An Examination of User-Focused Context-Gathering Techniques in Recommendation Interfaces

Eoin Hurrell
CLARITY: Centre for Sensor Web Technologies
School of Computing, Dublin City University
Dublin 9, Ireland
eoin.hurrell@computing.dcu.ie

Alan F. Smeaton
CLARITY: Centre for Sensor Web Technologies
School of Computing, Dublin City University
Dublin 9, Ireland
alan.smeaton@dcu.ie

ABSTRACT

Attempts to capture context within applications take a wide variety of forms. While it is generally accepted that a user's current context shapes how they perceive and interact with a system such as a recommender we here explore a novel method of interacting with the user to gain a conceptual understanding of their own frame of reference. By drawing on a more human-centric approach we show that users accept and participate in sharing of context readily as part of an interactive system.

General Terms

Human Factors, Experimentation

Keywords

Recommender systems, etc.

1. INTRODUCTION

The idea of somehow capturing and using a user's context as s/he uses some computer system spans multiple disciplines, including psychology, philosophy, anthropology as well as the technical aspects in engineering and computer science. Generally the term context-awareness denotes the ability to ambiently capture and make use of the user's context without interfering with the task the user is trying to accomplish [4]. Each field that has explored context tends to take a different approach to the subject, with anthropologists and sociologists conducting ethnographic studies [5] and a great deal of computer science and engineering work concerned with the methodology of collecting and using directly sensed data from the subject.

The importance of knowing context in any kind of user interaction cannot be overstated, as it is the means by which users and systems come to a mutual understanding. Derrida, whose field of deconstruction probes the context of works, said "There is nothing outside the text" [3], which he later explained as "There is nothing outside context". From a HCI perspective this can be seen as foreshadowing the usefulness of contextual data in driving the over-arching narrative of interaction within a system.

Context-awareness is a key requirement of human-centric computing systems, allowing them to adapt and to form meaningful interactions by accounting for the user's current needs, task, environment, etc.. Yet there exists an issue; purely sensed context needs a great deal of data to infer patterns of usage and meaning, for example GPS coordinates could tell that a user visited a shop twice, which could either mean they are a frequent customer or they bought something that was faulty and had to be returned, meanings that imply vastly different levels of customer satisfaction for example.

Barkhuus and Dey [2] explored and defined three levels of user interactivity related to context-awareness: personalisation, passive context-awareness, and active context-awareness. Personalisation makes use of user settings, whereas context-aware applications make more dynamic use of context or sensor information. Active context-aware systems automatically make context-based changes, which Barkhuus and Dey found through evaluation to be preferable to passively offering the option to change. Our work explores the collection of this data.

2. RELATED WORK

From a HCI perspective the dynamic adaptation of systems according to user circumstances is becoming increasingly desirable to create adaptive designs around which users' experiences of a system can be said to be truly personal. Much focus has been given to context-gathering in the mobile application space, as smartphones and the like come equipped with many easily-used sensors. But while so much raw data can be acquired from sensors and usage patterns can be learned and formed automatically we wish to examine whether it could be beneficial to give users the power to literally put this data into context.

Since context is such an abstract concept, information that forms a context can be represented in various formats. Much work has been done in computer science to provide middleware[1] to fuse the multitude of contextual sources a system might need in order to be fully context-aware. Here we look at giving the user a method to express the meaning of their own contextual data collected, providing semantics at the point of collection, rather than after collecting enough data to determine if there are patterns. Our focus in this paper is a recommendation system that collects contextual information about a user. Contextual recommendation has a rich background of related work, making use of sensed data such as location or time to improve the quality of the items recommended.

While the distinction between "active" and "passive" modes of context use is made clear by Barkhuus and Dey in [2], here we explore "transparent" and "opaque" modes of context *collection*. Gathering context from sensors transparently and ambiently so the user does not even have to be made aware of the collection process and where it does not interfere with the user's task, is the current standard. In an attempt to aid the definition of semantic meaning around this context-sensing data we built a system to test a method of querying the user prior to system interaction, opaquely gathering the reason behind the data gathered.

3. GATHERING CONTEXT

Our experiment in context-gathering made use of a recommender application to help users find movies that might be of interest to them, a system we now describe.

