p(y)}{d(x, y)} \quad (1)$$

where $p(x)$, $p(y)$ are the positions of a user-item pair, $d(x, y)$ is their current Euclidean distance, $dd(x, y)$ is their desired distance (based on the rating value r), $\frac{p(x)-p(y)}{d(x,y)}$ is the unit vector that gives the direction node x should move, and δ controls the method's convergence, since it is the fraction of distance node x is allowed to move toward its ideal position. This process is performed iteratively for each known user-item pair (i.e. for which there is a known rating r).

Upon algorithm convergence (i.e all users and items in the N-dimensional space rarely change their positions any more), the Euclidean distance between user u and an unrated (by user u) item i provides the basis for a prediction for the preference of user u for item i. The algorithm is deemed to have converged when the training RMSE change after a number of updates falls below a pre-defined threshold.

$$RMSE = \sqrt{E\{(R - \widehat{R})^2\}} \quad (2)$$

The main input of the algorithm is a list of known (u, i, r) triplets which are used to train the model. In addition, some straight forward execution parameters need to be defined. These are:

- The number of dimensions of the Euclidean space
- The minimum and maximum values allowed for ratings (r)
- The termination threshold of the algorithm.

TABLE I. RMSE of the eight recommendation algorithms for the four datasets

Dataset	SmallNetflix	ML	Jester2	Jester3
ALS	1.160	0.964	0.943	1.380
ALS-CCD++	1.140	0.932	0.917	1.300
SGD	0.961	0.898	0.872	0.906
Bias-SGD	0.958	0.897	0.872	0.909
Bias-SGD2	0.967	0.888	0.861	0.923
SCoR	**0.940**	**0.875**	**0.854**	**0.894**
RBM	0.941	0.900	0.880	0.912
SVD++	0.989	0.944	0.910	0.953

In its original publication, SCoR was compared against seven state-of-the-art recommender systems. In Table I,the comparison of SCoR is shown, in terms of accuracy (RMSE), against the following seven state-of-the-art algorithms [26]:

- **ALS**, the Alternating Least Squares algorithm [27].
- **ALS-CCD++**, the Parallel Coordinate Descent approach to Matrix Factorization [28].
- **SGD**, the Stochastic Gradient Descent method [29].
- **Bias-SGD**, the Biased Stochastic Gradient Descent method [30].
- **Bias-SGD2**, an improved alternative of the Biased Stochastic Gradient Descent method [30].
- **RBM**, the Restricted Bolzman Machines [31].
- **SVD++** method [30].

The datasets used to compare the recommendation algorithms were the following four real world datasets, with diverse structural characteristics (density, average degree, etc):

- **SmallNetflix** a reduced version of the original Netflix Prize dataset.
- **ML**, the MovieLens dataset by GroupLens website (https://grouplens.org/datasets/movielens/)
- **Jester2** and **jester3** datasets (http://www.ieor.berkeley.edu/ goldberg/jester-data/)

The algorithm's performance is measured using the Root Mean Squared Error (RMSE) [32], [33]. The RMSE values are computed using a "ground-truth" set of known user ratings, whose values are treated as unknown by the algorithms. For each rating in the "ground-truth", each algorithm calculates a predicted recommendation value. These are compared against the actual ratings using (2), where R is the set of "ground-truth" values and $\widehat{R}$ are the corresponding predicted values computed by each algorithm. A lower RMSE value indicates a more accurate prediction. Table I shows how SCoR outperformed all other algorithms for each dataset.

IV. Tour Planning System

The developed tour planning system is aimed at tourist support with information about interested for the tourist attractions based on his/her preferences and context situation in the region location. To take into account tourist preferences the SCoR system described in the prevoous section is used. Context in modelled based on ontology management technique [34].

The system consists of the two main modules: tourist smartphone and cloud (see Fig. 1). The tourist interacts with the smartphone via the graphical user interface and access to the attraction that are ordered for him/her based on the ratings and the accessibility possibilities. The tourist sets these ratings to estimate previously attended attractions. Mobile application installed to the tourist smartphone is responsible for providing to him/her the following information: list of attractions with description; media information about interested attraction; ratings of attractions. SCoR module is used to access to the SCoR system to calculate the closeness of a tourist to an attraction based on the estimations of other tourists in the system (see section III). Offline attraction database keeps information about all known attractions.

