J Ambient Intell Human Comput (2016) 7:889–900
DOI 10.1007/s12652-016-0357-4

ORIGINAL RESEARCH

Testing and evaluating recommendation algorithms in internet of things

Ibrahim Mashal[1] · **Osama Alsaryrah**[1] · **Tein-Yaw Chung**[1]

Received: 5 November 2015 / Accepted: 20 February 2016 / Published online: 7 March 2016

Abstract Technological revolution in communication and embedded computing has led to the Internet of Things (IoT) where all objects are connected together to provide users with services. Nowadays, many third party service providers are providing a large number of IoT services. Suggesting suitable services to IoT users based on objects they own has not been tackled yet. In this paper, we investigate the possibilities of leveraging recommendation algorithms, especially graph-based, to IoT. We propose a graph-based model for IoT systems and conduct experiment in which analyze and explore correlations between performances of different algorithms. We show that the graph-based recommendation algorithm can be used to develop an effective recommender system for the IoT. Moreover, we show that some algorithms perform reasonably well and produce high quality results.

Keywords Internet of Things · Recommender system · Tripartite graph · Collaborative filtering · Internet of things recommender system · Hyper-edge

1 Introduction

Internet of Things (IoT) has emerged as a widely recognized paradigm that aim to build a global network to connect virtual world with physical world to provide human useful services (Atzori et al. 2010; Mashal et al. 2015a). The IoT integrates technologies such as RFID, wireless sensor networks and many other underlying technologies. More specific, the IoT attempts to connect uniquely identified and addressed objects to Internet based on standard communication protocols. Examples of things include smartphones, power meters, heart beat monitors, temperature meters, and various sensors that are equipped with processor, memory, and storage. Analysts predict that the number of interconnected objects will reach 212 billion by 2020 (Al-Fuqaha et al. 2015).

Nowadays, third party service providers offer IoT based applications and deliver innovative and valuable services. Third party value-added IoT services can be published, repositioned, resold or bundled with other services by another service provider. For instance, TELUS has launched its IoT marketplace in December 2014 featuring 75 different services and solutions (Telus). TELUS services range from smart restaurants to intelligent stores to connected farms. The TELUS IoT Marketplace enables users to quickly identify and request a TELUS-approved IoT solution to be added with monthly charge. Another example of IoT marketplace is Libelium (Asin 2011) which has listed 54 IoT applications grouped by 12 vertical markets including: urban and remote environments, agriculture and farming, water quality, security and emergencies, retail, logistics, domestic automation and e-health.

As time goes on, more and more smart objects will be connected and countless services will be introduced. Many users will subscribe or own more of these value-added services in the near future. This causes users to encounter unprecedented difficulties in finding ideal service from the overwhelming large number of services based on the smart objects they own. To resolve this complex problem, recommender systems (RSs) offer an effective solution (Bobadilla et al. 2013; Lu et al. 2015).

✉ Osama Alsaryrah
osanaji@yahoo.com

Tein-Yaw Chung
csdchung@saturn.yzu.edu.tw

[1] Innovation Center for Big Data and Digital Convergence, Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan

RSs are software system that analyzes information about items, users, and interactions between them in order to recommend the most suitable items to users by predicting his interest in a particular item. RSs have demonstrated its effectiveness in different domains, especially in the e-commerce domains. However, it is not examined and fully studied for the IoT. This paper is the first promising step towards addressing the problem of recommending IoT third party services to users and developing a novel service recommendation in the IoT. In this paper, we investigate and evaluate recommendation algorithms in the IoT. Through experimental analysis of recommendation algorithms, we critically compare the performance of different algorithms. We model IoT systems by a tripartite graph with hyper-edges among users, objects, and services. Then we formalize the service recommendation problem as a link prediction problem based on a graph approach and we analyze various entities and heterogeneous relationships to uncover correlations between objects, services, and users. After that, we present Internet of Things Service Recommender System (IoTSRS) by considering different relations in the IoT. Finally, we implement various existing recommendation algorithms and their combinations to study their behavior in IoT service recommendation. We evaluate the usefulness of existing recommending algorithms on IoT service recommendation based on the some well-known metrics such as recall and precision. The results illustrate that existing schemes worked decently but further extension is required to meet the challenges of IoT service recommendation. To the best of our knowledge, our work is novel and there hasn't been any prior study on designing service recommender system in the IoT.

The rest of the paper is organized as follows. Section II introduces related work and recommender system preliminary. In Section III, we highlight the special characteristics of the IoT and the challenges face when developing IoTSRS. Section IV depicts the general system architecture. Section V presents a motivating example to illustrate the urgent need for IoTSRS. Section VI introduces the proposed tripartite graph-based model that employs information of services and ambient objects for services recommendation. Furthermore, we give formal representation of the IoT. Section VII introduces formal model of the IoTSRS algorithm. Section VIII describes in detail our dataset and the evaluation metrics used to validate IoTSRS. Preliminary results and evaluations are given in Section IX. Finally, Section X concludes the paper.

2 Related work

Traditional RSs use different techniques include content-based filtering (CB), collaborative filtering (CF), and hybrid system techniques. As its name suggests, CB requires textual information about the items and historical records of the user (Lops et al. 2011). CB recommends items similar to the items that the active user has previously consumed or liked. It is based on description of item characteristics and a profile of the user's preference. However, it requires a mechanism to associate content to many heterogeneous networked objects. Moreover, in CB filtering, only very similar items to previous items consumed by the user are recommended which creates a problem of overspecialization (Pazzani and Billsus 2007). In the IoT, analysis of objects and services content may not be a good idea since it will cost complex and expensive computation of similarity and take a long time.

In contrast to the CB filtering, CF does not rely on items representations and its content (Su and Khoshgoftaar 2009). Instead, it relies on the opinions of other people who share similar interests or have similar tastes. CF requires additional rating system to capture and store users' ratings. It has been found that CF is a powerful technique that produces high quality recommendations with high levels of accuracy. Fundamentally, CF can be classified into user-based and item-based approaches (Lü et al. 2012). In user-based CF, the K most similar users which have similar rating are found, and then use the ratings from those users to calculate a prediction for the active user. Items-based takes into account the similarity between items themselves not the users based on ratings given by those users who have rated both items (Sarwar et al. 2001). There are many approaches to find similarity, the most popular similarity metrics are Pearson Correlation Coefficient (PCC) and Vector Space Similarity (VSS). PCC often achieves better performance than VSS (Xue et al. 2005).

Traditional CF techniques are sound and have been successfully applied in many different domains. However, in the context of the IoT, it encounters many challenges. CF requires an up-to-date dataset of users and their preference, which is difficult to gather for large number of objects. Moreover, computing similarity between every pair of users or services may take too much time and thus cannot make decision within acceptable time. Furthermore, traditional CF algorithms consider all services to compute services' rating similarities while most of them are different to the target service.

