Experiments in Dynamic Critiquing

Kevin McCarthy, James Reilly, Lorraine McGinty & Barry Smyth
Adaptive Information Cluster
Department of Computer Science
University College Dublin
Belfield, Dublin 4
Ireland
{kevin.mccarthy, james.d.reilly, lorraine.mcginty, barry.smyth}@ucd.ie

ABSTRACT

Conversational recommender systems are commonly used to help users to navigate through complex product-spaces by alternatively making product suggestions and soliciting user feedback in order to guide subsequent suggestions. Recently, there has been a surge of interest in developing effective interfaces that support user interaction in domains of limited user expertise. Critiquing has proven to be a popular and successful user feedback mechanism in this regard, but is typically limited to the modification of single features. We review a novel approach to critiquing, *dynamic critiquing*, that allows users to modify multiple features simultaneously by choosing from a range of so-called *compound critiques* that are automatically proposed based on their current position within the product-space. In addition, we introduce the results of an important new live-user study that evaluates the practical benefits of dynamic critiquing.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Human Factors

Keywords

critiquing, recommender systems, user-interfacing/feedback

1. INTRODUCTION

Developing effective product recommendation systems is an important and challenging problem [1, 13]. It is made difficult for a variety of reasons. Very often users are not familiar with the details of a particular product domain, or may not fully understand or appreciate the trade-offs that exist between different product features. Conversational recommender systems are commonly used to help users to navigate through complex product-spaces. The user is guided through a sequence of recommendation cycles in which one or more products are recommended based on some evolving model of the user's requirements. During each cycle the user is offered the opportunity to provide feedback in order to steer the recommender in the direction of their desired product. Unfortunately users rarely provide complete or accurate product specifications to begin with and their feedback can be inconsistent and contradictory.

One feature of intelligent user interfaces is an ability to make decisions that take into account a variety of factors, some of which may depend on the current situation [3]. Consequently, it is crucial that user interfaces provide appropriate feedback mechanisms for the domain and users in question. Recently researchers have begun to consider the use of different forms of feedback in recommender systems along a variety of dimensions. From an interfacing standpoint, different forms of feedback assume different degrees of domain expertise and require different levels of user effort. For example, *value elicitation*, where users indicate a precise feature value — "I want a digital camera with 512Mb of storage", for example — assumes that users have detailed domain knowledge and that they are willing to indicate the precise requirements on a feature by feature basis. In contrast, *preference-based* feedback asks the user only to indicate a preference for one suggestion over another.

In this paper we are interested in a form of feedback known as *critiquing*; see [4]. Critiquing can be viewed as a compromise between the detail provided with value elicitation and the ease of feedback associated with preference-based methods. To critique a product a user indicates a directional change to a specific feature. For example, a digital camera shopper might ask for a camera that is *more expensive* that the current suggestion; this is a critique over the *price* feature. We present a novel approach to critiquing known as *dynamic critiquing* which automatically presents combinations of critiques to users as feedback options, thus facilitating the simultaneous critiquing of multiple features. We have previously introduced this critiquing strategy but our earlier evaluations were limited to artificial user studies [7, 11, 12]. In this paper we introduce a recent prototype system and interface that incorporates dynamic critiquing. Furthermore we present the results of a live-user trial to show the practical benefits of dynamic critiquing in terms of overall recommendation performance.

IUI'05, January 9–12, 2005, San Diego, California, USA.

Critiquing was first introduced as a form of feedback for recommender interfaces as part of the FindMe recommender systems [4, 5], and is perhaps best know for the role it played in the Entree restaurant recommender (see Figure 1). During each cycle Entree presents users with a fixed set of critiques to accompany a suggested restaurant case, allowing users to *tweak* or critique this case in a variety of directions; for example, the user may request another restaurant that is *cheaper* or *more formal*, for instance, by critiquing its *price* and *style* features. A similar interface approach was later adopted by the RentMe (see Figure 2) and Car Navigator (Figure 3) recommenders from the same research group [4, 5].

Figure 1: A sample screen-shot from the Entree Restaurant recommender illustrating the standard critiquing approach.

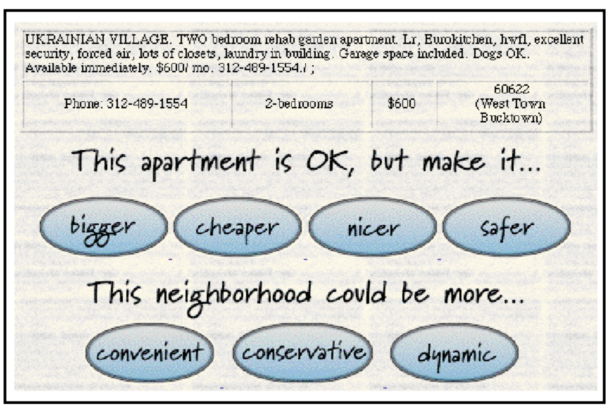

Figure 2: A sample screen-shot from the Rent Me recommender illustrating the standard critiquing approach.

