

Adaptive Tradeoff Explanations in Conversational Recommenders

Li Chen

Department of Computer Science
Hong Kong Baptist University, Hong Kong

lichen@comp.hkbu.edu.hk

ABSTRACT

The completeness and certainty of a user's preferences may vary during her preference construction process in a conversational recommender. In order to more effectively support users to uncover their hidden criteria and/or solve preference conflicts, we propose to generate adaptive tradeoff explanations in organization-based recommender interfaces, to be conditional on the user's contextual needs. An experiment shows the adaptive element's higher potential to improve recommendation efficiency, relative to methods without this feature.

Categories and Subject Descriptors

H5.2 [Information interfaces and presentation]: User Interfaces – graphical user interfaces (GUI), user-centered design.

General Terms

Design, Human Factors.

Keywords

Conversational recommenders, explanation, preference elicitation.

1. INTRODUCTION

Recommenders that involve users in a conversational interaction model to incrementally elicit their preferences are commonly referred as conversational recommenders, which are especially usable for solving multi-objective decision problems. For example, while searching for high-risk products (e.g. PCs, digital cameras, cars), users inherently have multiple objectives to reach for targeting at their desired product, like criteria on computers' price, processor speed, memory, hard capability, etc.

Most of recent works on conversational recommenders have emphasized on how to generate critiques in order to allow for effective feedback mechanism during each interaction cycle. As a typical example, the dynamic-critiquing system can propose a set of compound critiques (e.g. Different Manufacturer, Lower Resolution and Cheaper") for the user to choose one to improve the current recommended product [13]. The suggested critique is also regarded as a kind of explanation, explaining to users the recommendation opportunity in the remaining dataset [9]. However, the critique is computed purely according to the

availability of products in the dataset, without dynamically adapting its generation process to the condition of the user's current preferences (such as whether the user would prefer the suggested critique and whether it can further assist users to complete their requirements or solve preference conflicts).

As a matter of fact, according to adaptive decision theory [10], people are usually unable to accurately state their preferences up front especially when they are confronted with an unfamiliar product domain or overwhelming information. We call such preference condition as "incomplete preferences" which can be concretely reflected with un-specified criteria on the product's attributes and will typically appear during a user's beginning interaction cycles. Moreover, "conflicting preferences" phenomenon will happen when no option is best on all of a user's stated attribute preferences, that she may have probably established initially or in the later interaction rounds when being more familiar with the product catalog. Indeed, *conflicting* has long been recognized as a major source of decision difficulty and many researchers have argued that making tradeoffs between more of one thing and less of another is a crucial aspect of high-quality and rational decision making [4,10,11]. However, users have been found often avoiding explicit tradeoffs primarily due to their cognitive or emotional limitation [8].

It is hence meaningful to make recommendations explicitly serve people with different initial preference certainty levels and even tailor to various preference conditions of a single user during her whole preference construction process. Additionally, it is of critical value to correspondingly explain these recommendations so as to guide users to understand the recommending rational and hence complete their preference model for achieving an accurate decision. In recent years, explanations have been increasingly studied while mainly for collaborative filtering based or content-based recommendations [6,14]. Few works have actually paid attention to developing explanations to expose tradeoffs (i.e. pros and cons) of recommendations in a conversational recommender, although much earlier works on user-adaptive systems have indicated the importance of the tradeoff exposure [2,5]. In the following, we first briefly summarize our previous findings, based on which we will introduce our newly designed adaptive tradeoff explanation interfaces in a prototype recommender.

1.1 Contribution of Our Work

We have previously proposed a method to organize recommendations into categories, so called the organization-based recommender interface empirically proven with two practical benefits [12,3]: one is that it can more effectively build users' trust in recommendations than the traditional interface where recommendations are listed one by one even each with a "why" explanation; and another is that it can more accurately predict

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RecSys '09, October 23-25, 2009, New York, New York, USA.

