SERVICE RECOMMENDATION ON WIKI-WS PLATFORM

ANDRZEJ SOBECKI

Academic Computer Centre, Gdansk University of Technology
Narutowicza 11/12, 80-233 Gdansk, Poland

(received: 4 June 2015; revised: 6 July 2015;
accepted: 14 July 2015; published online: 1 October 2015)

Abstract: The article presents the issues of recommendation services for users. The commonly used solutions shown include known classes of recommendation systems and information about their area of use and the most frequently used algorithms. The issue of choosing the services has been described based on the Wiki-WS platform including a model of cooperation with the recommendation system. The conclusion from the analysis of the proposed model and available algorithms is that there is a need to create a hybrid service recommendation algorithm in order to fulfill the requirements of those using a service platform like Wiki-WS.

Keywords: recommendation system, Wiki-WS, web services, web platform, recommendation algorithms

1. Introduction

Every day each one of us is a participant in many processes in which specific activities are done. For some activities we are using applications that automate the process or support these activities. The available applications may solve the problem, basically by changing input data to output data, in accordance with an earlier agreed scenario, where it was also defined how to perform the calculations. The latest trend in applications is to use an application scenario to change requirements without using programming languages. Activities in the application scenario are done by components which are called services. The service is performing an activity on the provider server in accordance with an implemented algorithm. The advantage of this solution is user independence in choosing the way of performing the activities, limited by the range of available services. In the Internet the available services are described and grouped into catalogues which have a different functionality. The register is the simplest catalogue of services because it offers the ability of describing and a simple mechanism of searching the services based on keywords and belonging to the category. Another type of service catalogues are repositories which provide higher

functionality than the registers for f.ex. development and monitoring services, managing of the distribution process, contact of developers with service users. An example of a repository is the Wiki-WS platform [1] which, using a virtual work team, provides development of services in accordance with the Agile methodology. Wiki-WS is also a trade platform between providers and users which also provides a set of tools to manage publishing services on servers, including monitoring of the quality of published services. The Wiki-WS platform, similarly like a service register, can provide only simple searching based on keywords and belonging to the category. The searching process should be improved in order to increase the availability of services registered on the Wiki-WS platform. The proposed solution is to use a recommendation system which will detect the requirements of users and which will also recommend a set of services that are adequate to the searched requirement. Moreover, recommended services will be able to meet the required level of qualitywhich is specified in the application scenario. Recommendation systems [2, 3] are commonly used in popular Internet systems like Amazon[1], Youtube[2], Netflix[3], TripAdvisor[4], Google[5], IMDb[6], iTunes (Genius). Each of the recommendation systems uses a set of algorithms that have been selected and adjusted to the specific context and field of use. This article presents popular classes of recommendation systems and commonly used algorithms. We propose a model of cooperation between the recommendation system and the Wiki-WS platform, including basic requirements of the proposed recommendation system. Finally, we estimate the range of usefulness of existing recommendation algorithms for the proposed recommendation system and also a hybrid recommendation system is proposed.

2. Recommendation systems

The aim of the recommendation system (RS) is to find services that are potentially interesting to the user [4]. The recommendation systems are a subclass of systems that filter information and estimate the value of rating that the user would give to a service [5]. Information about the services and users is retrieved by appropriate algorithms from the research area of *Information Retrieval* [6]. To detect the services that meet the user requirements, solutions associated with the Data Mining research area [7, 8] were used. The recommendation system, independently from the used solutions, uses three types of objects:

1. http://amazon.com
2. http://youtube.com
3. http://netflix.com
4. http://tripadvisor.com
5. http://google.com
6. http://imdb.com

- Service model – description of the recommended item which, depending on the recommendation systems, has a different complexity. In general, the service model is called the service profile;
- User model – description of the user preferences in the structured form, adopted to the field of use. The user model is called the user profile;
- Relation – describes the compatibility of models: service to user, service to service and user to user.