1.1 Critiquing, Usability & Performance

As a form of feedback critiquing has many advantages. From a user-interface perspective it is relatively easy to incorporate into even the most limited of interfaces. For example, the typical *"more"* and *"less"* critiques can be readily presented as simple icons or links alongside an associated product feature value and can be chosen by the user with a simple selection action. Contrast this to value elicitation approaches where the interface must accommodate text entry for a specific feature value from a potentially large set of possibilities, via drop-down list, for example. In addition, critiquing can be used by users who have only limited understanding of the product domain. For instance a digital camera buyer may understand that greater resolution is preferable but may not be in a position to specify a concrete target resolution.

While critiquing enjoys a number of significant usability benefits, as indicated above, it can suffer from the fact that the feedback provided by the user is rarely sufficiently detailed to sharply focus the next recommendation cycle. For example, by specifying that they are interested in a digital camera with a *greater resolution* than the current suggestion the user is helping the recommender to narrow its search but this may still lead to a large number of available cases to chose from. Contrast this with the scenario where the user indicates that they are interested in a *5 megapixel* camera, which is likely to reduce the number of product options much more effectively. The result is that critiquing-based recommenders can suffer from protracted recommendation sessions, when compared to value elicitation approaches.

Consequently researchers have begun to look more closely at ways to improve the performance of this form of feedback. For example, [9, 10] show how the performance of critiquing can be improved by incorporating diversity into the recommendation process. Very briefly, the approach taken by [9, 10] proposes that critiques can not only tell a recommender system about the direction a user would like to go in the product space, but that they can also be used to determine whether the recommendations presented during the current cycle have been an improvement on those from the previous cycle. If they are an improvement then the recommender is likely to be narrowing its search in the right way. Otherwise it may indicate that the recommender is incorrectly focused and that the search should be broadened out during the next cycle. In this way the recommendation strategy uses a mixture of similarity and diversity to guide the recommendation process and experimental results indicate that significant reductions in average session length can be achieved.

1.2 Compound Critiques

The critiques described so far are all examples of, what we refer to as, *unit* critiques. That is, they express preferences over a single feature; Entree's *cheaper* critiques a *price* feature, and *more formal* critiques a *style* feature, for example. This too ultimately limits the ability of the recommender to narrow its focus, because it is guided by only single-feature preferences from cycle to cycle. Moreover it encourages the user to focus on individual features as if they were independent and can result in the user following false-leads. For example, a price-conscious digital camera buyer might be inclined to critique the price feature until such time as an acceptable price has been achieved only to find that cameras in this region of the product space do not satisfy their other requirements (e.g., high resolution). The user will have no choice but to roll-back some of these price critiques, and will have wasted considerable effort to little or no avail.

An alternative strategy is to consider the use of what we call *compound* critiques [11]. These are critiques that operate over multiple features. This idea of compound critiques is not novel. In fact the seminal work of Burke *et al.* [4] refers to critiques for manipulating multiple features. For example, Figure 3 shows a screen shot from the Car Navigator recommender[4].

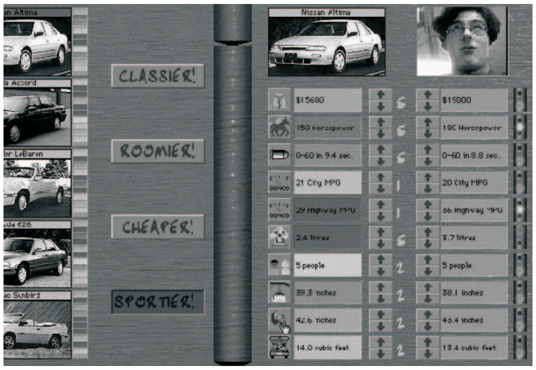

Figure 3: A sample screen-shot from the Car Navigator recommender illustrating the *static* compound critiquing approach.

The right-hand page of the interface presents the user with a range of individual features and their associated unit critiques whereas the left hand page shows a set of compound critiques. For instance, the *sportier* critique combines multiple unit critiques over the *horsepower, acceleration* and *engine size* features. Similarly we might use a *high performance* compound critique in a PC recommender to simultaneously increase *processor speed, RAM, hard-disk capacity* and *price* features.