Database management service is located in the cloud environment and responsible for attraction database formation from different Internet services. The OpenStreetMap platform is used for generating list of attractions in the tourist location. Wikipedia platform provides attraction description and media information, and Google Place platform provides the rating information about found attraction in the tourist region. Data management service provides web interface for the tourism experts who can view and manage the attraction in the database as soon as create new region. Database management service creates the offline attraction database and shares it with the tourit mobile device.

Tourist smartphone provide to the tourist a tour that consists of the set of attraction that have been determined that is most interesting to see for him/her as soon as these attractions are reachable by the reasonable time based on the context situation in the tourist location region. Context includes the location, time, traffic situation, weather. Tourist has possibility to see text and media information about attraction. Media information includes photos, videos, and / or audio. After the browsing the attraction the tourist has possibility to estimate it by providing the rating from the following interval (0, 5]. The rating is provided for the current context that includes weather situation and company of the tourist. Weather situation is actual for the outside attractions (such as parks, monuments, bridges, and etc.). In case of rain the rating can be completely different compare with sunny weather. The same situation for company. If the tourist is travelling alone he/she can have one preferences but in case of travelling with family the preferences can be different and more oriented to the children.

V. Evaluation

A. Data Extraction

To conduct the experiment the attraction information for St. Petersburg, Russia has been processed. The attraction data extraction from the OpenStreetMaps and Wikipedia platforms is described in details in the paper [35]. The general scheme of attraction rating extraction is presented on Fig. 2.

Firstly the ratings for the attractions were taken from the Google Places service (https://cloud.google.com/maps-platform/places). This service allows users to fetch various information about attractions around the world with possibility of free provision of a certain amount of data. For the experiment, the authors gather the attraction geo-data, title, and rating. To reduce the number of calls to the Google Place service, several points in the city with the greatest number of attractions have been selected. For each point, a search has been performed within a radius of 50 kilometers on the presence of various attractions. Data was requested in Russian and English languages to increase attraction coverage.

After that, the additional search has been done by using the Google Knowledge Graph service (https://developers.google.com/knowledge-graph). This service can be described as a semantic knowledge database with possibility of the linked data search. Each item from Google Place data was processed with additional search by the following template: {`title: place_title, types: ['Place', 'TouristAttraction'], languages: ['en', 'ru']`}. The `types` field describes the item type in knowledge graph. For some attraction the Wikipedia references is accessible that provides possibilities to merge attraction found in Wikipedia and attraction found in Google Place Service. As result, the authors get a array with merged data from both services.

At the end, the mapping process between attractions from the offline database and attractions from Google Places and Knowledge Graph is started. Firstly, the search process for the references to Wikipedia articles was performed among the Google data. This references let the authors match different attractions, because every attraction from offline database contains link to the Wikipedia about itself. Secondly, the geographical search was performed in the offline database with filtering data by using attractions' titles similarity based on Levenshtein distance.

B. Ratings Generation

To conduct the experiments the personalized ratings are required that can characterize the user's interests. State-of-the-art analysis of the existing tourism portals and research papers showed that such databases are not accessible. In this case authors propose in the paper the following rules for the personalized rating formation (see table II). The general idea is to create the unique group of people with own attraction preferences. It is proposed to create 8 groups. Each group contains 10 people who create ratings for each attractions in according to pattern described in the table II. Variable $rating$ in the table represents overall rating from the user and belongs to the following range: $rating \in [0, 5]$. If the result is out of bounds in the generation process, it is reduced to the minimum or maximum values.

User groups I–III constructs their ratings based on the ratings acquired from the Google Places service. Variable G_r represents the original rating from the Google Places service belongs to the following range: $G_r \in [0, 5]$. Group I takes the non-zero rating and adds a value from the $S_r() = \text{random}(-0.5, 0.5)$ function. This function takes a pseudo-random value from $[-0.5, 0.5]$ range. Group II choose the opposite rating from the G_r and Group III divides incoming rating by 2.

