J. Myanmar Acad. Arts Sci. 2021 **Vol. XIX. No.2**

KNOWLEDGE-BASED RECOMMENDATION SYSTEM FOR TRAVEL DESTINATION

Tin Nilar Win[1], Wint Pa Pa Kyaw[2]

Abstract

Recommendation systems are software tools and techniques providing suggestions for items to be of interest to a user and can help to overcome the problem of information overload. Tourism is a primary application area for mobile applications to support the travellers before, during and after the travel. A mobile based travel recommendation system can effectively help the customers to find their trip destination according to their interests. Knowledge-based systems generate recommendations using specific domain knowledge about how certain item features meet user preferences, and ultimately, how the item is of interest to the user. A specific type of knowledge-based recommenders are constraint-based recommenders. In this research, the system suggests personalized travel locations to a user by exploiting predefined knowledge bases that contain explicit rules about how to relate customer requirements with item features based on knowledge-based filtering method.

Keywords: Recommendation System, Constraint-based, Knowledge-based Filtering

Introduction

The explosive growth of online environments has made the issue of information search and selection increasingly cumbersome; users are overwhelmed by options which they may not have the time or knowledge to assess. Recommendation systems have proven to be a valuable tool for online users to cope with the information overload. Recommendation systems use details of registered user profiles and habits of the whole user community to compare available information items against reference characteristics in order to present item recommendations. Typically, a recommendation system compares a user profile to some reference attributes and seeks to predict the 'rating' or 'preference' that a user would give to an item she has not yet considered.

Recommendation systems originally found success on e-commerce web sites to present information on items and products that are likely to be of interest to the user (e.g. films, books, news, web pages etc). Lately, they have been increasingly employed in the field of electronic tourism (e-tourism), providing services like trip and activities advisory, lists of points of interest (POIs) that match user preferences, recommendations of tourist packages, etc. Existing recommendation systems in e-tourism typically emulate services offered by tourist agents where prospective tourists refer to seeking advice for tourist destinations under certain time and budget constraints. The users typically state their needs, interests and constraints based upon selected parameters. The system then correlates user choices with catalogued destinations annotated using the same vector of parameters.

A relatively recent development in e-tourism lies in the use of mobile devices as a primary platform for information access, giving rise to the field of mobile tourism. In the tourism field, travel recommendation systems aim to match the characteristics of tourism and leisure resources or attractions with the user needs. The travel companies have to aware of these preferences from different tourists and serve more attractive packages to get more business and profit. Therefore, the demand for intelligent tour services, from both travellers and tour companies, is expected to increase dramatically. The unique characteristics of mobile tourism bring forward new challenges and opportunities for the evolution of innovative personalized

[1] Dr, Assistant Lecturer, Department of ICT, Co-operative University (Thanlyin)
[2] Dr, Associate Professor, Department of Computer Studies, University of Yang

services which have no place in the field of e-tourism. Since recommendation systems have been successfully applied to enhance the quality of service for customers in a number of fields, it is natural direction to develop knowledge-based recommendation system for travel destination.

Recommendation Filtering Methods

The use of efficient and accurate recommendation method is very important for a system that will provide good and useful recommendation to its individual users. Based on a particular prediction method, recommendation systems have been classified into three main categories. These are collaborative filtering, content-based filtering and knowledge-based filtering methods.

(1) In Content-Based Filtering, where the system makes use of the user’s profile to recommend items that exhibit similar characteristics to what he has liked in the past.

(2) In Collaborative Filtering, the recommendation compares the user’s past ratings with those of other users to find users with similar taste. Highly rated items by these neighbors will be recommended.

(3) Knowledge-Based recommendations make use of domain specific information to match user interests with items.

Knowledge-Based Filtering

Knowledge-based recommendation systems are particularly useful in the context of items that are not purchased very often. Examples include items such as real estate, automobiles, tourism requests, financial services, or expensive luxury goods. In such cases, sufficient ratings may not be available for the recommendation process. As the items are bought rarely, and with different types of detailed options, it is difficult to obtain a sufficient number of ratings for a specific instantiation (i.e., combination of options) of the item at hand.

