Collaborative Filtering With Adaptive Information Sources

Neal Lathia
Department of Computer Science
University College London
Gower Street, WC1 E6BT, UK
n.lathia@cs.ucl.ac.uk

Xavier Amatriain, Josep M. Pujol
Telefonica Research
Via Augusta 177
Barcelona 081290, Spain
xar, jmps@tid.es

Abstract

Collaborative filtering (CF) algorithms, which generate recommendations for web users by predicting user-item ratings, are often evaluated according to their predictions; in this context the problem of generating recommendations can be formulated as one of fitting a community of users to the best set of predictors. However, the data used to perform CF is sparse, and accuracy is limited by both the quantity and quality of information available. Mining the web has the potential to address these issues: the quality and quantity of ratings can be incremented by collecting external sources of rating information. In this work we introduce a method to perform CF with external data sources; furthermore, we show that a community of users can be partitioned according to what external source acts as a better predictor of each user's preferences. In particular, we find that a single kNN predictor can achieve remarkably high prediction accuracy if the data sources are selected optimally: designing a recommender system can thus be approached with the focus on data quality rather than algorithmic method.

1 Introduction

Recommender systems, based on collaborative filtering (CF), are displaying an evermore important and pervasive presence on the web. The problem of generating recommendations has been described as a *prediction* problem: based on a profile of user ratings, the system needs to predict future user ratings for other content in the future. The approaches adopted to perform CF can be broadly divided into two categories. The first are *statistical* approaches; these draw on the assumption of like-mindedness between users and therefore focus on a variety of classifiers that operate on the user-rating data; the most prominent candidates being based on matrix factorisation and neighbourhood methods [Koren, 2008][Herlocker *et al.*, 2004]. The second approach is based on *user modeling*; these methods augment statistical approaches by reasoning on the context and behaviours that emerge when people use recommender systems; recent examples include the rising interest in trust modeling for collaborative contexts, including [O'Donovan and Smyth, 2005].

Traditional CF suffers from the problem of *data sparsity*; the ability that a system has to make predictions for a user or item is limited by the lack of rating information. The data also has very high dimensionality; for example, the Netflix dataset[1] includes about half a million users and about twenty thousand movies. The mere size of the data implies that generating recommendations is a very expensive process that is difficult to scale to large communities. The current focus of much CF research is on improving the accuracy of the *algorithms* applied to generate recommendations. In particular, a number of successful statistical methods [Koren, 2008] combine an ensemble of predictors to produce higher accuracy. However, improving the classification method does nothing to improve the data that is being used when predicting user's preferences, and a fundamental limitating factor of any learning algorithm applied to the CF domain is the sparsity and potential inaccuracy of the data being used.

Similarly, algorithm-centric research also deters from fully modeling the implicit ways in which people form their opinions. While sociologists often model preference formation according to the principles of *homophily* (like-mindedness) and *social influence* (adopting the same preferences as influential members) [Axelrod, 1997][McPherson *et al.*, 2001], CF research has mainly centred its assumptions on the former theory. Although the task of identifying the source of influence in a set of user ratings seems daunting, this theory carries with it the assumption that there are a *range* of sources where users may form their opinions; in particular, not all users form their opinions by eliciting information from similar neighbours. The problem is thus how to model the way people form their opinions.

Mining the web for publicly available ratings has the potential to address the sparsity problem by drawing on the assumptions of *social influence*: there are a great number of online resources that contain a vast amount of ratings that may be accessed by users as they form their opinions. In this work we therefore propose to explore four different source datasets and evaluate the predictive power they have on a test set of user-movie ratings. Two of these source datasets were collected from the web, while the second two are de-

[1] http://www.netflixprize.com

Dataset	Users	Ratings	Sparsity (%)
Flixster	77	585,293	79.02/0.01
Rotten Tomatoes	1,651	151,949	98.87
Netflix Training	9,980	1,432,259	99.19
Netflix Test	8,877	19,476	N/A

Table 1: Dataset Information

rived from the a set of training data, based on *neighbours* and *power users*; Section 2 describes these datasets, and Section 3 highlights the statistical features that emerge between the sets. In Section 4 we introduce the method we implement to perform cross-dataset predictions, and Section 5 reports and analyses the results when each source dataset is used to make predictions on a common test set.

