Collaborative Filtering Is Not Enough? Experiments with a Mixed-Model Recommender for Leisure Activities

Nicolas Ducheneaut[1], Kurt Partridge[1], Qingfeng Huang[1], Bob Price[1], Mike Roberts[1], Ed H. Chi[1], Victoria Bellotti[1], Bo Begole [1]

[1] Palo Alto Research Center, 3333 Coyote Hill Road,
Palo Alto, CA 94304, USA
{nicolas, kurt.partridge, qingfeng.huang, bob.price,
michael.roberts, echi, victoria.bellotti, bo.begole}@parc.com

Abstract. Collaborative filtering (CF) is at the heart of most successful recommender systems nowadays. While this technique often provides useful recommendations, conventional systems also ignore data that could potentially be used to refine and adjust recommendations based on a user's context and preferences. The problem is particularly acute with mobile systems where information delivery often needs to be contextualized. Past research has also shown that combining CF with other techniques often improves the quality of recommendations. In this paper, we present results from an experiment assessing user satisfaction with recommendations for leisure activities that are obtained from different combinations of these techniques. We show that the most effective mix is highly dependent on a user's familiarity with a geographical area and discuss the implications of our findings for future research.

Keywords: Recommender systems, hybrid models, evaluation.

1 Introduction

The variety of opportunities available today makes it difficult to find items that suit an individual's tastes. On the Internet, people can enter search terms to seek items but this only works for items that a person knows how to describe in some form (usually simple keyword lists) and potentially misses a variety of items that the person might like if only she knew of them. To address this issue, recommendation systems use a model of a person's interests to suggest items that are likely to be close to an individual's tastes. They have proven effective at recommending content such as movies, books, music, and other kinds of products [18].

Early recommender systems were mostly designed for access from a personal computer [8, 10]. But more recently the explosion in the number of mobile devices has led researchers to examine how recommender systems should be designed for use on-the-go [15]. With mobility, recommender systems are used in contexts that change frequently, as the user moves about and engages in a variety of activities. The incorporation of this contextual information into the recommendation process has been identified as a central challenge for the recommender systems community [2]. As such there has been some recent work to extend the power of recommender systems to leverage context [1] in the physical world primarily for providing tourist information (e.g., [6, 16]), as well as some early efforts at providing mobile restaurant recommendations (e.g., [17, 21]).

Another common thread in recommender systems research is the need to combine recommendations techniques to achieve peak performance [5]. Collaborative filtering is currently the most familiar and most widely implemented technique [13]. It aggregates ratings assigned to items by its users, recognizes commonalities between them on the basis of these ratings, and generates new recommendations based on inter-user comparisons. But a variety of other approaches are available: content-based, demographic, utility-based, and knowledge-based techniques have all been actively researched and combined [2, 5], with varying degrees of success. The challenge here is to find a combination of algorithms that would best satisfy a user, given her needs and circumstances.

We recently designed and implemented a mobile, leisure-time recommender system (codenamed Magitti) to address these two challenges. Magitti delivers recommendations that consider both the user's contextual data (location, weather, current reading patterns, etc.) and their tastes (see [3] for more detail on the system's implementation). To determine how to best combine all these information sources, we conducted experiments to find out how to make the most effective recommendations. In this paper, we present results from these experiments indicating which combination of recommender techniques might lead to the highest user satisfaction when using such mobile, context-aware recommender systems.

2 Magitti: Recommending Leisure

The prototype for Magitti serves needs broadly similar to those addressed by other mobile recommendation systems such as CitySearch, Yelp, and Zagat. In addition, Magitti performs sophisticated user modeling that combines various kinds of user preference data, contextual data, and activity inference. The activity inference mechanism, which is described elsewhere [3], infers which general category of information the user is most likely to be interested in. The categories are "EAT" (restaurants and cafes), "SHOP" (retail stores), "SEE" (theaters and museums), "DO" (parks and sporting events), and "READ" (news and lifestyle articles). The inference is not perfect, but the user can explicitly correct wrong inferences.