To achieve higher performance and overcome the drawbacks of aforementioned techniques, hybrid technique has been proposed (Burke 2002). Hybrid RSs combine collaborative and content information by one of seven basic hybridization mechanisms to combines features of two or more recommendation techniques.

In recommender systems, graph-based approach has been used in many different domains. For example, tag recommenders construct a graph with users, resources and tags and recommend a set of tags for a given user based on previously used and assigned tags (Zhang et al. 2011; Zhou et al. 2012). Numbers of meaningful research works on tag recommendation have been proposed in recent years (Milicevic et al. 2010). A well-known tag recommender approach is the FolkRank (FR) (Hotho et al. 2006; Jäschke et al. 2008) which adapts the Google PageRank algorithm to rank the nodes within a graph based on their importance in the network. A different mechanism is based on the classical CF approach that has been adopted for tags recommendation in (Hamouda and Wanas 2011; Marinho and Schmidt-Thieme 2008). Collaborative approaches exploit the relations between users, resources and tags of the folksonomy graph to select the set of recommended tags. Jäscke et al. (Jäschke et al. 2007) evaluate and compare user-based collaborative filtering, graph-based, and counting co-occurrences algorithms for tag recommendation. Other studies focus on using different approaches such as Rendle et al. (Rendle and Schmidt-Thieme 2010); Wetzker et al. (Wetzker et al. 2010), Lops et al. (Lops et al. 2013), and Rawashdeh et al. (Rawashdeh et al. 2013). Rawashdeh et al. applied the Katz measure to weighted, undirected tripartite graph to provide tag recommendations for individual users.

Research work on recommendation for the IoT is in its infancy. To the best of our knowledge, there are very few articles that discussed recommendation in the IoT environment, however, not in much detail. For example, the work in (Yao et al. 2016; Yao et al. 2014) addressed things recommendation in the IoT. In particular, they propose a framework to recommend the right thing to use at the specific time by exploring users' relations and things correlations. However, they do not consider recommending third party services. Compared with this research effort, we construct the services-things correlation graph capturing similarities between services.

In general, semantic models and semantic technologies (i.e. ontologies) play a role in service specification and discovery. In particular, ontologies such as OWL-S has been used in the past to semantically represent services (Martin et al. 2005) and several approaches have been defined to address service discovery hybridizing techniques of information retrieval and description logic (Fenza et al. 2008; Klusch et al. 2009). Analogously, in the area of the IoT, SSN ontology (Compton et al. 2012; Janowicz and Compton 2010) and IoT service modeling (De et al. 2011) may be used to better address IoT service recommendation. An interesting approach was presented in (Fenza et al. 2012) where the authors present an architecture for context-aware service discovery in a healthcare scenario that exploits synergy among intelligent agent technology, semantic web models and computational intelligence techniques. The architecture emphasizes the need of synergic approach between ontology and fuzzy logic to model the user's context. Nevertheless, semantic models and semantic technologies are beyond the scope of this paper.

3 IoT characteristics

As mentioned before, the IoT is a network of physical objects that feature an IP address for internet connectivity between objects and other Internet-enabled devices and systems. The IoT has special characteristics that add more complexity and more challenges to the IoT applications and services. First, the IoT is highly heterogeneous network that deploys large number of different types of objects and implement wide variety of different wireless technologies and communication protocols. Even though IoT objects may have similar functionalities, they still have unique properties. Second, the IoT must enforce interoperability not only between heterogeneous technologies but also between various services. Moreover, it must enforce interoperability between human society and physical objects. Third, the IoT communication is real-time communication that supports data analysis and exchange between applications and services. Fourth, the IoT infrastructure is deployed in many geographically distributed places and the services are Location-Based Services (LBS). Fifth, as a consequence of the wide geographic distribution and the huge number of objects in the IoT, scalability is a critical issue that must be enforced in the IoT systems. Sixth, another source of complexity in the IoT is mobility. Mobility allows objects to be organized in changing structures. Thus, the IoT systems must support reliable services delivering between moving objects.

The aforementioned characteristics make some applications suffer from some shortcomings and may become incapable of achieving efficient tasks in the IoT. Correspondingly, IoTSRS faces new challenges and is much more complex than traditional recommender systems for many reasons:

- IoTSRS must be able to analyze the heterogeneous relationships between different objects and services which add more complexity to IoTSRS. These relationships will require comprehensive analysis and identification in order to produce accurate recommendation results.
- IoTSRS must incorporate contextual information, such as time and place. However, contextual information is hard to specify. For instance, applying the semantic technologies to the IoT is a challenge since the resource-constrained nature of the IoT imposes many constraints that must be taken into consideration. Similarly, IoT services descriptions are usually incomplete and ambiguous.

4 Architecture of the IoTSRS

Multimedia social web of things (MUL-SWoT) is a lightweight RESTful platform for developing the IoT and SWoT applications and services (Chung et al. 2014; Chung et al. 2013). MUL-SWoT establishes connection between users and their smart objects to allow smart objects to communicate with Social Networks Sites (SNS). The platform collects, stores, and analyzes data from users' smart objects and then posts the data on users' SNS. Thus, it allows users to share objects and services with their friends and people they know. The MUL-SWoT design includes number of modules and sub-modules, as shown in Fig. 1.

The first component is the RESTful API module, which represents and provides the 3rd party services provider with interface to access MUL-SWoT functions. The security module is a critical part of the MUL-SWoT because it implements the authentication, authorization, and accounting functionalities (AAA). The SNS API module provides access to social networks and imports clients' relations and profiles. The SNSmanagement module imports users' friends list and their profiles from SNSs.Finally, the publisher module posts messages to users' SNSs.

The core component of the platform is the MUL-SWoT management module that manages all other components. MUL-SWoT management module is divided into a number of sub-modules: Communication manager sub-module handles all communications through the MUL-SWoT. Configuration registry sub-module manages and stores all configurations about services and objects. Device management sub-module is responsible for managing all objects registered on the MUL-SWoT. Control and monitor sub-module controls objects and services and monitors their actions based on stored configurations.