Our main result is that the accuracy of a CF prediction algorithms heavily depends on the *quality* of the information used to generate predictions, and the most appropriate source is user-dependent. In particular, matching users to the correct source of rating information has the potential to produce highly accurate recommendations when using a simple user-based kNN algorithm. We evaluate a number of benchmark methods that attempt to achieve this goal in Section 6; we thus introduce a novel perspective to CF, where the focus should not be so much on the *method* applied, but on the *data* that is used.

2 Information Sources

In this work we ran experiments using a subset of the Netflix prize data. Our subset consists of $10,000$ randomly selected Netflix users from the training set, and each of these user's probe ratings as a test set. To compliment this dataset we crawled two different sources of rating profiles: Rotten Tomatoes[2] and Flixster[3]. Based on how ratings are input into each of these systems, we call these sources *experts* and *enthusiasts* respectively:

Experts: The Rotten Tomatoes portal aggregates a number of cinema critic reviews from a wide range of web sources, including newspapers, specialized websites, and magazines. The critics use different rating scales; some range from 1-10 stars, others 1-5, and some use a 100-point scale. However, all of these ratings can be normalised. For example, a 9 out of 10 star rating is the same as 4.6 out of 5; we adopt a simple linear transpose to re-intepret ratings from one scale to another.

Enthusiasts: Flixster is one of the largest movie-oriented social networks, and therefore contains ratings given by the site's movie-enthusiast subscribers. We collected the profiles of the top-100 users from Flixster. However, not all users set their profiles to public: this reduced our collected dataset to 77 users. The Flixster users rate movies on a 1-5 star scale, but also have a further two options available: "want to see" (WS), and "not interested" (NI). In fact, the majority of ratings in the data fall into one of these two latter categories.

[2]http://www.rottentomatoes.com

[3]http://www.flixster.com/

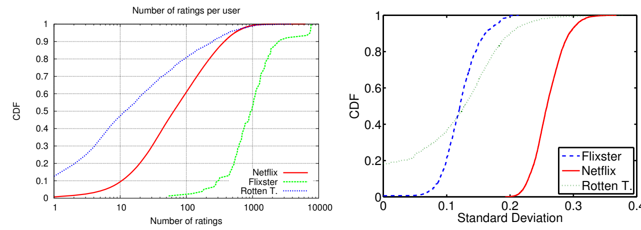

Figure 1: CDF of Ratings and Standard Deviation Per User

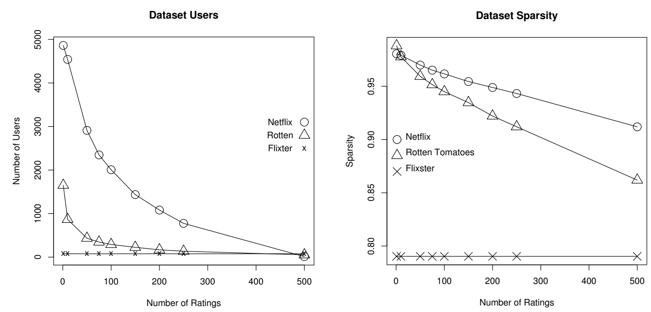

Figure 2: Number of users (left) and dataset sparsity (right) with rating threshold

Due to string-matching inconsistencies between the movie titles in Netflix and the crawled datasets, our datasets contain $6,088$ out of the $17,770$ available in Netflix; furthermore, to accomodate for this, we also cut any Netflix users who had no training ratings within this set of movies. A summary of the size of each dataset is given in Table 1. The analysis in Section 3 is based on this number of movies, which could be identified in *all three* datasets. We also compare the predictive performance of the external data sources to two other sets, each derived the the Netflix training subset we use:

Neighbours: The benchmark performance that we compare the above sources to is the approach adopted by traditional user-based kNN; there is no distinction made between users, who all come from the same community.

