A Game-with-a-Purpose for Recommender Systems

Barry Smyth, Rachael Rafter, Sam Banks
Insight Centre for Data Analytics,
University College Dublin,
Dublin, Ireland.
rachael.rafter@insight-centre.org, barry.smyth@insight-centre.org

Abstract

Recommender systems learn about our preferences to make targeted suggestions. In this paper we outline a novel *game-with-a-purpose* designed to infer preferences at scale as a side-effect of gameplay. We evaluate the utility of this data in a recommendation context as part of a small live-user trial.

Introduction

Recommender systems help us choose what to read, watch and buy, and are now a common feature of online services. They do this by learning about our preferences and relationships in order to generate targeted suggestsions that we are likely to find relevant (Adomavicius and Tuzhilin 2005; Desrosiers and Karypis 2011). The data on which they rely can be difficult to collect at scale and in this paper we consider a novel approach to collecting this data by using a so-called *game-with-a-purpose* (GWAP); see (von Ahn and Dabbish 2004; Law and von Ahn 2009; Salvador et al. 2013; Gligorov et al. 2011; Cooper et al. 2010). In the past GWAPs have been used for challenging tasks such as object recognition and protein folding; might they also be used to help build better recommender systems?

To explore this we describe a single-player matching game in which players attempt to match movies with their friends. These matches can be used to infer the strength of relationships between users and the predicted level of interest a user might have for a particular movie. Both of these types of data can be useful in a recommendation context. We evaluate the utility of this data as part of a live-user trial.

The Recommendation Game

Our game is a simple Facebook app in which a player (p) attempts to match movies (m) by dragging movie icons (posters), floating across the screen, to the avatars of their friends ($Friends(p)$); see Figure 1(a). Each match is rewarded by an audio and graphical flourish and a score. The score is based on whether the friend (f) is known to like the movie, based on their Facebook likes ($m \in Likes(f)$), and how often other players have matched the same movie with the same friend ($PastMatches(m, f)$); see Equation 1.

$$score(p, m, f) = 10 \bullet \left(\alpha \bullet 1[m \in Likes(f)] + \frac{1 - \alpha}{1 + PastMatches(m, f)} \right) \quad (1)$$

A game comprises 5 friends and 18 movies chosen from a mixture of friends' likes and the Rotten Tomatoes "Popular Movies" list. This ensures a mix of movies that are known to be liked and some unkown movies. When movie m is matched with friend f by player p ($match(p, m, f)$), it suggests that the p believes f will like m. If $m \in Likes(f)$ then the match is an indicator that p has a good sense of $f's$ movie likes. We can estimate how well p knows f by the number of these *known* movies that p matches with f ($KnownMatches(p, f)$) as a proportion of all movies that p matches with f ($Matches(p, f)$); see Equation 2.

$$knows(p, f) = \frac{|KnownMatches(p, f)|}{|Matches(p, f)|} \quad (2)$$

If m is unknown to f ($m \notin Likes(f)$) then m is a potential recommendation candidate for f. And, we can estimate f's likely interest in m based on how often other players match m with f and how well these players appear to know f; see Equation 3.

$$interest(m, f) = \sum_{\forall p: match(p,m,f) \wedge m \notin Likes(f)} knows(p, f) \quad (3)$$

Making Recommendations

We use this type of recommendation knowledge to implement two basic recommendation strategies to generate and rank a set of candidate suggestions for a target user u_t. The *crowdsourced* strategy (CS) selects, as recommendation candidates, those movies that were matched with u_t during gameplay but that are not already likes of u_t. These are then ranked in terms of $interest(m, u_t)$.

The alternative *content-based* strategy (CB) generates candidates for u_t based on the set of movies returned by the Rotten Tomatoes API call, $movie\_similar(m)$, for each of the movies that u_t already likes. These movies are selected based on a similarity metric used by Rotten Tomatoes using features such as genre, director, actors etc. We rank these by