3.1 Interactive Recommendation Approach

Our recommendation approach centres around the idea of users choosing their area of interest. We provide a means to give feedback based on the reaction, either reasoned or reactionary, of "I don't think I'd like that" or "I'm interested in that". While this reasoning may initially seem to be fussy, imprecise, and difficult to capture it is nonetheless an important part of decision-making for users. In contrast to early work on case-based conversation [6] this is not the same a expressing "I'm interested in more like this", rather the process proceeds like a conversation in which indicating a preference produces potentially entirely new recommendations. Our approach also differentiates a person's *immediate* interests, i.e. at a given point in the present interactive session's preference indications, from their *continuing* or long-standing interests collected when they rated items and as such it reflects the constantly-evolving nature of users' information needs as they continuously update both their knowledge and their needs as they are presented with new recommended items. This approach creates a system with an in-built expectation of interaction, novel for this reason and allowing us to incorporate a short survey at the beginning of each recommendation session in order to gather contextual information about the user. The responses to this question formed part of the evaluation in Section 4.

The strength of collaborative filtering (CF) recommendation lies in using rating information to understand users in comparison to others, to place them in a neighbourhood of peers or find items similar to the ones they like. Our interactive approach uses this understanding of items through ratings, by focusing on how popular an item is, and how well it is rated. The popularity of an item ($Pop(i)$) for our purpose is its rating coverage, i.e. the number of people who have rated it, while the measure of how well-rated it is comes from the average rating:

$$Pop(i) = \sum ratings(i)$$
$$Rated(i) = Avg(ratings(i))$$
$$Point(i) = (Pop(i), Rated(i))$$

From this, any item in the collection of items can be represented on a graph of popularity against average rating. This graph is a representation of the collection that is equally valid in all areas to user tastes. That is to say that aficionados of items such as books or film can understand that there are audiences for both well-rated niche items and items that everyone has seen but wouldn't be their favourite.

Our approach works iteratively. A session begins with a user having access to the entire collection of items. Two indicative movies are randomly picked from the collection, one to represent popular items and another to represent highly rated ones. The *popular* indicative movie is chosen from the movies with at least half the average number of ratings, while the *highly rated* one is chosen from movies with at least half the average rating of the collection. These are chosen from the movies considered to be *of interest* to the user, the set that they are working to decrease at each iteration. The two options are shown to the user who is asked "Which do you prefer?". Additionally, a list of recommendations from the collection is generated and the top five recommendations are shown below the question, both to give users a sense that their interaction is having a meaningful effect and to show new suggestions they may be interested in. Once the user chooses either option, the set of items from which recommendations and interface choices are generated is partitioned. We use a bounding technique here, which has been explored in search tasks [7] but not in recommendation, especially as a means by which conversation can occur. Here we use lower rather than upper bounds, to signify *least acceptable value*.

A new pair of options, with a new list of recommendations, is posed to the user. The degree to which the items are partitioned depends on the density of the collection and our aim is to reduce the set to produce a visible change in recommendations through every action thus developing the user's sense that each item of feedback is making a difference. This continues until the user stops answering questions or there are less than ten items to choose from, at which point all ten are presented. The user refines the recommendation provided for them by culling from the collection, movie items which they feel are of no interest to them. The system asks "which of the following two items do you think you would prefer?", to which the user provides a preference which can be used to narrow their possible recommendations. In order to do this without intrinsic knowledge of the items themselves, as CF sees items we have explored using the information provided by ratings.

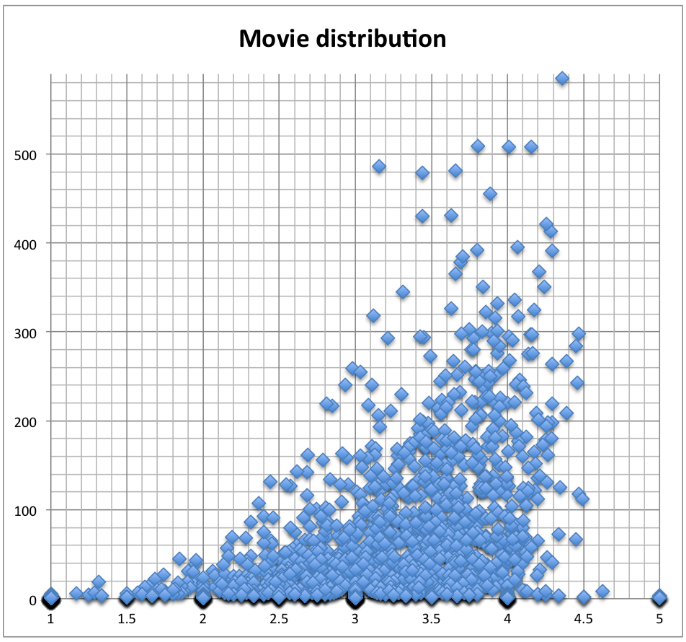

Figure 1: Distribution of items in the MovieLens dataset when plotted using our measurements.