Power Users: This group is a subset of the "neighbours" group, the user profiles that, based a simple measure of profile size, are deemed to carry a significant amount of *reliable* information, and represent a sub sample of the community that may offer powerful predictions for the rest of the users. The idea of power users has been explored in the past [Cho *et al.*, 2007], and usually relies on identifying users based on a number of pre-defined heuristics. In this work we focus on profile size; that is, we assume that users who are proactively rating more items are following different behaviours to the casual rater [Herlocker *et al.*, 2004]. Note that all of the above groups strongly differ to results that would be obtained from clustering algorithms: users are grouped either based on profile attributes (rather than the value of their ratings), or based on where their ratings were crawled from. It is thus not guaranteed that users in the same group will agree with each other, whereas clustering algorithms tend to group users based on a notion of similarity.

3 Comparing Information Sources

In this section, we compare the three datasets. As introduced above, they can already be differentiated from one another according to a broad characterisation of the end-users of each system; however, in this section we examine the extent that ratings from different sources will differ in terms of summary statistics: the number and distribution of ratings, the sparsity, and rating deviation between per user.

Number of Ratings: With less than $1/20th$ of the users, the Flixster data contains over 5 times the number of ratings than the Rotten Tomatoes data. The same feature can be observed in Figure 1, which shows the cumulative distribution (CDF) of the number of ratings in each dataset. As the plot shows, 60% of the Rotten Tomatoes experts have about 30 or less ratings, 60% of the Netflix set users have 100 ratings or less, but the same proportion of Flixster reaches up to $1,000$ ratings.

Sparsity: Table 1 reports the sparsity values for each dataset; once again, Rotten Tomatoes and Netflix share similar sparsity values, while the Flixster dataset is a much denser set of ratings. The table also reports two separate sparsity values for the Flixster dataset. The first value, though excluding both WS and NI ratings, shows that the dataset is 79.02% sparse. Including all these extra ratings reduces the sparsity to 0.01%: in both cases, the user-rating matrix contains a much larger amount of ratings than Netflix alone. We also measured how the sparsity fluctuates as different sub samples of users are selected, reflecting the evaluation of *power users* we report in the Section 5. If we only select users who have rated more than α movies, both the number of users and resulting dataset sparsity will change. Figure 2 shows how these changes are affected by the rating threshold. The plots show that there is an uneven distribution of ratings amongst the users themselves, reinforcing the notion that users will behave differently as they interact with the system. The plots also confirm what was observed in Figure 1: the Netflix dataset, while having the highest number of users, also is also the sparsest of the three datasets.

Standard Deviation: Looking at this aspect aims to see the extent that each group of users agrees with each other by capturing the *spread* of ratings around each movie mean. As Figure 1 shows, the distribution of standard deviation values is very different from one community to the next. There are also a very small proportion of Flixster profiles that appear to be *outliers*: their profiles are full of the same rating for nearly *all* content. This causes the standard deviation over their ratings to be less than 0.05. These Flixster outliers will not be able to contribute useful information to any prediction, and can therefore be safely ignored.

4 Collaborative Filtering With External Data

There are a number of methods that can be implemented in order to use the ratings of the above datasets to predict the test set. In particular, one may simply combine all of the datasets into a single, larger training set that can be fed into any learning algorithm. However, in this work we aim to evaluate the potential that disparate sources have to predict a common test set: how well do *experts* predict the crowds? Do *enthusiasts* do better? In this light, and due to the semantics of cross-dataset prediction, we focus on a *single* method: the kNN algorithm.

The kNN can be built according to either the item-based or user-based paradigm. Both methods operate in very similar ways, and differ only in how they assume the underlying data is structured. Here we only consider the user-based approach. Once again, this makes our cross-dataset prediction highly explainable and transparent, which is a key aspect in building recommender systems that users trust [Herlocker *et al.*, 2004].

