# Context-aware Recommendation System with Anonymous User Profile Learning

Yan Liu
*School of Software Engineering*
Tongji University
Shanghai, China
Email: yanliu.sse@tongji.edu.cn

Yangyang Xu
*School of Software Engineering*
Tongji University
Shanghai, China

Mei Chen
*Decision and System Group*
United Technologies
Research Center (China)
Shanghai, China

***Abstract*—Recommendation system requests huge personal data, including personal information, purchase history, social tag/network, professional/personal preference, etc. Privacy preservation gets more and more concern in modern recommendation system. In this paper, we use surfing data in single session with context-aware learning to generate an anonymous user profile for recommendation, which yields very encouraging results. User profile is generated based on pre-learned hotel profile with pre-assigned weights. Two major behaviors are captured to learn the temporary user profile, which are search and view functions. A novel factor called "irrelevance" is created to measure the sensitivity of user to each item of hotel profile based on the surfing behaviors. A case study on a flight/hotel inquiring and booking website with different application scenarios and results are analyzed.**

***Keywords–Context awareness; recommendation system; e-service; user profile***

## I. Introduction

A recommendation system (RS) is a Web technology that proactively suggests item(s) to user based on side information, which could be user historical records or explicitly stated group preferences. It has been studied for more than ten years in various application areas, including e-commerce, e-health, and social network. Algorithms as content-based filtering [1], [3], [4], collaborative filtering [2], and context-aware prediction are widely applied [5], [7].

The main aim of a recommendation system is to support the website adherence on its user and attract new customers, which might be one of the most critical parameter for modern e-business. Current recommendation system uses user historical data to predict the blanks in the utility matrix (content-based filtering) or historical data of group users to understand potential options based on similar group of people (collaborative filtering). Users might be not preferred to be predicted or do not agree with the prediction, especially when the search varies or the aim is ambiguous. Many papers have discussed the challenge on privacy preservation [7], [8], which raise a big problem for learning algorithm - how can we learn the user profile in an anonymous way without or with limited historical data?

In this paper, we discuss a novel user profile learning method, which conduct recommendation with no user historical data. We utilize temporary user interaction (search / view) data, hotel profile, and environment information with a context-aware learning method to understand user's intention, and develop anonymous but effective user profile for recommendation. This anonymous user profile learning recommendation system is applied in hotel recommendation for a travel booking web site and obtained good results. The article is organized as follows: first, we briefly discuss existing research and challenge on recommendation system. In section III we introduce the booking system and three application scenarios, as well as the major problems and challenges for hotel recommendation. Proposed learning methods and recommendation algorithm is discussed in section IV with results and comparison. Conclusion and future work are in section V.

## II. Background

Content-based Recommender System focus on properties of items, where the recommendation on items is based on learning the user preference and constraints. It created user specific item profiles (important characteristics of an item) and calculate the similarity of items. It predicts items that user is most likely to be interested in or has highest tendency will accept. Collaborative-filtering RS focus on the relationship between users and items. It measures user similarity for any items to establish a group profile, which recommend items to a user by voting on the group users. Content-based RS needs historical data for single user, and collaborative-filtering RS requires historical data from group users. Both systems require large data for profile learning, and might not work when the utility matrix is sparse.

Context-aware RS attracts more attentions as people realized that taking into consideration on any contextual information, such as time, place, is important. It might be critical to incorporate the contextual information into the recommendation process, especially under certain circumstances (i.e. location related recommendation). Context is a multifaceted concept that has been studied across different research disciplines [5], where RS utilizes the concept from data mining, e-commerce personalization, information retrieval, and other directly related fields. When $R: User \times Items \times Context \rightarrow Rating$, selecting proper item for specific user at set up context environment will generate very

DOI reference number: 10.18293/SEKE2015-065

different rating. This is most interesting but challenging part for a context-aware recommendation system.

## III. Application Scenario, Problem and Challenges

In this paper, we develop an effective anonymous hotel recommendation system for a travel booking website who delivers recommendation through email. The recommendation is based on transition information obtained within an interaction session, which can be a flight ticket booking procedure, or an ambiguous hotel inquiry. The main aim is to increase the hotel booking rate and check-in rate by applying certain recommendation system.