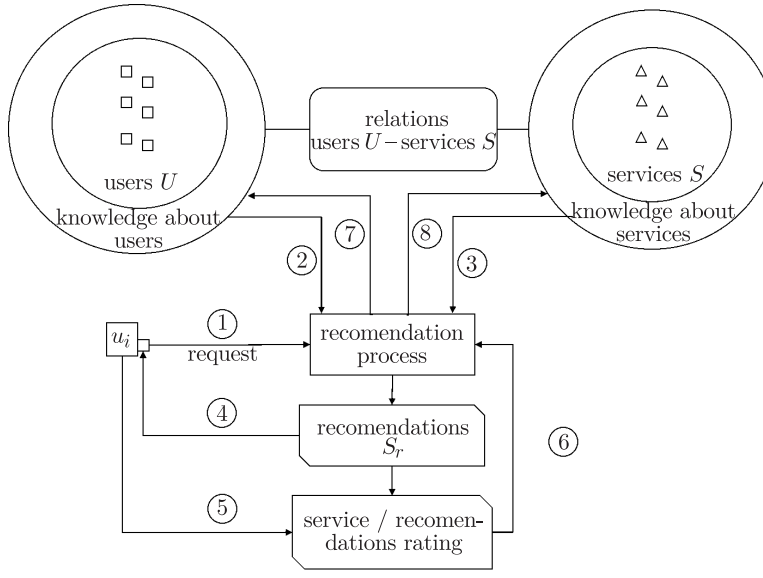

Figure 1. General scheme of the recommendation system

Methods used to create the above-mentioned objects depend on the recommendation system. The generalized flow diagram of the recommendation system is shown in Figure 1. The diagram shows the following operations (1) receiving request, (2) retrieving the user profile, (3) retrieving of the service profile, (4) preparing recommendations for the user profile, (5) the user's opinion about the recommendations, (6) interpretation of the user's opinion, (7) update of the user profile, (8) update of the service profile. The knowledge about users shown in the diagram presents the information collected in the user profiles. Knowledge about services is analogous information contained in the profiles of services. The recommendation systems are different because of the algorithm used in the recommendation process, including methods of creating and using knowledge about services and users. R. Burke in [4] has proposed a general classification of recommendation systems based on the used source of information. The interpretation of the mentioned classification is shown in Table 1. The table uses the following abbreviations:

- **CBR** (Content-Based Recommender) [9] – systems using similarity of content describing the services in order to find similar services to those accepted by the user;
- **KBR** (Knowledge-Based Recommender) [10] – systems using rules defined by the tools of knowledge engineering;
- **CFR** (Collaborative-Filtering Recommender) [11] – systems using similarity of opinions defined by the users;
- **ComBR** (Community-Based Recommender) [12] – services are recommended based on the results of the analysis of social relations of users in the system;
- **DBR** (Demographic-Based Recommender) [13] – systems using similarity of demographic data that describe users.

Table 1. Recommendation system classification based on the source of information used

<table>
<tr><th>Information source</th><th>Description of the needs of a user defined in the request</th><th>Ratings given by the user</th><th>Demographic information about the user</th></tr>
<tr><td>Expert domain knowledge</td><td rowspan="2">KBR</td><td>–</td><td>–</td></tr>
<tr><td>Information about services</td><td>CBR</td><td>–</td></tr>
<tr><td>Information about ratings</td><td>–</td><td>CFR</td><td rowspan="2">DBR</td></tr>
<tr><td>Demographic information</td><td>–</td><td>–</td></tr>
<tr><td>Information about social relationships</td><td>–</td><td>ComR</td><td>–</td></tr>
</table>

The above-mentioned classes of recommendation systems are characterized in Table 2.

In addition to these classes a hybrid recommendation system that uses a combination of algorithm and sources of information from each of the mentioned-above classes also stands out.

Regardless of the RS class in the recommendation process, some kind of algorithm or a set of algorithms are used in order to find the services compliance with the user requirements. Popular algorithms used in RS are divided into two groups:

- Memory-based – in order to predict the value of ratings the algorithm uses all known relations; the accuracy of prediction depends on the size of the relation collections;
- Model-based – information about user behavior is transformed into a model describing his/her preference; the created model is used to estimate the level of compatibility between the services and the user.