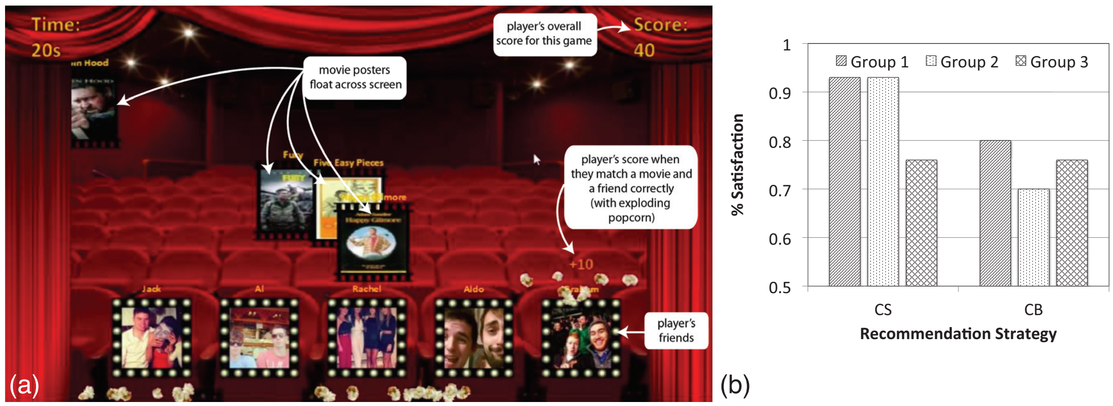

Figure 1: (a) The Recommendation Game in action. (b) Preliminary evaluation results.

giving priority to movies appearing more frequently in these similar movie sets.

Preliminary Evaluation

Our pilot involved 3 groups of graduate and undergraduate, students (males and females); there were 15 in Group 1, 6 in Group 2, and 6 in Group 3. Groups 1 and 2 were mutual friends while Group 3's members did not know each other's movie tastes well; thus Group 3 acted as a control when it came to understanding the impact of relationship strength on the outcome. For each participant, we had their Facebook movie likes and participants acted as both players and friends. Each participant played multiple games and the data collected was used to generate 6 recommendations for each player using CS and CB. We asked participants to rate these recommendations as either satisfactory or unsatisfactory.

The satisfaction results in Figure 1(b), for each group and strategy, highlight a benefit for CS for Groups 1 and 2. In these groups the CS strategy delivers satisfaction scores above 90% compared with between 70% and 80% for CS in Group 3 (not close friends) and for all groups in the CB strategy. This suggests that the data generated during gameplay is a useful source of recommendation data among friends.

Conclusions

The work as presented is limited in many ways. The small scale of our evaluation offers little more than a proof-of-concept for what we are trying to achieve. Nevertheless we believe it serves to highlight the potential for new ways to think about recommendation data and recommender systems while contributing to a growing interest in the role of crowdsourcing in recommender systems research[1].

Acknowledgements

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

[1]See http://crowdrecworkshop.org

References

Adomavicius, G., and Tuzhilin, A. 2005. 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):734–749.

Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M.; Leaver-Fay, A.; Baker, D.; Popović, Z.; and Players, F. 2010. Predicting Protein Structures with a Multiplayer Online Game. *Nature* 466(7307):756–760.

Desrosiers, C., and Karypis, G. 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In Ricci, F.; Rokach, L.; Shapira, B.; and Kantor, P. B., eds., *Recommender Systems Handbook*. Springer Berlin Heidelberg. 107–144.

Gligorov, R.; Hildebrand, M.; van Ossenbruggen, J.; Schreiber, G.; and Aroyo, L. 2011. On the Role of User-generated Metadata in Audio Visual Collections. In Musen, M. A., and Corcho, O., eds., *Proceedings of the 6th International Conference on Knowledge Capture - K-CAP '11*, 145–151. Banff, Canada: ACM.

Law, E., and von Ahn, L. 2009. Input-Agreement: A New Mechanism for Collecting Data using Human Computation Games. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems - CHI 09*, 1197–1206. Boston, Massachusetts, USA: ACM Press.

Salvador, A.; Carlier, A.; Giro-i Nieto, X.; Marques, O.; and Charvillat, V. 2013. Crowdsourced Object Segmentation with a Game. In *Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia - CrowdMM '13*, 15–20. Barcelona, Spain: ACM.

von Ahn, L., and Dabbish, L. 2004. Labeling Images with a Computer Game. In Dykstra-Erickson, E., and Tscheligi, M., eds., *Proceedings of The ACM Conference on Human Factors in Computing Systems*, 319 – 326. Vienna, Austria: ACM.