Obviously compound critiques have the potential to improve recommendation efficiency because they allow the recommender system to focus on multiple feature constraints within a single cycle. In addition, it has also been argued that they carry considerable explanatory power because they help the user to understand common feature interactions [8, 11]; in the PC example above the user can easily understand that improved CPU and memory comes at a price.

Importantly, in the past when compound critiques have been used they have been hard-coded by the system designer so that the user is presented with a fixed set of compound critiques in each recommendation cycle; in the Car Navigator example above, the compound critiques are fixed and static, for example. These compound critiques may, or may not, be relevant depending on the cases that remain at a given point in time. For instance, in the example above the *sportier* critique would continue to be presented as an option to the user despite the fact that the user may have already seen and declined all the relevant car options.

2. DYNAMIC CRITIQUING

This research is motivated by the need to develop a more dynamic approach to critiquing in which compound critiques are generated on-the-fly, during each recommendation cycle, by mining commonly occurring patterns of feature differences that exist in the remaining cases [11]. Figure 5 shows a Digital Camera recommender system that we have developed (based on the *FindMe* recommender systems [5]) to explore usability and efficiency issues in a real-world setting. The screenshot shows the information presented to the user during a single recommendation cycle: unit critiques are presented alongside each individual feature of the case description, and 3 dynamically generated compound critiques are presented below the current case description. In the subsections that follow we will describe how dynamic critiques are generated and selected for presentation as part of each recommendation cycle.

2.1 Critique Patterns

The first step in critique discovery is to generate a set of so-called *critique patterns* from the cases that remain in the current cycle. Each critique pattern reflects the differences between a remaining case and the current recommended case as a set of unit critiques.

	Current Case	Case *c* from CB	Critique Pattern
Manufacturer	Canon	Sony	!=
Model	Powershot S500	DSC-V1	!=
Format	Ultra Compact	Ultra Compact	=
Resolution (M Pixels)	5.1	5.0	<
Optical Zoom (X)	3	4	>
Digital Zoom (X)	4.1	4	<
Weight (grams)	215	298	>
Storage Type	Compact Flash	Memory Stick	!=
Storage Included (MB)	32	16	<
Price (Euro)	443.00	455.00	>

Figure 4: Illustration of how a sample critique pattern may be generated the digital camera domain.

Figure 4 illustrates what we mean with the aid of an example. It shows the current case that has been selected for recommendation to the user as part of the current cycle and also a case, *c*, from the case-base. The resulting critique pattern reflects how case *c* differs from current case in terms of individual feature critiques. For example, the critique pattern shown includes a "<" critique for *resolution*— we will refer to this as $[Resolution <]$—because the comparison case has a lesser resolution than the current recommendation.

2.2 Critique Generation

Recurring sets of unit critiques within the critique patterns are ideal candidates for compound critiques. Discovering these recurring subsets is equivalent to the so-called *market-basket analysis* problem, where the task is to find regularities in the shopping behaviour of customers [6] by identifying sets of products that tend to be purchased together: each critique pattern is equivalent to the shopping basket for a single customer, and the individual critiques correspond to the items in this basket. The combinatorics of a typical task domain—thousands of products and customers—make this a challenging problem, leading to an explosion in the number of possible groups of recurring items. However, it is not so acute in our critiquing scenario because there are only a limited number of possible critiques. For instance, each numeric feature can have a $<$ or a $>$ critique and each nominal feature can have a $=$ or a $!=$ critique, so there are only $2n$ possible critiques in a case-base where the cases are made up of n individual features.

Previously [7, 8, 11] we have shown how the well-known Apriori algorithm [2, 6] can be used to efficiently discover compound critiques as association rules of the form $A \rightarrow B$: from the presence of a certain set of critiques (A) one can infer the presence of certain other critiques (B). For example, one might learn that from the presence of the critique, $[Price <]$, we can infer the presence of $[Memory <]$ with a high degree of probability; in other words the pattern $\{[Price <],[Memory <]\}$ is commonplace.

We use Apriori, during each recommendation cycle, to generate a collection of compound critiques (frequent itemsets over the pattern-base), and to then select a small subset of the *best* of these compound critiques for presentation to the user to complement the standard unit critiques. How we use the Apriori algorithm to select what compund critiques to present to the user is discussed in Section 2.3.

2.3 Critique Selection

We can expect a large number of compound critiques, of different sizes, to be generated during a typical recommendation cycle. From a user interface and availability viewpoint it is obviously not practical to present large numbers of different compound critiques, as feedback options, to the user, during each cycle. For this reason a filtering strategy is needed so that we can select the most useful critiques, say the top 3, for presentation purposes. We have previously shown how the *support* values that Apriori generates can be used to select a subset of these compound critiques for presentation to the user [7, 11].