Once the activity category is determined, the recommender engine ranks the items in the chosen category by combining results from a variety of models to compute each item's *utility*. We adopt a hybrid approach [5] for the recommender in order to address the issues introduced by mobility and the leisure domain. Indeed, it has been proposed that different recommender algorithms are better suited to certain information seeking tasks. For instance, CF may generate more serendipitous recommendations, while a content-based recommender might produce more homogeneous results with high similarity [13]. Since Magitti needs to support both serendipitous discovery and directed search and planning, it makes sense to combine

Table 1: Models used to calculate the ranking of recommended items.

Model Name	Details
Collaborative Filtering Model	Unknown ratings are estimated from user similarity on known ratings using the Pearson correlation coefficient.
Stated Preferences Model	Explicit preferences for attributes (e.g., "Japanese cuisine") are multiplied by a particular item's attributes, summed, and normalized.
Learned Preferences Model	Time-dependent attributes are learned from proximity to other items in historical traces for each user. E.g., frequent visits to Mexican restaurants would automatically infer a preference for them [3].
Distance Model	Utilities are scored according a thresholded exponential decay function of the distance from the user.
Content Preference Model	Keywords from viewed web-page content are extracted using TF/IDF [2], and item scores are determined using cosine similarity to the keywords in the item's text description.
Future Plans Model	Calendar entries and planning-related natural language expressions in messages influence particular attributes (e.g., the message "How about pizza tonight?")

different recommendation techniques. A hybrid approach also allows us not only to integrate contextual models when such information is available, but also to return useful results when it is not.

Currently, Magitti combines the six models in Table 1. These models can be combined in a variety of ways. In the current implementation we use a simple weighted linear combination for its flexibility. With this approach, it is easy to perform post-hoc credit assignment and adjust the contribution of each model to the total utility accordingly [5]. This lets the recommender not only adapt to the results of the evaluation, but also, given enough time and data, learn a weight for each model that is independent of each model's time-varying parameters.

3 Evaluation

To better understand how to combine recommendation techniques to be most satisfying to users, we conducted a qualitative evaluation in which users assessed the usefulness and serendipity of recommendations obtained from these models, both individually and in various combinations.

3.1 Method

We recruited 16 participants through a mailing list internal to our organization (confidentiality agreements prevented us from recruiting external users). The participants were offered a $20 gift certificate for completing the evaluation. The participants ranged in age from 20 to 60 years old; 11 were male, 5 were female.

To facilitate participation and data collection we created a web-based system that reproduced the main features of Magitti's recommendation list. This way, participants were free to complete the evaluation tasks at work or from their home, at their own pace. We were also able to collect much more data than would have been possible in an *in situ* study. Since our focus was on the quality of recommendations and not on activity detection, we limited our content to information about one activity, eating, in order to allow comparison between lists of recommendations for the users.

The participants started by rating a list of local restaurants that they had already visited on a scale from 1 to 5. These ratings were necessary to bootstrap the CF model. Prior to the experiment, we had 20 members of our team and other employees rate the same set of restaurants, which produced enough comparison data to generate collaboratively filtered recommendations for each participant.

The participants then entered explicit preferences to be used by the Stated Preferences model. They were presented with a list of all possible attribute/value pairs available in our restaurant database and asked to assign a value to as many as they wished. The participants were told that scores should lie between 0 and 1.

After the initial setup phase we asked the participants to select one of five street locations and assume they were being teleported to it on the next coming Friday at 6pm to have dinner there. The locations were all well-known neighborhood dining "hotspots." The system then sequentially presented them with five different lists of ten restaurants, each generated by one or several of our algorithms. The restaurants were sorted in decreasing order of utility, and the algorithm used to generate the list and each restaurant's utility score was not shown to the participants. For each list, the participants answered "Would you be interested in dining at this restaurant?" using one of three choices: "(1) Definitely," "(2) Maybe," or "(3) Probably not." The participant could also decline to answer. We also asked them to say whether they had already been to each restaurant they chose to evaluate.