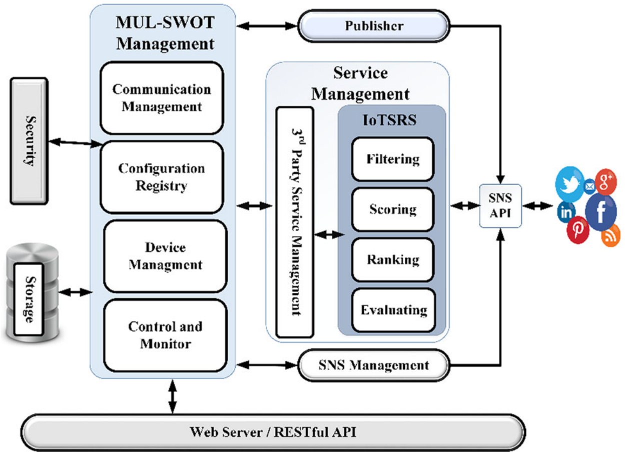

Fig. 1 MUL-SWoT architecture

An important component of MUL-SWoT is the service management module which is responsible for managing and storing information about service. Two sub-modules are found in the service management, namely Third party service management and IoTSRS. Third party service management sub-module controls and manages third party accounts and all services registered on the MUL-SWoT. Moreover, it generates service ID for the third party service provider.

IoTSRS is a sub-module of the service management module. The aim of this module is to assess which are the best services and returns top-ranked services that best matching users' needs based on things they own. The recommender runs a large number of recommendation algorithms and is implemented through a four-step process consists of filtering, scoring, ranking, and evaluating. Filtering excludes services that do not match users' needs. Scoring assigns a numeric value to each service. Ranking returns an ordered list of services based on filtering and scoring results. Evaluating uses standard Information Retrieval (IR) metrics to evaluate recommended services and recommendation algorithms. Based on the result from evaluation component, the top three recommendation algorithms that have the highest performance are then combined and integrated in attempt to produce more accurate and useful recommendations.

5 Motivating example

To illustrate the importance of IoTSRS, we give the following example. Consider a restaurant and food producers scenario where the owner Bob wants to monitor food conditions in order to comply with the food safety regulations. Traditionally, Bob and the employees collect manually readings about temperature in refrigerators, freezers, and ovens to decide whether food is safe or not. In order to improve the accuracy and reduce time, Bob installed numbers of sensors to measure and report the refrigerator temperature, air temperature, food temperature, and humidity. However, this is not enough to satisfy his needs. Without a powerful service that can collect and store these readings to be analyzed, Bob cannot be sure of his business and that his restaurant reputation is protected. Bob asks suggestions to our IoTSRS about possible services that can make use of the objects he owns. IoTSRS recommends him a SafeFood services offered by blueRover Inc. which is best meets his needs. SafeFood enables Bob to monitor real-time information about food safety status and receive a real-time notifications and alerts on their smart phones or any other devices if anything goes wrong (e.g., a cold storage failure).Thus, SafeFood creates a safer work environment for employees, and safe food for customers.

6 IoT formal model and problem definition

IoTSRS must draw users' attentions to new services they do not subscribe yet. These services must be suitable for the objects owned by the users. Intuitively, objects may be used by more than one service, which mean that the service that shares some objects should not be excluded from recommendation. In order to develop IoTSRS the first step is to create a model of the IoT.

In the IoT, the relation between users, objects, and services can be modeled as a tripartite graph with hyper-edges between them as show in Fig. 2. A tripartite graph is a graph with its vertices partitioned into three disjoint sets: $U = \{U_1, U_2, \ldots, U_m\}$ denote a set of m users, $O = \{O_1, O_2, \ldots, O_n\}$ denote a set of n objects, and $S = \{S_1, S_2, \ldots, S_k\}$ denote a set of k services (Mashal et al. 2015b). We define Y as a ternary relation between these three components that represent users' subscription of services based on the objects they own. Y is defined in Eq. 1.

$$Y \subseteq \{u, o, s{:}u \in U,\ o \in O, s \in S\} \tag{1}$$

Thus, the IoT systems can be defined as a tuple that describes the users U, services S, objects O, and the ternary relation between them. The tuple is given in Eq. 2.

$$I = (U; S{:}\ O{:}\ Y) \tag{2}$$

The hyper-graph can be projected into three bipartite graphs that represent relations between objects-services (OS), users-services (US), and users-objects (UO). The relation between objects and services (OS) is given in Eq. 3, where the weights represent the number of users who subscribe service with their object.

$$OS(o, s) = |\{y = u, o, s \in Y : u \in U\}| \tag{3}$$

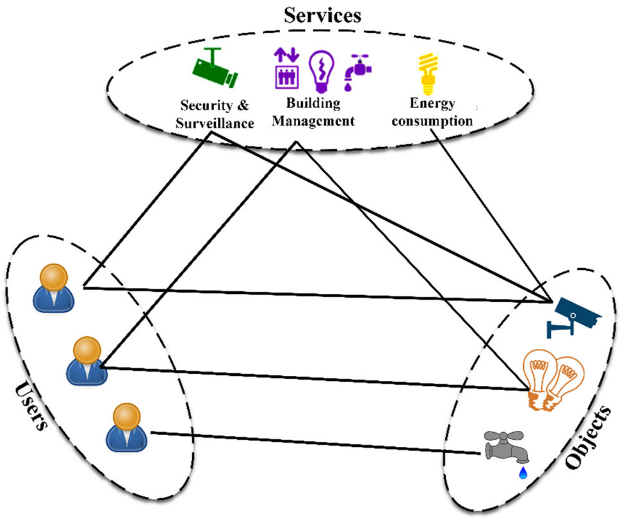

Fig. 2 Service recommendation based on a set of users who have subscribed services

The relation between users and services (US) is given in Eq. 4, where the weights represent the number of objects used to subscribe a service.

$$US(u, s) = |\{y = u, o, s \in Y : o \in O\}| \tag{4}$$

The relation between users and objects (UO) is given in Eq. 5, where the weights represent the number of services share the same objects for user.

$$UO(\mathrm{u}, o) = |\{y = u, o, s \in Y : s \in S\}| \tag{5}$$

7 Internet of things service recommender systems (IoTSRS)

The task of IoTSRS is to rank all services provided by the 3rd party, filter services, and recommend a set of services $s \in S$ for a given user $u \in U$ and a given object $o \in O$. It takes a set of users, a set of objects, and a set of services and outputs a list of ranked services. IoTSRS implements several existing recommendation algorithms developed for recommendation in tripartite graph. In this subsection we review these techniques.