Copyright 2009 ACM 978-1-60558-435-5/09/10 ... \$10.00.

critiques that users truly intend to make, compared to other typical critique generation methods.

Despite of these benefits, one limitation of our previous approach as observed from user studies is that it is lack of adaptability to the variety of user requirements. For instance, some users commented that they felt helpless about how to revise their preferences when there are conflicts, since the recommended products including their explanations seemed not directly corresponding to such situation.

Thus, in this paper, we propose an approach to addressing the limitation, which is to integrate “adaptive tradeoff explanations” into our preference-based organization interface. In essence, the adaptive element works by automatically capturing the current user’s preference model at a time, analyzing its property (e.g. whether it is “incomplete” and/or “having conflicts”) and then adjusting the organization algorithm being conditional on the property. For example, when the system observes that the user’s preferences are incomplete respecting some un-stated attributes, suggestions will be made and explained as “these products have bigger hard drive capability and larger screen size that you may like”. In another case when there are preference conflicts, the explanation for a set of products will be like “they satisfy your preferences on price and processor speed, but not on memory type”. In order to test the new algorithm’s efficiency, we have conducted a simulation to compare it with our previous method and a baseline approach.

2. ADAPTIVE TRADEOFF EXPLANATIONS

The tradeoff explanation is intended for two purposes: one is to show the tradeoff knowledge that exists between product attributes (e.g. higher processor speed is often correlated to higher price) for the user to get to know their relationship; another is to explain the conflict (if there is) occurring in a user’s stated criteria so as for her to make proper weight adjustment. We are still based on the association rule mining to discover the tradeoff relationship, like in our previous organization algorithm. However, the definition of inputs to Apriori (the association rule mining tool [1]) is fundamentally different from the original method. That is they are adaptively changeable conditional on the user’s current needs, rather than configured with a uniform definition and format (see our prior publications [3, 12]).

In truth, our new algorithm has been developed with the purpose of explicitly tailoring its generation of recommendations and explanations to different user preference conditions such as “incomplete preferences” or “preference conflicts”. More concretely, in different conditions, the recommendations along with their explanations will be either adapted to stimulate users to uncover hidden needs, or solve conflicting values, or with both objectives. The user’s reaction to the displayed items will be reflected in her refined preference model for the system to compute a new set of recommendations in the next cycle. The conversational process can continue until the user’s preferences are maximally complete and precise, at which time the best matching product should be her target choice. Figure 1 shows an example of the algorithm’s control flow at a cycle.

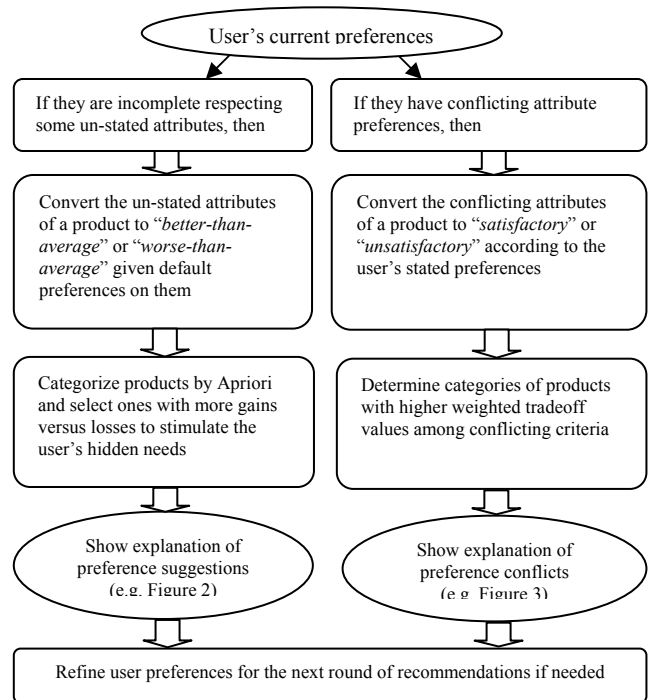


Figure 1. Control flow of the adaptive organization algorithm provided with two conditions.