Table 2. Simplified characterization of recommendation systems

RS class	Feature	Description
CFR	Principle of recommendation	Recommendation for active user services evaluated positively by other users with similar tastes
	Usage areas	Popular recommendation services used and evaluated by multiple users
	Used knowledge	Opinions indicating compliance services and user requirements (user-user relations)
CBR	Principle of recommendation	Recommendation services similar to those accepted by a user in the past
	Usage areas	Recommendation services described in a format allowing automatic feature extraction (*e.g.* text format, XML, JSON)
	Used knowledge	Representations of services and user profiles as a set of weighting factors and features extracted from the content describing the service
KBR	Principle of recommendation	Use of domain knowledge including individual meaning of attributes describing the service, and the rules for comparing attributes in order to adjust the function used to calculate similarity of services to user preferences
	Usage areas	Recommendation of rare or unpopular services for which there is a problem with collecting the required number of opinions/ratings and services that are described with many attributes that differentiate and require domain knowledge
	Used knowledge	Domain knowledge; User requirement for each attribute; Set of criteria
ComBR	Principle of recommendation	Recommendation for the active user, services rated positively or recommended by other users in the relationship (family, friends, co-workers)
	Usage areas	Services comprehensible and usable for a specified community; Detection of relationships
	Used knowledge	Relationships between users and their opinions
DBR	Principle of recommendation	Recommendation for active user services rated positively by a user with similar demographic features (age, education, localization, number of children)
	Usage areas	Services usable for specified group: trainings, delivery of food, entertainment, insurance
	Used knowledge	Demographic data and opinions of users

A list of the algorithms used in the most popular classes of recommendation systems is shown in Table 3 [14]. Different algorithms based on the history and the model are analyzed in [15].

These algorithms use information contained in the user and service profiles and information about their relationship. The range of available information depends on the field and context of use of the recommendation system. The selection of the algorithm depends also on the user needs including the defined set

Table 3. Division of algorithms used in popular recommendation systems

RS class	Used algorithm	
	History-based	Model-based
CBR	TF-IDF Clustering	Bayes classifier Clustering Decision tree Neural network
CFR	k-NN Clustering Graph theory	Bayesian network Clustering Neural network Linear regression Probabilistic models
HR	Linear combination of predicted ratings Different patterns of voting Concluding one component as part of heuristics in another	Including one component like an element of the other model Construction of a unified model

of criteria for services. In order to specify requirements for the recommendation system to the Wiki-WS platform, it should define a model of cooperation and describe users and their requirements.

3. The cooperation model between the recommendation system and the Wiki-WS platform

The traditional process of the service selection by the user comprises the following steps: (1) reduce the set of service candidates and (2) manually verify the compatibility of the service and undefined criteria. In the first step, the user uses the available tools that allow him/her to define the criteria in order to automatically filter inconsistent services. In the second step, the user manually verifies the service candidacy from the collection returned from the first step. Verification enables one to eliminate services inconsistent with the requirements that were not defined by the user in the first step because there were not enough tools. The presented service selection process is characterized by low effectiveness and usefulness, because it only in a small part supports the user and every time requires him/her to define the set of requirements. Due to the simplicity of the implementation, the shown service selection process is available in every service register.

The presented service selection process is insufficient in the case of platforms such as Wiki-WS because the platform is used by users which can generally be described as the service providers and the service customers. The service providers include: (1) developers who create new services using a programming language and (2) scenario developers who create application scenarios using the

existing services. Service developers expect from the recommendation system to find similar services in order to compare their solutions. Scenario developers are a service provider and a service customer at the same time because they create new services (scenarios) using the existing services. Scenario developers expect from a recommendation system to find services compatible with a specified set of attributes and a specific goal *e.g.*: (1) find services on the same level of quality, (2) replace the indicated service into the scenario of a service that is cheaper/faster/more reliable, (3) improve the quality of the scenario (*e.g.* the level of reliability, availability, *etc.*), (4) lower the cost of the scenario, (5) reduce the working time of the scenario. Depending on the purpose chosen by the user, the selection process should filter the set of services presented to the user in a different way. The service consumers expect also to automate the above process of introducing their search criteria. In this case, user preferences should be automatically read from him/her past opinions. Any other criteria entered manually by the user should automatically replace the detected user preferences.

The proposed model of cooperation, the Wiki-WS platform and the RS are shown in Figure 2.

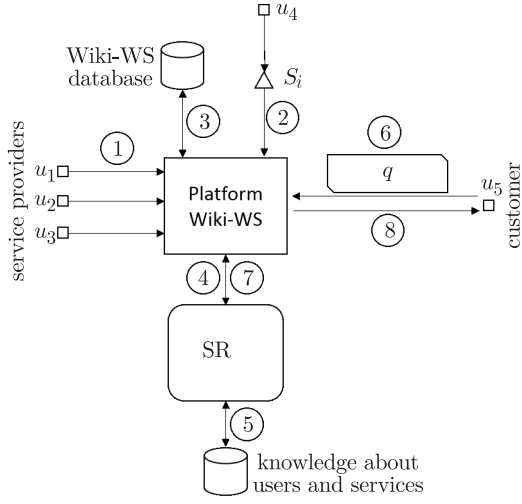

Figure 2. General diagram of cooperation of Wiki-WS with the recommendation system

The following steps are shown in the diagram: (1) the registration/modification service by the service provider u_1, u_2, u_3 on the Wiki-WS platform, (2) the use of service s_i by user u_4, (3) saving information in the Wiki-WS database, (4) forwarding information about services and users into the recommendation system, (5) saving/reading the knowledge about services and users, (6) request for finding the service for user u_5 using criteria q, (7) forwarding the request from Wiki-WS to the recommendation system and receiving recommendations, (8) presentation of personalized recommendations for the user.