7.1 Individual service recommendation algorithms

1. *Most-Popular-Service* (*MPS*) This approach is the most basic approach that recommends most popular services to users. For any user $u \in U$ and any object $o \in O$ the same set of services S(u, o) is recommended. This set of services is weighted by count frequency in all service subscription Ys. MPS is given in Eq. 6.

$$S(u,o) = \underset{s \in S}{\mathrm{argmax}}(|Y_s|) \tag{6}$$

2. *Most-Popular-Service-User* (*MPSU*) It suggests the most frequent service within the services user subscribed regardless of the objects. It is personalized extension of MPS. We define MPSU as in Eq. 7.

$$S(u,o) = \underset{s \in S}{\mathrm{argmax}}(|Y_{s,u}|) \tag{7}$$

3. *Most-Popular-Service-Object* (*MPSO*) Suggest services that are most frequent with a particular object regardless of the users. We define MPSO as in Eq. 8.

$$S(u,o) = \underset{s \in S}{\mathrm{argmax}}(|Y_{s,o}|) \tag{8}$$

4. *Most-Popular-Service-User-Object* (*MPSUO*) This algorithm mixes the Most-Popular-Service-User with the Most-Popular-Service-Object. MPSUO is given in Eq. 9.

$$S(u,o) = \underset{s \in S}{\mathrm{argmax}}\left(\beta|Y_{s,u}| + (1-\beta)|Y_{s,o}\right) \tag{9}$$

where β used to balance the influence of both components; the user and the object components.

5. *ServRank* (*SR*) ServRank is inspired by the famous algorithm called FolkRank. ServRank computes preference vector from the graph $G = (V, E)$, where $\mathrm{V = U \cup S \cup O}$ is the set of all vertices in the graph, which is composed of users, objects, and services. E is the set of the edges in the graph, which is defined by the three projections of the hyper-graph OS, UO, and US presented in the previous sub-section. ServRank vector is given by Eq. 10.

$$\mathrm{w = dAw + (1{-}d)p} \tag{10}$$

where A is the normalized column-stochastic version of the adjacency matrix w of graph G, p is the preference vector, and the dampening factor $0 < \mathrm{d} \leq 1$ determines the influence of p. The ServRank vector is taken as a difference between two computations: one with a preference vector and one without the preference vector. The ServRank vector is defines in Eq. 11.

$$\mathrm{w = w_1{-}w_0.} \tag{11}$$

6. *User-Based Collaborative Filtering* (*UBCF*) IoTSRS also implements the commonly used algorithm CF. However, traditional CF cannot be applied directly and must be modified to cope with the ternary relations in the IoT tripartite graph. CF is based on finding similarity between users u and v using different similarity matrix. In IoTSRS, user u is modeled as vector over set of services where the weight is the projection $US(u, s)$ and neighbors N_u^k of a user u are formed based on the set of services in the user profile Y_u and only the subset V_s of users that have subscribed a service for active object o are taken into account when calculating the user neighborhood. Several techniques are used to calculate similarity. In IoTSRS, Jaccard's similarity is used. The set of n recommended services can then be determined based on this neighborhood as given in Eq. 12.

$$\mathrm{S(u,o)} = \underset{\mathrm{s \in S}}{\mathrm{argmax}} \left(\sum_{\mathrm{v \in N_u^k}} \mathrm{sim(Y_u, Y_v)} * \delta\mathrm{(v,o,s)} \right) \tag{12}$$

where $\delta(v, o, s) = \begin{cases} 1 & \mathit{if}(v, o, s) \in Y_s \\ 0 & \mathit{else} \end{cases}$

UBCF scales well with very large datasets since UBCF considers only those users that have subscribed same services with similar objects and thus the number of similarities to calculate is drastically reduced and thus reducing computation burden. However, the algorithm may not be able to recommend some services since user's neighbors did not subscribe these services.

7. *Object-Based Collaborative Filtering* (*OBCF*) In IoTSRS, objects o is modeled as vector over set of services where the weight is the projection $OS(o, s)$. When a user selects an object to subscribe a service, the cosine similarity between this object and every object is calculated and neighbors N_o^k are then constructed. We define OBCF in Eq. 13.

$$\mathrm{S(u,o)} = \underset{\mathrm{s \in S}}{\mathrm{argmax}} \left(\sum_{\mathrm{j \in N_o^k}} \mathrm{sim(Y_i, Y_j)} * \delta\mathrm{(u,j,s)} \right) \tag{13}$$

where $\delta(u, j, s) = \begin{cases} 1 & \mathit{if}(u, j, s) \in Y_s \\ 0 & \mathit{else} \end{cases}$

7.2 Hybrid service recommendation algorithm (HSR)

Some of the aforementioned algorithms do not perform well when it runs alone. Combining different algorithms together will produce an effective and enhanced recommender algorithm that outperforms its constituent parts. IoTSRS combines several individual recommenders together to produce hybrid recommenders using two different hybrid methods: Weighted Average and Simple Multiplication.

1. *Weighted Average Hybrid Service Recommendation* (*WAHSR*) In WAHSR, The top three individual algorithms are combined together to get three WAHSR. Individual recommender algorithms are combined on weighted hybrid recommenders where each model is trained separately. WAHSR final result is normalized of each recommendation algorithm on the scale of 1. The WAHSR is given in Eq. 14.

$$\mathrm{S(u, o)} = \underset{\mathrm{s \in S}}{\mathrm{argmax}}(\beta * \mathrm{S_a(u, o)} + \alpha * \mathrm{S_b(u, o)}) \tag{14}$$

where β and α are used to control the contribution of the two recommenders where $\beta = 1 - \alpha$. Intuitively, When α is set to 0, recommender a acts alone and recommender b will have no effect on the final result. In the case that α is set to 0.5, each recommender contributes equally to the final result.

2. *Simple Multiplication Hybrid Service Recommendation* (*SMHSR*) The top three individual algorithms are combined together again but this time with different tactic to get three SMHSR. Simple multiplication is a simple method that multiplies the scores of different individual algorithms and recommends service with highest scores after combination. SMHSR is given in Eq. 15.

$$\mathrm{S(u, o)} = \underset{\mathrm{s \in S}}{\mathrm{argmax}}(\mathrm{S_a(u, o)} * \mathrm{S_b(u, o)}) \tag{15}$$

7.3 Experimental setup

In this section we describe in detail the dataset, the evaluation method and the metrics used to validate IoTSRS. The code we used for IoTSRS implementation is based on the open-source code provided by TagRec (Trattner et al. 2015).

7.4 Dataset

To perform reliable experiments, it is ideal to use a large scale datasets which are important for effective evaluation of recommendation algorithms. Unfortunately, there is a lack of publicly accessible well-known large scale database for IoT that contains information about users, objects, IoT services, and user subscription in services. The acquisition of datasets for testing recommendation algorithms used in IoTSRS is a challenge task. First, we create service profiles to comprise 80 services and 100 objects which are used in those services. This information collected from Libelium, TELUS, and blueRover catalogs from their websites.