Implementing a kNN CF algorithm can be decomposed into three steps: (a) *neighbourhood formation*, where the top neighbours are computed for each user, (b) *rating aggregation*, where ratings for an item are collected and used to make a predicted rating, and finally (c) *recommendation and feedback*, where users update their profiles by responding to the recommendations they are given. When using external data sources, neighbours for a user in the training dataset are found from the source set; similarly, ratings for items in the test set will be predicted using the ratings in the selected source set. We identify neighbours using a *weighted* Cosine Similarity: the similarity $sim(a, b)$ between two users a and b is scaled according to how many items $N_{a \cup b}$ their profiles share in common. We base neighbour selection on a similarity threshold: any neighbour with similarity greater than zero is included. A predicted rating is then computed as a weighted average of deviations from each neighbour's mean [Herlocker *et al.*, 2004]. The only limitation we impose is a measure of prediction *confidence*: if less than 10 neighbour ratings have been found for a prediction, the prediction is set to the user mean. Setting the similarity threshold at zero and the confidence at 10 may not be optimal values for each dataset; in this work we focus on evaluating the ability to use adaptive *information sources* when generating predictions, rather than simply tweaking the *algorithm* itself for optimal performance.

We also noted that the Flixster dataset contains two additional ratings, NI and WS. It is not immediately transparent how these ratings should be transposed onto the scale in the target dataset, since they are difficult to place on an ordinal scale of ratings; however, a relationship between the NI rating and the movie average emerges in a few select cases. For example, consider two different movies that each have 1000 ratings. The first has a very high average, $4.8/5$, and only 10 NI ratings. The second has a very low average, $1.1/5$, and 800 NI ratings: it seems possible to assume that NI is roughly equivalent to a form of *negative feedback* provided by the user. However, we decided to ignore these ratings in the neighbourhood formation part of our algorithm.

We did test methods to include these ratings in the prediction step. Drawing from the assumption that NI may act as a form of *negative feedback* and WS represents a potential *positive opinion*, a Flixster neighbour n's NI and WS ratings for item i being predicted for user u (who has mean rating $\bar{r}_u$ and standard deviation σ_u) would map to:

$$NI_{n,u,i} = (\bar{r}_u - \sigma_u) \tag{1}$$

$$WS_{n,u,i} = (\bar{r}_u + \sigma_u) \tag{2}$$

Threshold (α)	# Users	RMSE	Proportion
0	9980	0.9700	22.82
50	5746	0.9709	9.09
100	3936	0.9722	9.26
200	2249	0.9748	8.81
300	1420	0.9778	9.33
400	928	0.9810	12.62
500	602	0.9854	28.08

Table 2: Power User Group RMSE Results

Results when both using and ignoring these kind or ratings are reported in the following section.

5 Evaluation: Transparency of The RMSE

We measure the accuracy of predictions using the Root Mean Squared Error (RMSE) [Herlocker *et al.*, 2004]. We divide our experiments into two parts; in the first, we report the results when using different power user groups from within the Netflix dataset as sources. We then compare performance across the three source datasets.

Predicting With Power Users: Table 2 shows the relationship the rating threshold α used to define power users, the number of power users found who match this criteria, and the RMSE achieved when each group is used as the source set for predictions. The table shows that as the rating-threshold is increased, accuracy worsens. However, there are two points to note here: (a) as above, the algorithm has not been fully tuned for optimal performance (which is a dataset-dependent problem, subject to both the similarity metric and rating aggregation method implemented), and (b) relying on an aggregate error measure like the RMSE does not highlight the performance that is being achieved on a per-user basis. In other words, a single RMSE value does not show whether some users are better suited to certain information sources than others.

Based on the Table 2 alone, it seems that removing non-power users from the dataset results in a loss of prediction accuracy. To explore the veracity of this impression, we built a second matrix; in this case, each row corresponded to a user, and each column represented a source set of power users (according to a rating threshold). Each entry (i, j) in the matrix is the RMSE achieved on user i's test ratings with column j's source set. From this matrix, we were able to compute the proportion of users who best "affiliated" with each subset of power users. In other words, we could determine which source dataset was the most appropriate per user. Table 2 also shows the proportions of the $8,877$ test users dataset who affiliated best with each subgroup of power users. As the plot shows, there is in fact a very large proportion of users whose best source of predictions is the set of power users who have rated more than 500 movies. An algorithm which could manage to select the best power set group for each user would improve the aggregate accuracy from 0.97 to 0.914: users, therefore, associate differently with different sets of power users, and simply targetting a global optimal does not achieve the best possible accuracy. Conversely, it is also possible to infer from these experiments that the traditional nearest-neighbour model, based on selecting the *best* k neighbours for each user, is not optimal: improved accuracy is obtained when user groups are made apriori, and users then find their neighbours within those groups.

