Collaborative Filtering For Recommendation In Online Social Networks*

Steven Bourke, Michael P O'Mahony, Rachael Rafter, Kevin McCarthy, Barry Smyth

Abstract In the past recommender systems have relied heavily on the availability of ratings data as the raw material for recommendation. Moreover, popular collaborative filtering approaches generate recommendations by drawing on the interests of users who share similar ratings patterns. This is set to change because of the *unbundling* of social networks (via open APIs), providing a richer world of recommendation data. For example, we now have access to a richer source of ratings and preference data, across many item types. In addition, we also have access to mature social graphs, which means we can explore different ways of creating recommendations, often based on explicit social links and friendships. In this paper we evaluate a conventional collaborative filtering framework in the context of this richer source of social data and clarify some important new opportunities for improved recommendation performance.

1 Introduction

Today recommender systems provide a mature approach for addressing the information discovery challenge facing online users and offer service providers a considerable advantage when it comes to promoting content, products, and services directly to users based on their preferences. Much of the success of recommender systems can be traced back to collaborative filtering techniques [1] that rely on large quantities of ratings or transactional data.

Recently researchers have begun to question some of the basic assumptions that inform collaborative filtering. For example, [3, 23] studied the importance of neigh-

Steven Bourke, Michael P O'Mahony, Rachael Rafter, Kevin McCarthy, Barry Smyth
Clarity: Center for Sensor Web Technologies, University College Dublin, Dublin 4, Ireland e-mail: {steven.bourke, michael.omahony, rachael.rafter, kevin.mccarthy, barry.smyth}@ucd.ie

* Supported by Science Foundation Ireland under Grant No. 07/CE/11147 CLARITY

M. Bramer and M. Petridis (eds.), *Research and Development in Intelligent Systems XXIX*,
DOI 10.1007/978-1-4471-4739-8_24, © Springer-Verlag London 2012

bours when making a ratings prediction to conclude that in fact neighbours can play a relatively minor role in prediction accuracy or user satisfaction. Elsewhere researchers have started to consider other sources of recommendation knowledge to compliment ratings-based similarity data. For example, with the rise of the social web there has been considerable interest in modelling the reputation of users to bias future recommendations/predictions with respect to users who are both relevant *and* reputable; see for example [11, 18]. On the topic of leveraging social web data for recommendation, research such as [8] explores the use of microblogging services like Twitter as a new source of product data and user opinions, showing how even this noisy information can be used to make reliable recommendations. And finally, the social web has proven to be a fertile ground for new recommendation tasks whether recommending helpful product reviews to users [20, 22], or tags [24], or even suggesting connections and friends in social networks [7, 9].

In this paper, rather that rush to develop a new type of recommender system, we felt it worthwhile to return to recommendation basics by evaluating conventional collaborative filtering in the context of new sources of social recommendation data. This is important if it helps to establish a baseline for future research as well as providing an opportunity to reconsider some basic recommender systems assumptions in the context of the availability of new types of social data. These new data sources have been made available as major social networks such as Facebook and Twitter have provided API access to user data. This data is interesting on two fronts. First it can provide access to large quantities of ratings-like preference data, across a variety of item types. Secondly it can provide access to mature social graphs based on explicit social links and real-world friendships. The work of [2, 3, 15], for example, concluded that people are more likely to respond well to recommendations from their friends suggesting that recommender systems should take advantage of explicit social connections where possible; see also the work of [11, 17] on a social recommendation technique based on trust propagation in social networks.

The central contribution of this work is an experiment designed to evaluate the effectiveness of collaborative filtering under a number of data conditions. This includes varying the way in which the user neighbourhoods and candidate item sets are formed during collaborative filtering; we compare traditional approaches based purely on ratings similarity across a user population to approaches that rely on more constrained populations of users, such as friends and friends-of-friends. We also vary the items used during profiling and recommendation. The results are interesting. For instance they reveal that significant improvements in recommendation quality can be achieved by using more constrained populations of users, at least across most item types. The remainder of this paper is organised as follows. In the next section we describe the relevant background material, after which we describe the experimental setup in terms of the dataset, algorithms, and methodology. Section 4 will describe and discuss the results and finally we will discuss the implications of this work for future recommender systems research.