This problem is also encountered in the context of the cold-start problem, when sufficient ratings are not available for the recommendation process. Furthermore, the nature of consumer preferences may evolve over time when dealing with such items. For example, a user who went to the beach, coast, and island for his or her holiday trip doesn’t want to go there again, as a result of which the preferences of user may show a corresponding evolution.

In other cases, it might be difficult to fully capture user interest with historical data such as ratings. A particular item (tour package) may have attributes associated with it that correspond to its various properties, and a user may be interested only in items with specific properties. For example, a tour package may have several properties such as type (drive or fly), night stay, depart date, return date, number of travellers, the cost of trip, and things to do while travelling. User interests may also be regulated by a very specific combination of these options. Thus, in these cases, the tourism domain tends to be *complex* in terms of its varied properties, and it is hard to associate sufficient ratings with the large number of combinations at hand.

Such cases can be addressed with knowledge-based recommendation systems, in which ratings are not used for the purpose of recommendations. Rather, the recommendation process is performed on the basis of similarities between user requirements and item descriptions, or the use of constraints specifying user requirements. The process is facilitated with the use of *knowledge bases*, which contain data about rules and similarity functions to use during the retrieval process.

In fact, the knowledge bases are so important to the effective functioning of these methods that the approach takes its name from this fact. The explicit specification of requirements results in greater control of users over the recommendation process. In both collaborative and content-based systems, recommendations are decided entirely by either the user's past actions, ratings of user's peers, or a combination of the two. Knowledge-based systems are unique in that they allow the users to explicitly specify what they want. Figure 1 shows the paradigm of knowledge-based recommendation.

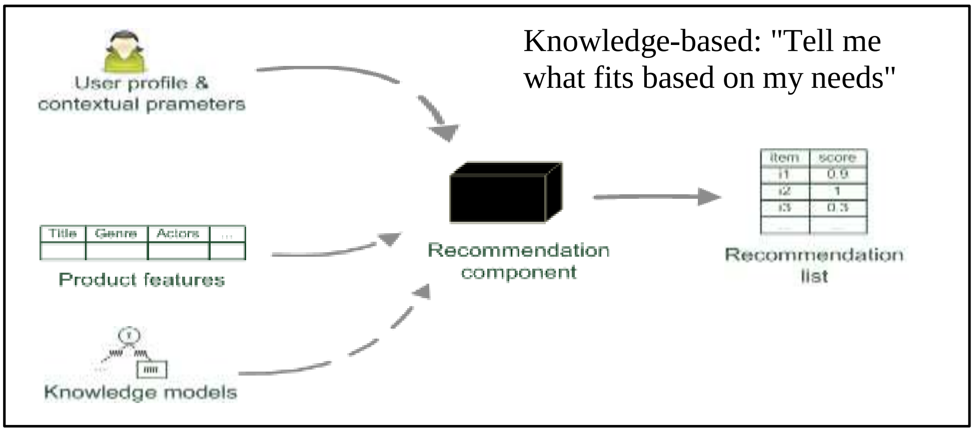

Figure 1 Paradigm of Knowledge-based Recommendation System

Constraint-based Recommendation System

In constraint-based systems, users typically specify requirements or constraints (e.g., lower or upper limits) on the item attributes. Domain-specific rules are used to match the user requirements to item attributes. These rules represent the domain-specific knowledge used by the system. Such rules could take the form of domain-specific constraints on the item attributes. Furthermore, constraint-based systems often create rules relating user attributes to item attributes. In such cases, user attributes may also be specified in the search process. Depending on the number and type of returned results, the user might have an opportunity to modify their original requirements. For example, they might relax some of their constraints when too few results are returned, or they might add more constraints. This search process is interactively repeated until the user arrives at her desired results.