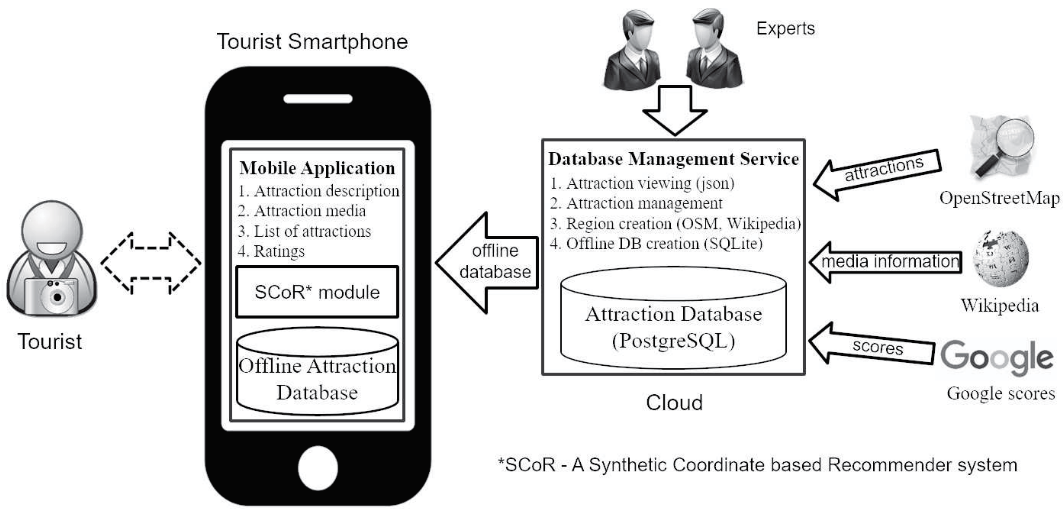

Fig. 1. Developed Tour Planning System: A Reference Model

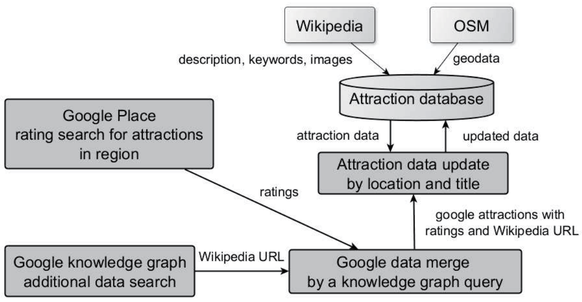

Fig. 2. General scheme of attraction data mapping

Groups IV–VIII shares another approach: people in these groups choose a certain kind of attraction and rate only that types of attractions. Variable R_c is a base constant, which is equal 4.5, tag is a attraction type from OpenStreetMaps data. For a suitable attraction, the value of the function $S_r()$ is added to the R_c constant. For the current steps, only museums, theatres, parks, monuments and bridges are processed.

TABLE II. EXPERIMENT DATASETS

User group	Explanation
I	$rating = \begin{cases} G_r + S_r(), \text{if } g\_rating > 0 \\ \text{None, otherwise} \end{cases}$
II	$rating = \begin{cases} 5 - G_r, \text{if } g\_rating > 0 \\ \text{None, otherwise} \end{cases}$
III	$rating = \begin{cases} G_r/2, \text{if } g\_rating > 0 \\ \text{None, otherwise} \end{cases}$
IV	$rating = \begin{cases} R_c + S_r(), \text{if tag} == \text{'museum'} \\ \text{None, otherwise} \end{cases}$
V	$rating = \begin{cases} R_c + S_r(), \text{if tag} == \text{'theatre'} \\ \text{None, otherwise} \end{cases}$
VI	$rating = \begin{cases} R_c + S_r(), \text{if tag} == \text{'park'} \\ \text{None, otherwise} \end{cases}$
VII	$rating = \begin{cases} R_c + S_r(), \text{if tag} == \text{'monument'} \\ \text{None, otherwise} \end{cases}$
VIII	$rating = \begin{cases} R_c + S_r(), \text{if tag} == \text{'bridge'} \\ \text{None, otherwise} \end{cases}$

C. SCoR Integration

Described in the section III SCoR recommendation system has been adapted for smartphone utilization. The SCoR algorithm was originally implemented as a stand-alone application [36]. In order to meet the needs of the current work, this implementation has been modified into a Java library, which can be integrated into any Android application. Its API exposes the following methods:

`void` **`trainModel`**`(Iterable<Rating> train, File coordsFile, String coordsOutputFile)` — trains the SCoR Model.