2 Background

In this section we review relevant background material related to our work. Specifically we look at recommender systems with a particular focus on collaborative based systems. We then move onto recommender systems which use some form of social information in the recommendation process, and then finally some newer social networking based recommender systems. Recommender systems can be divided into two categories, content and collaborative filtering based techniques. A content based recommender system will usually leverage some form of meta data alongside user and item transaction information. For example, meta data could include genre or actor information in a movie recommender. How this information is used in a content based recommendation can vary significantly, for instance, in [21] the authors use vector space models to better represent content when generating recommendations; the authors find the vector space model helps to improve the overall accuracy. In [5] the authors represent all content in a hyper graph so that it can be combined alongside implicit data. This allows the authors to treat content based information the same way as transactional information within their recommender system.

The second style of recommendation is known as collaborative filtering, which avoids the need for meta data, and relies solely on user and item transaction information to expose the underlying preferences users have for items. Collaborative filtering techniques can be divided into two categories, known as model-based and memory based techniques [4]. Memory based techniques initially gained popularity due to the seminal work of the GroupLens research group [13], in which the transaction data is represented as a sparse matrix and then either a user-based [10] or an item-based technique [12] is employed to generate recommendations. User-based techniques work by analysing common user preferences for items in order to generate recommendations. This is done by measuring profile similarity between users to form *k*-nearest neighbourhoods. Then a ranking technique is used to score items from the neighbourhood for recommendation to a particular target user. Alternatively, item-based recommendation analyses the similarity between items, expressed in terms of the shared opinions of users who have consumed the items, in order to generate recommendations. Candidate items are ranked based on their similarity to the target user's previously rated items [12]. Improvements in accuracy via the use of extensions for user-based and item-based collaborative filtering are discussed in [4] and [10].

A model-based collaborative technique generally involves the application of a machine learning technique to the user,item matrix, for example clustering or Bayesian networks [4]. More recently, in the Netflix prize [6] model-based techniques based on matrix factorisation have gained a lot of popularity due to the improvements in accuracy they offer over typical memory based approaches on the Netflix dataset [14]. Matrix factorisation involves reducing the user item matrix into a dense latent feature space for both users and items. The latent features represent how strongly an item relates to a certain feature, and the extent to which a user is partial to that feature. Predicting relevant items for the target user is done by calculating the dot product of a user and item feature vector. However, while matrix factorisation has proved to be quite accurate in the Netflix competition, related research has suggested that it may not always be the most accurate approach. For instance in [16] the authors find that user-based recommendation outperforms matrix factorisation in the context of an online auction environment.

The incorporation of social network information into recommender systems is a relatively new direction in this field, prompted by the *unbundling* of networks such as Facebook and Twitter (via open APIs) which reveal rich social data and mature social graphs. For example [8] and [9] exploit social network information in content-based recommendation scenarios. Alternatively, [25] propose a graph based approach for memory-based collaborative filtering that uses transitive similarity metrics to calculate similarities between users based on their social graph, leading to improvements compared to standard user-based collaborative filtering. For model-based collaborative filtering there are approaches such as that proposed in [11] which looks at incorporating trust from social networks into the recommendation process. The authors report improvement over standard matrix factorisation. Finally, in [17], social information is incorporated into the regularisation process of their model training. In our work we focus specifically on the recommendation challenge presented in finding relevant content for users of online social networks. Our work measures the performance of known recommendation techniques in social environments. We add additional social filtering to the process as well as measure cross domain recommendations. To date we are unaware of anyone that has done a detailed study into the effectiveness of recommendation in these types of online social networks.

3 Experimental Setup: Data, Algorithms & Conditions

The data for this study was obtained using the Facebook API from Sept. to Oct. 2011 to collect 42,550 user profiles containing links (URLs), videos and check-ins (location ids) shared by these users. In other words, these are unary profiles containing only a user's *positive* preferences, in the form of items that they have actively shared.

We consider a number of basic collaborative filtering style algorithms for the purpose of this study. Specifically we evaluate a user-based and an item-based recommendation algorithm. For our user-based algorithm we use a standard approach described in [19]. To generate a set of recommendations for a target user, the steps involved are as follows:

- The algorithm first finds neighbours using a similarity thresholding approach where users with non-zero similarity to the target user are selected
- User similarity is computed using the Jaccard index over the ratings vector of each user.
- Next, assuming unary ratings, the score for each of the neighbour items (less those that appear in the target user profile) is calculated from the sum of the similarities of the neighbours that contain that item. The *k* items with the highest

scores are then recommended to the target user. We will refer to this algorithm as the standard user-based approach.