For users' objects and services they subscribe, we attempted to use real-world data. Data was collected from 60 individual users. Each participant was asked to select the number of objects he owns, or wishes to own from an objects list. In addition, the users were asked to arbitrarily select IoT services they are interested in from the default list. Unfortunately, the IoT is still in its infancy stage where IoT objects and services are not commonly used among people, which has a severe influence on collected data and the debate on accuracy of the results. For example, only fewer than one in 10 of those surveyed said they know what the IoT is. Moreover, 97 percent of those surveyed responded with Smart Phone for the question "What smart objects you own?", while only 1.5 percent responded with Smart TV. When it came to the question "What smart objects you wish to own in the near future", 95 percent of those surveyed responded with smart blood pressure, smart refrigerator, and smart bulb. The results showed that those surveyed think that eHealth and Smart Home are the most important services among all IoT services.

As a consequence, we write a computer program to generate randomly dataset contains information about users and objects they own and services they subscribe. The dataset contains 1000 users, each user is assigned a different number of objects with minimum of one and maximum of 9 objects. Based on service profiles, some objects are assigned with 12 services while other objects are assigned with only two services. We followed a standard procedure in recommender research and divided the dataset as follows: 80 % for training set which is used to generate recommendation lists and 20 % for the test set which is used to verify the quality of the recommendations.

7.5 Evaluation method

To evaluate prediction quality and performance of different algorithms implemented in IoTSRS, we compared the top recommended services using a set of various well-known evaluation metrics as follows.

- *Recall* (*R*) Is a metric for completeness of recommendation result. Recall is calculated as the number of correctly recommended services divided by the number of relevant services. Recall is defined in Eq. 16.

$$R@K = \frac{1}{|U|}\sum_{u\in U}\left(\frac{|S_u^k \cap S_u|}{|S_u|}\right) \tag{16}$$

  where s_u^k denotes the top k recommended services and S_u the list of relevant services of user $u \in U$.
- *Precision (P):* Is a metric for exactness of the recommendation results that is calculated as the number of correctly recommended services divided by the number of recommended services. Precision is defined in Eq. 17.

$$P@K = \frac{1}{|U|}\sum_{u\in U}\left(\frac{|S_u^k \cap S_u|}{|S_u^k|}\right) \tag{17}$$

- *F-measure* Is another metric to compare the performance of algorithms. It is a measure of a test's accuracy that combines precision and recall into one score. F-measure is defined in Eq. 18.

$$F - measure = \frac{1}{|U|}\sum_{u\in U}\left(2 \times \frac{P@K \times R@K}{P@K + R@K}\right) \tag{18}$$

- *Mean Reciprocal Rank* (*MRR*) Is the sum of the reciprocal ranks of all relevant services in the list of the recommended services. This means that a higher MRR is achieved if the relevant services occur at the beginning of the recommended list. MRR is defined in Eq. 19.

$$\mathrm{MRR} = \frac{1}{\mathrm{|U|}}\sum_{\mathrm{u=1}}^{\mathrm{|U|}}\left(\frac{1}{\mathrm{|S_u|}}\sum_{s\in S_{\mathrm{u}}}\frac{1}{\mathrm{rank(s)}}\right) \tag{19}$$

- *Mean Average Precision* (*MAP*) Is an extension of the precision metric that also looks on the ranking of the recommended services. MAP is defined in Eq. 20.

$$MAP = \frac{1}{|U|}\sum_{u=1}^{|U|}\left(\frac{1}{|S_u|}\sum_{k=1}^{||S_u^k||} B_k \times P@K\right) \tag{20}$$

  where B_k is 1 if the recommended service at position k is relevant.
- *Discounted Cumulative Gain* (*DCG*) In DCG, the gain of a recommended item is discounted logarithmically

with respect to its position in the recommendation list (Yang et al. 2014). The DCG of a list of k services is defined as in Eq. 21.

$$\mathrm{DCG@K(u)} = \sum_{j=1}^{K} \frac{g_{u,s(j)}}{\max(1, \log_b j)} \tag{21}$$

where $g_{u,s(j)}$ is the gain of user u when service s is recommended. denotes the j-th service in the recommendations ordered list, and the logarithmic base b is suggested to be 2 to ensure all positions are discounted. A normalized version of DCG, called NDCG is given in Eq. 22.

$$\mathrm{NDCG@K(u)} = \frac{\mathrm{DCG@K(u)}}{\mathrm{DCG^{*}@K(u)}} \tag{22}$$

where $\mathrm{DCG^{*}@K(u)}$ is the ideal $\mathrm{DCG@K(u)}$

8 Experimental result

In order to validate the efficiency and evaluate IoTSRS, we run a number of experiments. The preliminary results of our evaluation of the dataset are based on Precision/Recall plots. To perform experiment, several variables are tuned. First, we fix the neighborhood number K in UBCF and OBCF to 20. This value was chosen after extensive experimentation of k incrementally from 0. Recall and precision both suffer from diminishing returns as k increases beyond 20. Second, for the SR algorithm the parameter d was set to 0.7 and the number of iterations was set to 10. Finally, β was set to 0.5 in MPSUO.

8.1 Individual algorithm performance evaluation

The preliminary results of our evaluation of the dataset are based on Precision/Recall plots. Figure 3 reports the Precision/Recall of the individual recommendation algorithms.

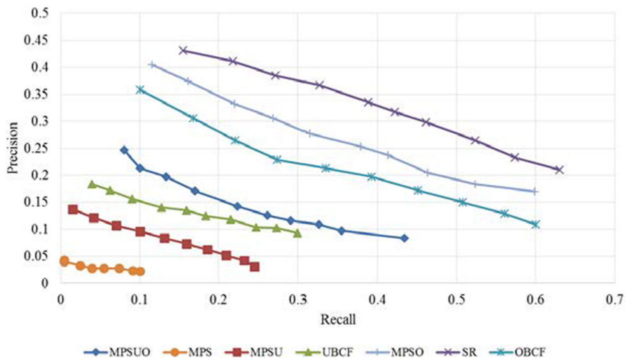

Fig. 3 Recall/precision *plots* for individual recommendation algorithms in IoTSRS

Comparing individual recommendation algorithms, SR is the strongest individual recommender with the highest level of Precision/Recall estimation. The second most successful recommender algorithm is MPSO, meaning that frequency-based algorithms perform better when they consider objects. MPSO is able to provide relevant recommendations specially when there is no much information available about users and services they subscribed. MPSO outperforms all other frequency-based algorithms. OBCF comes third of the individual recommender algorithms. OBCF performs far better that UBCF, which reveals that services are better modeled by objects than by users. Other algorithms perform moderately well. However, all the recommender algorithms perform far better than the un-personalized MPS algorithm which simply employs the popularity of a service.