- **train:** A Collection of the available known user preferences ((u, i, r) triplets).
- **coordsFile:** The location of previous training data. If this parameter is not provided, the model will be trained from scratch.
- **coordsOutputFile:** The desired location to store the new (updated) training data

`double` **`getRecommendation`**`(String userID, String itemID)` — provides rating prediction for a single user,item pair

- **userID:** The ID of the user in question.
- **itemID:** The ID of the item in question.
- **Returns** the predicted rating of the user for that item.

`ArrayList<RecResult>` **`getRecommendation`**`(String userID)` — provides rating predictions for a single user on all available items.

- **userID:** The ID of the user in question.
- **Returns** a sorted Collection of the predicted ratings of the user on all items.

In addition, the following variables/parameters can be set by the developer, prior to training:

- **dims:** The desired number of dimensions of the Euclidean space.
- **min:** The minimum allowed value of any rating.
- **max:** The maximum allowed value of any rating.
- **threshold:** The value of the termination threshold.
- **nrOfThreads:** The total number of threads to be used during the Model training, for parallel execution.

D. Implementation

The authors use Android prototype of the offline tour trip planning system (described in paper [34]). Recommendations are generated locally at the client side. Google Places ratings are stored into the offline sqlite database and generated dataset is attached in APK as a JSON file. This file contains an array of dictionary objects with attraction id as key and user rating as value.

The training process of the recommendation model is performed once on the application startup in a background thread. During the training the user gets information about attraction in according with the trained last time model. This allows continuous user experience with the application. As input, the SCoR recommendation system requires an iterable array which contains data about user rating for an attraction. After the training the recommendation system can predict the tourist preferences for other attractions. In case of adding new ratings to attractions, the model is retrained again to present the tourist on-the-fly recommendations based on his/her preferences.

The prototype screenshots of the proposed tour planning service are presented on Fig. 3 and Fig. 4. Each time if the user enters main screen, the recommendations based on users scores and preferences are presented in the ”The best attractions in the location” part of the screen. Each entry in section contains the following information: attraction image and title, distance from the tourist to the attraction, attraction rating from the Google Places, and synthetic score calculated by the SCoR system. Fig. 3 shows the situation when the tourist open the service. In this moment he/she has preferences about museums and theaters (calculated by SCoR system). Fig. 4 shows the reordered list of attractions when the tourist preferences have been updated (scores have been calculated after the user estimates brifgrs and parks. It should be noted that additional training process (in the case of user preferences update) does not require model training from scratch as previous training results are used in the new training in order to significantly reduce training time.

E. Performance Experiments

The performance experiments have been conducted in the Android emulator. The computer specification is the following: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz process, 16 Gb RAM, 1 TB HDD. Based on the principles from subsection V-B, the datasets were generated. Each dataset contains the different amount the user group: 10 people, 100, 250, 500, 750 and 1000 accordingly. The time of the initial training (Fig. 5) and time of recommendations retraining after user rating addition (Fig. 6) were measured. The retraining speed is significantly faster than initial training speed, however, it is recommended to manipulate with model in the background thread for improving user experience with the application.