For our item-based approach our technique is based on the the work proposed in [12]. The algorithm works as follows:

- To generate a set of recommendations firstly an *item-item* similarity matrix is built using the Jaccard index against co-rated items.
- The algorithm identifies the most similar items to the target user's items, these are considered the candidate items for recommendation
- The candidate items are then ranked for recommendation by comparing their overall similarity with the previously rated items in the target user's profile.

Next we define two variations of the standard collaborative filtering configuration used in both the user-based and item-based algorithms, by changing the source of users from whom items can be recommended. In the *FRIEND* algorithm we consider only items from friends of the target user, instead of the full set of all users. This allows us to consider the utility of direct social connections as a means of identifying neighbours, rather than using anonymous users with similar ratings patterns. Second, we define *FoF*, this time only considering items from the target user's friends of friends. When applying the FRIEND or FoF variation to the item-based algorithm, only items that exist in either the FRIEND or FoF respectively are considered as candidates for recommendation, i.e. the co-ratings can only come from users that exist within that subgroup. In the case of the user-based variations, only users that exist in the FRIEND or FoF configuration can be used as neighbours when generating recommendations. In both variations once the neighbours have been selected, the candidate items are ranked as in the standard algorithm.

Finally, we also consider a number of input/output variations by changing the rules regarding the types of items that may appear in a profile (input) or be recommended (output). Using the Facebook data we can consider links, videos, location check-ins or any combination thereof as either input, output, or both. This leads to 16 input/output combinations to test with our 3 algorithms (CF, i.e. standard collaborative filtering, FRIEND and FoF).

4 Results

4.1 Methodology

From the 42,550 users in the Facebook dataset we identified a subset of 3,419 users whose profiles contained at least 5 items of *each* type (links, videos, check-ins) as the *target users* (the users for whom we will make recommendations); see Table 1. Then, for any other user, in order to qualify as a potential neighbour when recommending a particular type of item, a user must have at least 5 of those items in their profile (although they may have fewer than 5 of the other types of items). Table 1 reports the dataset statistics for each recommendation task (links, videos, or check-ins).

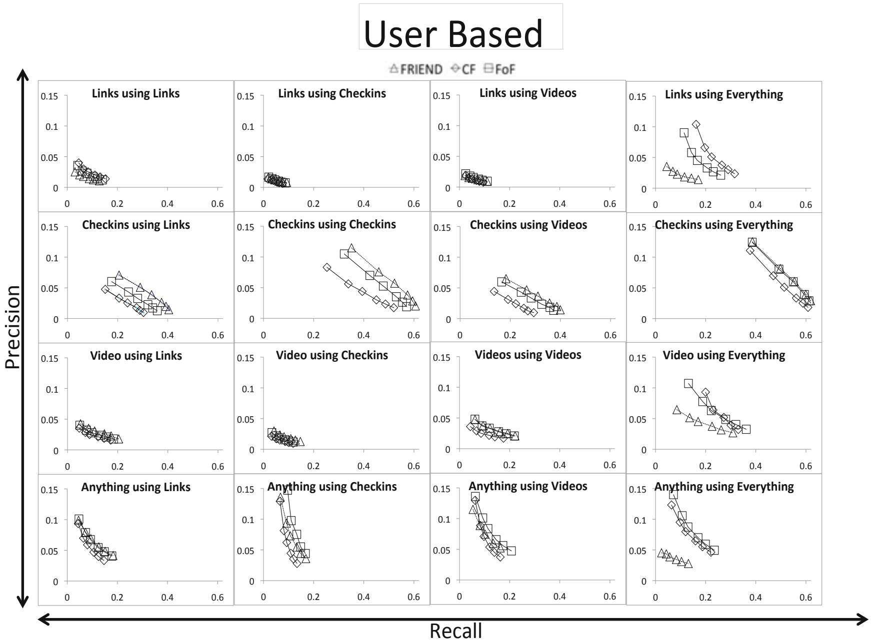

Fig. 1 Precision vs. Recall for recommendation lists sizes of 5, 10, 15, 25, 35, 50.