We guide the user through a series of decisions that subdivides the possible recommendation space according to their relative preferences using a pair of lower bounds, reducing the portion of the collection we define as *of-interest* to the user. This differs from critiquing, where the conversation is based on domain-specific traits. Our approach therefore works with any collection of items that do not have descriptive metadata, making it useful in situations where none exists.

3.2 The MovieQuiz Application

We developed an application to evaluate our method using the well-known MovieLens 100K dataset which contains 100,000 ratings from 1,000 users on 1,700 movies. We use this as the seeding data for recommendations, with actual user interaction and rating data collected from live users. Our example application uses movies, where "blockbuster" films and "indie hits" represent equally valued possible recommendations. Prior to engaging with the conversational interface users were asked to rate a set of 10 randomly-selected most-popular films in the collection from a list presented to them.

We use a k-NN item-based collaborative-filtering algorithm to form recommendations. This algorithm is used for traditional recommendation and we adapt it here for our conversational approach as detailed above, to recommend from a subset. The adaptation is conceptually straightforward, in that we modify it to recommend only films with an average rating greater than or equal to X and with Y ratings, where X and Y are determined by the user's interactions with the conversational interface on a per-session basis. Any recommendation algorithm that can be so altered could be used for this approach.

In order to enable traversal of large datasets by the user, the affordance of the interface we develop must allow interaction while informing the user of the current best recommendations. Our basic layout, as shown in Figure 3, is to prompt the user with two candidate preferences. Not shown below the choices is a list of the top five recommended films from the collection according to the current partitioning. Users are given the title and genres of the movie, along with a poster and links to searches for the film on IMDB[1] and on YouTube[2].

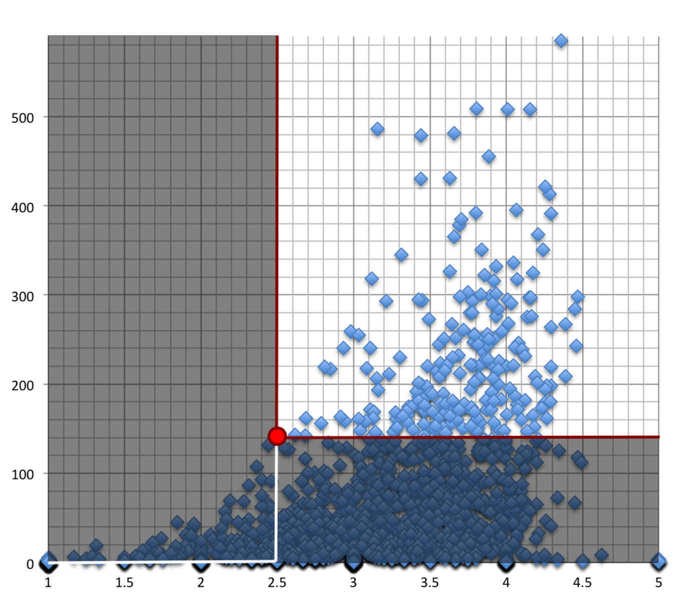

Figure 2: The collection partitioned according to a user's choices in the system

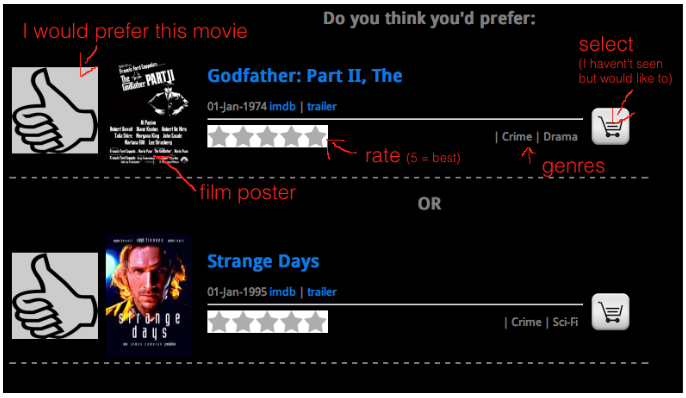

Figure 3: The MovieQuiz application

4. RESULTS

4.1 Gathered Context

During an on-line evaluation of our system, users logged into the website to use the recommendation system. They were presented with a survey prior to each series of interactions (of which there were multiple per session) which asked them the purpose of the recommendation.