In general, knowledge-based systems rely on detailed knowledge about item characteristics. Roughly speaking, the recommendation problem consists of selecting items from this catalog that match the user's needs, preferences, or hard requirements. The user's requirements can, for instance, be expressed in terms of desired values or value ranges for an item feature, such as "the cost should be lower than $100" or in terms of desired functionality, such as "the activity should be suited for art and culture" in travel domain.

A constraint-based recommendation problem can, in general, be represented as a *constraint satisfaction* that can be solved by a constraint solver or in the form of a *conjunctive query* that is executed and solved by a database engine.

Knowledge Representation and Reasoning

In general, knowledge-based systems rely on detailed knowledge about tour package characteristics. A snapshot of such tour packages list is shown in Table 1 for the travel domain. Roughly speaking, the recommendation problem consists of selecting tour packages from this table that match the user's needs, preferences, or hard requirements. The user's requirements can, for instance, be expressed in terms of desired values or value ranges for a tour package feature.

Table 1 Sample Tour Package Assortment

id	Type	Depart City	Stay	Depart Date	Return Date	Tra veler	Activity	Cost (ks)	Tour Place
p1	drive	yangon	2	4/8/20	6/8/20	35	hill & mountain, heritage	15000	Kyaik tiyo
p2	drive	yangon	2	4/8/20	6/8/20	15	beach & coast, honeymoon	30000	Ngwe Saung
p3	fly	manda lay	3	4/8/20	6/8/20	35	food & drink, hill &mountain	150000	Myit kyina
p4	fly	yangon	3	11/8/20	13/8/20	15	beach & coast, surfing	180000	Nga pali
p5	drive	manda lay	3	11/8/20	13/8/20	35	honey moon, hill & mountain	50000	Hpa-An
p6	drive	manda lay	3	11/8/20	13/8/20	35	adventure, mountain biking	35000	Taung gyi
p7	fly	yangon	2	13/8/20	16/8/20	15	art & culture, heritage	170000	Mrauk -U
p8	drive	yangon	2	13/8/20	16/8/20	35	art & culture, heritage	50000	Bagan
p9	drive	yangon	2	16/8/20	19/8/20	35	beach & coast, surfing	30000	Chaung Thar
p10	drive	yangon	3	16/8/20	19/8/20	35	adventure, hill & mountain	120000	Nat-Ma-Taung

Constraints

A classical constraint satisfaction problem (CSP) can be described by a-tuple (*V*, *D*, *C*) where *V* is a set of variables, *D* is a set of finite domains for these variables, and *C* is a set of constraints that describes the combinations of values the variables can simultaneously take. A solution to a CSP corresponds to an assignment of a value to each variable in *V* in a way that all constraints are satisfied.

Table 2 Sample Recommendation Task (V_C, V_{PK}, C_R, C_F, C_{PK}, REQ) and the Corresponding Recommendation Result (RES)

V_C	{*type*(…), *depart_city*(…), *stay*(…), *depart_date*(…), *return_date*(…), *traveler*(…), *activitiy*(…), *max_cost*(…)}
V_{PK}	{*type* (drive or fly), *depart_city* (yangon, mandalay,…), *stay* (1…10), *depart_date* (4/8/20…16/4/20), *return_date* (6/8/20…19/8/20), *traveler* (1…35), *activitiy* (heritage, art & culture, surfing, beach & coast, adventure, mountain biking, sky diving, hill & mountain, honeymoon, food & drink), *cost* (15000…180000)}
C_F	{*activitiy* = art & culture, *stay*=3)} (*activity* is customer property and *stay* are tour package property)
C_R	{activitiy =hill & mountain, max_cost<=50000)} (*activitiy* and *max_cost* are customer properties)
C_{PK}	{(*id*=p1, *type*=drive, *depart_city*=yangon, *stay*=2, *depart_date*=4/8/20, *return_date*= 6/8/20, *traveler*=35, *activitiy* = hill & mountain, heritage, *cost*=15000, *tour_place*=Kyaiktiyo) … (*id*=p10, *type*=drive, *depart_city*=yangon, *stay*=3, *depart_date*=16/8/20, *return_date*= 19/8/20, *traveler*=35, *activitiy* = adventure, hill & mountain, *cost*=120000, *tour_place*=Nat-Ma-Taung)}
REQ	{*type*=drive, *depart_city*= yangon, *stay*=2, *depart_date*=4/8/20, *return_date*=6/8/20, *traveler*=1, *activitiy* = hill & mountain, max_cost<=30000}
RES	{*type*=drive, *depart_city*= yangon, *stay*=2, *depart_date*=4/8/20, *return_date*= 6/8/20, *traveler*=1, *activitiy* = hill & mountain, heritage, max_cost<=50000, id=p1, *type*=drive, *depart_city*= yangon, *stay*=2, *depart_date*=4/8/20, *return_date*=6/4/20, *traveler*<=35, *activitiy* = hill & mountain, heritage, *cost*=15000, *tour_place*=Kyaiktiyo}