Apriori is a multi-pass algorithm, where, in the k^{th} pass, all large itemsets of cardinality k are computed. Initially *frequent itemsets* are determined. These are sets of items that have at least a predefined minimum support. Then, during each iteration those itemsets that exceed the minimum support threshold are extended. Very briefly, "support" is the percentage of patterns for which the rule is correct; that is, the number of patterns that contain both A and B divided by the total number of patterns. For instance, we would find that the rule $[Resolution >] \rightarrow [Memory >]$ has a support of 0.2 if there are a total of 100 critique patterns but only 20 of them contain $[Resolution >]$ and $[Memory >]$. We have shown that presenting critiques with low support values provides a good balance between their likely applicability to the user and their ability to narrow the search (see [7, 11] for further details).

2.4 Compound Critiques as Explanations

The core hypothesis is that compound critiques help the user to better understand the *recommendation opportunities* that exist beyond the current cycle by helping them to appreciate common interactions between features. We believe that in many recommender domains, where the user is likely to have incomplete knowledge about the finer details of the feature-space, that compound critiques will help to effectively map out this space. For instance, with standard critiquing in the digital camera domain a user might naively select the $[Price <]$ unit critique in the mistaken belief that this may deliver a cheaper camera that satisfies all of their other requirements. However, reducing price in this way may lead to a reduction in resolution that the user might not find acceptable and, as a result, they will have to backtrack. This problem is less likely to occur if the compound critique $\{[Price <], [Resolution <]\}$ is presented because the user will come to understand the implications of a price-drop prior to selecting any critique.

The point is that in many recommender domains, where the user is likely to have incomplete knowledge about the finer details of the feature-space, and perhaps little or no knowledge of detailed feature interactions, then compound critiques may help to clarify these interactions. This will help the user to better understand the options that are available beyond the current recommendation cycle. For this reason we believe that users will actually find it easier to work with compound critiques than unit critiques and this may, for example, help the user to make fewer critiquing errors.

3. PROTOTYPE SYSTEM

Figures 5(a-c) present a series of screenshots from a prototype application of our dynamic critiquing approach in the context of an online digital camera store. The screenshots present a sequence of recommendation cycles and in each we see the currently recommended case, its features and their unit critiques, plus a set of 3 compound critiques, and their associated explanations. Each compound critique is translated into natural language and can be chosen directly (via the 'pick' option) or elaborated further (via the 'explain').

After the user has provided some initial information they are presented with a high-end Canon camera for 995 Euro with 512MB of memory and 6.2 mega-pixels, as shown in Figure 5(a). The user can critique any of the individual features, such as *manufacturer, memory, resolution* etc. by selecting the appropriate critique icon on either side of the feature value fields that are displayed for the current camera; the up-arrow indicates a *greater-than* critique, down-arrow indicates a *less-than* critique and the cross indicates a *not-equal-to* critique. In addition, just below these features, three compound critiques are displayed and the user can select one of these to be applied directly (the 'pick' option). In our opinion, these compound critiques serve as partial explanations as to some of the remaining retrieval options. For example, the first compound critique in Figure 5(a) explains to the user that there are cheaper cameras available but that they come with less memory and lower resolution.

In addition, the user can request a more detailed explanation of the compound critique via the 'explain' option. This explanation is presented in the pane to the right of the feature values, and defaults to be an explanation of the first compound critique. For example, in Figure 5(b) the user asks for further explanation of the third compound critique (*"Different Manufacturer, Lower Resolution and Cheaper"*). The resulting explanation tells the user that there are 87 remaining cameras that satisfy this critique—that is, there are 87 cameras that are cheaper, with a lower resolution, and made by a different manufacturer, than the currently recommended Sony camera. In addition, the explanation provides information about the ranges of values for these critiqued features. For instance, the user is told that these 87 cameras are made by manufacturers such as Canon, Fuji, Kodak and Nikon, that they have resolutions from 1.4 to 4.8 mega-pixels, and that their price ranges from 125 to 399 Euro.

