MATEC Web of Conferences 44, 01005 (2016)
DOI: 10.1051/matecconf/20164401005
© Owned by the authors, published by EDP Sciences, 2016

An Agent Framework of Tourism Recommender System

Zhi Yang Jia[1,a], Wei Gao[2] and Yi Jin Shi[1]

[1] *Yunnan University, Tourism and Culture College, 674100 Lijiang, China*
[2] *Yunnan Normal University, School of Information Science, 650500 Kunming, China*

Abstract. This paper proposes the development of an Agent framework for tourism recommender system. The recommender system can be featured as an online web application which is capable of generating a personalized list of preference attractions for tourists. Traditional technologies of classical recommender system application domains, such as collaborative filtering, content-based filtering and content-based filtering are effectively adopted in the framework. In the framework they are constructed as Agents that can generate recommendations respectively. Recommender Agent can generate recommender information by integrating the recommendations of Content-based Agent, collaborative filtering-based Agent and constraint-based Agent. In order to make the performance more effective, linear combination method of data fusion is applied. User interface is provided by the tourist Agent in form of webpages and mobile app.

1 Introduction

Over the last few years, communication services via the Internet are springing up at an amazing rate. The world is revolving around information which can be easily access to a massive load of information [1] through the Internet. The big data era [2], despite all its competitive advantages and extensive prospect, also comes with the disadvantage of information overload, making the search of precise information an overwhelming and time consuming task [3]. A personal recommender system [4] is considered as the effective way to overcome this problem by supporting people in daily choices and pointing out alternatives.

For the tourism suppliers, the Internet and mobile communication technology provides multiple ways to globally sell their products and services to potential tourists at any time. The suppliers can remotely manage their Web & WAP servers to display their marketing content at a high speed. For tourists, the Internet enables them to communicate directly with tourism suppliers to obtain information, and to search and purchase services at any time and any place online [5].Now recommender systems are partially applied in the travel industry helping tourists to make personalized vacation plans. Personal preferences of people and their interests play a decisive role in choosing their supplier considering the fact that on vacations. In such a customer sensitive industry the application of personal travel recommender systems become a necessity, not a luxury. This has become more important for survival of tourism companies for the fact that economic slowdowns have curtailed excursion budgets of people. Moreover, demand for package travel has declined and people prefer personalized trips, therefore recommender systems have attracted attention of business and academic community.

Conventional recommendation techniques, content-based filtering and collaborative filtering [6], for instance, are particularly well suited for the recommendation of quality and taste products such as books, movies, or pop music. However, for products from other categories such as cars, computers, apartments or tourist attractions these conventional approaches are not as efficient. The reasons are many, for example, tourist attractions are not visited very frequently which makes it rather infeasible to collect numerous ratings for one specific item. Ratings from users and similar data are required by collaborative recommendation algorithms. Furthermore, customers who use recommendation applications would not be satisfied with recommendations based on years-old item preferences.

The two major alternative knowledge-based recommender systems are called case-based and constraint-based recommendation systems [7]. In both approaches the principles of exploiting knowledge are similar: User requirements are collected, and recommendation results are explained. The major difference between them is in the method solutions are calculated. Knowledge-based approaches have the advantage of direct inclusion and mapping of expert's knowledge, for that neither a cold-start phase nor large user communities is required. As with all knowledge-based systems however, the development of such a knowledge intensive application requires some additional ramp-up effort in both knowledge acquisition and engineering, as well as in the development of an interactive user interface.

[a] Corresponding author: jdyjazz@gmail.com

Article available at http://www.matec-conferences.org or http://dx.doi.org/10.1051/matecconf/20164401005

An agent-based model [8] is one of a class of computational models for simulating the actions and interactions of autonomous agents with a view to assessing their effects on the system as a whole. In artificial intelligence, an intelligent agent is an autonomous entity which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals. Multi-agent based frameworks [9] are considered as potential effective solutions for tourism recommender system. A series of knowledge-based agents that include a knowledge base and an inference system can be parts of the multi-agent framework.

2 Architecture of the Agent framework

The framework proposed in the paper integrates Collector Agent, Tourist Agent, Content-based Agent, Collaborative Filtering-based Agent and Constraint-based Agent into a multi-agent system through the Internet to providing tourism recommender service. The design of this mechanism is based on three kinds of recommendation system [10], which can perform respective recommendation filtering algorithm. The framework can integrated the recommendations generated by the three Agents, and then presents the results to the tourist. The Framework is depicted in Figure 1.