Beyond its name and address, our Web interface also displayed the following information about each entry, just like Magitti would: (1) its relative distance to the participant's chosen location, (2) the average past user ratings, (3) pricing information, and (4) (if available) three user comments. To help investigate various biases we also asked participants two questions about each list:

(1) "To what degree did the user comments affect your decision (from 1 [not at all] to 10 [very much])?"

(2) "To what degree did the existing ratings influence your decision (from 1 [not at all] to 10 [very much])?"

We also asked the participants to complete the same tasks for other locations among the five, if time permitted.

3.2 Algorithms tested

The nature of our experiment prevented us from testing all the recommendation models available in Magitti. Models relying on historical information would have required extensive use of the mobile device to generate useful recommendations for the participants, which we could not achieve during the timeframe of this study. As such, we did not evaluate Learned Preferences and Future Plans. Moreover, privacy concerns about collecting participants' Web usage data prior to the experiment prevented us from bootstrapping the Content Preference Model. Without such data, the model is effectively deactivated. Consequently, the five algorithms studied were: 1) Collaborative filtering; 2) the Stated Preferences model; 3) the Distance model; 4) a mixed algorithm using (1-3) with even weights; 5) a mixed algorithm using (1-3) with respective weights of (0.2, 0.75, 0.25).

This set of algorithms is diverse and interesting for comparison purposes. CF is a good representative of a high-serendipity model with high dependence on user input. The Stated Preferences model is a good example of a static utility-based approach that produces results consistent with a user's explicit tastes, but without much variability. And finally the Distance model illustrates how contextual data (in this case, location) can be integrated with other recommendation techniques. The weights for Algorithm 5 were determined empirically by the experimenters after using the system over several months—they are closest to what we felt was the "ideal" combination, based on our usage patterns.

Magitti has a facility for automatically learning weights over time that maximizes the models that best predict the items that the user will select. However, it requires more usage data than we could conveniently collect for this study, and therefore the weights of each model were kept constant for the duration of each experimental session.

3.3 Results

Our 16 participants evaluated a total of 99 recommendation lists. Most completed the tasks for one location only, a few did two. Twelve (out of the 16) participants rated results generated by all five algorithms. This resulted in a total of 940 restaurant recommendation assessments. Some did not assess all items in a list, which was allowed by the protocol.

Usefulness of Recommendations. In order to evaluate the usefulness [13] of each algorithm we first computed its average "score" by assigning a value of 1 to "definitely", 0.5 to "maybe" and 0 to "probably not" for each user assessment of an algorithm's recommended items. Table 2 summarizes the results. The results indicate that CF seems to dominate, offering on average recommendations that are more likely to be followed than other algorithms. The difference is statistically significant, $F(2.3816) = 13.0211$, $p<0.01$.

It is possible, however, that users might favor familiar places. To explore this possible confound we repeated the above analysis, this time excluding places that had already been visited (see Table 3). The results are again statistically significant,

Table 2: Average usefulness score for each algorithm.

Algorithms	N ratings	Average	Variance
CF	110	0.75	0.13
PREFS	220	0.54	0.15
DISTANCE	200	0.43	0.15
ALL EQUAL	180	0.56	0.14
CUSTOM WEIGHTS	210	0.56	0.14

Table 3: Average usefulness scores, excluding already visited locations.

Algorithms	N ratings	Average	Variance
CF	46	0.57	0.14
PREFS	154	0.46	0.12
DISTANCE	139	0.37	0.13
ALL EQUAL	112	0.48	0.12
CUSTOM WEIGHTS	135	0.45	0.13

$F(2.3872) = 3.4395$, $p<0.01$. CF retains its edge but the effect is less dramatic than before.