In order to realize the proposed model of cooperation we should create a recommendation system that specifies a set of user requirements based on the users' opinions about the used services and composed scenarios. The recommendation system should find services compatible with the detected requirements. Also, in order to optimize, the system should select only the set of services that is recommended by other users and based on their opinions. Achieving the objectives of the proposed model requires algorithms with the following features: (1) detecting the preferences of users, (2) clustering users and services and (3) computing the compliance of the service attributes and the user needs.

To fulfill the above defined requirements it is necessary to create a hybrid algorithm that will consist of algorithms selected from Table 3 and will be similar to the solution presented in the dissertation [16].

4. Conclusion

The selection of the algorithm for a recommendation system depends on the field of use because it is determined by the available knowledge about the services and the users. In fact, there are many different recommendation systems *e.g.* Youtube, Netflix, TripAdvisor, Google. This article highlights the need to create a hybrid recommendation algorithm that would expand the range of functionality of the existing service recommendation algorithms. The versatility of the solution in the specified fields of use depends on the range of information that is transformed into the knowledge which is then used by the recommendation system. Information about the services that is available on the Wiki-WS platform includes: service description in natural language, cost attributes, keywords, QoS attributes. These attributes create a general description of the service. This description is understandable in most service platforms. Developing a system that conforms with the generalized service description will provide the proposed system with the compatibility and the possibility of working with other service platforms like KASKADA [17] and similar.

Acknowledgements

NIWA Centre of Competence. A project co-funded by the European Union from the European Regional Development Fund within the Innovative Economy Programme, "Subsidy for innovation" Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland.

References

[1] Krawczyk H and Downar M 2012 *Commonly Accessible Web Service Platform – Wiki-WS*, Intelligent Tools for Building a Scientific Information Platform, Springer 251
[2] Goldberg G, Nichols D, Oki B M and Terry D 1992 *Communication ACM* **35** 61
[3] Mahmood T and Ricci F 2009 *Improving recommender systems with adaptive conversational strategies*, ACM 73
[4] Burke R 2007 *Hybrid web recommender systems*, *The Adaptive Web* 377
[5] Ricci F, Rokach L and Shapira B 2011 *Introduction to Recommender Systems Handbook*, Springer 1

[6] Frakes W 1992 *Information Retrieval Data Structures and Algorithms*, Prentice-Hall
[7] Fayyad U, Piatetsky-Shapiro G and Smyth P 1996 *American Assocation for Artifical Intelligence Magazine* 37
[8] Witten I H and Eibe F 2005 *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann
[9] Pazzani M J and Billsus D 2007 *Content-based recommendation systems*, The adaptive web, Springer 325
[10] Bridge D, Göker M, McGinty L and Smyth B 2006 *The Knowledge Engineering review* **3** (20) 315
[11] Schafer J B, Frankowski D, Herlocker J and Sen S 2007 *Collaborative filtering recommender systems*, *The Adaptive Web, Springer* 291
[12] Kamahara J, Asakawa T, Shimojo S and Miyahara H 2005 *A community-based recommendation system to reveal unexpected interests*, Proceedings of the 11th International Multimedia Modelling Conference (MMM 2005) 433
[13] Pazzani M J 1999 *Artificial Intelligence Review* **13** (5–6) 393
[14] Adomavicius G and Tuzhilin A 2005 *IEEE Transactions on Knowledge and Data Engineering* **17** (6) 734
[15] Herlocker J L, Konstan J A, Terveen L G and Riedl J T 2004 *ACM Transactionall Information Systems* **22** (1) 5
[16] Sobecki A 2015 *Hybrydowy system rekomendujący usługi do scenariuszy aplikacji wszechobecnych*, PhD Dissertation
[17] Krawczyk H and Proficz J 2010 *Kaskada-multimedia processing platform architecture*, Proceedings of the 2010 International Conference on Signal Processing and Multimedia Applications (SIGMAP) 26