Constraint-based recommendation systems can build on this formalism and exploit a *recommender knowledge base* that typically includes two different sets of variables ($V = V_C \cup V_{PK}$), one describing potential user requirements and the other describing product properties. Three different sets of constraints ($C = C_R \cup C_F \cup C_{PK}$) define which items should be recommended to a user in which situation. Sample for such variables and constraints for tour package recommendation is shown in Table 2 where the corresponding recommendation result according to the user's constraints are queried from Table 1.

(a) Customer properties (V_C) describe the possible customer requirements. The customer property *max_cost* denotes the maximum cost acceptable for the customer, the property *activities* is the type of activities such as heritage, surfing, adventure, etc.

(b) Package properties *(V_{PK})* describe the properties of tour packages in an assortment; for example, *type* denotes possible means of transport in tour package.

(c) Filter conditions *(C_F)* define under which conditions which tour package should be selected – in other words, filter conditions define the relationships between customer properties and package properties. An example filter condition is *heritage, art & culture activities require cost should be at least 50000.*

(d) Compatibility constraints *(C_R)* define allowed instantiations of customer properties – for example, *if beach & coast activity is required, the minimal accepted cost must be greater than or equal to 30000.*

(e) Package constraints *(C_{PK})* define the currently available tour package assortment. Each conjunction in this constraint completely defines a package (item) – all package properties have a defined value.

The task of identifying a set of products (tour packages) matching a customer's wishes and needs is denoted as a *recommendation task*. The user requirements *REQ* can be encoded as unary constraints over the variables in V_C and V_{PK} – for example, *max_cost* = 50000.

Formally, each solution to the CSP ($V = V_C \cup V_{PK}, D, C = C_R \cup C_F \cup C_{PK} \cup REQ$) corresponds to a consistent recommendation. In many practical settings, the variables in V_C do not have to be instantiated, as the relevant variables are already bound to values through the constraints in *REQ*. The task of finding such valid instantiations for a given constraint problem can be accomplished by every standard constraint solver.

Conjunctive Queries

A slightly different way of constraint-based item retrieval for a given item (tour package) catalog is to view the item selection problem as a data filtering task. The main task in such an approach, therefore, is not to find valid variable instantiations for a CSP but rather to construct a conjunctive database query that is executed against the item catalog. A *conjunctive query* is a database query with a set of selection criteria that are connected conjunctively.

For example, *σ* [*activity*=hill & mountain, *max_cost*<=50000] (*P*) is such a conjunctive query on the database table *P*, where *σ* represents the selection operator and [*activity*=hill & mountain, *max_cost*<=50000] the corresponding selection criteria. Conjunctive queries for item selection purposes is exploited, V_{PACK} and C_{PACK} are represented by a database table *P*. Table attributes represent the elements of V_{PACK} and the table entries represent the constraint(s) in C_{PACK}.