We asked three multiple-choice questions of users to put their next interactions in context within the system. These questions were tailored to the task in order to greater understand the users' need and actions and are shown in Table 1. Importantly, the questions demonstrate the intent behind a context, i.e. "I am here to browse", distinct from the sensed details of "I am in a shop" or even "I am in the large music shop on Y street in X city". This was in order to supplement any automatically-sensed data and provide a more conceptually accurate context.

At the start of each session we also recorded GPS location, operating system used on the device, browser and IP address. Depending on the browser security settings, a user could choose to not share their sensed data with the system. The summary data is shown in Table 2.

From the figures in Table 2 we see that users more readily answered the survey than shared sensed data. In less than 25% of cases the user choose to share sensed data, indicating an issue of trust with the system. The survey generated a large number of responses as it was a key step in the system. Almost 30% of the collected survey answers are different from the default, indicating the need for good defaults that make sense. In our case we allowed for the possibility that the user placed no special value on their current context.

After the online evaluation we asked 34 of the users about the system. 28 said they would use it again, showing a general acceptance for this sort of mechanism for captur-

[1]http://www.imdb.com
[2]http://www.youtube.com

Table 1: Survey questions

Question	Possible Answers			
What are you here for ?	just browsing	looking to buy	sharing my opinion	
Are you in a group ?	just me	me and a friend	part of a couple	party or big group
Where are you ?	nowhere important	point-of-purchase	researching	

Table 2: Context statistics

247	users
614	sessions
4.1	average context entries per person
149	entries of sensed context
30	different operating system/browser combinations
864	entries of surveyed context

ing context via dialogue. Our method of conceptual context shows potential for framing a single use of a recommender system as part of a larger narrative, for example “This user likes vastly different films when they are browsing with their partner”. By focusing the user on interacting with the system they are comfortable sharing beneficial information that they are unwilling to share through direct sensor activity, and have some understanding of how context is viewed by the system. User trust in context-gathering is an area that needs to be further explored.

5. CONCLUSIONS AND FUTURE

When users respond to recommendations with ratings or other straightforward interactions such as “likes” this can represent a missed opportunity to capture what could be a deep personal expression of an opinion on a recommended item. From the preliminary work that we have reported we found that giving users a method by which we can provide a frame of reference for these opinions and allowing a richer kind of user feedback appears to be a positive thing, as long as the system is careful not to impose meaningful context when none is perceived by the user.

For our future work we will continue to research this facility for allowing user explanation of circumstances or contexts surrounding recommendation in other domain areas. In work to date on recommender systems and user feedback, the items to be recommended are almost always atomic items, such as a movie in the experiments reported in this paper. In our own future work we will focus on instances where the actual item to be recommended is built up through dialogue with the user, thus extending the kind of mutual understanding of context between user and system introduced here.

Acknowledgments

This work is supported by Science Foundation Ireland under grant 07/CE/I1147.

6. REFERENCES

[1] D. Athanasopoulos, A. V. Zarras, V. Issarny, E. Pitoura, and P. Vassiliadis. Cowsami: Interface-aware context gathering in ambient intelligence environments. *Pervasive and Mobile Computing*, 4(3):360 – 389, 2008.

[2] L. Barkhuus and A. K. Dey. Is context-aware computing taking control away from the user? three levels of interactivity examined. In *UbiComp 2003: Ubiquitous Computing, 5th International Conference*, volume 2864, pages 149–156, 2003. http://www.odysci.com/article/1010112985191527.

[3] J. Derrida. *Of grammatology / by Jacques Derrida ; translated by Gayatri Chakravorty Spivak.* Johns Hopkins University Press, Baltimore :, 1st american ed. edition, 1976.

[4] A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, pages 4–7, 2001. http://www.odysci.com/article/1010112991760792.

[5] C. Goodwin and A. Duranti. *Rethinking context : language as an interactive phenomenon / edited by Alessandro Duranti and Charles Goodwin.* Cambridge University Press, Cambridge [England] ; New York :, 1992.

[6] L. McGinty and B. Smyth. Evaluating preference-based feedback in recommender systems. In M. O’Neill, R. Sutcliffe, C. Ryan, M. Eaton, and N. Griffith, editors, *Artificial Intelligence and Cognitive Science*, volume 2464 of *Lecture Notes in Computer Science*, pages 209–214. Springer Berlin / Heidelberg, 2002.

[7] A. F. Smeaton and C. J. van Rijsbergen. The nearest neighbour problem in information retrieval: An algorithm using upperbounds. In *Proceedings of the Fourth SIGIR Conference*, pages 83–87. ACM, 1981.