This booking website provides flight and hotel inquiring and booking services. It uses email to conduct recommendation, which is quite efficient for following application scenarios:

- New coming users, most of them are unwilling to register or just have a quick search without booking. We ask these users to leave their email address before leaving. We believe that the user who has intent to book a flight/hotel will leave a valid email. We already observed it during daily operation.
- Registered user without log in, which is almost the same situation as the new user until we found the email address was registered. Actually, we found that it made no difference no matter whether the registered user log in or not. It is hard and almost impossible to conduct effective on-line recommendation (very low hit rate make the recommendation annoy).

  The potential reasons lie in 2 aspects: one is that the historical record is sparse for most user which make the prediction matrix high sparsity with large uncertainty, and large intra-group variance make the recommendations deviate from real intention significantly. Sometime, the users themselves might not have clear target hotels before searching.
- Users inquiring but did not booking would like to receive recommendations especially with promotion, which means the recommendation through email gets attention if it hits the needs truly. This is another observed practice.

### A. Scenarios

There are three different application scenario might trigger the recommendation:

1) Promotion proposed by hotel, the RS will send email to target users.
2) User search flight and finally book one or more itineraries. This means that the user has logged in.
3) Anonymous user search flight or hotel information but did not book anything. The email address will be asked before inquiry and the email input is optional.

Scenario (1) and (2), user profile (UP) learning with personal information and historical data is applicable. Price sensitive customers with previous vacation trip(s) are target user for scenario (1). RS sends out emails to potential interested customer, which we call a passive RS. Applicable strategies are:

- select price sensitive users;
- select users having past trips for holiday or vacation within a time range (e.g. 6 months);
- focus on promotion before public holiday.

Scenario (2) and (3) are active RS, where the recommendation is triggered by user actions. For scenario (2), static UP will be created for recommendation based on historical data. A context understanding model will generate a dynamic UP, and the final recommendation will be rated based on hybrid static & dynamic profiles. Scenario (1) & (2) will not be discussed as we are interested in learning UP with no historical data.

### B. Problem and Challenges

We believe that scenario (3) is more applicable and preferred by user, as most users do want a good recommendation without leaking too much personal information, especially when RS learns their profiles. We claim that an anonymous UP learned with context in (3) can be sufficient for RS. That is why we choose scenario (3) as our typical case for analysis, and the percentage of scenario (3) is dominant when analyzing the web visits.

In scenario (3), the setup conditions are: a) no historical data, b) the user is anonymous, c) the destination city is known, and d) side information such as viewed flights or viewed hotels is also given. There are two kinds of inquiry behaviors in scenario (3): 3.a) searching the flight itineraries; or 3.b) searching hotels in a city (several cities).

In (3.a), useful inputs are: destination, viewed flight(s), itineraries date/time, and the search date/time. We use $BT$ for "Business Trip" and $PT$ for "Private Trip" in following analysis. An item profile is created to learn the intention with probability of "Business Trip" versus "Private Trip" based on context understanding:

- Destination and/or any event related to destination (i.e. a commercial show in the destination city) are used to calculate $P(BT|Destination, Events)$ and $P(PT|Destination, Events)$.
- Flights being viewed suggests the acceptable and preferred class and price level. This helps to determine $P(BT|FlightClass, ItineraryTime)$, and $P(PT|FlightClass, ItineraryTime)$.
- Price sensitivity can be inferred from viewed flights if the user viewed several itineraries. The difference between itineraries tells the priority of price vs. time, and the itinerary date tells the flexibility on the trip. $P(BT|PriceSensitivity, TripFlexibility)$,

and $P(PT|PriceSensitivity, TripFlexibility)$ are learned.
- Viewed itineraries time also helps to learn the $P(BT|Distance\ on\ Itineraries\ Time)$, $P(PT|Distance\ on\ Itineraries\ Time)$ as the business trip is more time sensitive than price.
- Searching date/time, esp. date help calculating $P(BT|Weekday/Weekend, Daytime/Nighttime)$, and $P(PT|Weekday/Weekend, Daytime/Nighttime)$.

All the user specific items help to understand the search purpose and determine whether this is a user who might be interested in hotel recommendation (with/without promotion). We found that most user search flight itinerary will make final booking, which suggests that (3.a) can be merged to scenario (2) by learning context-aware item profile.