The Novelty Factor. While CF appears to stand out in terms of the usefulness of its recommendations, it did not generate as many novel recommendations as we had expected. The differences between Table 2 and Table 3 already pointed in this direction and we therefore decided to investigate the issue in more depth by computing the average number of novel items returned by each algorithm.

Table 4 clearly shows that CF yields the smallest number of novel items, despite its reputation as the most serendipitous algorithm [13]. The results are statistically significant, albeit a bit weaker than before: $F(2.5571) = 2.8553$, $p<0.05$. It is interesting to note that models based on distance or stated preferences greatly outperform CF. As a consequence the two mixed models also score higher on novelty since they mix the three other algorithms (CF, Preferences, and Distance) in various proportions. The custom weights model (0.2, 0.75, 0.25) scores much better than an even mix and it is even fairly close to the highest score (7.00 vs. 7.92). This confirms our intuition, based on system usage, about the overall balance that needs to be achieved between models to generate the most serendipitous recommendations, which was the main goal of our design. In particular, it looks like while CF is clearly useful (especially for people unfamiliar with an area) and needs to be taken into account, its overall contribution needs to be downplayed against more "static" models like stated preferences and distance to optimize novelty. But these results also show that this mix is still not ideal and could be further improved upon.

Table 4: Average number of items classified as "novel" by the participants.

Algorithms	N users	Average	Variance
CF	8	4.25	5.36
PREFS	12	7.92	7.72
DISTANCE	11	6.55	3.47
ALL EQUAL	12	5.83	7.06
CUSTOM WEIGHTS	12	7.00	7.64

Another interesting discovery is that the number of items rated "definitely worth visiting" is inversely correlated with the number of novel items generated by an algorithm (r=0.79). This might be a reflection of some risk aversion from our participants when choosing restaurants: while exotic places are attractive, they might also disappoint. This is an interesting illustration of the tension created by mixing the results from heterogeneous models in the same list and potentially argues for more user control over which model should be on or off, as suggested by Schaefer et al [19].

Predictive Accuracy. We initially designed our system with a Model Weights Learner that automatically adjusts the contribution of each individual model based on a user's implicit feedback. However, this only works if at least some of the models predict a utility for an item that is close to the user's true rating of this item. We therefore looked into the predictive accuracy of each algorithm.

Table 5 lists the average error for each algorithm and each participant. For each recommended item, we subtracted the user's assessment (by again mapping "definitely" to 1, "maybe" to 0.5, and "probably not" to 0) from the item's predicted utility. We then averaged these values over all items recommended by a given algorithm. This gives us a sense of how much an algorithm over- or underestimates item values.

Several interesting trends can be seen. First, it is clear that the accuracy of any algorithm varies greatly across users. For instance, the Stated Preferences model overestimates the value of items for P1 (0.21), but also underestimates the same value for P12 (-0.71). This is a clear argument in favor of a dynamic model weights learner that would adjust each model's contribution over time on a per-user basis (using the same examples, the learner should downplay Stated Preferences for P1, but increase its contribution for P12). Second, the average error across all participants indicates that some models tend to over- or underestimate a user's value of an item. For some algorithms this is not surprising: for example, the Distance model returns maximum utility (that is, 1 or close to it) for all items that are within range, but the probability that all of them will be useful to the user is low—consequently, the model often overestimates the value of an item (mean error: 0.32). More surprisingly, it looks as if CF overestimates value overall (mean error: 0.17) while Stated Preferences underestimates (mean error: -0.15). The differences are statistically significant, $F(2.5306) = 19.1794$, $p<0.01$. Since the three models appear to "pull" in opposite

Table 5: Average error (deviation from user rating) for each algorithm and each participant. Cells are blank if too little data was available to compute the error.