4. EVALUATION

Previously we have described a series of experiments designed to evaluate the effectiveness of dynamic compound-critiques compared to their more traditional unit-based counterparts [7, 11, 12]. The results were extremely promising. For example, we predicted that users would likely select a compound critique in about 20%-25% of cycles and that these compound critiques would help to deliver significant reductions in overall session length, with reductions of up to 45% noted. However these earlier studies were based on artificial users rather than real, live users and as such could not

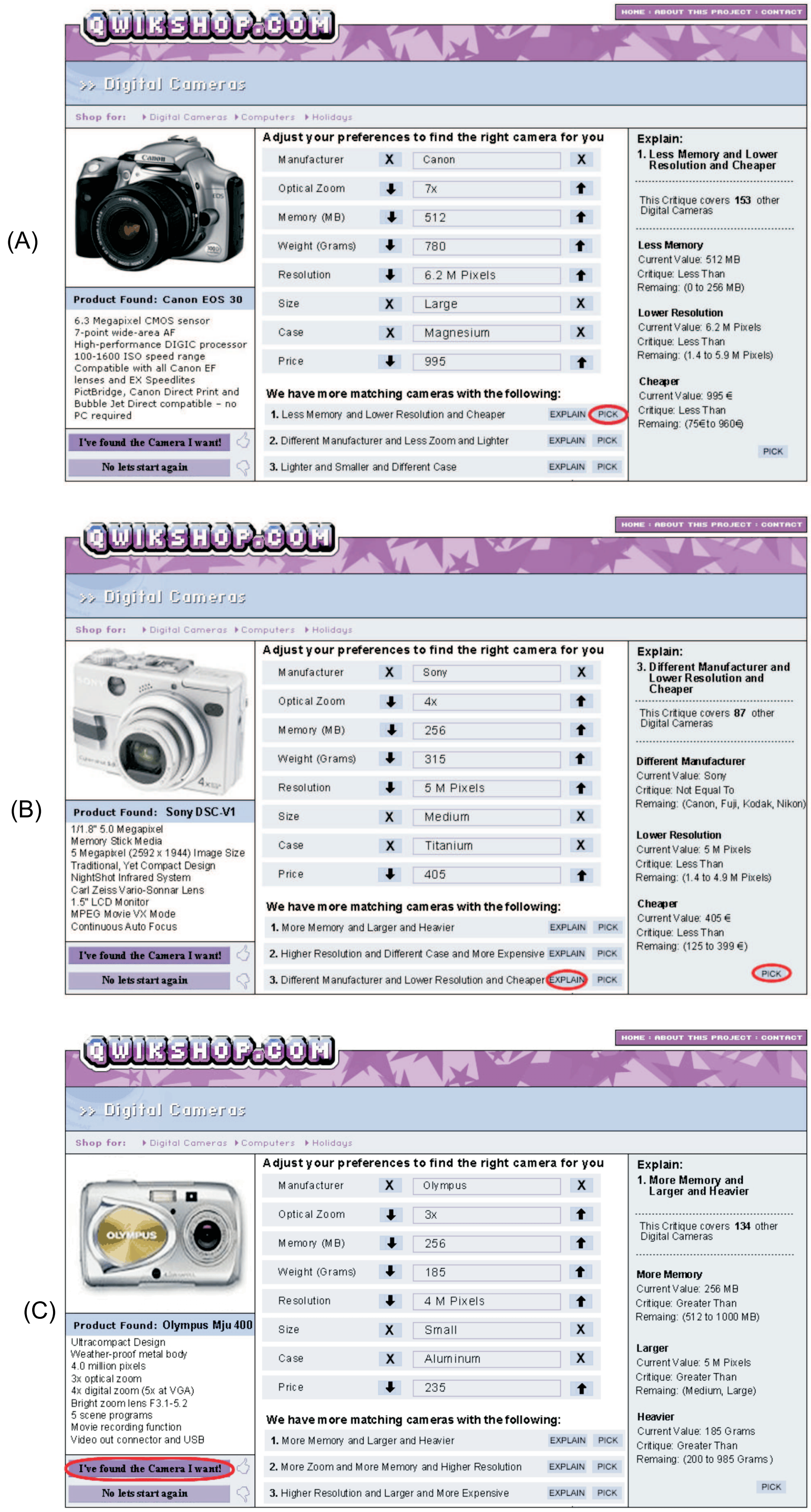

Figure 5: A sequence of recommendation cycles in our prototype digital camera recommender. In (A) the the user is presented with a suggestion and invited to critique this recommendation in line with their requirements. The user seeks a recommendation that has less memory, lower resolution and is cheaper than the current suggestion. In (B) the user is presented with an alternative and provides feedback as before. Finally, in (C) the user indicates that they have found a suitable camera for purchase.

be forwarded as definitive proof of the benefits of dynamic critiquing. Thus, in this section we describe the results of a recent live-user trial using our digital camera recommender.

4.1 Setup

The digital camera recommender system was developed to serve as an application prototype in order to conduct live-user trials of our critiquing approaches. It is a fully functioning critique-based content recommender comprising a case-base of some 250 up-to-date digital camera descriptions with each camera case composed of 10 features (e.g. *manufacturer, price, resolution, optical zoom*, etc.). The system provides a user interface that allows users to navigate through the product space using a combination of unit and compound critiques (see Figure 3).