Table 1 Data Statistics

Type	# Target Users	# Other Users	# Items
Links	3,419	29,517	529,257
Video	3,419	23,731	171,901
Check-in	3,419	4,776	15,259

For our experiments we adopted a standard *leave-one-out* testing approach. Each of the 3,419 target users is considered in turn for testing. For each target user, its profile is randomly divided into 20% *test* and 80% *training* data; this is repeated 5 times for each user. Each set of training data is used as the basis for identifying recommendations, which are then compared to the test data items using precision, recall and the F1 measure. We also examine the item coverage in each case. Results are averaged for different length k of recommendation list ($k = 5, 10, 15, 25, 35, 50$).

To measure significance, we firstly analyse the results using Krustal Wallis, then Tukey's is used as the post-hoc analysis to measure for pair-wise significance amongst the different techniques. The p–value is set to 0.05.

The detailed results of these experiment are given in Figs. 1 and 2 (user-based) and Figs. 3 and 4 (item-based). Figs. 1 and 3 show the precision and recall, while Figs. 2 and 4 show the F1 measure, coverage and significance results. All figures contain 16 graphs, one for each of the 16 profile/recommendation variations above. In each graph we indicate the profile/recommendation combination in the graph title; e.g. *Links using Check-ins* indicate links being recommended but using check-in data to select the neighbourhoods. In each graph we present a set of results for the three algorithmic variations. For example, in Figs. 1 & 3 we present the precision and recall results for CF, FRIEND, and FoF for varying recommendation list sizes ($k = 5, 10, 15, 25, 35, 50$). In Figs. 2 & 4 each graph depicts a bar chart of the mean F1 (across k) for each of the algorithmic variations, and a line graph showing recommendation *coverage* (the percentage of recommendation trials where at least one recommendation could be made). Finally, statistical significance is indicated by the pattern of the bar in the F1 charts, if the internal pattern for a condition is unique that means that the technique was statistically significant, if it is not unique that represents a non statistically significant result. The F1 charts in Figs. 2 & 4 give us a general performance of each technique. As there are too many charts to discuss individually, we will now highlight what we believe to be the most interesting insights.

4.2 User Based Results

From these results we can make some interesting observations. Coverage favours standard CF, which beats FoF and FRIEND in each of the 16 conditions, albeit marginally. This is not surprising since the CF condition enjoys a much larger population of candidate neighbours (some 42,550 users) than FoF (median of 6,500 users) or FRIEND (median of 190 users).

We also see that CF rarely outperforms FRIEND and FoF; in fact only when links are recommended and only then when profiles contain only links, or everything (links, videos, check-ins); Notice too the FRIEND algorithm tends to beat CF and FoF when recommending check-ins (see the 2^{nd} row of graphs in Figs. 1 and 2). This makes sense because check-ins are physical locations and thus more likely to be accurately recommended by people who share these locations, and a higher density of these people are likely to be a user's friends. The FoF and FRIEND algorithms outperform CF when profiles contain everything (links, videos, check-ins) and when recommending anything; see the *Anything using Everything* graph in Figs. 1 and 2. It is also noted that when using a richer profile (The *Everything* condition) accuracy is always the best performing approach.