A pre-filtering process is first conducted to retrieve products that exactly match all of the user’s attribute preferences. If the retrieval is successful and the number of matching products exceeds a pre-defined threshold p (e.g. $p = 20$), we mainly consider this retrieval set for the organization. At this point, if the user’s preferences are found “incomplete”, the organization is aimed at suggesting preferences on un-stated attributes (e.g. “these products satisfy all of your stated preferences. In addition, they have bigger memory that you may like” where the memory is an un-stated attribute that the user has not specified any criterion). Specifically, we convert all retrieved products into a set of tradeoff vectors. Each vector indicates the corresponding product’s tradeoff properties on un-stated attributes, by comparing each of the attribute values with its average across all products (“better-than-average” abbreviated as *BTA* and noted as “↑”, or “worse-than-average” abbreviated as *WTA* and noted as “↓”). As an example, one product can be formalized as {(processor speed, ↑), (weight, ↑), (memory, ↓)} meaning that it has higher processor speed, lighter weight but smaller memory relative to these attributes’ average values in the retrieval set. The determining of *BTA* or *WTA* property is concretely based on the attribute’s default preference. That is, as for each attribute, a default preference is pre-defined (e.g. for “processor speed”, it is the higher, the better). These tradeoff vectors are then inputted to the Apriori algorithm to reveal how the attributes are frequently associated between one another, based on which the organization is made. As a sample interface, Figure 2 shows how the set of matching products is displayed in the organization interface, where each category contains products with the same properties on suggested attributes. Thus, it can be seen that the tradeoff explanation here is to disclose what additional benefits along with compromises the user may be interested in, so as to stimulate them to discover hidden needs.






Show top 22 products satisfying all of your stated preferences, in addition	
They have faster processor speed, but are slightly more expensive	
	Toshiba Portege M200 Tablet PC \$ 2349 Toshiba, Microsoft Windows XP Tablet PC, 4.34 hours battery, 12.1 in screen, 80 GB, 512 MB Ram, Intel Pentium M Processor (Centrino), 2 GHz, 4.4 lbs
	Toshiba Portege M200 Tablet PC \$ 2189 Toshiba, Microsoft Windows XP Tablet PC, 4.34 hours battery, 12.1 in screen, 60 GB, 256 MB Ram, Intel Pentium M Processor (Centrino), 2 GHz, 4.4 lbs
5 products View All	
They have longer battery life and lighter weight, but are from different manufacturers	
	ElectroVaya Scribbler SC-2000 Tablet PC \$ 1988.87 ElectroVaya, Microsoft Windows XP Tablet PC, 9 hours battery, 12.1 in screen, 30 GB, 256 MB Ram, Intel Pentium M Processor (Centrino), 1.2 GHz, 3.5 lbs
	Fujitsu Stylistic ST5020 Tablet PC \$ 2149 Fujitsu, Microsoft Windows XP Tablet PC, 6 hours battery, 12.1 in screen, 60 GB, 512 MB Ram, Intel Pentium M Processor, 1.1 GHz, 3.5 lbs
6 products View All	
They have larger display size and more installed memory	
	Fujitsu LifeBook T4010D Tablet PC \$ 1661 Fujitsu, Microsoft Windows XP Tablet PC, 5 hours battery, 14.1 in screen, 40 GB, 1024 MB Ram, Intel Pentium 4 Processor M, 4.0 GHz, 4.9 lbs

Figure 2. Explanation of preference suggestions to stimulate hidden needs.