Scenario (3.b) is the most critical case and will be studied in this paper. An anonymous user profile is developed by combining information from hotel profile (HP) and temporary user interaction data through a context-aware learning schema. Major problems and challenges in (3.b) are as follows:

1) High variation and uncertainty on searching content, with several times or dozens times of search.
2) There are 1000+ or 2000+ hotels in metropolis or megapolis, such as Beijing, Shanghai, etc.
3) Hotel number varies from dozens to thousands in different cities, where the room type and price range vary for same star hotel in different cities.
4) City functionality and characteristics have high variance.
5) Identify target users from non-target users.
6) Learn user profile and understand user intention for accurate recommendation.

For the challenge related to city own functionality, we are not able to tell or utilize the city profile in our model and we will not count this as side information.

## IV. System, Model and Results

Figure (1) shows the workflow on this anonymous user profile learning system using hotel profile and temporary user data. Hotel profile, including static and dynamic profile, will be updated in a pre-set period. In parallel, the user interaction data will be used to learn the user behaviors and responds according to different hotel profiles. Through the context understanding we can learn the user intention and select target hotel(s) for recommendation.

In (3.b), the designed features for HP based on static data and side information are: 1) GIS/Business Zone; 2) Hotel Star; 3) Price; 4) Facility; 5) Transportation; 6) Rating; 7) Room; 8) Promotion; 9) Event. Static and dynamic HP will be learned from these features, where static feature could be Business Zone, Hotel Star, etc.; and price, promotion are dynamic features.

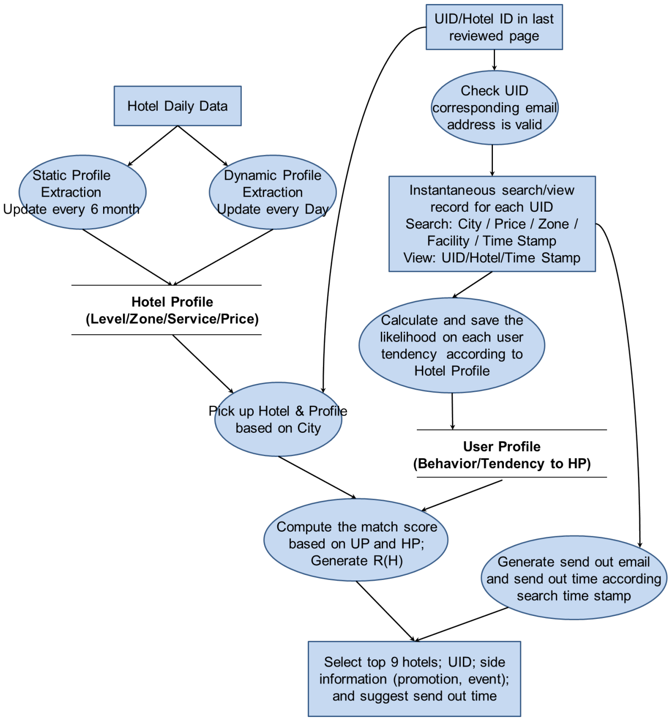
UID/Hotel ID in last reviewed page
Check UID corresponding email address is valid
Hotel Daily Data
Static Profile Extraction Update every 6 month
Dynamic Profile Extraction Update every Day
Instantaneous search/view record for each UID Search: City / Price / Zone / Facility / Time Stamp View: UID/Hotel/Time Stamp
Hotel Profile (Level/Zone/Service/Price)
Calculate and save the likelihood on each user tendency according to Hotel Profile
Pick up Hotel & Profile based on City
User Profile (Behavior/Tendency to HP)
Compute the match score based on UP and HP; Generate R(H)
Generate send out email and send out time according search time stamp
Select top 9 hotels; UID; side information (promotion, event); and suggest send out time

Figure 1. Content-Aware User Profile Learning Recommendation System Architecture.

The inferred UP will be used to answer two questions: 1) whether we are going to send out email with hotels recommendation; 2) what hotels should be recommended. Before creating HP and UP we need to filter out inquires made by agent software such as web crawler. The inquiry with number lager than 100 times contains most random inquires (such as random selected city names), which is very likely to be the scans made by the agent software, and should be removed. This kind of inquiry will be outliner in learning UP and introduce large deviation to real user intention.