IoT characteristics have a great impact on the performance of recommender algorithms. It is clear that not all algorithms can cope with IoT characteristics. As mentioned before, IoT is highly heterogeneous network that deploys a large number of different types of objects and produces a large number of different services. Thus, IoTSRS must analyze the heterogeneous relationships between different users, objects, and services. As can be seen in Fig. 3, SR performance is the best due to its nature which makes it the most suitable for the IoT systems. It is intuitive that SR achieves more accurate predictions and more reasonable recommendation than other algorithms. SR is a powerful tripartite graph tool that is able to capture, identify, and comprehensively analyze the complex relations between nodes in the IoT systems.

Moreover, SR handles another critical feature of the IoT system. The IoT produces vast amount of data and services from a huge number of objects. Recommendation algorithms based on the graph model has an advantage on solving the sparsity problem with both high accuracy and little computation time. The SR algorithm in IoTSRS effectively mitigates the sparsity problem by creating more paths between users, objects, and services nodes.

Another reason for the superiority of SR is its independency of contextual information. In the IoT, the resource-constrained nature imposes many constraints that make contextual information hard to specify, even with the use of semantic technologies. Moreover, the IoT services descriptions are usually incomplete and ambiguous.

Table 1 shows a comparison of results for MRR, MAP, and NDCG for both K = 5 and 10. The results clearly show that both SR and MPSO algorithms outperform all other individual algorithms.

8.2 HSR performance evaluation

HSR combines the best three individual recommender algorithms earlier discovered. SR, MPSO, and OBCF are

Table 1 Comparison of results for MRR, MAP, and NDCG

Algorithm	Measure					
	MRR@5	MAP@5	NDCG@5	MRR@10	MAP@10	NDCG@10
MPS	0.086788	0.09399	0.141666	0.102341	0.118519	0.193908
MPSU	0.230302	0.261417	0.335579	0.248704	0.300707	0.394514
UBCF	0.267604	0.294258	0.353345	0.271717	0.305076	0.367613
MPSUO	0.504488	0.570987	0.653029	0.528267	0.631598	0.734624
OBCF	0.533274	0.63664	0.719554	0.551486	0.692901	0.785377
MPSO	0.569914	0.702872	0.780856	0.582504	0.749283	0.829517
SR	0.574332	0.704967	0.789098	0.58843	0.75125	0.832668

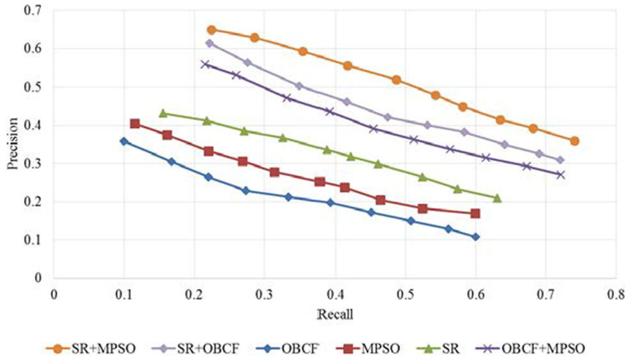

Fig. 4 Recall/precision *plots* for WAHSR algorithms

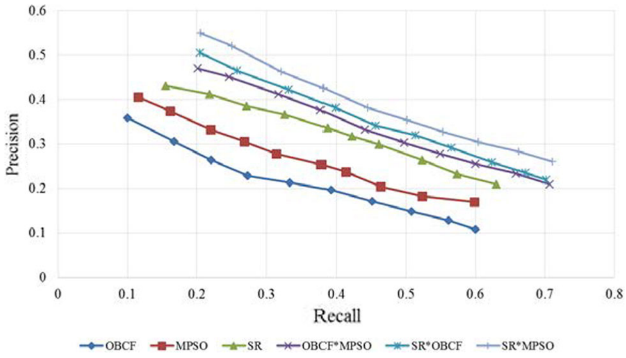

Fig. 5 Recall/precision *plots* for SMHSR algorithms

well represented in the HSR algorithms. Figure 4 shows the results from WAHSR, we see that the three WAHSR algorithms dramatically surpass the individual components. We evaluated the weights α and β in 0.05 increments to consider every possible combination. For OBCF + MPSO WAHSR best results were found when α = 0.20 and β = 0.80, meaning that MPSO contributes 80 % of the WAHSR and OBCF contributes 20 %. Best results for SR + OBCF were found when α = 0.85 and β = 0.15, meaning that SR contributes 85 % of the hybrid model and OBCF contributes 15 %. When looking at their Precision/recall, OBCF + MPSO and SR + OBCF achieve nearly identical results. This is due the fact that OBCF is the worst algorithm among these three. OBCF not only contributes little in the WAHSR, but also degrades the overall result of WAHSR. SR + MPSO achieves the best performance of the WAHSR when α and β were set to 70 and 30 % for SR and MPSO, respectively. SR + MPSO adds to the strength of SR the strength of the frequency-based algorithms which is based on frequency counting and thus achieve higher score.

The same results are introduced from SMHSR. Again, the three SMHSR algorithms surpass their individual components. The performance of both OBCF*MPSO and SR*OBCF was nearly identical. And they continue to lag behind the SR*MPSO. The results of SMHSR are shown in Fig. 5.

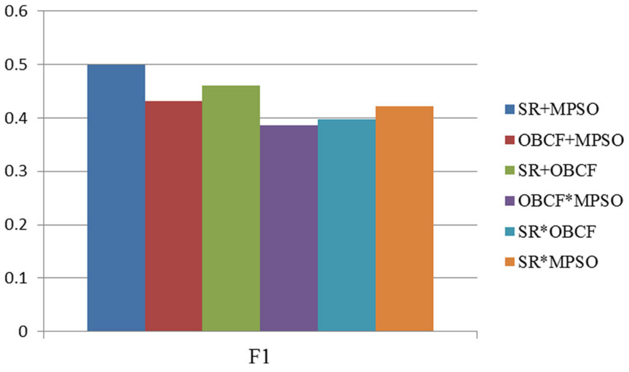

Fig. 6 WAHSR Vs SMHSR

Comparing WAHSR to SMHSR shows that, in general, WAHSR outperforms SMHSR for most α and β settings. It is worthy to notice that OBCF + MPSO in its best performance barely outperforms SR*MPSO. For most α and β settings, SR*MPSO outperforms OBCF + MPSO which emphasize the importance of SR and its impact on HSR. Figure 6 shows the comparison of F1for WAHSR and SMHSR.