No product satisfying all of your stated preferences	
They satisfy your preferences on price and weight, but not on processor speed	
	Fujitsu LifeBook T4010D Tablet PC \$ 1639 Fujitsu, Microsoft Windows XP Tablet PC, 5 hours battery, 12.1 in screen, 60 GB, 512 MB Ram, Intel Pentium M Processor (Centrino), 1.3 GHz, 4.3 lbs
	HP (Hewlett-Packard) Compaq TC4200 Tablet PC \$ 1459 HP (Hewlett-Packard), Microsoft Windows XP Tablet PC, 5.5 hours battery, 12.1 in screen, 60 GB, 512 MB Ram, Intel Pentium M Processor (Centrino), 1.1 GHz, 4.49 lbs
6 products View All	
They satisfy your preferences on manufacturer and processor speed, but not on price	
	Toshiba Portege M200 Tablet PC \$ 2189 Toshiba, Microsoft Windows XP Tablet PC, 4.34 hours battery, 12.1 in screen, 60 GB, 1024 MB Ram, Intel Pentium M Processor (Centrino), 2 GHz, 4.4 lbs
	Toshiba Tecra M4-S115TD Tablet PC \$ 2379 Toshiba, Microsoft Windows XP Tablet PC, 6 hours battery, 14.1 in screen, 40 GB, 256 MB Ram, Intel Pentium M Processor (Centrino), 1.6 GHz, 6.4 lbs
5 products View All	
They satisfy your preferences on memory and price, but not on weight and processor speed	
	Toshiba Satellite R10-S802D Tablet PC \$ 1259 Toshiba, Microsoft Windows XP Tablet PC, 4 hours battery, 14.1 in screen, 40 GB, 1024 MB Ram, Intel Mobile

Figure 3. Explanation of partially satisfied products to solve preference conflicts.

In another condition when no available product matches all of the user's stated preferences or the retrieved number is less than the threshold p , a partial satisfaction set will be organized and returned. The goal is to propose different tradeoff directions for the user to decide which attribute constraint(s) she would be willing to relax, to exchange for gains on more important ones. Based on the Multi-Attribute Utility Theory (MAUT) [7], each product is computed with a score representing its weighted satisfying degree with the user's preferences, and the set of products with higher scores is then returned. Each product in the set is formally converted into a tradeoff vector indicating which attribute preference(s) it matches (i.e. "satisfactory" noted as \uparrow) and which one(s) it does not (i.e. "unsatisfactory" as \downarrow). For example, one product is formalized like $\{(display\ size, \uparrow), (weight, \uparrow), (processor\ speed, \downarrow)\}$, indicating that "this product satisfies your preferences on display size and weight, but not on processor speed". Therefore, the explanation at this point is to expose the tradeoff relation between conflicting preference values that the user specified, so as to support her to make informed weight adjustment to refine preferences. Please note that although it is also called "tradeoff vector" in this case, the meaning is different

from when suggesting preferences: *satisfactory* against *unsatisfactory* in accordance with the user's stated preferences, instead of *BTA* or *WTA* on un-stated attributes. Again, the association rule mining tool is applied to uncover frequent and recurring patterns among inputted tradeoff vectors. Products are thereafter organized into categories in the form of "these products satisfy your preferences on ..., but not on ..." (see Figure 3). If "incomplete preferences" are additionally observed along with "preference conflicts", they will be also reflected in the generated category (e.g. "these products satisfy your preferences on In addition, they have bigger hard capability", where "bigger hard capability" is a suggested preference).

Since Apriori normally creates a number of category candidates (each candidate is a discovered association rule), we primarily select ones with higher overall utilities, while the utility is computed differently dependent on the current preference condition. For instance, it is formally calculated as the category's gains versus losses relative to the user's potential needs, if the purpose is to suggest preferences, but a weighted tradeoff value between conflicting criteria when it is to assist users in solving preference conflicts. In either case, diversity is further involved in order to return categories as diverse as possible in respect of both their titles (i.e. the discovered rules) and associated products (please refer to [3] in this regard). As a result, k categories is selected and each category is representative of several products that share the same pattern (e.g. $\{(price, \uparrow), (weight, \uparrow), (memory, \downarrow)\}$). A pre-designed base of explanation templates is then used to explain each selected category in a right manner. Specifically, the property (i.e. \uparrow and \downarrow) assigned to each attribute is concretized regarding its actual meaning. For example, if it is "better-than-average" regarding one attribute (e.g. processor speed), the explanation will be like "higher processor speed" as a suggested preference. On the other hand, if it is "satisfactory" referring to an attribute value satisfying the user's stated criterion (e.g. on price), the explanation will be like "satisfy your preference on price". Please see more explanation examples (i.e. categories' titles) in Figures 2 & 3.