Source	RMSE	Proportions
Rotten Tomatoes	1.070	26.62
Flixster-WI/NS	1.207	25.25
Netflix	0.970	48.13
Flixster	1.259	N/A
User-Matched	0.856	N/A
Item-Matched	0.936	N/A
User-Item Matched	0.776	N/A

Table 3: RMSE When Predicting With External Sources & RMSE if users, items, and user-items were matched to the best source

Predicting With External Data: The accuracy results when using all the external datasets are reported in Table 3. As the table shows, the overall accuracy when making predictions with an external sources is worse than simply using neighbours; from these values it would appear that exclusively using external data sources is not a viable option when designing a CF algorithm. The only point worth noting is that including the NI/WS ratings when making predictions using Flixster as a source provided an improvement. However, once again we constructed the user-data source RMSE matrix, and were able to extract the performance that a user-based kNN predictor would achieve if it were able to perfectly match users, items, or user-item pairs to the correct source data set. This time, each column of the error-matrix represented a different source set. The improvement is remarkable: assigning users to the correct data source provides (with this subsample of the data) an accuracy *below* the target of the Netflix prize. Repeating the above analysis across the three datasets also shows that there is no absolutely dominant source, the right column of Table 3 shows. The semantic interpretation of these results is that the Netflix dataset optimally predicts only half of the sampled population; movie critics and enthusiasts are more accurate for the others.

6 Benchmark Methods

Based on the above work it becomes apparent that classifiers like user-based kNN can achieve very high accuracy in the context of recommender systems if users are paired with the correct source of data; the main problem is thus how to infer, given a user profile, the correct source. Our first attempt considered various qualities of each user's profile, such as profile size, mean rating, rating standard deviation, and average agreement of the user's profile items with the movies' mean ratings. However, none of the individual components correlated strongly with each user's selection of optimal data source.

Our current work therefore focuses on how to infer what the best data source is for each user. In this work we propose and evaluate benchmark results derived from two methods. The first is based on *linear combinations* of the dataset

Type	Method	RMSE
Weight	Equal (1/3)	1.0215
Weight	Avg Similarity	0.9851
Weight	Group-Mean RMSE	1.0253
Weight	Training Set RMSE	1.013
Select	Max Avg. Similarity	0.9829
Select	Min Group-Mean RMSE	1.1243
Select	Best Training RMSE	1.208
Select	Most Training Confidence	0.9701

Table 4: Benchmark Method Performance

predictions, where a prediction of item i for user u is generated by each source, and the final prediction is computed as a *weighted* average of each score. The second method pre-classifies each user to a particular dataset, and only computes one prediction per user-item pair using the *selected* dataset. This way, we can evaluate the method both in terms of the aggregate RMSE and the precision/recall metric related to how well the method paired each user with the appropriate dataset. The results we report here can be broadly categorised into two groups. The first are **structural properties**: We weight (or select) datasets based on emergent structural properties of the kNN algorithm; in particular, we measure the role that the similarity function plays when correlating a user to each set, by looking at the average positive similarity the users share with each source set. The second, **user-fit RMSE**, includes methods that weight (or select) the best source set according to how well each user *fits* the three sources, or how well each source predicts the user's training profile. This process entails a two-fold use of each user's training set of ratings; it is first used to compute similarity weights with members of each source set, and a second time to measure how well each source predicts the user's profile. The motivation for the latter group is as follows. Any given user u will have three potential neighbourhoods: Netflix (N), Rotten Tomatoes (RT), and Flixster (F). Each of these neighbourhoods will contain varying proportions of co-rated items with u's training set ratings, and the RMSE between these co-rated movies can be collected (ϵ_N, ϵ_{RT}, ϵ_F). Assuming that a relationship exists between each source's RMSE on the user's training profile and the predictive performance on the same user's test ratings, we can either *select* the source that provides the lowest RMSE, or weight the contribution of each source S proportionally to its accuracy on the training items:

$$w_S = \frac{(\Sigma_i \epsilon_i) - \epsilon_S}{\Sigma_i \epsilon_i} \quad (3)$$