### A. City and Hotel Profile

We generate HP with different feature sets and weights according to the city. City type also determines how likely we will send out the recommendation to potential user. Table I shows the city category, size, classification rule, features, and methods for HP generation. Category of a city determines which feature set we would like to apply in HP for hotels in that city. Here we use megapolis city as example, which will have all features as we mentioned before.

The HP development is a score calculating and weighting process, where we treat each feature independently. The correlation will be considered during learning the UP based on HP:

- Hotel star is a simple but typical static feature, which

TABLE I

Category on hotel profile features according to the destination city.

| Category Rule | Size | Classification | Features | Hotel Profile Generation |
|---|---|---|---|---|
| $C_0$ | megapolis | Beijing, Shanghai, Shenzhen, Guangzhou, Tianjin, Chongqing | All Features | CB + CF |
| $C_1$ | metropolis | $N_1 < Num_{Hotel} < N_0$ | No GIS/Business Zone | CB + CF |
| $C_2$ | city | $N_2 < Num_{Hotel} < N_1$ | No GIS/Business Zone/Room | Rule-based filtering |
| $C_3$ | town | $Num_{Hotel} < N_2$ | No GIS/Business Zone/ Transportation/ Room/Rating | Rule-based filtering |
| $C_4$ | special hot spot | Hongkong, Macau, Sanya, or seasonal hot spots | Transportation/Room/ Promotion /Event | CB |

is grouped into 4 ranges naturally ('two star & below' are put together in a single range). Each hotel can be only assigned in one of the range.

- Price is a typical dynamic feature; and to be simple, we put hard edge on the price range. There are total 10 ranges from 0 to $\infty$. Each hotel can have room in multiple price range, and all occupied price range will be marked for a hotel daily.
- Transportation is important but hard to quantize. We use the time to any transportation center as score, where this parameter has less impact than the GIS/Business Zone, especially in metropolis or megapolis city. In our case study, we will not use this feature for HP generation.
- Normalized rating score collected from Hotel Evaluation Website is used directly. This feature has small impact for HP.
- Facility only counts in WiFi, breakfast, parking, which has $0/1$ value, corresponding to yes/no.
- Business Zone (BZ) is a unique feature used in this travel booking website, which can be treated as a demographic GIS area. There are about 50 to 100 BZ in metropolis or megapolis city. Each BZ has one unique index number (in each city), where the index does not have numerical meaning and cannot be grouped based on the value.

### B. User Profile and Ranking Model

We select user whose inquiry time is 10 - 50 as target user. We define 'behavior' as the operator conducted in the web page. In each successful inquiry, the UP will have accumulated score updated based on the user's 'behavior'. The detailed calculation will be discussed late, and 'behaviors' combined with HP will generate different scores. 'Behaviors' are:

1) 'Search': defined as $U_{search}()$. Considered parameters inclue price range (p), star (s), business zone (z), and facility (f) as $U_{search}(p, s, z, f)$. The parameters and the $U_{search}()$ expressions are:
   - $U_{search}(P)$, where $P = [min_{price}, max_{price}]$ is the price range from min to max.
   - $Usearch(star_2) + Usearch(star_3)$ if multiple star hotels are selected.
   - $U_{search}(z)$, where zone only has single value.
   - facility has 3 categories, and the parameters can be written as $Usearch(facility_1) + Usearch(facility_2) + Usearch(facility_3)$.
2) 'View': defined as $U_{view}()$ with only single parameter 'Hotel' as there is no specific information provide by the user. We will use hotel profile in this parameter.
3) 'Order': contains historical data, which will not be used in UP learning but for validation.
4) 'Count': defines the number of a specific 'behavior' happened in a user. The parameter is $U_{search}()$ and $U_{view}()$.

Users have both view and search record in single inquiry (one event), and the score calculated by $U_{search}()$ will have larger weight. Score tells how far the user is interested in this feature and will be calculated for each feature based on $U_{search}()$ and $U_{view}()$. The calculated individual feature score will be normalized, weighted with user sensitivity on this feature, and then summarized for final ranking. There are 2 different score calculation methods:

- Unique feature for a hotel (i.e. star, zone), the search will have twice counting on 'search' than 'view'. For example, the score for star $i$: $Star_{i,Score} = 2 \times num(Usearch(star_i)) + num(Uview(Hotel.star = i))$, where $i \in [2, 5]$.
- Features having multiple parameters (i.e. price):
  1) Each $U_{search}(p)$, a 10 element vector $a$ $(1 \times 10)$ is created based on $price(minPrice, maxPrice)$. The searched or viewed price ranges falling in the $(minPrice, maxPrice)$ is marked with 1 in corresponding elements in $a$.
  2) Each $U_{view}(Hotel)$, vector $b$ $(1 \times 10)$ is created based on hotel profile, where the price ranges learned from HP is marked with 1 in corresponding elements in $b$.