As noted before, a user's preference construction is in nature an incremental process. After each recommendation cycle (if the user does not quit), the user's preferences will be automatically refined according to the action she did, so as for the computation of new items in the next cycle. More concretely, default value function and weight will be assigned to an attribute and included in the user's preference model, if she selected a near-target product with suggestion on this previously unconsidered attribute. Weight adjustment will be performed when there are preference conflicts. The weight of "satisfactory" attribute(s) appearing in the chosen product will be increased by a certain degree (e.g. β), and the weight of "unsatisfactory" one(s) will be decreased by β (β is default set as 0.25). If the weight of an attribute is reduced to 0, it will be removed from the user's preference model (implying that the user is indifferent to it any more).

3. EXPERIMENT

3.1 Experiment Design

In order to understand whether the integration of the "adaptive element" can experimentally improve the algorithm's recommendation efficiency, i.e. saving users' interaction cycles in locating their target choice, we performed a simulation to

compare the new algorithm with our previous method that is without the adaptive feature [3] (so termed as Non-ADPT), in addition to a standard related approach which is also based on the association mining to categorize but without consideration of both user preferences and adaptability (the dynamic-critiquing [9,13], henceforth STD).

A PC dataset (with 120 PCs) was used for the offline evaluation. Each product is described by 9 main attributes including price, type, manufacturer, processor speed, memory, etc. A similar *leave-one-out* methodology to [13] was adopted in our experiment. That is, at each time, we randomly chose one product (called the base item) in the dataset used for two purposes: the product most similar to it was determined as a simulated user's target choice, and the random subset of its attribute values was based to generate the user's initial preferences. The tested algorithm returns k categories of recommendations during each cycle, and the category with the product closest to the "user's" ideal preferences (i.e. the base item) is selected. The preference model is refined according to the category's properties, and a new set of recommendations is computed for the next round. The interaction session terminates when the "user's" target choice is identified.

3.2 Results

Each product in the dataset was left out at 10 times and we ran the *leave-one-out* test for all products in the dataset (so totally 120x10 times of tests on each algorithm). The number of recommendation cycles for finding the target choice was recorded. Figure 4.a shows the average cycles respectively through the three methods. It can be seen that the adaptive organization algorithm (ADPT) demands the least amount of cycles, which is reduction of 33% relative to the standard organization method (STD) and 12% to our previous approach (Non-ADPT).

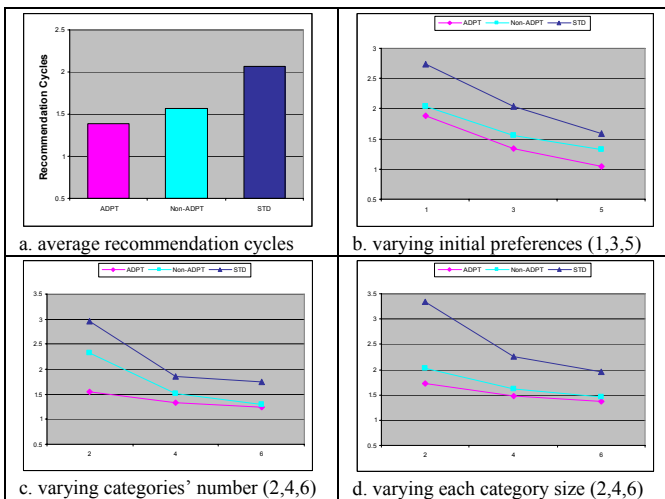


Figure 4. Experimental results.

