LEARNING NEIGHBOURHOOD-BASED COLLABORATIVE FILTERING PARAMETERS

J. Griffith[1], C. O'Riordan[1] and H. Sorensen[2]
[1]*College of Engineering and Informatics, National University of Ireland, Galway, Ireland*
[2]*Department of Computer Science, University College Cork, Cork, Ireland*

Keywords: Collaborative filtering, Neighbourhood-based approach, Genetic algorithms.

Abstract: The work outlined in this paper uses a genetic algorithm to learn the optimal set of parameters for a neighbourhood-based collaborative filtering approach. The motivation is firstly to re-assess whether the default parameter values often used are valid and secondly to assess whether different datasets require different parameter settings. Three datasets are considered in this initial investigation into the approach: Movielens, Bookcrossing and Lastfm.

1 INTRODUCTION

Since Herlocker et al.'s comprehensive investigation of collaborative filtering parameters for the Movielens dataset (Herlocker et al., 2002), there has been mostly general acceptance that these parameters are the best for the Movielens dataset. The space of possible parameter values, and their combinations, explored by Herlocker et al. were large and the exploration was done in a brute-force manner. In recent years there are a number of new collaborative filtering datasets available, with potentially different characteristics to the Movielens dataset. It is not clear if the results from the previous work by Herlocker et al. are applicable to different datasets.

The approach outlined in this paper uses a genetic algorithm to learn the best set of collaborative filtering parameters for three datasets (Movielens, Bookcrossing and Lastfm). Genetic algorithms are stochastic search techniques that evaluate a population of solutions (individuals) over a number of iterations (generations) and at each iteration, evaluate how good (or fit) each solution is (Holland, 1975). Based on this evaluation, some simple operations are performed on the solutions to create a new, "better" population for the next iteration. The process continues until a satisfactory solution is found or until a set number of iterations have been reached. The genetic algorithm approach has been applied successfully in other areas of Information Retrieval (Trotman, 2005). The motivation of the work is two-fold: firstly to validate, or improve upon, the default settings commonly used for the Movielens dataset (Herlocker et al., 2002); secondly to ascertain if the set of parameter values found for the Movielens experiment are also the optimal parameter values for two other datasets that display different characteristics to the Movielens dataset (Bookcrossing and Lastfm).

2 PREVIOUS WORK

A large body of work has concentrated on different collaborative filtering techniques and their evaluation and comparison. Within the popular neighbourhood-based approaches, much early work empirically evaluated variants of the approach (Breese et al., 1998). Herlocker et al. tested a classic neighbourhood-based collaborative filtering algorithm using the standard Movielens dataset and the mean absolute error metric (MAE) was used to compare results. The overall recommendations from the work for the Movielens dataset were (Herlocker et al., 2002):

1. to use Pearson correlation for the similarity measure.
2. to dampen similarity scores between users who have co-rated a small number of items. A devaluation term above 50 did not appear to improve results. The devaluation term was used by multiplying two user's correlation by $\frac{n}{d}$ where n is the number of co-rated items between the two users and d is the devaluation value.

Griffith J., O'Riordan C. and Sorensen H..
LEARNING NEIGHBOURHOOD-BASED COLLABORATIVE FILTERING PARAMETERS.
DOI: 10.5220/0003657404440447
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2011), pages 444-447
ISBN: 978-989-8425-79-9

3. to normalize user ratings by "deviation from the mean".
4. to use the *TopN* best neighbours (highest similarity to the test user) for neighbourhood selection. The potential best range of neighbours (*N*) was found to be between 20 and 60.
5. to weight neighbour contributions when forming predictions.

More recently, work has shown that results using a combination of a mean-square difference metric and the Jaccard coefficient between users outperforms the commonly-used approach using Movielens and Netflix datasets (Bobadilla et al., 2010). Two parameters from Herlocker et al.'s work - similarity between users and normalisation of user ratings - were re-evaluated using the Movielens and Netflix dataset (Howe and Forbes, 2008). It was found that Pearson correlation is not necessarily the best similarity metric to use and different parameterisations work better for different datasets.

Some work has applied genetic algorithms in the collaborative filtering domain. However, the approaches which learn per user are very computationally expensive. Hwang uses a genetic algorithm, per user, to learn an optimal weighting scheme for the collaborative filtering system for each user (Hwang, 2010). Both collaborative and inferred content information is used. In comparison to a traditional collaborative filtering approach, using the metrics of precision, recall and the f1 measure, improvements were seen with the genetic algorithm approach. Ko et al. first classify items into groups using a Bayesian classifier to reduce the dimensions of the space. A genetic algorithm is used to cluster users in this new space (Ko and Lee, 2002). Ujjin et al. use a genetic algorithm to find the best "profile" that describes each user in the dataset (Ujjin and Bentley, 2002). The Movielens dataset is used and 22 features from the dataset are used to create a profile for each user using the movie ratings and user and movie details. The weights for each feature are evolved, per user, using a genetic algorithm. Similarity is found between profiles.