    3) Price score is the summary of all vectors: $Upricescore = a_i + b_j,\ i = 1, \dots, M,\ j = 1, \dots, N$.

- Hotel rating is summarized based on search and view hotels, and normalized. Similar work for transportation.
- Other binary features (promotion and event) take value of 0/1 based on hotel/city daily updated status, meaning Yes/No.

Two weights $w_i$ and $w_j$ are applied to modeling the user true intention upon the calculated feature scores; where $w_i$ represents user sensitivity and $w_j$ represents confidence. We use ('zone', 'star', and 'price') as example to calculate $w_i$ and $w_j$ and explain the main ideas. Let's assume a megapolis city, which has zone index (1-70), hotel star (2-5), and price range (1-10). A user UID '001' has 20 view records.

- $w_i$: no previous information on user preference, we assume each feature set is uniform distributed.

  Use 'zone' as example, the sensitivity can be calculated as: $w_{i,zone} = \frac{\sum_{r=1}^{R} I_{zone}(i,r)}{\sum_{rr=1}^{R} \sum_{k=1}^{N} I_{zone}(k,rr)}$, where $N = 70$, $R = 20$, and $I_{zone}$ is the indicator function with $I_{zone}(i,r) = 1$ when the $i^{th}$ zone is selected in the $r^{th}$ record.

  $w_i$ for 'star' and 'price' are calculated in the same way.
- $w_j$: variance level works as confidence.
    1) Feature 'star' and 'price' have numerical meaning on their values. We use inversed $L_2$ distance between selected parameters as confidence. $w_{j,star} = \frac{1}{\sum_{m,n=1}^{20} (star_m - star_n)^2}$.
    2) Feature 'zone' has no numerical meaning on its index. We use occurrence vector and standard deviation to model the zone confidence. For example, we have $zone_j$ view record $m_j = \{1,1,0,0,0,0,0,1,1,0,0,1,1,0,1,1,0,0,0,0\}$, where 1/0 means selected/no. Then the standard deviation $D_j$ represents how reliable this user like $zone_j$. Averaged all 70 $D_j$ we can final confidence $D$ as $w_{j,zone}$.

Ranking model is to calculate the likelihood of each hotel that user might be interested. Given

- $w_i$,
- $w_j$,
- features for destination city ($F_{ct} := \{F_j, j = 1, \dots, n_{ct}\}$);

where $n_{ct}$ is the total number of features. We calculate $R(H) = \sum_{j=1}^{n_{ct}} w_j \times \sum_{i=1}^{N_j} w_i \times HP(F_{ct})$; where $HP(F_{ct})$ is the hotel profile (given city and its specific features).

### C. Validation and Results

Performance validations are conducted in two categories: 1) user inquires hotels and make the booking at the same day (noted as $T_{user} \equiv 0$); 2) user inquires hotels, does not make booking instantaneously but book the hotel within 10 days (note as $T_{user} \equiv 1$). We use $T_{user} \equiv 2$ to refer user never make the booking (including user did not input email address). Category (2) is verified with UID and associated email address, where the email address was obtained during inquiry and late booking. Only users used same email address are considered as same user and will be used for validation as $T_{user} \equiv 1$. The comparison is made between the real booked hotel and our recommended top 9 hotels. It means anyone of our recommended 9 hotels is the same hotel as the user booked hotel, we counted as a successful hit.

Table II shows the number of people with 3 situations stated above in a 7 days' record. A clear pattern of weekday vs. weekend is shown, especially for $T_{user} \equiv 2$. This information will be used in determine sending email time.