We further did the comparisons by varying initial preferences' completeness levels, the number of returned categories (i.e. k) and the number of products displayed within each category (i.e. q), respectively. It shows that ADPT won in all of the scenarios. Specifically, when the initial preferences became more incomplete, all of the three algorithms turned out with increasing session lengths, but ADPT relatively required the least interaction effort (see Figure 4.b). The adaptive algorithm was also stable in obtaining the highest efficiency, especially when less categories

were returned ($k = 2$ or 4), or less products were shown in each category at a cycle ($q = 2$ or 4). On average, it reaches 37% and 39.6% reductions of cycles compared to STD, and 20% and 10.6% to Non-ADPT (see Figures 4.c and 4.d).

4. CONCLUSION AND FUTURE WORK

In this paper, we introduced and evaluated an adaptive element that can explicitly tailor the process of retrieving and explaining recommendations to the condition of user preferences, taking into account of both "incomplete preferences" and "preference conflicts" phenomena. Positive experimental results imply its potential practical impacts. Therefore, for the next step, a user study will be carefully designed and performed in order to not only verify the simulation's findings, but also measure users' decision quality and subjective perceptions when they are in reality interacting with the adaptive tradeoff explanation interface. We believe that with this adaptive element, a conversational recommender should have higher chance to allow its users to efficiently obtain an accurate and confident decision.

5. REFERENCES

- [1] Agrawal, R., Imielinski, T., and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD '93* 207–216.
- [2] Ardissono, L., Goy, A., Petrone, G., Segnan, M., and Torasso, P. 2003. INTRIGUE: personalized recommendation of tourist attractions for desktop and handset devices. In *Applied Artificial Intelligence* 17(8-9), 687-714.
- [3] Chen, L., and Pu, P. 2007. Preference-based organization interfaces: aiding user critiques in recommender systems. In *Proc. UM'07*, 77-86.
- [4] Frisch, D., and Clemen, R. T. 1994. Beyond expected utility: rethinking behavioral decision research. *Psychological Bulletin* 116, 46-54.
- [5] Jameson, A., Schäfer, R., Simons, J., and Weis, T. 1995. Adaptive provision of evaluation-oriented information: tasks and techniques. In *Proc. 14th IJCAI*, 1886–1893.
- [6] Herlocker, J. L., Konstan, J. A., and Riedl, J. 2000. Explaining collaborative filtering recommendations. In *Proc. CSCW'00*, 241-250.
- [7] Keeney, R., and Raiffa, H. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press.
- [8] Luce, M. F., Payne, J. W., and Bettman, J.R. 1999. Emotional trade-off difficulty and choice. *Journal of Marketing Research* 36, 143-159.
- [9] McCarthy, K., Reilly, J., McGinty, L., and Smyth, B. 2004. Thinking positively – explanatory feedback for conversational recommender systems. In *Workshop of 7th ECCBR*, 115-124.
- [10] Payne, J. W., Bettman, J. R., and Johnson, E. J. 1993. *The Adaptive Decision Maker*. Cambridge University Press.
- [11] Pu, P., and Chen, L. 2005. Integrating tradeoff support in product search tools for e-commerce sites. In *Proc. 6th ACM EC*, 269-278.
- [12] Pu, P., and Chen, L. 2006. Trust building with explanation interfaces. In *Proc. IUI'06*, 93-100.
- [13] Reilly, J., McCarthy, K., McGinty, L., and Smyth, B. 2004. Dynamic critiquing. In *Proc. ECCBR '04*, 763-777.
- [14] Vig, J., Sen, S., and Riedl, J. 2009. Tagsplanations: explaining recommendations using tags. In *Proc. IUI'09*, 47-56.