3 METHODOLOGY AND TEST SETS

The collaborative filtering technique used is a standard neighbourhood-based test approach where a portion of users are chosen as the test users (10%) and a portion of their items are withheld as test items (up to 10%). The task is to generate predictions for the withheld test items for the test users. Using a similarity function, users similar to the test users are found (their neighbours). Deviation from the mean is used to normalise user ratings. Similarity scores between users are "dampened" if the number of items co-rated by two users is below a certain significance threshold. Using a prediction formula, predictions for test items are calculated using a function based on the neighbour's ratings for the test items, the neighbour's similarity score with the test user, the neighbour's mean ratings and the test user's mean rating. The accuracy of the predictions are calculated based on the predicted ratings produced by the system and the actual ratings given to the test items in the withheld set using mean absolute error (MAE).

For the genetic algorithm, the parameters chosen are based on a subset of those tested in the work by (Herlocker et al., 2002). The flow of control of the genetic algorithm experiment is as follows:
For each of 20 generations:

1. Pick test users and test items. A new set of test users and items are picked for each new generation to avoid over-fitting.
2. Randomly generate a population of individuals, of a fixed size (size is 50 in these experiments).
3. Calculate the fitness of each individual by setting all of the collaborative filtering parameters to the values indicated in the individual and running the collaborative filtering component. The average MAE is calculated and returned as the fitness score of the individual. The genetic algorithm for this experiment is required to *minimise* the fitness score.
4. Perform the genetic algorithm operators of crossover and mutation and selection. The crossover operator used is single point crossover and the crossover rate is 80%. The mutation rate is set at 5%. The selection operator used is roulette wheel selection.

The parameters tested per position in the chromosome are:

- *sigT*, the *significance threshold*, which is an integer in the range 0 to 100. This is used when calculating the similarity between users to dampen the similarity between two users if the number of co-rated items between the users is less than this threshold. The dampening used is that already outlined from the work by (Herlocker et al., 2002). 100 was chosen as the limit as it does not seem reasonable to dampen a similarity score if the number of co-rated items is greater than 100.
- *sim*, the *similarity option*, which is an integer value in the range 0 to 2. This indicates which

Table 1: Comparison of Datasets.

	ML	BX	Lfm
Domain	Movies	Books	Music
Num Users	943	77805	3080
Num Items	1682	185968	30520
%Sparsity	87.66%	99.9%	99.1%
Rating range	1-5	1-10	1 to 7939

similarity function should be used to find the similarity between users. The options are: Spearman rank correlation (0), Pearson correlation (1) or Cosine similarity (2).

- *P*, the *predict option*, which is an integer value in the range 0 to 3. This indicates which version of a prediction formula is used. There are two main differences in the prediction formulas currently tested: when selecting the neighbours whose ratings will be used to generate the predictions, whether the $topN$ best (most similar) neighbours are chosen (option 1 and 3) or whether all neighbours whose similarity is above a certain threshold (the correlation threshold) are chosen (option 0 and 2); and when calculating the average ratings of users and neighbours, whether these averages are calculated over all the ratings a user or neighbour has given (option 2 for correlation thresholding and option 3 for $topN$ neighbour selection) or only over the ratings given to co-rated items between the current test user and the current neighbour (option 0 for correlation thresholding and option 1 for $topN$ neighbour selection).
- *N*, the *topN value*, which is an integer in the range 0 to 300. This is used when the predict option of using $topN$ (option 1 or 3) is chosen and indicates the number of neighbours that will be used to form a prediction.
- *corrT*, the *correlation threshold value*, which is a real value in the range $[0.0 - 0.35]$. This is used when the predict option of correlation thresholding (option 0 or 2) is chosen. The limit of 0.35 was chosen as in reality user similarities would rarely be greater than this.

Three data sets are used in the experiments: Movielens (ML), Bookcrossing (BX) and Lastfm (Lfm). Each dataset has slightly different characteristics. A summary of the datasets is outlined in Table 1.

The Lastfm data differs to the other two in that the number of times a user listens to a music track (the playcount), is stored rather than a discrete rating for an item. The normalisation used for the experiments in this paper maps the playcounts to discrete values in the range $[1 - 6]$ based on 6 "buckets", where the first 5 buckets contain playcounts in steps of 10, e.g. playcounts in the range $[1 - 10]$ are mapped to 1, playcounts in the range $[11 - 20]$ are mapped to 2, etc. The final bucket (with value 6) contains playcount values from 51 upwards.