	CF	PREFS	DISTANCE	ALL EQUAL	CUSTOM WEIGHTS
P1		0.21		0.05	0.01
P2	0.02	0.08	0.32		0.04
P3			0.34		-0.40
P4		-0.42	0.47	0.03	-0.17
P5		-0.35	0.23	-0.11	-0.63
P6	0.13				
P7		-0.22	0.29	0.00	-0.45
P8	0.25	0.02	0.21	-0.22	-0.12
P9	0.09	-0.10	0.16	-0.08	-0.15
P10	0.50	-0.06	0.48	0.24	-0.09
P11	-0.07	-0.08	0.24	-0.09	-0.38
P12	0.30	-0.71	0.38	-0.23	-0.50
P13		-0.42	0.48	-0.02	-0.39
P14	0.30	0.07	0.19	0.12	-0.08
P15				-0.06	-0.36
P16	-0.02	0.05	0.37	0.08	0.02
MEAN	0.17	-0.15	0.32	-0.02	-0.24
SD	0.19	0.26	0.11	0.13	0.21

directions, the algorithm mixing them all with equal weights is overall fairly accurate, deviating from a user's assessment by only -0.02 on average.

Influence of Comments and Ratings. Magitti does not limit itself to displaying a ranked list of recommended items based on information internal to each algorithm. As we mentioned earlier, the device's interface also displays the average user rating for each item, as well as detailed comments from other users. We wanted to assess the influence of this additional information on a user's final decision since earlier research shows it can have a significant impact [7].

Table 6 summarizes our participants' answers to the two questions they received after assessing each recommendation list. Several interesting trends can be seen.

First, it looks as if comments have more influence than ratings (the difference is significant, $F(3.8893) = 46.4293$, $p<0.01$). It is clear that users pay attention to reviews written by other users and that these reviews affect their decision, probably not to the point of totally overriding an algorithm's ranking but still with some significance (the average reported influence is close to 6/10). The detailed nature of these reviews (many are several paragraphs long) and the fact that they come from human beings and not from the system probably both play a role in their impact. Conversely, a simple average rating is probably too limited to have a great influence.

Table 6: Average influence of comments and ratings for each user.

	Influence of…	
Participant	COMMENTS	RATINGS
P1	2.5	1.7
P2	5.0	3.6
P3	6.0	1.0
P4	6.4	6.6
P5	6.5	2.5
P6	7.0	7.2
P7	7.7	3.8
P8	3.3	5.0
P9	2.9	1.1
P10	6.0	4.0
P11	7.0	1.0
P12	8.0	5.8
P13	6.0	4.8
P14	10.0	1.6
P15	6.0	1.7
P16	9.5	4.0
MEAN	5.7	3.2
SD	2.8	2.4

Second, it is clear that the impact of this additional information varies greatly across users. For instance, while P1 claimed being almost "immune" to the effect of comments (2.5/10), P10 was greatly swayed by them (10/10). The differences between users are significant both for comments ($F(1.7885) = 8.1037$, $p<0.01$) and ratings ($F(1.7907 = 17.1247$, $p<0.01$). Again, this great variability illustrates how difficult it can be to satisfy all users with a "one size fits all recommender," which argues in favor of rich interfaces and dynamic, adjustable recommender systems.

Another way to look at the same data is to average the influence of comments and ratings by algorithm type rather than participants. This is shown in Table 7. There does not appear to be much variability across algorithms. The differences are not significant either for comments ($F(2.4685) = 0.6253$) or ratings ($F(2.4685) = 0.6373$). User-generated data like comments appear therefore to have a purely subjective influence that does not affect one type of recommender algorithm more than another.

Table 7: Average influence of comments and ratings for each recommender algorithm.