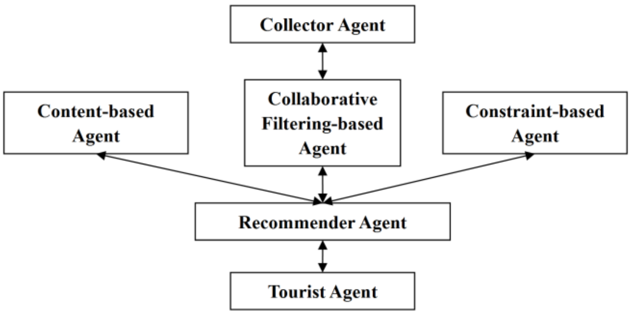

Figure 1. Agent framework for attraction recommendations.

3 Developments of the Agents

3.1 Collector Agent

The Collector Agent of the framework is consisted of some web crawlers [11] that follow hyperlinks throughout a Web site, retrieving and indexing pages to document the site for collecting tourist information. In other words, it is an intelligent topic crawler focusing on the tourist field. Based on the tourist ontology designed for parsing the unstructured Web, the Collector Agent can also discover emerging Web pages related to tourism on the Web, and extract the text information after downloading the Webpages.

3.2 Recommender Agent

As shown in Figure 1, Recommender Agent can generate recommender information by integrating the recommendations of Content-based Agent, Collaborative Filtering-based Agent and Constraint-based Agent. And then the recommender information can be transferred to the Tourist Agent. In order to make the performance more effective, linear combination method of data fusion [12] is applied.

Assumed that, the recommendations sent to the tourists is specified as $List_n$, where n is the amount of the total items. And $List_{n1}$ is the recommendation generated by Content-based Agent; And $List_{n2}$ is the recommendation generated by Collaborative Filtering-based Agent; And $List_{n3}$ is the recommendation generated by Constraint-based Agent. So that the final recommendation $List_n$ can be:

$$List_n = \alpha \cdot List_{n1} + \beta \cdot List_{n2} + \gamma \cdot List_{n3} \quad (1)$$

Where α , β and γ are weights, and they are set to be 1/3 initially.

Depending on the feedback of Tourist Agent, the three weights which stored in the Tourist Agent can be updated dynamically according to the preferences of the tourists.

Neighbour tourists generated mainly based on the similarity between each user. However, a larger number of users have become null and void, so the system needs to pre-process [13] the user data before generating neighbours. The pre-processing process consists of data cleaning, data integration, data conversion and data reduction. The system then can get a more streamlined and representative list of users which are usually often logged on and recently recorded users, which are recognized as active users. For users already have travel records, the system can use the visiting history of tourist attractions to calculate the similarities between users. For the user there is no travel records, the system can use the user's basic information (such as logging times, sex, professional, vehicle, etc.) as the basis for the calculation of similarity [14].

3.3 Content-based Agent

Items that can be recommended to the tourist are represented by a set of features, also called attributes or properties. For example, in a tourist attraction recommendation application, features adopted to describe a movie are: actors, directors, genres, subject matter. When each item is described by the same set of attributes, and there is a known set of values the attributes may take, the item is represented by means of structured data. In this case, many ML algorithms can be used to learn a user profile. In most content-based filtering systems, item descriptions are textual features extracted from Web pages, emails, news articles or product descriptions. Unlike structured data, there are no attributes with well-defined values. Textual features create a number of complications when learning a user profile, due to the natural language ambiguity. The problem is that traditional keyword-based profiles are unable to capture the semantics of user interests because they are primarily driven by a string matching operation. If a string, or some morphological variant, is found in both the profile and the document, a match is made and the document is considered as relevant. The result is that, due to synonymy, relevant information can be missed if the profile does not contain the exact keywords in the documents while, due to polysemy, wrong documents could be deemed relevant.

Most content-based recommender systems use relatively simple retrieval models, such as keyword matching or the Vector Space Model (VSM) with basic TF-IDF weighting [15]. VSM is a spatial representation of text documents. In that model, each document is represented by a vector in an n-dimensional space, where each dimension corresponds to a term from the overall vocabulary of a given document collection.