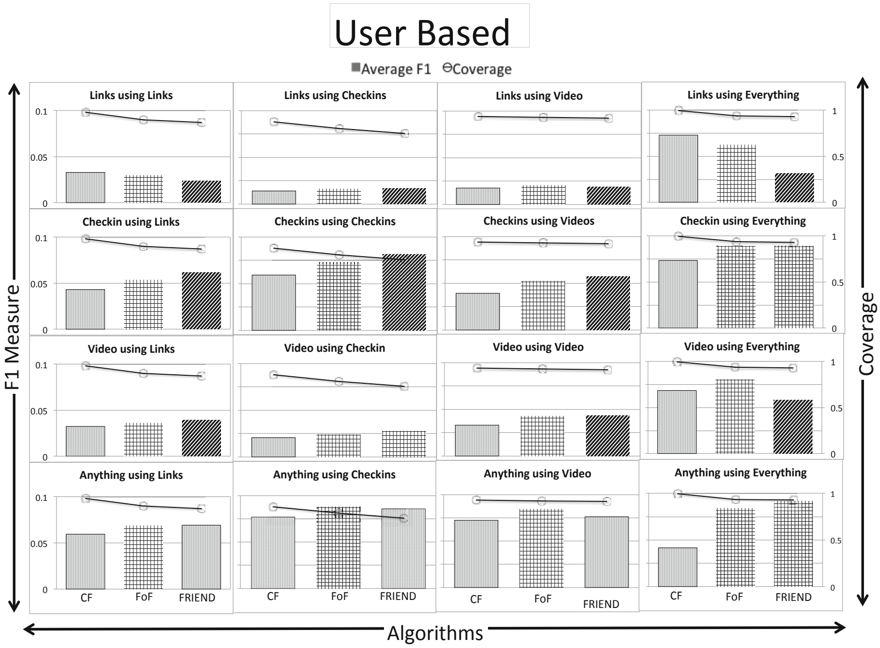

Fig. 2 User Based: Average F1 and Coverage results.

4.3 Item Based Results

In our item-based results we can firstly note that the level of coverage amongst the different techniques and item combinations varies throughout the experiment compared to the user-based coverage. The general trend is that CF has the highest rate of coverage where as FoF and FRIEND drop off in instances where the profile consists of *Links*,*Videos* or *Checkins*.

We can see that in most cases using FRIEND and FoF outperform CF in terms of accuracy. The only occasion where CF actually outperforms any social based technique is in the case of recommending *Links* and using *Check-ins*. We can note that in general item-based recommendation seems to have a more difficult time accurately predicting recommendations in the *Links* domain under item-based compared to *Videos*, *Checkins* or *Anything*. The highest F1 score is 0.028 when using FoF with *Links* using *Everything* (Fig. 4). Links are difficult for the item-based recommender to accurately predict due to the sheer number of candidate items which reduces the likelihood of two items being co-rated, thus and low levels of similarity between links due to a comparatively low rate of co-rating between items (e.g the number of users who co-rated $item_i$ and $item_j$ is 2). The sparsity of our item-item matrix for links is 0.0037. When recommending check-ins we can see that using FRIEND is the best performing technique in all cases except for *Checkins using Videos* (Fig. 4), but if we look at the overall precision and recall (Fig. 3) we note that while

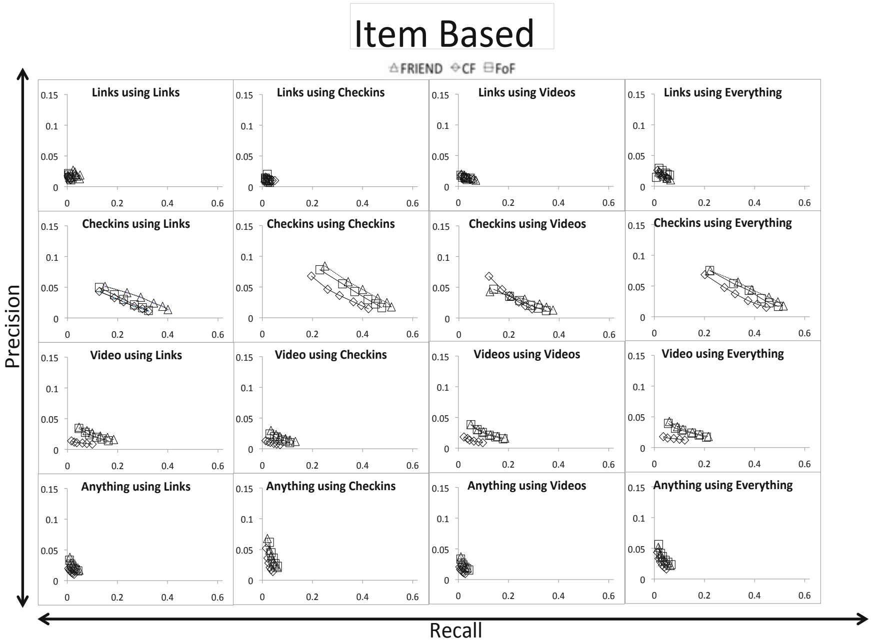

Fig. 3 Item Based: Precision vs. Recall for recommendation lists sizes of 5, 10, 15, 25, 35, 50.