Weighted Combinations We first tried a variety of linear combinations of each sources' predictions for each user's test items. As shown in Table 6, these ranged from weighting each source equally, to weighting each source according to the average shared similarity with the target user, weighting according to how well the target's profile fits the movie means generated from each source, and weighting according to how well each target fits the neighbourhood in each source. All the linear combinations of each sources' predictions failed to produce more accurate results on the test set than using the Netflix source alone. One of the primary reasons for this was that, in many cases, each source produced diverging predictions from the next: linear combinations of polarising predictions therefore hurt the overall results.

Group	Precision	Recall
Min Avg Similarity		
Netflix	0.457	0.867
Rotten Tomatoes	0.269	0.114
Flixster	0.277	0.018
Min Group-Mean RMSE		
Netflix	0.410	0.024
Rotten Tomatoes	0.261	0.546
Flixster	0.255	0.381
Min Training Set RMSE		
Netflix	0.307	0.002
Rotten Tomatoes	0.247	0.121
Flixster	0.263	0.834
Training Set Confidence		
Netflix	0.457	0.999
Rotten Tomatoes	0.0	0.0
Flixster	0.2	$8.2e^{-4}$

Table 5: User-Dataset Classification Performance

User-Source Classification The second set of experiments were performed in two steps. The first step assigns each user to a source by generating a mapping from each user to the the categorical set of sources, while the second step uses the mapping to generate predictions for each user's test set with the assigned source. As above, we tried classifying users according to how well they fit the source's mean ratings, their neighbourhood in each source, or by selecting the group that the user shares the highest amount of similarity with. We complimented these with a classifier that operated on how much *confidence*, or number of ratings, each source has about the target user's profile; the idea being that a user's behaviour mimics that of a particular source if both consistently rate the same items. Based on this methodology, we can measure two results: (a) the RMSE achieved on the test set after pre-classifying each user, and (b) how well the pre-classification step works. We evaluated the latter based on the precision and recall metrics.

In this case, we find that the RMSE results are more encouraging: they differ from when only using the Netflix dataset by less than 0.012 using the average-similarity classification, and by 0.001 when the confidence-based classifier is implemented. However, exploring the precision and recall metrics in Table 6 highlights why these results were obtained: in the latter case, nearly *all* the users have been mapped to the Netflix source, thus producing the same results. Majority of the recall values are low, indicating a high proportion of misclassifications. Examining the results highlighted the fragility of the pre-classification step, and the dependence it had on the sources. In other words, users who were *wrongly* assigned to the Netflix source did not contribute as much error as those who were wrongly mapped to the smaller Flixster and Rotten Tomatoes datasets.

7 Related Work

The idea of using *experts* has been used before in CF research. The work by [Su *et al.*, 2007] defines experts as the *algorithms* that can be used to produce predictions; the authors construct a hybrid CF algorithm that outputs a weighted average of multiple CF algorithms. This significantly departs from the definition we apply here, where expertise is a quality of the *data* and not of the *method* applied to generate predictions using it. On the other hand, [Cho *et al.*, 2007] define experts as a subset of the users of a community based on a number of heuristics. In particular, expertise in an a domain is based on how many items a user has rated in that domain. This definition is closer to the way we identify *power users*, based on rating frequency, although we do not differentiate between domains within the items that can be rated.