As shown in these figures, SR + MPSO shows a clear edge over other HSR both in WAHSR and SMHSR on precision and recall. More specific, SR + MPSO achieves overall improvement on recall with less than 1 %

Table 2 Comparison of Recall among different algorithms

Recall	SR + MPSO	SR + OBCF	OBCF + MPSO	SR*MPSO	SR*OBCF	OBCF*MPSO	SR	MPSO	OBCF
SR + MPSO	–	0.0084	0.0444	0.0670	0.0467	0.0763	0.2462	0.4308	0.3710
SR + OBCF	–	–	0.0356	0.0580	0.0378	0.0672	0.2357	0.4187	0.3594
OBCF + MPSO	–	–	–	0.0215	0.0021	0.0304	0.1931	0.3698	0.3126
SR*MPSO	–	–	–	–	0.0190	0.0087	0.1679	0.3409	0.2849
SR*OBCF	–	–	–	–	–	0.0282	0.1906	0.3670	0.3098
OBCF*MPSO	–	–	–	–	–	–	0.1578	0.3293	0.2738

Table 3 Comparison of Precision among different algorithms

Precision	SR + MPSO	SR + OBCF	OBCF + MPSO	SR*MPSO	SR*OBCF	OBCF*MPSO	SR	MPSO	OBCF
SR + MPSO	–	0.1650	0.2708	0.3037	0.4663	0.5196	0.5502	0.8378	1.3725
SR + OBCF	–	–	0.0908	0.1190	0.2586	0.3044	0.3306	0.5774	1.0364
OBCF + MPSO	–	–	–	0.0258	0.1538	0.1957	0.2198	0.4461	0.8668
SR*MPSO	–	–	–	–	0.1247	0.1656	0.1891	0.4096	0.8198
SR*OBCF	–	–	–	–	–	0.0363	0.0572	0.2533	0.6180
OBCF*MPSO	–	–	–	–	–	–	0.0201	0.2093	0.5612

compared to SR + OBCF and more than 4.45 % compared to MPSO + OBCF. Comparing precision, the SR + MPSO precision rises by more than 16.5 and 27 % compared to SR + OBCF and OBCF + MPSO, respectively. Such improvements prove the effectiveness of SR + MPSO over other WAHSR and over the individual recommendations algorithms. Moreover, an overall improvement in performance is reported comparing WAHSR with SMHSR. A complete comparison of overall recall of these different algorithms is given in Table 2 while overall precision comparison is given in Table 3.

Apparently the precision and recall would be different with different number of recommended services. Figure 7 presents the relation between recall and the number of recommended services. To illustrate the influence of the number of recommended services on various recommendation algorithms, we take the recall as an example. For the SR + MPSO, when the service number changes from two to ten, the recall increases about 1.8 times. While for the other WAHSR algorithms the increment is between 1.6 and 1.7 times for MPSO + OBCF and SR + OBCF. Such difference implies that SR + MPSO is more complete than the other WAHSR algorithms. SMHSR algorithms have higher increment in recall values when moving from two to ten recommended services. The increment is 1.9, 1.8, and 1.7 times for OBCF*MPSO, SR*MPSO, and SR*OBCF, respectively. However, the SMHSR precision degrades when moving from two to ten recommended services which implies that they suffer from imprecision of recommendations results.

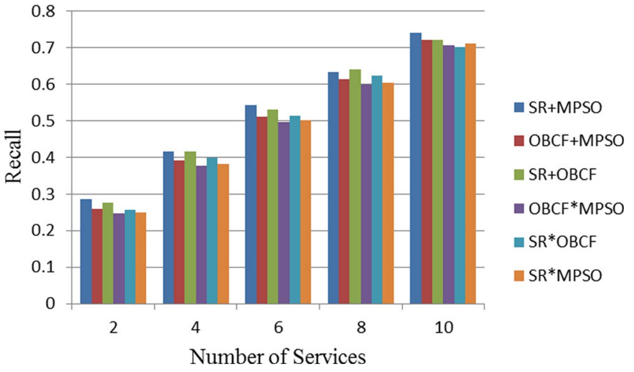

Fig. 7 Recall for WAHSR and SMHSR against different number of recommended services

From the experimental results and its analysis we can see that WAHSR recommendation algorithms, especially SR + MPSO, not only outperform the individual recommendation algorithms, both frequency-based algorithms and the more effective graph-based algorithms, but also the SMHSR algorithms. However, both WAHSR and SMHSR algorithms are computationally efficient and thus are suitable for deployment in large networks such as the IoT.

9 Conclusion

Recommendation of services in the IoT is crucial to facilitate popularity of the IoT. In this paper we examined the possibilities of leveraging graph-based recommender system to the IoT. We have proposed a graph based model for the IoT systems and we have illustrated how graph-based recommendation algorithms can be extended to recommend services in the IoT. The results of our experiments not only show that techniques used for the tripartite graph recommendation can be used to develop an effective recommender system for the IoT, but also show that some algorithms perform reasonably well and produce high quality results. At present, one limitation of IoTSRS is that it is based on simple widely used algorithms. Moreover, the graph model presented cannot depict some important features of the IoT services, such as incremental service, mobile sensors, etc. In future, we plan to develop more powerful model and improve IoTSRS to make more accurate recommendations by extending existing algorithms. Moreover, we also plan to study performance evaluation of IoTSRS by qualitative investigation of user satisfaction.

Acknowledgments This work was supported by the Ministry of Science and Technology of Republic of China, Taiwan, under contract number MOST 104-2221-E-155-012.

References

Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols and applications. IEEE commun surv tutor 17(4): 2347–2376. doi:10.1109/COMST.2015.2444095

Asin A (2011) Smart cities from libelium allows systems integrators to monitor noise, pollution, structural health and waste management. http://www.libelium.com/smart_cities/. Accessed Feb 2016

Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54:2787–2805. doi:10.1016/j.comnet.2010.05.010

Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowledge-Based Sys 46:109–132. doi:10.1016/j.knosys.2013.03.012

Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User Adap Inter 12:331–370. doi:10.1023/A:1021240730564

Chung TY, Mashal I, Alsaryrah O, Huy V, Kuo WH, Agrawal DP (2013) Social web of things: a survey. In: Parallel and Distributed Systems (ICPADS), 2013 International Conference on, 15–18 Dec. 2013. pp 570–575. doi:10.1109/ICPADS.2013.102

Chung TY, Mashal I, Alsaryrah O, Chang CH, Hsu TH, Li PS, Kuo WH (2014) MUL-SWoT: a social web of things platform for internet of things application development. In: Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing(CPSCom), IEEE, 1–3. pp 296–299. doi:10.1109/iThings.2014.53

Compton M et al (2012) The SSN ontology of the W3C semantic sensor network incubator group. Web Semant Sci Serv Agents World Wide Web 17:25–32. doi:10.1016/j.websem.2012.05.003