Formally, every document is represented as a vector of term weights, where each weight indicates the degree of association between the document and the term. Let $D=\{d_1,d_2,...,d_n\}$ denote a set of documents or corpus, and $T=\{t_1,t_2,...,t_n\}$ be the dictionary, that is to say the set of words in the corpus. T is obtained by applying some standard natural language processing operations, such as tokenization, stop words removal, and stemming [16]. Each document d_j is represented as a vector in an-dimensional vector space, so $d_j=\{w_{1j},w_{2j},...,d_{nj}\}$, where w_{kj} is the weight for term t_k in document d_j .

Document representation in the VSM raises two issues: weighting the terms and measuring the feature vector similarity. The most commonly used term weighting scheme, TF-IDF (Term Frequency-Inverse Document Frequency) weighting, is based on empirical observations regarding text :

- Rare terms are not less relevant than frequent terms (IDF assumption),
- Multiple occurrences of a term in a document are not less relevant than single occurrences (TF assumption),
- Long documents are not preferred to short documents (normalization assumption).

In other words, terms that occur frequently in one document (TF=term-frequency), but rarely in the rest of the corpus (IDF=inverse-document-frequency), are more likely to be relevant to the topic of the document. In addition, normalizing the resulting weight vectors prevent longer documents from having a better chance of retrieval. These assumptions are well exemplified by the TF-IDF function [17]:

$$TF-IDF(t_k,d_j)=TF(t_k,d_j)\cdot\log\frac{N}{n_k} \qquad (2)$$

Where N denotes the number of documents in the corpus, and nk denotes the number of documents in the collection in which the term tk occurs at least once.

$$TF(t_k,d_j)=\frac{f_{k,j}}{\max_z f_{z,j}} \qquad (3)$$

Where the maximum is computed over the frequencies fz,j of all terms tz that occur in document dj .

In order for the weights to fall in the [0,1] interval and for the documents to be represented by vectors of equal length, weights obtained by

Equation (2) is usually normalized by cosine normalization [18]:

$$w_{k,j}=\frac{TF-IDF(t_k,d_j)}{\sqrt{\sum_{s=1}^{|T|}TF-IDF(t_k,d_j)^2}} \qquad (4)$$

As stated earlier, a similarity measure is required to determine the closeness between two documents. Many similarity measures have been derived to describe the proximity of two vectors; among those measures, cosine similarity is the most widely used:

$$sim(d_i,d_j)=\frac{\sum_k w_{ki}\cdot w_{kj}}{\sqrt{\sum_k w_{ki}^{\;2}}\cdot\sqrt{\sum_k w_{kj}^{\;2}}} \qquad (5)$$

The Content-based agent is relying on VSM where as both tourist profiles and items are represented as weighted term vectors. Predictions of a tourists' interest in a particular item can be derived by computing the cosine similarity.

3.4 Collaborative filtering-based Agent

The starting point of collaborative filtering-based agent is: Suppose that tourists with similar interests should favourite to the same items as each other. So, as long as the maintenance of a database on the tourists' preference, the neighbour tourists with similar interests can be calculated by analysing the stored preference, and then it can be recommended to the tourist based on the neighbour tourists' interests.

On the basis of collaborative filtering principle, the recommendation process of tourist attractions can be divided into three steps [19].

- The representation of tourist information. Tourist visiting history of attractions needs to be analysed and modelled.
- The generation of neighbour tourists. The similarity of tourists can be computed according to the visiting history data and the collaborative filtering algorithm presented by us. A neighbour tourist list can be calculated on the basis of known similarities.
- The generation of attraction recommendations. Top-N attractions will recommended to the tourist according to the visiting history of his neighbours.

According to above steps, user's basic information and past travel history can be used to calculate the user list of neighbours which are recorded in the corresponding record in the user database. When users log into the system, the recommendations of tourist attractions can be presented based on the travel experiences of his neighbours.

The process of recommendation is shown in Figure 1. In order to increase the efficiency of the system, the system calculated users' neighbours when offline for each tourist, and calculated at regular intervals

Neighbour users generated mainly based on the similarity between each user. However, a larger number of users have become null and void, so the system needs to pre-process the user data before generating neighbours. The pre-processing process consists of data cleaning, data integration, data conversion and data reduction. The system then can get a more streamlined and representative list of users which are usually often logged on and recently recorded users, which are recognized as active users. For users already have travel records, the system can use the visiting history of tourist attractions to calculate the similarities between users. For the user there is no travel records, the system can use the user's basic information (such as logging times, sex, professional, vehicle, etc.) as the basis for the calculation of similarity [20].