For our live-user study we asked 38 post-graduate students to interface with the system and locate a digital camera that they would be prepared to purchase. They were provided with a brief description of the recommender interface, highlighting the use of unit and compound critiques during product navigation. In an associated questionnaire they were asked to rate the digital camera knowledge on a scale of 1 (low) to 5 (high). The average rating was 3.56, with 14 participants rating their knowledge as 3 or above; 12 of the participants reported to having bought a digital camera in the past.

During the trial the full selection and critiquing behaviour of each participant was recorded. In addition at the end of the trial participants were asked to rate their level of satisfaction with the camera they decided to purchase. In the following subsections we describe an analysis of trial data focusing on:

1. the application frequency of unit/compound critiques
2. the average session lengths, and
3. user satisfaction levels at the end of each session.

4.2 Application Frequency

Figure 6 presents a user-by-user account of the frequency with which compound critiques were selected during each session and the length of these sessions. For example, user 8 has an application frequency of 25%—meaning that they selected compound critiques in 25% of recommendation cycles and unit critiques in the remaining cycles—and a session length of 8.

The first thing to notice is that there is considerable variation in the application frequency among the trial participants. The average application frequency is 25%—which, incidentally, is entirely consistent with the results of our previous artificial-user trials—with one user making use of compound critiques in two-thirds of their session (user 22) but another user never using any compound critiques (user 15). It appears likely that these application frequency levels could be improved. It seems that not all users fully understood or appreciated the role of compound critiques in the short time they spent with using the prototype system; each user only took part in a single recommendation session after reading a brief explanation of the interface and experiment objectives.

Upon comparing the compound critique application frequency and session length we see a clear relationship: high compound critique application frequencies appear to be correlated with short session lengths. In fact the correlation

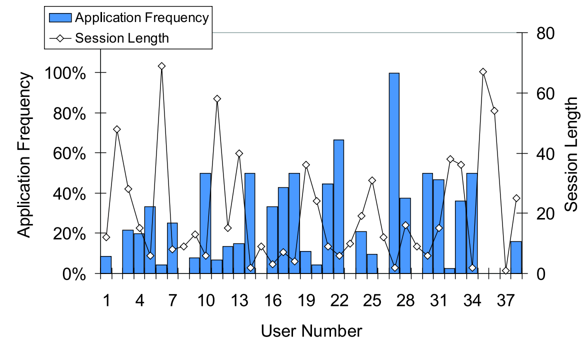

Figure 6: Compound critique application frequency & associated session length on a user-by-user basis.

between these two measures across the 38 trialists is -0.57. This is an important result as it suggests that those participants that availed of compound critiques tended to find their target camera in relatively few cycles. For example, user 6 availed of compound critiques only 4% of the time and presented with the longest session length of 69 cycles. In contrast, users such as user 3, 10 and 14 selected more compound critiques and benefited from much shorter recommendation sessions.

4.3 Recommendation Efficiency

The above relationship between compound critique application frequencies and average session length is consistent with the results of our previous artificial-user trials. One of the key problems with an over reliance on unit critiques is that the user makes relatively slow progress through the product space, and often follows false leads as they focus on one feature to the detriment of others. For instance, we often find that users critique features such as *price* until such time as they have received a recommendation that matches their target price, only to find that other features have been compromised (e.g. *resolution* or *memory*). As a result users may have to roll-back on the price critiques in order to find a more acceptable compromise between sets of features. Compound critiques attempt to reduce this problem by allowing the user to simultaneously critique multiple features in a manner that exposes their inter-dependencies.

To better understand the concrete efficiency benefits of compound critiquing we divided the users into two groups during the analysis phase: the *low-frequency* users are those who selected compound critiques less than 25% of the time; the *high-frequency* users are those who selected compound critiques 25% or more of the time. The average session length for each of these user groups is presented in Figure 7 and shows a clear and significant advantage for those users who tended to select proportionally more compound critiques. The low-frequency users used compound critiques just under 9% of the time and required almost 29 cycles per session before they found a satisfactory camera. In contrast, the high-frequency group used compound critiques 44% of the time and located a satisfactory camera with an average of less than 6 cycles. This represents an reduction in average cycle length, for the high-frequency compound critique users, of more than 80%, relative to the low-frequency users.