Algorithms	COMMENTS	RATINGS
CF	5.2	2.5
PREFS	6.3	3.7
DISTANCE	5.1	3.2
ALL EQUAL	5.8	2.9
CUSTOM WEIGHTS	5.8	3.3

4 Discussion

Collaborative filtering algorithms have been at the core of most recommender systems for more than a decade. As we argued in our introduction, however, the move towards mobile and context-sensitive systems seems to require more complex, hybrid approaches. Our experiments provide evidence as to how much improvement these approaches provide, and what combination of models is most promising.

It is important to mention at first that CF, by itself, still generates recommendations rated as the most useful by our participants – a testament to the value of this algorithm and proof that simplicity is often a virtue. However, our data allowed us to refine this picture and shed light on CF's limitations for mobile and context-aware systems. In particular, it appears that CF is most useful for people unfamiliar with an area. This would make it particularly well suited to applications like a tourist guide.

The problem, however, is that overall, each individual algorithm has a tendency to under- or overestimate utility. Our data shows that, since they each pull in opposite directions, a mix of models with even weights tends to be fairly accurate and assigns a utility to each item that is close to a user's eventual rating. For users highly familiar with an area, such a mix of models generates recommendations that are as useful as CF alone, but more diverse. However, the most diverse and novel recommendations are obtained by combining the models using the custom weights we had chosen based on our own usage. While users would be less likely to follow-up on all of the recommendations from the latter algorithm (it is more error-prone), it is clearly the most serendipitous and probably closest to our design intentions.

Our analyses also clearly show that, to perform well, a hybrid recommender should be customized for each user. There is simply too much variability in the accuracy of each algorithm across users for a "one size fits all" approach to be satisfying. Moreover, our data also indicates that the eventual mix of models and their weights could benefit from a user's input in order to reduce the tension between getting items that are either too exotic or too familiar. This argues in favor of an automated approach to weights learning, but with the addition of transparency and control [9] so that the user can override some of the estimated parameters if desired.

Finally, it is worth noting the considerable influence that user-generated content, like comments about restaurants, can have on some users. For this subpopulation, it is almost as if the system could dispense with complex recommendation techniques in favor of user reviews, since the latter become the ultimate decision factor.

5 Conclusion

Mobile location-based systems are increasingly becoming a practical means for providing context-specific information. Recommending information takes the effort out of location-based information retrieval so that users need no longer shoulder the entire burden of searching for information relevant to them. The most effective solutions will likely combine explicit user input with implicit and/or contextual data.

To understand which combination might be the most useful, this paper has presented results from an evaluation of a context-aware recommendation system that uses a mixture of models to make recommendations. Our data indicates that, for systems targeted at users highly familiar with the intended geographical area of use, collaborative filtering alone is not enough. Indeed, our data shows that a hybrid approach combining collaborative elements with more static preferences and contextual data like a user's location has the potential to generate recommendations that are more novel and useful. However the contribution of each model to such a combination appears to be highly dependent on the user of the system, which argues in favor of an individual, automated model weights learner – provided the weights and types of models are made transparent to the user, in order to allow them to alter the mix of models if they feel the need. Indeed, the hybrid approach has the potential to generate results that are either too exotic (and therefore not acted upon) or too familiar (and therefore ignored). Putting the user in the loop would help ensure the right balance is achieved. The addition of user-generated content like restaurant reviews also helps put the recommendations into context, which some users find very influential.

We believe that these results are broadly applicable to the design of future mobile, activity-based recommendation systems. More research is needed to refine our understanding of the design of such systems—in particular, it would be interesting to conduct an experiment like ours on a larger scale, with the addition of other models, and over a longer usage period. While we plan to investigate these issues in future work, we also hope this paper inspires others researchers interested in the intersection between recommender systems, mobility, and context-awareness.

Acknowledgements. Dai Nippon Printing Co., Ltd. 'Media Technology Research Center' and 'Corporate R&D Division' sponsored this research. Diane Schiano conducted extensive analysis to provide the prior probabilities needed for bootstrapping our activity detection module.