Previous work [Aciar *et al.*, 2007] has also considered the problem of source selection; however, Aciar *et al* address the problems of identifying, selecting, and retrieving unstructured information from the web in order to produce recommendations. Sources are selected based on quantifiable relevance and considering how complete, diverse, and timely the data the sources contain is. The authors therefore propose a trust model to effectively select data sources. However, they adopt the broader goal of producing recommendations, while the work above centres on improving the accuracy of recommender system algorithms with a basic model of social influence. Examining how the quality of data relates to performance has also been discussed in the context of *computational trust*. In particular, [O'Donovan and Smyth, 2005] considers that users are more trustworthy sources of information if they tend to provide ratings that are good predictors of neighbour preferences. In our case we seek to identify the most trustworthy *source* of data per user in the Netflix community. Measuring trust based on a history of accurate predictions is similar to the baseline experiment in Section 6 that focused on how well users fits each source.

8 Conclusion

The primary motivation of this work was to highlight the dependence of CF algorithm's performance on the *quality* of the data that is being used to predict user preferences. We therefore explored the potential that a variety of datasets from the web have to predict a sample set of Netflix users. In doing so, we proposed a framework for cross dataset prediction, including methods to normalise data and interpret non-numeric ratings (NI and WS) on an ordinal scale. First, we examined the effect of learning to classify items based on a dense subset of the available training data, by extracting *power users* from the Netflix training set. We then analysed the predictive potential of *external* data sources, based on a collaborative method that generates a neighbourhood for a Netflix user composed of Flixster or Rotten Tomatoes profiles. We identified that the predictive power of both the power-user subsets and external sources is user-dependent; there are some users who are best predicted by power users, others by experts, enthusiasts, or neighbours. The two experiments, however, are not mutually exclusive. In fact, power users can also be identified and exploited within the Rotten Tomatoes dataset, and performing a further crawl of Flixster would supply the kNN algorithm with a richer set of enthusiast movie raters. The main focus of our future work will be combining the above results, in order to match to the best subset of a source dataset.

The potential of mining the web for rating information thus shifts the focus of building an accurate CF algorithm away from the *algorithm* and toward matching users to the appropriate *information* sources. The problem can thus be formulated as follows: given a user profile u, what profile *features* and emergent-*structural* properties of the kNN algorithm can be used to match the user to the best dataset? The preliminary experiments we report in Section 6 are promising, but still lack in the desired performance. In fact, alternative classification methods, with varying levels of dependence on the quality of the rating data, may perform better. We found that the strongest improvement was measured when data sets were adaptively selected for each user: the main result we observed is that classification accuracy is strongly related to the data source that is used, and improvement to the aggregate, global RMSE is proportional to how well users and data sources are linked. A viable option for building a CF system, therefore, need not rely on a combination of predictors [Koren, 2008], but rather on an optimal combination of data sources.

References

[Aciar *et al.*, 2007] S. Aciar, J. L. de la Rosa i Esteva, and J. L. Herrera. Information Sources Selection Methodology for Recommender Systems Based on Intrinsic Characteristics and Trust Measure . In *Proceedings of The 5th Workshop on Intelligent Techniques for Web Personalization*, Vancouver, Canada, 2007.

[Axelrod, 1997] R. Axelrod. The dissemination of culture: A model with local convergence and global polarization. *Journal of Conflict Resolution*, (41):203–226, 1997.

[Cho *et al.*, 2007] Jinhyung Cho, Kwiseok Kwon, and Yongtae Park. Collaborative filtering using dual information sources. *IEEE Intelligent Systems*, 22(3):30–38, 2007.

[Herlocker *et al.*, 2004] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. In *ACM TOIS*, volume 22, pages 5–53. ACM Press, 2004.

[Koren, 2008] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *ACM SIG KDD Conference*, 2008.

[McPherson *et al.*, 2001] M. McPherson, L. Smith-Lovin, and J.M. Cook. Homophily in social networks. *Annual Review of Sociology*, (27):415–444, 2001.

[O'Donovan and Smyth, 2005] J. O'Donovan and B. Smyth. Trust in recommender systems. In *IUI '05: Proceedings of ACM IUI*, pages 167–174. ACM Press, 2005.

[Su *et al.*, 2007] Xiaoyuan Su, Russell Greiner, Taghi M. Khoshgoftaar, and Xingquan Zhu. Hybrid collaborative filtering algorithms using a mixture of experts. In *Web Intelligence*, pages 645–649, 2007.