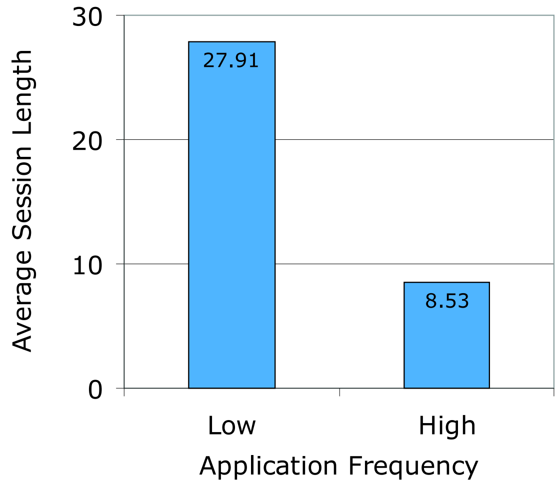

Figure 7: Average session length for low-frequency and high-frequency users.

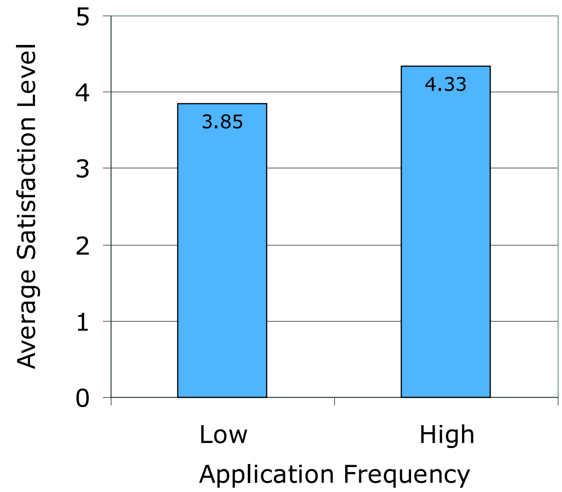

Figure 8: Average satisfaction rating for low-frequency and high-frequency users.

4.4 Recommendation Quality

While the above efficiency results are indeed striking, and a significant improvement on the 40% session length reduction predicted by the artificial-user trials, it is not yet clear whether it is fair to compare these user groups in such a direct manner. For example, it may be the case that the low-frequency users are simply more difficult to satisfy, or that longer sessions might result in more satisfied users.

To investigate this issue we compared the two groups according to their average satisfaction rating, provided as part of a post-trial questionnaire. The results are presented in Figure 8 and clearly show that not only are the high-frequency users every bit as satisfied with their target cameras as the low-frequency users, they actually appear to be marginally more satisfied. The high-frequency users reported a 13% improvement in their satisfaction levels compared to the low-frequency users.

5. DISCUSSION

The results of our live-user trial suggest that dynamic critiquing has the potential to help users find their desired products more efficiently that traditional unit critiques. However, while these results are promising, during the course of the trial a number of important issues were raised by the participants. In its current guise our prototype system presents users with an interface that combines static and dynamic elements according to the principle of *partitioned dynamicity*; see [14]. Accordingly the static interface elements (which include the camera description and unit critiques) are augmented by a separate panel for the dynamic, compound critiques. In this way it was possible to provide a level of interface adaptability (changing and context-sensitive compound critiques) without obscuring the core functionality provided by the primary interface elements. In practice this approach seems to have worked reasonably well. However a number of users identified two important problems with the current dynamic critiquing strategy, discussed below.

5.1 Critique Labeling

Some users complained that the system sometimes presented them with compound critiques that were made up of collections of unrelated features; for example, a compound critique *"Different Strap Type and Different Storage Format"*. The point was well-made that compound critiques made up of unrelated features did not really help the user to focus their search. In response to this we are looking at a number of solutions that make it possible to consider inter-feature relationships during critique generation. For example, one possibility is to look at defining *critique templates* that provide for more meaningful groupings of unit critiques. This approach also has the added benefit that it allows for more useful or intuitive labels to be provided for the compound critiques; so rather than presenting a compound critiques as a collection of unit critiques, we can present a more intuitive high-level label. For instance, the critique template *more professional* could be applied to a set of unit critiques (e.g., *greater resolution, more memory, greater zoom, SLR format*. During critique selection, those compound critiques that matched critique templates could be selected ahead of non-matching critiques and the template titles could be used to provide more meaningful labels for the user.

5.2 Critique Continuity

Other users commented that sometimes the recommender system appeared to 'forget' about their earlier critiques. For example, one user indicated that they had asked for at least a 2x optical zoom during one cycle but that later, when the asked for an increased digital zoom, the recommender suggested a camera with a lower optical zoom. This problem stems from the fact that in the prototype application critiques are not tracked from cycle to cycle; each critique (compound or unit) is applied in isolation. This can lead to a number of *continuity* problems, especially when users are unsure of their requirements. To cater for this we have recently developed and evaluated an extended approach to dynamic critiquing called *incremental critiquing* which successfully solves this problem, and which has been shown to offer further potential efficiency improvements [12]. Very briefly, incremental critiquing maintains a history of a users critiques within a given session and new recommendations are selected not only because of their compatibility with the current critique but also on the basis that they match as many previous critiques as possible. This combination of factors during recommendation improves means that suggests that conflict with past critiques are less likely and helps the recommender to better focus its search through the product-space.