FRIEND is the best overall approach there are minor differences in overall precision between FRIEND and FoF. Once again we believe that using a persons' friends is more favourable for check-ins because the likelihood of sharing similar physical preferences amongst friends is more likely. When recommending *Anything* using item-based recommendation, we can see that check-ins give a similar performance as to using *Everything*. When we look at precision on its own, Figs. 1 & 3, we can see that the user-based algorithm across different configurations can achieve at least 0.1 in terms of precision where as the item-based algorithm never achieves this.

4.4 Summary Analysis

Given the detail in the above charts, it is useful to summarise the results by averaging the results over the different profile and recommendation types. For example, in Fig. 5(a) we present a clustered bar chart of the mean F1 for all list sizes. For each of the three user-based algorithms, results are averaged over each profiling condition (links (L), check-ins (C), videos (V), everything (E)). For each cluster of bars we also show the mean F1 averaged over CF, FRIEND and FoF. In this case we see that CF is consistently beaten by both FRIEND and FoF. It is also clear that there is a benefit to using all three types of profile data (the E condition), with an overall mean F1 of

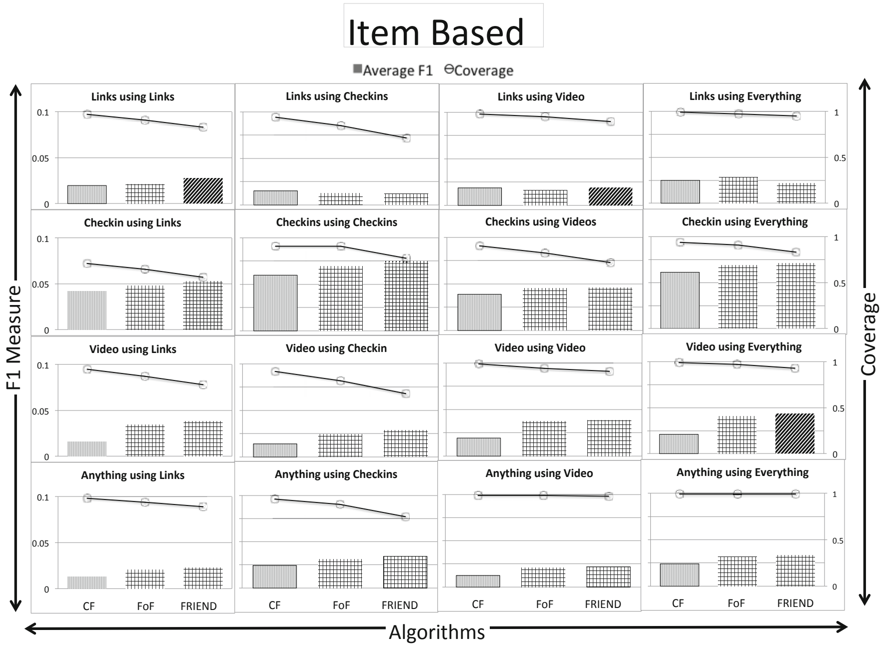

Fig. 4 Item Based: Average F1 and Coverage results For Item based.

0.07 compared to 0.05 for any individual profiling technique. In general FRIEND and FoF beat CF by about 15% to 20% for each of the L, C, and V conditions, but there is a win for FoF when profiles contain everything (E condition) where we see FoF beating CF and FRIEND by about 25%.

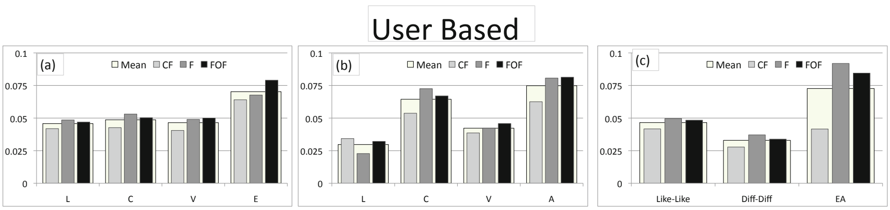

Fig. 5 (a) F1 when recommending items *using* Links, Check-ins, Videos and Everything; (b) Average F1 when *recommending* Links, Check-ins, Videos or Anything; (c) F1 based on within-domain cross-domain variations.