Queries can be defined that select different item subsets from *P* depending on the requirements in *REQ*. Such queries are directly derived from the filter conditions (C_F) that define the relationship between customer requirements and the corresponding item properties. The filter condition *activity =beach & coast → stay=2* denotes the fact that if customers want to have *beach & coast activity*, the *stay* must be 2. If a customer defines the requirement *activity =beach & coast*, the corresponding filter condition is active, and the consequent part of the condition will be integrated in a corresponding conjunctive query. The existence of a recommendation for a given set *REQ* and a product assortment *P* is checked by querying *P* with the derived conditions (consequents of filter conditions). Such queries are defined in terms of selections on *P* formulated as *σ*[*criteria*](*P*).

Tour Packages Design for Knowledge-based Recommendation

Table 3 shows the tour packages list in travel recommendation system. In this table, there are twenty-five tour packages for the month of August, 2020 and also corresponding their attributes such as drive or fly (0 means "drive", 1 means "fly" in database table), depart from (city), night stay, depart date, return date, things to do, and trip price (cost) and trip name, etc. Knowledge-based system uses this knowledge sources and recommends tour packages according to the user's constraints.

Table 3 Tour Packages List in Travel Recommendation System

id	drive_or_fly	depart_from	night_stay	depart_date	return_date	total_seats	thing_todo	trip_price	trip_name
1	0	Yangon	3	16-08-20	19-08-20	35	#Horse Riding #Scuba Driving #Beaches & Swimming	50000	Ngwe Saung
2	0	Manda lay	3	16-08-20	19-08-20	35	#Heritage #Art & Culture	30000	Bago Travel
3	0	Yangon	3	16-08-20	19-08-20	35	#Heritage #Art & Culture	70000	Mrauk U Travel
4	1	Yangon	3	16-08-20	19-08-20	35	#Hills & Mountains #Adventure #Mountain Biking	180000	HpaAn (Kyauk Ka Lat)
5	0	Manda lay	3	16-08-20	19-08-20	35	#Adventure #Hills & Mountains #Wildlife & Nature #Heritage	20000	Kyaik htiyo Travel
6	0	Yangon	3	16-08-20	19-08-20	35	#Hills & Mountains #Honeymoon & Romance	120000	Lake Inle
7	0	Mandalay	3	16-08-20	19-08-20	35	#Art & Culture #Heritage	50000	Bago
8	0	Yangon	3	16-08-20	19-08-20	35	#Honeymoon & Romance #Food & Drink	70000	Pyin Oo Lwin Travel
9	0	Mandalay	3	16-08-20	19-08-20	35	#Heritage #Art & Culture #Hills & Mountains	70000	Pindaya Caves
10	0	Yangon	3	16-08-20	19-08-20	35	#Art & Culture #Heritage #Food & Drink	80000	Mingun
11	1	Yangon	2	24-08-20	26-08-20	35	#Wildlife & Nature #Mountain Biking #Heritage #Adventure	120000	Mount Popa
12	0	Yangon	2	24-08-20	26-08-20	35	#Heritage #Art & Culture #Food & Drink	65000	Amara pura
13	0	Naypyi taw	2	24-08-20	26-08-20	35	#Heritage #Art & Culture	52000	Shwebo Travel
14	1	Yangon	2	24-08-20	26-08-20	35	#Adventure #Wildlife & Nature #Coasts & Islands	250000	Myit kyina Travel
15	0	Yangon	2	24-08-20	26-08-20	35	#Coasts & Islands #Beaches & Swimming #Surfing #Food & Drink	80000	Satse Beach
16	0	Yangon	2	24-08-20	26-08-20	35	#Beaches & Swimming #Coasts & Islands #Food & Drink	70000	Gaw Yin Gyi Island