6. CONCLUSIONS

The ability to learn from user feedback is a vital feature of conversational recommender systems. Critiquing has proven to be a useful form of feedback that is especially well-suited to many recommendation tasks because it offers a unique balance between the level of user effort required and its ability to effectively focus search. Usually critiquing-based recommender systems make use of static unit critiques, allowing users to adjust a single feature at a time. In this paper we describe the *dynamic critiquing* approach - an effective user feedback interfacing mechanism that increases a users depth of understanding of the product-space. Essentially, we argue in favour of *compound critiques*, that allow the user to simultaneously adjust multiple features in order to better focus the recommender system. Moreover, we have described a technique for automatically generating compound critiques during each cycle of a conversational recommendation session, and devised a strategy for selecting a small number of high-quality compound critiques for presentation to the user.

In this paper we present a live-user evaluation of our dynamic critiquing approach based on a prototype recommender system. Although the results indicate that not every user is immediately inclined to use compound critiques instead of unit critiques, the results show that those users that do take advantage of compound critiques benefit from significantly shorter recommendation sessions that lead to higher quality purchases.

We argue that dynamically generated compound critiques serve both as a valuable form of feedback and as a type of explanation in the sense that a compound critique explains something about the remaining cases to the user and may help them to make more informed decisions in subsequent recommendation cycles.

7. ACKNOWLEDGMENTS

Sincere thanks is extended to the postgraduate students, from the Department of Computer Science at University College Dublin, Ireland, for participating in the evaluation study described by this paper.

This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361.

8. REFERENCES

[1] Half-day tutorial presented with John Riedl and Joseph Konstan at IJCAI 2003, the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15 2003.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast Discovery of Association Rules in Large Databases. *Advances in Knowledge Discovery and Data Mining*, pages 307–328, 1996.

[3] J. Anthony, B. Großmann-Hutter, L. Marchand, R. Rummer, T. Bohnenberger, and F. Wittig. When Actions Have Consequences: Empirically Based Decision Making for Intelligent User Interfaces. *Knowledge-Based Systems*, 14:75–92, 2001.

[4] R. Burke, K. Hammond, and B. Young. Knowledge-based navigation of complex information spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 462–468. AAAI Press/MIT Press, 1996. Portland, OR.

[5] R. Burke, K. Hammond, and B. Young. The FindMe Approach to Assisted Browsing. *Journal of IEEE Expert*, 12(4):32–40, 1997.

[6] Z. Hu, W. Chin, and M. Takeichi. Calculating a New Data Mining Algorithm for Market Basket Analysis. *Lecture Notes in Computer Science*, 1753:169–175, 2000.

[7] K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems. In P. D. Bra, editor, *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH-04)*, 2004. Eindhoven, The Netherlands.

[8] K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. Thinking Positively - Explanatory Feedback for Conversational Recommender Systems. In P. Cunningham and D. McSherry, editors, *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop*, pages 115–124, 2004. Madrid, Spain.

[9] L. McGinty and B. Smyth. The Role of Diversity in Conversational Systems. In D. Bridge and K. Ashley, editors, *Proceedings of the Fifth International Conference on Case-Based Reasoning (ICCBR-03)*. Springer, 2003. Troindheim, Norway.

[10] L. McGinty and B. Smyth. Tweaking Critiquing. In *Proceedings of the Workshop on Personalization and Web Techniques at the International Joint Conference on Artificial Intelligence (IJCAI-03)*. Morgan-Kaufmann, 2003. Acapulco, Mexico.

[11] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Dynamic Critiquing. In P. G. Calero and P. Funk, editors, *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04)*. Springer, 2004. Madrid, Spain.

[12] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Incremental Critiquing. In *Proceedings of The 24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence. (AI-04)*, 2004. Cambridge, UK.

[13] J. Schafer, J. Konstan, and J. Riedl. Electronic Commerce Recommender Applications. *Journal of Data Mining and Knowledge Discovery*, 5, 2000.

[14] D. Weld, C. Anderson, P. Domingos, O. Etzioni, T. Lau, K. Gajos, and S. Wolfman. Automatically Personalizing User Interfaces. In *Proceedings of the 18th International Joint Conference Artificial Intelligence (IJCAI-03)*, 2003. Acapulco, Mexico.