Suppose that the set of all tourists $T=\{T_1, T_2...T_n\}$, for each tourist T_i (i=1, 2...n), the system can calculate the neighbours list including the top N tourists which similarity is higher than the given threshold θ.

The core of collaborative filtering algorithms is the calculation of users' similarity. The similarity between the tourist T_i and T_j is measured as follows: First the system needs to retrieval all the ratings of tourist T_i and T_j over the attractions, and then calculate the similarity between the tourist T_i and T_j through different similarity models, denoted as $sim(T_i,T_j)$. There are mainly three ways to measure the similarity between tourists, including Cosine method, Correlation similarity method and Adjusted Cosine method. We have adopted the Cosine method in this paper. Tourist ratings are viewed as a vector of n-dimensional term space. If the tourist has ever visited the attraction, the rating is set to 1, otherwise set to 0. And the cosine similarity between tourists is computed by the angle between vectors. The similarity between tourist T_i and T_j can be calculated by:

$$sim(\mathrm{T}_i,\mathrm{T}_j)=\frac{R_{T_i}\bullet R_{Tj}}{\left\|R_{T_i}\right\|\times\left\|R_{Tj}\right\|} \tag{6}$$

When computing recommendations for a particular tourist, the very basic approach is to select the objects favoured by other tourists that are similar to the target tourist. When new tourists enter the system, there is usually insufficient information to produce recommendation for them, because there is no visiting history of the new tourists. The usual solution of the cold start problem is similarity calculation between each user by profile information, such as age, sex, professional, vehicle, income, etc.

3.5 Constraint-based Agent

Main objective of constraint-based agent is to generate tourist attraction recommendations, and the task is viewed as a constraint satisfaction problem (CSP) [21]. Formally, a constraint satisfaction problem is defined as a triple of $\langle V,D,C\rangle$, where:

- $V = \{V1,\ldots,Vn\}$ is a set of variables,
- $D = \{D_1,\ldots, D_n\}$ is a set of the respect domains of values, and,
- $C = \{C_1,\ldots, C_m\}$ is a set of constraints.

Each variable V_i can take on the values in the nonempty domain D_i. Every constraint $C_j \in C$ is in turn a pair $\langle t_j,R_j\rangle$, where $t_j \subset X$ is a subset of k variables and R_j is an k-ary relation on the corresponding subset of domains d_j. An evaluation of the variables is a function from a subset of variables to a particular set of values in the corresponding subset of domains. An evaluation v satisfies a constraint $\langle t_j,R_j\rangle$ if the values assigned to the variables t_j satisfies the relation R_j.

An evaluation is consistent if it does not violate any of the constraints. An evaluation is complete if it includes all variables. An evaluation is a solution if it is consistent and complete; such an evaluation is said to solve the constraint satisfaction problem.

Constraint Satisfaction Solutions for constraint satisfaction problems are calculated on the basis of search algorithms that use different combinations of backtracking and constraint propagation.

For the particular issue of recommendation task, we could define [22]:

$V=V_C \cup V_P$, where V_C is a set of single-valued or multiple-valued variables that represent the tourists' requirement, and V_P is a variable set represent the properties of tourist attraction.

D contains the domain definitions for variables in V.

$C=C_P \cup C_I \cup C_F \cup C_O$, where C_P is the constraints set of the values in V_P that describe the available tourist attractions; C_I is the constraints (capturing tourist requirements) set of the values in V_C; C_F is the set of filters' constraints; C_O is the set of other constraints (inconsistency constraints).

3.6 Tourist Agent

The Tourist Agent mainly acts as the intermediary service provider between the framework and tourists.

The emergence of the World Wide Web and mobile networks has given even the most mundane of applications a spiffy GUI. The Tourist Agent provides webpage UI and mobile APP UI to the tourist.

Tourist Agent saves the profile, preferences, Agent weights, and the use history of the tourist. It can send browse and enquiry request of tourist information to the Recommender Agent. The Recommender Agent response the Tourist Agent by recommendations based on the linear combination calculation. The tourist can feedback or rate the result recommendations presented by the Tourist Agent UI.