id	drive _or_ fly	depart _from	night _stay	depart _date	return _date	total _seats	thing_todo	trip _price	trip _name
17	0	Bago	2	24-08-20	26-08-20	35	#Coasts & Islands #Adventure #Hills & Mountains	50000	Mawtin sune Pagoda
18	0	Manda lay	2	24-08-20	26-08-20	35	#Food & Drink #Honeymoon & Romance #Coasts & Islands #Art & Culture	60000	Mawlam ying
19	0	Yangon	2	28-08-20	30-08-20	35	#Heritage #Art & Culture #Hills & Mountains #Adventure	20000	Kyaik Htee Yo
20	0	Yangon	2	28-08-20	30-08-20	35	#Art & Culture #Heritage #Food & Drink	60000	Sagaing Trip
21	0	Yangon	2	28-08-20	30-08-20	35	#Food & Drink #Heritage #Horse Riding	50000	Mawlam ying
22	0	Yangon	2	28-08-20	30-08-20	35	#Rafting #Beaches & Swimming #Surfing #Coasts & Islands	30000	Ngwe Saung Beach
23	0	Yangon	2	28-08-20	30-08-20	35	#Art & Culture #Food & Drink #Hills & Mountains	80000	Kayah State
24	0	Yangon	2	28-08-20	30-08-20	35	#Mountain Biking #Wildlife & Nature #Adventure #Hills & Mountains	80000	Goteik Bridge
25	0	Manda lay	2	28-08-20	30-08-20	35	#Art & Culture #Food & Drink #Heritage	70000	Shwe Ti Gone

Result and Discussion for Knowledge-based Recommendation

The following Table 4 shows the user's requirements and the corresponding recommendation result according to the constraints that are queried from Table 3. A consistent recommendation result (*RES)* for user's requirements (*REQ*) is depicted as follows.

Table 4 User Requirement (REQ) and the Recommendation Result (RES)

REQ1	{*drive_or_fly*=drive, *depart_from*= Yangon, *night_stay*=2, *depart_date*=28/8/20, *return_date*=30/8/20, *traveler*=2, *trip_price*= Between 20000 and 50000, *things_todo* = #Art & Culture #Heritage #Hills & Mountains#Beaches, Coast #Surfing}
RES1	{*id*=22, *drive_or_fly* =0, *depart_from* = Yangon, *night_stay* =2, *depart_date*=28/8/20, *return_date*=30/8/20, *traveler*<=35, *things_todo* = #Rafting #Beaches & Swimming #Surfing #Coasts & Islands, *trip_price*=30000, *tour_place*= Ngwe Saung Beach}
RES2	{*id*=21, *drive_or_fly* =0, *depart_from* = Yangon, *night_stay* =2, *depart_date*=28/8/20, *return_date*=30/8/20, *traveler*<=35, *things_todo* = #Food & Drink #Heritage #Horse Riding, *trip_price*=50000, *tour_place*= Mawlamying}
RES3	{*id*=19, *drive_or_fly* =0, *depart_from* = Yangon, *night_stay* =2, *depart_date*=28/8/20, *return_date*=30/8/20, *traveler*<=35, *things_todo* = #Heritage #Art & Culture #Hills & Mountains #Adventure, *trip_price*=20000, *tour_place*= Kyaik Htee Yo}
REQ2	{*drive_or_fly*=Drive, *depart_from*= Mandalay, *night_stay*=2, *depart_date*=24/8 /20, *return_date*=26/8/20, *traveler*=2, *trip_price*= Between 50000 and 100000, *things_todo* = #Adventure #Beaches, Coast #Food & Drink #Hills & Mountains #Surfing}
RES1	{*id*=18, *drive_or_fly* =0, *depart_from* = Mandalay, *night_stay* =2, *epart_date*=24/8/20, *return_date*=26/8/20, *traveler*<=35, *things_todo* = #Food & Drink #Honeymoon & Romance #Coasts & Islands #Art & Culture, *trip_ price*=60000, *tour_place*= Mawlamying}
REQ3	{*drive_or_fly*=Fly, *depart_from*= Yangon, *night_stay*=2, *depart_date*=24/8/20, *return_date*=26/8/20, *traveler*=2, *trip_price*= Between 100000 and 250000, *things_todo* = #Adventure #Art & Culture #Hills & Mountains #Beaches, Coast #Food & Drink}
RES1	{*id*=14, *drive_or_fly* =1, *depart_from* = Yangon, *night_stay* =2, *depart_ date*=24/8/20, *return_date*=26/8/20, *traveler*<=35, *things_todo* =#Adventure #Wildlife & Nature #Coasts & Islands, *trip_price*=250000, *tour_place*= Myitkyina Travel}
RES2	{*id*=11, *drive_or_fly* =1, *depart_from* = Yangon, *night_stay* =2, *depart_ date*=24/8/20, *return_date*=26/8/20, *traveler*<=15, *things_todo* = #Wildlife & Nature #Mountain Biking #Heritage #Adventure,*trip_price*=120000, *tour_place*= Mount Popa}