References

[1] Adomavicius, G., Sankaranarayanan, S.S.,Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems, 23 (1). (2005)

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

[3] Bellotti, V., Begole, J., Chi, E.H., Ducheneaut, N., Fang, J., Isaacs, E., King, T., Newman, M., Partridge, K., Price, B., Rasmussen, P., Roberts, M., Schiano, D.,Walendowski, A.: Activity-based serendipitous recommendations with the Magitti mobile leisure guide. In: Proceedings of CHI 2008, (2008)

[4] Breese, J.S., Heckerman, D.,Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of UAI 1998, Morgan Kaufmann, pp. 43-52. (1998)

[5] Burke, R.: Hybrid recommender systems: survey and experiments. User Modeling and User-Adapted Interaction, 12 (4), pp. 331-370. (2002)

[6] Cheverst, K., Davies, N., Mitchell, K., Friday, A.,Efstratiou, C.: Developing a context-aware electronic tourist guide: some issues and experiences. In: Proceedings of CHI 2000, ACM, New York, pp. 17-24. (2000)

[7] Cosley, D., Lam, S.K., Albert, I., Konstan, J.,Riedl, J.: Is Seeing Believing? How Recommender Systems Influence Users' Opinions. In: Proceedings of CHI 2003, pp. 575-592. ACM, New York, (2003)

[8] Goldberg, D., Nichols, D., Oki, B.M.,Terry, D.: Using collaborative filtering to weave an Information Tapestry. Communications of the ACM, 35 (12), pp. 61-70. (1992)

[9] Herlocker, J., Konstan, J.,Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of CSCW 2000, pp. 241-250. ACM, New York, (2000)

[10] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L.,Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM 40 (3), pp. 77-87. (1997)

[11] Kurtenbach, G., Sellen, A.,Buxton, W.: An empirical evaluation of some articulatory and cognitive aspects of "marking menus". Human Computer Interaction, 8 (1), pp 1-23. (1993)

[12] le Cessie, S.,van Houwelingen, J.C.: Ridge estimators in logistic regression. Applied statistics, 41 (1), pp. 191-201. (1992)

[13] McNee, S., Kapoor, N.,Konstan, J.: Dont look stupid: avoiding pitfalls when recommending research papers. In: Proceedings of CSCW 2006, pp. 171-180. ACM, New York, (2006)

[14] McNee, S., Riedl, J.,Konstan, J.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI 2006 Extended Abstracts, pp. 1097-1101. ACM, New York, (2006)

[15] Miller, B., Albert, I., Lam, S.K., Konstan, J.,Riedl, J.: MovieLens Unplugged: Experiences with a Recommender System on Four Mobile Devices. In: Proceedings of HCI 2003, British HCI Group, (2003)

[16] Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P.,Zipf, A.: CRUMPET: Creation of user-friendly mobile services personalised for tourism. In: Proceedings of 3G, pp. 26-28. (2001)

[17] Ricci, F.,Nguyen, Q.N.: Acquiring and revising preferences in a critique-based mobile recommender system. IEEE Intelligent Systems, 22 (3), pp. 22-29. (2007)

[18] Schafer, J.B., Konstan, J.,Riedl, J.: Recommender systems in e-commerce. In: Proceedings of EC99, pp. 158-166. ACM, New York, (1999)

[19] Schafer, J.B., Konstan, J.,Riedl, J.: Meta-recommendation systems: user-controlled integration of diverse recommendations. In: Proceedings of CIKM 2002, pp. 43-51. ACM, New York, (2002)

[20] Schiano, D., Elliott, A.,Bellotti, V.: Tokyo youth at leisure: towards the design of new media to support leisure planning and practice. In: CHI '06 Extended Abstracts, pp. 309-314. ACM, New York, (2006)

[21] Tung, H.-W.,Soo, V.-W.: A Personalized Restaurant Recommender Agent for Mobile E-Service. In: Proceedings of EEE 2004, pp. 259-262. (2004)

[22] Witten, I.H.,Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)