Table 2: Learning 5 parameters.

	sigT	P	N	corrT	sim	MAE
ML	18	1	199	0.319	1	0.627
BX	9	2	241	0.015	1	5.07
Lfm:	1	2	279	0.032	2	0.655

4 RESULTS

The goal of the experiment is to find the individual (set of parameter values) with the lowest MAE score. Table 2 outlines the results for all the parameters specified for the three datasets.

For Movielens, the prediction option chosen is $topN$ with co-rated means (option 1). The number of neighbours (*N*) is high at 199. For Bookcrossing and Lastfm, correlation thresholding (option 2), where means are not calculated over co-rated items, is chosen. The threshold values are low at .015 and 0.032. In the two different scenarios selected, there is likely to be a small difference between a $topN$ neighbourhood selection approach with a high *N* value and a correlation threshold approach with a low correlation threshold value.

The similarity option (option 1) of Pearson correlation is chosen for Movielens and Bookcrossing but cosine similarity (option 2) is chosen for Lastfm. An initial experiment using the same parameter values as in Table 2, except using $P = 0$, i.e., correlation thresholding with co-rated means, produced an equally good MAE (an average of 0.731 over 10 runs). This suggests that cosine similarity does seem a better similarity function than Pearson correlation for this normalised Lastfm dataset and it was not any of the other factors which accounted for the improved MAE.

A very low significance threshold value is selected for Bookcrossing and Lastfm. This indicates that dampening the similarity measure between users with a small number of co-rated items is not useful for the Bookcrossing and Lastfm dataset, whereas doing so is beneficial in the Movielens case. This makes particular sense for the Bookcrossing dataset where the dataset is extremely sparse and where any evidence, even between users with only a few co-rated items, is

better than the common case of having no evidence available to find similar users.

Each best set of parameters found were run 10 times (for 10 different test sets) in a collaborative filtering approach to test how accurate the best MAE found was. Table 3 shows the average MAE over ten runs in comparison to the best MAE found by the genetic algorithm for each of the best solutions found per dataset. It can be seen from Table 3 that the best MAE found is significantly better than the average MAE found when using these parameter values in a collaborative filtering system over 10 runs. Further runs need to be performed to test and analyse these results and to see if results are replicated over additional runs.

Table 3: Best MAE Vs Average MAE.

	Best MAE	**Average MAE**
ML	0.627	0.731
BX	5.07	5.91
Lfm	0.655	0.746

5 CONCLUSIONS AND FUTURE WORK

A genetic algorithm approach was used to learn an optimal set of parameters for three datasets in a nearest-neighbour collaborative filtering approach. The sample space of parameters and their possible values and the potential combinations of different parameters was considered too large and unwieldy to perform a brute force analysis of the problem. For this reason a genetic algorithm approach was adopted where each individual represented a set of values for parameters. The fitness of each individual was calculated by running a collaborative filtering approach on a test set using the parameter values specified in the individual and calculating the mean absolute error (MAE) of the results. Although the approach is computationally expensive it only needs to be carried out once per dataset. Results show that the genetic algorithm does converge to useful results which do not always agree with previous results. It could be argued in some instances that the field of recommendation has moved past a complete reliance on the neighbourhood-based model outlined in this paper and that the recent focus is on incorporating additional information that is available. Whilst this is undoubtedly an avenue of work which can potentially overcome many of the disadvantages associated with a pure collaborative filtering approach, there is still scope to continue investigation into the assumptions and parameter values chosen for the basic collaborative filtering approach. It is from such a perspective that the work outlined in this paper was undertaken.

REFERENCES

Bobadilla, J., Serradilla, F., and Bernal, J. (2010). A new collaborative filtering metric that improves the behaviour of recommender systems. *Knowledge-Based Systems*, 23:520–528.

Breese, J., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann.

Herlocker, J., Konstan, J., and Riedl, J. (2002). An empirical analysis of design choices in neighbourhood-based collaborative filtering algorithms. *Information Retrieval*, 5:287–310.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Howe, A. and Forbes, R. (2008). Re-considering neighbourhood-based collaborative filtering parameters in the context of new data. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*.

Hwang, C.-S. (2010). Genetic algorithms for feature weighting in multi-criteria recommender systems. *JCIT: Journal of Convergence Information Technology*, 5(8):126–136.

Ko, S. and Lee, J. (2002). User preference mining through collaborative filtering and content based filtering in recommender system. In *Third International Conference on Electronic Commerece and Web Technologies (EC-Web)*, pages 244–253.

Trotman, A. (2005). Learning to rank. *Journal of Information Retrieval*, 8(3).

Ujjin, S. and Bentley, P. (2002). Learning user preferences using evolution. In *4th Asia-Pacific Conference on Simulated Evolution and Learning*.