From the above Table 4, it can be concluded that the user requirement (REQ1) is the most similar to the recommendation results such as tour package id22 for RES1, id21 for RES2, and id19 for RES3. The user requirement (REQ2) is the most similar to the recommendation results such as tour package id18 for RES1. The user requirement (REQ3) is the most similar to the recommendation results such as tour package id14 for RES1, and id11 for RES2. From the fact that knowledge-based method gives consistent recommendation according to user constraints that fit his or her requirements.

Conclusion

Knowledge-based recommendation systems perform a needed function in a world of ever-expanding information resources. Unlike other recommendation systems, they do not depend on large bodies of statistical data about particular rated items or particular users. The knowledge component of these systems need not be prohibitively large, since we need only enough knowledge to judge items as similar to each other.

Further, knowledge-based recommendation systems actually help users explore and thereby understand an information space. Users are an integral part of the knowledge discovery process, elaborating their information needs in the course of interacting with the system. One need only have general knowledge about the set of items and only an informal knowledge of one's needs; the system knows about the tradeoffs, category boundaries, and useful search strategies in the domain. Knowledge-based recommendation systems are strongly complementary to other types of recommendation systems.

This paper proposes a recommendation system that offers knowledge-based recommendations for travel destinations to individuals. It can help overcome information overload problem by exposing users to interesting, novel, surprising and relevant items based on preferences users have expressed explicitly constraints. It can introduce users to new items that have not been known or have not been retrieved. So, these recommendations can help users in meeting their information needs for travel destinations.

Acknowledgement

I would like to express my profound gratitude to Dr. Soe Mya Mya Aye, Professor and Head of the Department of Computer Studies, Yangon University, for her kind permission to carry out this research. I would like to thank Dr. Ohn Mar San, Professor and Head of the Department of ICT, Co-operative University (Thanlyin), for her encouragement to do this research. I also would like to thank my supervisor Dr. Wint Pa Pa Kyaw, Associate Professor, Department of Computer Studies, Yangon University, for her close supervision and invaluable suggestions. I wish to thank all who help and give me advice on this research.

References

Aggarwal, C.C. (2016), *Recommendation System: The Textbook*, ©Springer International Publishing Switzerland, ISBN 978-3-319-29657-9

Alexander, F. (2008), *Persuasion in knowledge-based recommendation* in: Persuasive Technology, Springer, Page 71~82.

Friedrich, G., Jannach D., Zanker M. and Felfernig (2011), *Developing Constraint-based Recommenders*

Gavala, D., Konstantopoulos C., Mastakas K. and Pantziou, G. (2014), *Mobile Recommender Systems in Tourism*, Journal of Network and Computer Applications

Isinkaye, F.O., Folajimi, Y.O. and Ojokoh, B.A. (2015), *Recommendation systems: Principles, methods and evaluation*, Egyptian Informatics Journal, Vol.16, Page 261~273

Jannach D., Zanker M., Felfernig A. and Friedrich, G. (2011), *Recommendation Systems: An Introduction*, Cambridge University Press, ISBN 978-0-521-49336-9

Naaz, D. and Danish, M. (2018), *Knowledge Based Tourism Recommendation System Using Knowledge Base Filtering with Protégé Framework*, IJCSN- International Journal of Computer Science and Network, Volume 7, Issue 3, ISSN (Online: 2277-5420)

Ricci, F., Rokach, L., Shapira B., Paul, B. and Kantor (2011), *Recommendation Systems Handbook*, Springer New York Dordrecht Heidelberg London, ISBN 978-0-387-85819-7