De S, Barnaghi P, Bauer M, Meissner S (2011) Service modelling for the internet of things. In: Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, 18–21 Sept. 2011. pp 949–955

Fenza G, Loia V, Senatore S (2008) A hybrid approach to semantic web services matchmaking. Int J Approx Reason 48:808–828. doi:10.1016/j.ijar.2008.01.005

Fenza G, Furno D, Loia V (2012) Hybrid approach for context-aware service discovery in healthcare domain. J Comput Syst Sci 78:1232–1247. doi:10.1016/j.jcss.2011.10.011

Hamouda S, Wanas N (2011) PUT-Tag: personalized user-centric tag recommendation for social bookmarking systems. Soc Netw Anal Min 1:377–385. doi:10.1007/s13278-011-0028-6

Hotho A, Jäschke R, Schmitz C, Stumme G (2006) Information retrieval in folksonomies: search and ranking. In: Sure Y, Domingue J (eds) The semantic web: research and applications, vol 4011. Lecture notes in computer science. Springer, Berlin, pp 411–426. doi:10.1007/11762256_31

Janowicz K and Compton M (2010) The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. Paper presented at the Proceedings of the 3rd International Conference on Semantic Sensor Networks, Vol 668. Shanghai

Jäschke R, Marinho L, Hotho A, Schmidt-Thieme L, Stumme G (2007) Tag Recommendations in folksonomies. In: Kok J, Koronacki J, Lopez de Mantaras R, Matwin S, Mladenič D, Skowron A (eds) Knowledge discovery in databases: PKDD 2007, vol 4702. Lecture Notes in Computer Science. Springer, Berlin, pp 506–514. doi:10.1007/978-3-540-74976-9_52

Jäschke R, Marinho LB, Hotho A, Schmidt-Thieme L, Stumme G (2008) Tag recommendations in social bookmarking systems. Ai Commun 21:231–247

Klusch M, Fries B, Sycara K (2009) OWLS-MX: a hybrid semantic web service matchmaker for OWL-S services. Web Semany Sci Serv Agents World Wide Web 7:121–133. doi:10.1016/j.websem.2008.10.001

Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender systems handbook. Springer, Berlin, pp 73–105

Lops P, de Gemmis M, Semeraro G, Musto C, Narducci F (2013) Content-based and collaborative techniques for tag recommendation: an empirical evaluation. J Intell Inf Syst 40:41–61. doi:10.1007/s10844-012-0215-6

Lu J, Wu D, Mao M, Wang W, Zhang G (2015) Recommender system application developments: a survey. Decis Support Syst 74:12–32. doi:10.1016/j.dss.2015.03.008

Lü L, Medo M, Yeung CH, Zhang YC, Zhang ZK, Zhou T (2012) Recommender systems. Phys Reports 519:1–49. doi:10.1016/j.physrep.2012.02.006

Marinho L, Schmidt-Thieme L (2008) Collaborative tag recommendations. In: Preisach C, Burkhardt H, Schmidt-Thieme L, Decker R (eds) Data analysis, machine learning and applications. Studies in classification, data analysis, and knowledge organization. Springer, Berlin, pp 533–540. doi:10.1007/978-3-540-78246-9_63

Martin D et al. (2005) Bringing semantics to web services: the OWL-S approach. In: Cardoso J, Sheth A (eds) Semantic web services and web process composition: First International Workshop, SWSWPC 2004, San Diego, CA, USA, July 6, 2004, Revised Selected Papers. Springer, Heidelberg, pp 26–42. doi:10.1007/978-3-540-30581-1_4

Mashal I, Alsaryrah O, Chung TY, Yang CZ, Kuo WH, Agrawal DP (2015a) Choices for interaction with things on Internet and underlying issues. Ad Hoc Netw 28:68–90. doi:10.1016/j.adhoc.2014.12.006

Mashal I, Chung TY, Alsaryrah O (2015) Toward service recommendation in internet of things. In: Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on, 7–10 July 2015. pp 328–331. doi:10.1109/ICUFN.2015.7182559

Milicevic A, Nanopoulos A, Ivanovic M (2010) Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. Artif Intell Rev 33:187–209. doi:10.1007/s10462-009-9153-2

Pazzani M and Billsus D (2007) Content-based recommendation systems. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The Adaptive Web, vol 4321. Lecture Notes in Computer Science. Springer, Berlin, pp 325–341. doi:10.1007/978-3-540-72079-9_10

Rawashdeh M, Kim HN, Alja'am J, El Saddik A (2013) Folksonomy link prediction based on a tripartite graph for tag recommendation. J Intell Inf Syst 40:307–325. doi:10.1007/s10844-012-0227-2

Rendle S and Schmidt-Thieme L (2010) Pairwise interaction tensor factorization for personalized tag recommendation. Paper presented at the Proceedings of the third ACM international conference on Web search and data mining, New York

Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. Paper presented at the Proceedings of the 10th international conference on World Wide Web, Hong Kong

Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:1–19. doi:10.1155/2009/421425

Trattner C, Kowald D, Lacic E (2015) TagRec: towards a toolkit for reproducible evaluation and development of tag-based recommender algorithms SIGWEB Newsl:1–10 doi:10.1145/2719943.2719946

Wetzker R, Zimmermann C, Bauckhage C, Albayrak S (2010) I tag, you tag: translating tags for advanced user models. Paper presented at the Proceedings of the third ACM international conference on Web search and data mining, New York, New York

Xue GR, Lin C, Yang Q, Xi W, Zeng HJ, Yu Y, Chen Z (2005) Scalable collaborative filtering using cluster-based smoothing. Paper presented at the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador

Yang X, Guo Y, Liu Y, Steck H (2014) A survey of collaborative filtering based social recommender systems. Comput Commun 41:1–10. doi:10.1016/j.comcom.2013.06.009

Yao L, Sheng QZ, Ngu AHH, Ashman H, Li X (2014) Exploring recommendations in internet of things. Paper presented at the Proceedings of the 37th international ACM SIGIR conference on Research; development in information retrieval, Gold Coast, Queensland

Yao L, Sheng QZ, Ngu AH, Li X (2016) Things of interest recommendation by leveraging heterogeneous relations in the internet of things. ACM Trans Internet Technol (**to appear**)

Zhang ZK, Zhou T, Zhang YC (2011) Tag-Aware recommender systems: a state-of-the-art survey. J Comput Sci Technol 26:767–777. doi:10.1007/s11390-011-0176-1

Zhou X, Xu Y, Li Y, Josang A, Cox C (2012) The state-of-the-art in personalized recommender systems for social networking. Artif Intell Rev 37:119–132. doi:10.1007/s10462-011-9222-1