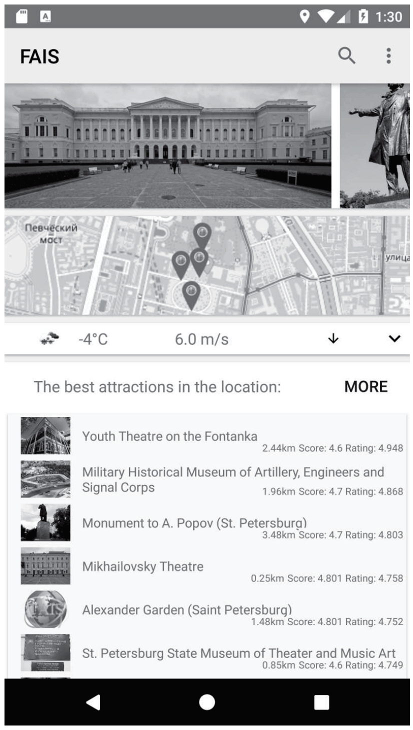

Fig. 3. Overview of the attraction recommendations screen

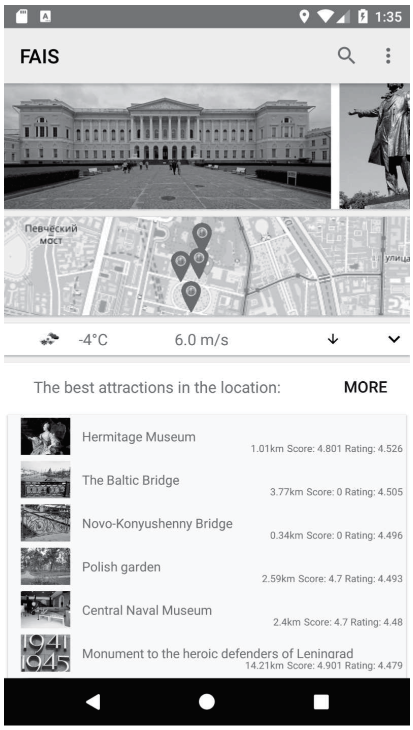

Fig. 4. Overview of the attraction recommendations screen after applying the tourist additional ratings

Fig. 5. First training Performance graph

Fig. 6. Retraining performance graph

VI. Conclusion

The paper presents an approach to context-driven Tour Planning based on synthetic coordinate recommendation. The paper shows joint work between teams from SPIIRAS, Russia and TEI of Crete, Greece. Integration of the SCoR recommender system developed by the TEI of Crete to the tourist trip planning system provides benefits for potential users. The preliminary experiments described in the section V-B shows that for such city as St. Petersburg with amount of attractions accessible in OpenStreetMaps the recommendations are implemented in real time. For the experiments 80 users have been generated with ratings based on information imported from Google Place. The tests show that based on the users ratings the system quickly retrain the model and reordering the attractions. In the future authors are going to extent the experiments.

Acknowledgment

The presented results are part of the research carried out within the project funded by grants # 18-37-00337 and 17-29-03284 of the Russian Foundation for Basic Research as well as by European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH CREATE INNOVATE (project code: T1EDK-02147).

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005. [Online]. Available: https://doi.org/10.1109/TKDE.2005.99

[2] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10 059 – 10 072, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417412002825

[3] R. C. Palacios, F. J. García-Peñalvo, V. Stantchev, and S. Misra, "Towards a social and context-aware mobile recommendation system for tourism," *Pervasive and Mobile Computing*, vol. 38, pp. 505–515, 2017.

[4] F. Santos, A. Almeida, C. Martins, R. Gonalves, and J. Martins, "Using poi functionality and accessibility levels for delivering personalized tourism recommendations," *Computers, Environment and Urban Systems*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971517302880

[5] X. Zhang, L. Yu, M. Wang, and W. Gao, "Fm-based: Algorithm research on rural tourism recommendation combining seasonal and distribution features," *Pattern Recognition Letters*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786551830936X

[6] A. H. Zins, "Exploring travel information search behavior beyond common frontiers," *Information Technology & Tourism*, vol. 9, no. 3-4, pp. 149–164, 2007. [Online]. Available: https://www.ingentaconnect.com/content/cog/itt/2007/00000009/f0020003/art00002

[7] I. MediaCT. (2014) The 2014 travelers road to decision. [Online]. Available: https://www.thinkwithgoogle.com/consumer-insights/2014-travelers-road-to-decision/

[8] A. Smirnov, A. Kashevnik, S. I. Balandin, and S. Laizane, "Intelligent mobile tourist guide," in *Internet of Things, Smart Spaces, and Next Generation Networking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 94–106.

[9] A. V. Smirnov, A. M. Kashevnik, and A. Ponomarev, "Context-based infomobility system for cultural heritage recommendation: Tourist assistant—tais," *Personal and Ubiquitous Computing*, vol. 21, no. 2, pp. 297–311, Apr 2017. [Online]. Available: https://doi.org/10.1007/s00779-016-0990-0

[10] C. C. Aggarwal, *Recommender Systems*, 01 2016, p. 493.