TABLE II
USER NUMBER AND DISTRIBUTION.

| DAY | TOTAL | $T_{user} \equiv 0$ | $T_{user} \equiv 1$ | $T_{user} \equiv 2$ |
|---|---|---|---|---|
| 1 | 1396 | 148 | 595 | 653 |
| 2 | 1329 | 175 | 591 | 563 |
| 3 | 1314 | 128 | 596 | 590 |
| 4 | 1311 | 142 | 573 | 596 |
| 5 | 1363 | 142 | 628 | 593 |
| 6 | 942 | 119 | 448 | 375 |
| 7 | 837 | 84 | 399 | 354 |

Table III shows the hit rate on hotels for 2 validation tests. From this table we find that we have obtained pretty good hit rate in a fully anonymous way, which means this system is valuable. Also, the hit rate of $T_{user} \equiv 0$ is less than $T_{user} \equiv 1$ (almost half) might due to fewer records for $T_{user} \equiv 0$. People make book at the same day always have clear target with less inquiries. Actually, for our RS, $T_{user} \equiv 1$ is our target people and the hit rate is fairly good. The calculated $R(H)$ values also give us a clear boundary on $T_{user} \equiv 1$ and $T_{user} \equiv 2$ people, which is useful to determine whether we need to send out recommendation.

TABLE III
HIT RATE FOR $Tuser = 0$ AND $Tuser = 1$.

| DAY | $T_{user} \equiv 0$ | $T_{user} \equiv 1$ |
|---|---|---|
| 1 | 21.62% | 41.17% |
| 2 | 28.00% | 47.55% |
| 3 | 26.56% | 47.48% |
| 4 | 26.35% | 42.11% |
| 5 | 23.24% | 42.19% |
| 6 | 26.05% | 44.19% |
| 7 | 26.19% | 41.85% |
| AVERAGE | 25.43% | 43.65% |

## V. CONCLUSION AND FUTURE WORK

In this paper, we address several major challenges in context-aware RS. An anonymous user profile-learning schema allows we provide a recommendation with privacy protection. Target users are separated from others successfully. The final hit rate from validation result indicates that it is feasible to design RS in an anonymous way under some circumstance. We also address challenges in context representation and semi-structure log data analysis, which are very critical in RS. Well-designed architecture ensures the RS work in an efficient way providing real time recommendation.

There are two major concerns for the future work. One is designing sophisticated HMI to understand the user intention (some initial work [9]), to explain the rationale behind recommendation to end-user. The recommendation could be a decision or action. The RS can have high risk in determining what to recommend, especially smart decision support. This requires integration on context awareness, user intention understanding, and recommendation expression as a whole picture. The other is Meta data design for map-reduce structure to handle big data challenge. For a big data flow on line processing system, it is necessary and important to have parallel processing capability. Immigrate this RS to a cloud based structure should be next step.

## ACKNOWLEDGMENT

This work was financially supported by the Science and Technology Commission of Shanghai Municipality (12dz1507000).

## REFERENCES

[1] Adomavicius, G., Tuzhilin, A.: Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. on Data and Knowledge Engineering, 17(6), 734–749 (2005)

[2] Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst., 22(1), 5–53 (2004)

[3] Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 325–341. Springer-Verlag, Springer (2007)

[4] Di Noia, T., Mirizzi, R., Claudio Ostuni, V., Romito, D., Zanker, M.: Linked open data to support content-based recommender systems. In: 8th International Conference on Semantic Systems(ACM), New York (2012)

[5] Adomavicius, G., Tuzhilin, A.: Context-Aware Recommendation Systems. In: Recommender systems handbook. pp. 217–253. Springer-Verlag, Springer (2011)

[6] Wang, S.L., Wu, C.Y.: Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. Expert Systems with applications. 38(9), 10831–10838 (2011)

[7] Verbert, K., Manouselis, N., Ochoa, X.,; Wolpers, M., Drachsler, H., Bosnic, I. ; Duval, E.: Context-aware recommender systems for learning: a survey and future challenges. IEEE Trans. on Learning Technologies. 5(4), 318–335 (2012)

[8] Jones, M. T.: Recommender systems, Part 1: Introduction toapproaches and algorithms. Technical report, IBM (2013) http://www.ibm.com/developerworks/library/os-recommender1/

[9] Pu, P., Chen, L., Hu, R., Hu: Evaluating recommender systems from the user's perspective: survey of the state of the art. User Modeling and User-Adapted Interaction. 22(4-5), 317–355 (2011)