
An Online Social Network-based Recommendation System

Jorge Aranda, Inmar Givoni, Jeremy Handcock, Danny Tarlow
Department of Computer Science
University of Toronto
10 King's College Road, Toronto, Ontario, Canada, M5S 3G4

Abstract

We have previously proposed to implement a social network-based recommendation system that uses data from and interfaces with the BoardGameGeek (BGG) website. The end result system will be usable by members of BGG to get recommendations given both their ratings history and their friend relationships. This report details the steps that we have taken so far and the steps that still need to be done, along with an expected timetable.

We have gathered most of the needed data from the BGG website, and we have some of the algorithmic implementation finished for our recommendation service. The next steps are to finalize data collection, finish and test the algorithm, run experiments to determine suitable parameter settings for the algorithm, and create a web-based application.

1 OVERVIEW OF PROJECT TASKS

Collect user data from the BGG website

This requires crawling the web-page since there is no available API for collecting the type of information we need. We need to create a database of games, users, users's social network (as defined by the "GeekBuddies" category) and users's game ratings. We decided to collect the data as a batch process, to store it in a database, and to then hand it over to the recommendation algorithm.

Design and Implement the recommendation algorithm

Decide on which approach to take for solving the problem, implement the algorithm in C++, define the interface that will allow the web-application to retrieve recommendation for a new user. Test the algorithm.

Experimental evaluation and parameter tuning

Once the algorithm is in place, one can think of different ways to use the friendship information (*e.g.* by using only immediate friendships, treating all friendships equally). We would like to test how these different choices affect the performance of the algorithm.

Create a web-based application that uses our collected data to provide any BGG user with game recommendations based on their previous ratings and on ratings made by their "GeekBuddies".

We will now describe what's been done and what's to be done for each of these tasks. Note that if all goes well, we have several additional modifications and enhancements we can explore (using additional information from the website, trying to engage real users to get actual feedback, etc.). Time permitting, we may be able to follow through on these ideas as well.

2 DATA COLLECTION

We intended to obtain our data directly from the BGG system administrators. We e-mailed them two times, but received no response from them. Therefore, we decided to write a set of crawlers to gather the data.

We have started collecting the data from the BGG website for our application. At this point, we have finalized crawling the following data:

- Games that any user has rated (about 30K)
- Users that have rated any games, and all the ratings they have given (about 35K of the 110K registered users, as well as their game collections and wish lists.)

A crawl to get the links between users ("GeekBuddies") is underway, and should finish by the end of this week.

To get the user information and ratings we used an API provided by the BGG website. However, getting

game and GeekBuddy data was more problematic, as it was only available to logged-in users. To overcome this our crawler had to “trick” the website, passing as a logged-in user.

We will save the collected data in a database, and we will write a script to allow an easy generation of a csv/tsv file with sparse representation of information to be used by the machine learning algorithm.

Timetable:

Nov 7th - Data collection finished

Nov 16th - Integration of DB with ML algorithm

3 ALGORITHM

We have chosen to implement a modified version of probabilistic matrix factorization (PMF [Minh 07]). Different variants of this approach are very popular among Netflix competitors [Netflix.com]. We have obtained a somewhat messy Matlab version of the code and are about half-way through re-implementing it in C++. To this implementation we need to add the module that incorporates user friendship relationships. We have developed a principled approach for doing that, and from an implementation point of view the modification to the algorithm is fairly straight forward (we hope).

Timetable:

Nov. 7th - Finalize code, finish debugging, test against the Matlab implementation, incorporate modification. Test on toy data/available data.

Nov 16th - Integration with DB

Possible problems: handling a very large dataset might require some additional hacking and tweaking.

4 EXPERIMENTAL EVALUATION

We will use some of the collected information as a test set and find out whether we can achieve any improvement in predicting user ratings compared to the unmodified algorithm. We would like to test different settings of how we construct the weights and types of friendships.

Nov. 7th - Prepare a set of Matlab scripts to test the above.

Nov. 20th - Run on real data, evaluate best setting for the friendship weights.

5 APPLICATION

The end result of our project will be a web-based application in which users of the BGG website can log in

and get recommendations from our algorithm.

The service will work as follows. We will have a job executing weekly to pull data from BGG, and passing it on to our recommendation algorithm. The algorithm will output weight matrices, which will be uploaded to our server. Whenever a user logs in to our website, the server will perform a matrix multiplication for that user, and present a set of recommendations for him. New users will be presented with a list of board games to rate, so that we can offer them some initial recommendations.

Ideally, linking this application with the BGG website would conveniently allow us to use dynamic data, and would be more convenient for BGG users. We will continue to attempt to communicate with the system administrators of the website to establish this link.

Nov. 23rd - Initial functionality

Dec 1st - A working application; if communication to BGG system administrators fail we will post a thread to let BGG users know about our service and encourage them to use it.

6 TIMETABLE

Here is the concentrated timetable of our project plan for the rest of the term:

1. Nov. 7th - Data collection finished
2. Nov. 7th - Finalize code, finish debugging, test against the Matlab implementation, incorporate modification. Test on toy data/available data.
3. Nov. 7th - Prepare a set of matlab scripts to test initial algorithm.
4. Nov. 16th - Integration of DB with ML algorithm
5. Nov. 20th - run algorithm on real data evaluate best setting for the friendship weights.
6. Nov. 23rd - Initial functionality of web application
7. Dec 1st - A working application and announcement to BGG community

References

A Minh, *et al.* Probabilistic Matrix Factorization Applied to the Netflix Rating Prediction Problem. *To appear in NIPS 2007* .

Netflix forum. <http://www.netflixprize.com/community/viewtopic.php>

Board Game Geek. <http://www.boardgamegeek.com/>