Summarising the results by recommendation type we obtain an analogous bar graph as shown in Fig. 5(b). This time we see that the best overall performance is achieved when recommending anything (the A condition). Again, both FRIEND and FoF outperform CF, with an average F1 of just over 0.08 compared to just 0.06 for CF, a relative increase of some 30%. Also interesting is the strong performance when recommending check-ins (the C condition); this time the FRIEND algorithm wins overall, beating CF by about 33%. This view of the data also clarifies those conditions where CF produces a higher average F1 than FRIEND or FoF: this only happens in the case where links are recommended (the L condition) where the F1 score for FRIEND is about 30% less than CF.

Next, we consider when the same type of items are used across profiling and recommendation (*Like-Like*) versus when different types of items are used (*Diff-Diff*), and compare these to when we profile using everything and recommend anything (*EA*). The average F1 results are shown in Fig. 5(c). We see superior performance under the *EA* condition. Clearly it is easier to make good recommendations by maximising the availability of profile data without restricting the type of items that can be recommended. While this might not be surprising, what is surprising is the scale of the benefits accruing to the FRIEND and FoF algorithms which deliver more than double the F1 performance of CF. Finally, while cross-domain recommendations are feasible (*Diff-Diff*), they are much less effective than within-domain recommendations (*Like-Like*); on average the F1 score for *Like-Like* is approximately 0.046 compared to 0.032 for *Diff-Diff*, an increase of about 40% for the former. And once again, for both *Like-Like* and *Diff-Diff*, we find that the FRIEND and FoF algorithms outperform CF.

In Figure 6 we performed a similar summary as presented in Figure 5 except that Figure 6 refers to our item-based recommendation results. For the different profiling conditions (Fig. 6(a)) we can see no distinct best performer. We do note that the E condition does slightly outperform the C condition, however the difference is insignificant with E performing at 0.039 and C performing at 0.033. Interestingly we do note that FoF and FRIEND always outperform standard CF. In Fig. 6(b) we see that Checkins(C) is by far the easiest item to recommend for when using a item-based recommender system. For C the F1 being 0.56 while the next best performing approach is Video(V) at 0.029. We do note with the exception of V that CF , FRIEND and FOF all perform to a similar degree, but with CF still being the least effective of the three. Lastly in Fig. 6(c) we can see that item-based recommendation performs best in the typical *Like-Like* configuration, with FRIEND and FoF outperforming CF. In the case of *Diff-Diff* and *EA* we see overall similar results both in terms of accuracy and also in regards to how the CF, FRIEND and FoF approaches perform.

From our summary analysis we can determine that user-based recommendation is generally more preferable in terms of accuracy compared to item-based recommendation. We also note that using either user-based or item-based recommendation we achieve higher accuracy when applying the FRIEND or FOF filters. One difference between user-based and item-based recommendation is that user-based performs well when using the Everything(E) profile (User based: Fig. 5(a), Item-based Fig. 6(a)). We also see that user-based is more accurate when recommending Anything (A) Fig. 5b compared to item-based Fig. 6(b). Finally that when recommending *Everything using Anything* (EA) user-based also more accurate than item-based (Fig. 5(c) & Fig. 6(c)).

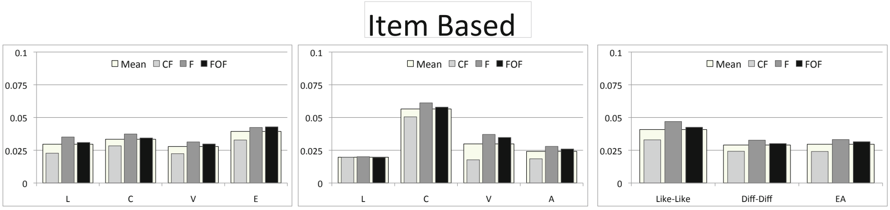

Fig. 6 (a) F1 when recommending items *using* Links, Checkins, Videos and Everything; (b) Average F1 when *recommending* Links, Checkins, Videos or Anything; (c) F1 based on within-domain and cross-domain variations.