[11] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," *SIGIR Forum*, vol. 51, no. 2, pp. 227–234, Aug. 2017. [Online]. Available: http://doi.acm.org/10.1145/3130348.3130372

[12] M. Elahi, F. Ricci, and N. Rubens, "A survey of active learning in collaborative filtering recommender systems," *Comput. Sci. Rev.*, vol. 20, no. C, pp. 29–50, May 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2016.05.002

[13] M. J. Pazzani and D. Billsus, "The adaptive web," P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer-Verlag, 2007, ch. Content-based Recommendation Systems, pp. 325–341. [Online]. Available: http://dl.acm.org/citation.cfm?id=1768197.1768209

[14] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Hum.-Comput. Interact.*, vol. 16, no. 2, pp. 97–166, Dec. 2001. [Online]. Available: http://dx.doi.org/10.1207/S15327051HCI16234_02

[15] G. Adomavicius and A. Tuzhilin, *Context-Aware Recommender Systems*. Boston, MA: Springer US, 2011, pp. 217–253.

[16] D. Herzog, C. Laß, and W. Wörndl, "Tourrec: a tourist trip recommender system for individuals and groups," in *RecSys*, 09 2018, pp. 496–497.

[17] C. Laß, W. Wörndl, and D. Herzog, "A multi-tier web service and mobile client for city trip recommendations," in *Proceedings of the 8th EAI International Conference on Mobile Computing, Applications and Services*, ser. MobiCASE'16. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 153–154. [Online]. Available: https://doi.org/10.4108/eai.30-11-2016.2267194

[18] J. N. Mete Sertkan and H. Werthner, "Mapping of tourism destinations to travel behavioural patterns," in *Information and Communication Technologies in Tourism*. Springer, Chams, 2018, pp. 422–434.

[19] J. Neidhardt, R. Schuster, L. Seyfang, and H. Werthner, "Eliciting the users' unknown preferences," in *Proceedings of the 8th ACM Conference on Recommender Systems*, ser. RecSys '14. New York, NY, USA: ACM, 2014, pp. 309–312. [Online]. Available: http://doi.acm.org/10.1145/2645710.2645767

[20] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borrs, "Sigtur/e-destination: Ontology-based personalized recommendation of tourism and leisure activities," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 633–651, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197612000516

[21] L. Dietz and A. Weimert, "Recommending crowdsourced trips on wondary," in *Proceedings of the ACM RecSys Workshop on Recommenders in Tourism*, ser. RecSys '18. ACM, 2018, pp. 13–17.

[22] M. Judith, *Group Recommender Systems: Aggregation, Satisfaction and Group Attributes*, 2015, pp. 743–776. [Online]. Available: https://doi.org/10.1007/978-1-4899-7637-6_22

[23] J. Castro, F. J. Quesada, I. Palomares, and L. Martínez, "A consensus-driven group recommender system," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 887–906, Aug. 2015. [Online]. Available: http://dx.doi.org/10.1002/int.21730

[24] H. Papadakis, C. Panagiotakis, and P. Fragopoulou, "Scor: A synthetic coordinate based system for recommendations," *Expert Systems with Applications*, vol. 79, pp. 8–19, 2017.

[25] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 15–26.

[26] D. Bickson. (2012) Collaborative filtering with graphchi. [Online]. Available: http://bickson.blogspot.com/2012/12/collaborative-filtering-with-graphchi.html

[27] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize," in *Proc. 4th Intl Conf. Algorithmic Aspects in Information and Management, LNCS 5034*. Springer, 2008, pp. 337–348.

[28] H.-F. Yu, C.-J. Hsieh, S. Si, and I. Dhillon, "Scalable coordinate descent approaches to parallel matrix factorization for recommender systems," in *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ser. ICDM '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 765–774.

[29] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[30] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 426–434.

[31] G. Hinton, "A practical guide to training restricted boltzmann machines." University of Toronto, Tech. Rep. UTML TR 2010-003, 2010.

[32] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

[33] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, 2012.

[34] S. Mikhailov and A. Kashevnik, "An ontology for service semantic interoperability in the smartphone-based tourist trip planning system," in *2018 23rd Conference of Open Innovations Association (FRUCT)*, Nov 2018, pp. 240–245.

[35] A. Kashevnik and S. Mikhailov, "Smartphone-based tourist trip planning system: a context-based approach to offline attraction recommendation," *MATEC Web of Conferences*, vol. 161, p. 03026, 01 2018.

[36] NetDiLab, TEI of Crete, "Scor algorithm implementation," 2017. [Online]. Available: http://netdil.epp.teicrete.gr/scor/