4 Conclusions

Traditional technologies of classical recommender system application domains, such as collaborative filtering, content-based filtering and content-based filtering are effectively adopted in the framework. In the framework they are constructed as Agent that can generate recommendations respectively. Recommender Agent can generate recommender information by integrating the recommendations of Content-based Agent, Collaborative Filtering-based Agent and Constraint-based Agent. In order to make the performance more effective, linear combination method of data fusion is applied. User interface is provided by the Tourist Agent in form of webpages and mobile app.

The emergence of the World Wide Web and mobile networks has given even the most mundane of applications a spiffy GUI. The Tourist Agent provides webpage UI and mobile APP UI to the tourist.

References

1. WP Eveland Jr, S Dunwoody. Examining information processing on the World Wide Web using think aloud protocols. MEDIAPSYCHOLOGY, **2** (2000)
2. B Brown, M Chui, J Manyika. Are you ready for the era of 'big data'. McKinsey Quarterly, **11** (2011)
3. Tuan-Dung Cao, Quang-Minh Nguyen. Semantic approach to travel information search and itinerary recommendation. International Journal of Web Information Systems, **8**, 3 (2012)
4. Paolo Cremonesi, Franca Garzotto, Sara Negro et al.. Comparative evaluation of recommender system quality. CHI '11 Extended Abstracts on Human Factors in Computing Systems, (ACM, New York, 2011)
5. GI Alptekin, Gülçin Büyüközkan. An integrated case-based reasoning and MCDM system for Web based tourism destination planning. Expert Systems with Applications,**38**,3(2011)
6. Xiaoyuan Su, Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. Advances in Artificial Intelligence, (2009)
7. Adomavicius, G., Tuzhilin, A.. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, **17**, 6(2005)
8. Linden, G., Smith, B., York, J. Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing, (IEEE Computer Society Press, Los Alamitos, 2003)
9. Jonathan L. Herlocker, Joseph A. Konstan, John Riedl. Explaining collaborative filtering recommendations. Proceedings of the 2000 ACM conference on Computer, (ACM, New York, 2011)
10. W. R. Gilks and P. Wild. Adaptive Rejection Sampling for Gibbs Sampling. Journal of the Royal Statistical Society. Series C (Applied Statistics), **2**, 41 (1992)
11. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of Dimensionality Reduction in Recommender System - A Case Study. Technical Report, (2000)
12. Robin Burke. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, **4**, 12(2002)
13. Lu, E.H.-C. ,Chih-Yuan Lin, Tseng, V.S.. Trip-Mine: An Efficient Trip Planning Approach with Travel Time Constraints. 2011 12th IEEE International Conference on Mobile Data Management (MDM) (IEEE Computer Society Press, Los Alamitos, 2007)
14. G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper and M.Pinkerton. Cyberguide: A Mobile Context Aware Tour Guide. Wireless Networks, **5**, 3 (1997)
15. T. Horozov, N. Narasimhan, V. Vasudevan. Using Location for Personalized POI Recommendations in Mobile Environments. Proceedings of International Symposium on Applications on Internet, (IEEE Computer Society Press, Los Alamitos, 2006)
16. Y. Zheng, X. Xie. Learning Travel Recommendations from User-Generated GPS Traces. ACM Transactions on Intelligent Systems and Technologies, **2**, 2 (2011)
17. V.-W. Soo, S.-H. Liang. Recommending a Trip Plan by Negotiation with a Software Travel Agent. Cooperative Information Agents(Springer Berlin Heidelberg , 2001)
18. Makoto Yokoo. Constraint Satisfaction Problem. Distributed Constraint Satisfaction, Springer Series on Agent Technology,(Springer Berlin Heidelberg , 2001)
19. Dietmar Jannach, Markus Zanker, Matthias Fuchs. Constraint-based recommendation in tourism: A multi-perspective case study, Information Technology & Tourism, 11 (2009)
20. Tsang, E. Foundations of Constraint Satisfaction, (Academic Press, London and San Diego, 1993)
21. A. Felfernig, G. Friedrich, D. Jannach, M. Zanker. Developing Constraint-based Recommenders, Recommender Systems Handbook, (Springer Berlin Heidelberg , 2011)
22. Ricci, F., Mirzadeh, N., Bansal, M.. Supporting user query relaxation in a recommender system. In: Proceedings of the 5th International Conference in E-Commerce and WebTechnologies - EC-Web, 2004:31-40.WP Eveland Jr, S Dunwoody. Examining information processing on the World Wide Web using think aloud protocols. MEDIAPSYCHOLOGY, **2** (2000)