5 Conclusions

In this paper we examined user-based and item-based collaborative filtering in the context of the social structures made available from social networks like Facebook. As a result of the large quantities of data being created and shared in these online social networks, users can struggle with the well known challenge of information overload, making it difficult for users to easily find interesting content. To help users with these challenges we performed a detailed evaluation of user and item-based collaborative filtering in comparison to variations that exploited a user's explicit friends or friends-of-friends as alternative sources of information. Furthermore we examined performance when utilising and recommending a variety of item types (links, videos and check-ins). A key result is that the FRIEND and FoF methods tend to outperform a typical collaborative filtering configuration. We also found that recommending check-ins proves to be easier than links or videos, possibly because of the potential for more limited variability and the requirement of a physical relationship to exist between the item and user. Our results confirm higher accuracies for same-domain versus cross-domain recommendations. We also found that in an environment where any type of item can be recommended, building a profile from *check-ins* to perform recommendations performs at a similar level of accuracy as building the profile from *everything*. Finally we have found that user-based collaborative filtering is more accurate than item-based collaborative filtering when recommending information in online social networks. Our work demonstrates that when using collaborative filtering based techniques, the use of friends, or friends of friends, as opposed to typical collaborative filtering returns higher levels of accuracy. Based on these results we conclude that for future work we should look to improve upon results achieved via user based techniques by more extensively leveraging the social graph information while also exploring ways to blend users interests in different item types.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.
2. P. Bonhard and M. A. Sasse. 'knowing me, knowing you' – using profiles and social networking to improve recommender systems. *BT Technology Journal*, 24(3):84–98, July 2006.
3. Steven Bourke, Kevin McCarthy, and Barry Smyth. Power to the people: exploring neighbourhood formations in social recommender system. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 337–340, New York, NY, USA, 2011. ACM.
4. John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
5. Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. Music recommendation by unified hypergraph: combining social media information and music content. In *Proceedings of the international conference on Multimedia*, MM '10, pages 391–400, New York, NY, USA, 2010. ACM.
6. Alex Conference Chair-Tuzhilin and Yehuda Conference Chair-Koren. Netflix '08: Proceedings of the 2nd kdd workshop on large-scale recommender systems and the netflix prize competition. August 2008.
7. Elizabeth M. Daly, Werner Geyer, and David R. Millen. The network effects of recommending social connections. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 301–304, New York, NY, USA, 2010. ACM.
8. Sandra Garcia Esparza, Michael P. O'Mahony, and Barry Smyth. Mining the real-time web: A novel approach to product recommendation. *Knowledge-Based Systems*, 29(0):3 – 11, 2012.
9. John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 199–206, New York, NY, USA, 2010. ACM.
10. Jon Herlocker, Joseph A. Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4):287–310, October 2002.
11. Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 135–142, New York, NY, USA, 2010. ACM.
12. George Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 247–254, New York, NY, USA, 2001. ACM.
13. Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, March 1997.
14. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.
15. Danielle H. Lee and Peter Brusilovsky. Social networks and interest similarity: the case of citeulike. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT '10, pages 151–156, New York, NY, USA, 2010. ACM.
16. Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*, page 959, New York, New York, USA, October 2010. ACM Press.
17. Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 287–296, New York, NY, USA, 2011. ACM.

18. Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 17–24, New York, NY, USA, 2007. ACM.
19. Andreas Mild. An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services*, 10(3):123–133, May 2003.
20. Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. Etf: extended tensor factorization model for personalizing prediction of review helpfulness. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 163–172, New York, NY, USA, 2012. ACM.
21. Cataldo Musto. Enhanced vector space models for content-based recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 361–364, New York, NY, USA, 2010. ACM.
22. Michael P. O'Mahony and Barry Smyth. Learning to recommend helpful hotel reviews. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 305–308, New York, NY, USA, 2009. ACM.
23. Rachael Rafter, Michael P. O'Mahony, Neil J. Hurley, and Barry Smyth. What have the neighbours ever done for us? a collaborative filtering perspective. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization*, UMAP '09, pages 355–360, Berlin, Heidelberg, 2009. Springer-Verlag.
24. Shilad Sen, Jesse Vig, and John Riedl. Tagommenders: connecting users to items through tags. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 671–680, New York, NY, USA, 2009. ACM.
25. Panagiotis Symeonidis, Eleftherios Tiakas, and Yannis Manolopoulos. Product recommendation and rating prediction based on multi-modal social networks. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 61–68, New York, NY, USA, 2011. ACM.