

Board Game Recommendation

Sam Helmich

February 12, 2015

1 Introduction

Board gaming, while being very popular among children, is a niche hobby beyond adolescence. Everyone may have grown up with Monopoly, but most people may have never heard of Citadels or Twilight Imperium. And as such, board gaming is a very difficult hobby to break into. "Gateway Games" such as Settlers of Catan or Ticket to Ride offer a gradual transition between the ultra-popular and mechanics-lite games like monopoly, to more rules-heavy games like Puerto Rico. The most difficult part of the transition, for most people, is actually finding new games to play. New games are often expensive for people who may be apprehensive about trying new games (a copy of Settlers of Catan goes for \$37.80 on Amazon.com). For this reason, a method to recommend games to people would be invaluable.

A few methods already exist, however, their usefulness varies. Boardgamegeek.com, a popular site among frequent board gamers, has a recommendation system that allows users to find a particular game they like, and then from there it will recommend more games that a person may like. A pretty thorough analysis of it and its issues can be found at http://boardgamegeek.com/wiki/page/Game_Recommendation. The biggest issue is that it cannot be tailored to an individual. You, an individual user, may have different preferences than the users as a whole, and it might be more useful to tailor recommendations to the individual.

A redditor identified this issue, and created a board game recommender on the subreddit boardgamerecommender. You can "summon" the recommender by posting your boardgamegeek username anywhere on reddit, preceded by "/u/boardgamerecommender". This will prompt the bot to go search through your ratings of games on boardgamegeek, and then it will make a post with 4 lists of games: New games you may enjoy, older games you may enjoy, games you may find underrated, and games you may find overrated. This is a huge step forward for individuals that are looking for games, as they can now rate games they've played, and can then get recommendations for games they might like. However, there are a few drawbacks. The process has a few possible user inputs (such as playtime, number of players, category, mechanic, etc) that will narrow down the games that will be returned to the user. However, with each specification, a new recommendation will be generated, a process that is said to take 10-30 minutes (FIND A BETTER ESTIMATE FOR THIS). Also, there is no published methodology, so we cannot view how these recommendations are made.

This being said, these are only two board game recommendation systems out of the many out there. In this paper, we seek to build a system that will recommend board games to users based on their own recommendations as well as the recommendations of others.

2 Data Collection

data scraped from boardgamegeek.com scraped in 2 sections - games and ratings pulled down whole pages due to extensive nature after pages were pulled down, code was written to extract

the pertinent information from each test data set was dumped into database database has a few tables - game key, player key, ratings, classifiers, other the structure of this database allows for easy functionality

3 Model

3.1 Model Selection

To begin implementing such a model, we can look to two papers on one of the most famous recommender systems of all: Netflix. In 2006, Netflix offered a \$1 Million prize for a improvement on their algorithm for predicting user rating on movies they had not seen. In 2008, two teams: BigChaos and BellKor (the eventual winner), published papers on their methodology. In the end, they ended up taking something of a shotgun approach, using a linear blend of many different models to come up with a single super-model.

There are a few things that make sense to model first. Certain games are more popular than other games, and certain people will naturally be more or less generous than others in their ratings. These two effects are very important to measure, as the first will allow us to provide estimates for people that have rated no games (and therefore we know nothing about), and the second will allow us to fine tune estimates for people who have rated many games. We will implement these in the same way that BigChaos implemented the Global Effects model (page 6, BigChaos paper). While they implemented 14 steps, we will only use the 2, due to the complexity of the calculations required and the time with which it is practical to fit a model.

After we fit the general effects, we wish to model the remaining residuals, ideally in some way that gives us informative numeric summaries about the individual users in the process. In addition, we'd also like to compare a users ratings to other users, and hopefully use the behavior of simiar users to predict new ratings for someone. A tempting first step is to, for a particular game, find other people who have rated that game who have rated games that you've rated. Then, you could find people who rated games similarly to you, and use their ratings of the new game as an estimate. This often doesnt work, as sometimes the number of people who have rated the same games as you can be relatively low, if not 0 (show a graph for this part).

To get around the problem of low crossover in game ratings, we turn to the characteristics of the games themselves. This kills two birds with one stone. We'll create, for each user, a rating for each game mechanic and genre. Then, since there are so many mechanics and genres of game, a player only needs to rate a small, albeit diverse set of games before we can correlate that player's habits with other players, even if they've never played any of the same games! Then we can use those to filter rating results, as we'll know what board game mechanics that a particular player will enjoy.

We now have a very simple two step model that will allow us to predict ratings for new board games that a person may not have even played! Also, it allows us to rate games that nobody has played before: We can use the overall mean as a starting point, then for an individual player, we can adjust the rating based on their ratings of similar games from other genres.

3.2 Model Fit

The specifics of the model follow very closely to those outlined in the Global Effects and knn sections of the BigChaos paper [need bib]. Here, we cover the specifics of both.

3.2.1 Global Effects

The motivations for using the formulas that follow can be found in Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights (make this a bib reference). The general

idea is to remove the global mean from all ratings to get a residual for each observation. We'll group these residuals by game, and then find an effect for that game. Subtracting this effect from the residual, we'll take those residuals and repeat the same for each user.

As we proceed $y_{u,i}$ will denote the rating by user u for game i . Then, we have

$$y_{u,i} = \mu_{global} + \mu_i + \mu_u + error$$

However, we'll approach this in steps, as follows:

$$y_{u,i} = \mu_{global} + error_{global} \tag{1}$$

$$error_{global} = \mu_i + error_{game} \tag{2}$$

$$error_{game} = \mu_u + error_{user} \tag{3}$$

And we'll use the estimation method outlined in the paper (need ref) for each, as follows below:

$$\mu_{global} = \frac{\sum_{u=1}^{N_u} \sum_{i=1}^{N_{games}}}{1} \tag{4}$$

4 Implementation

4.1 Individualized Summaries

4.2 Group Summaries

5 Conclusion

6 Data Management

7 Model

Let y_{ij} be the rating for the i^{th} player of the j^{th} game. We fit the model in two steps, first finding the global residual, and then using a k-means approach to estimation.

7.1 Global Residual

The goal of a global residual is to remove two logical biases: some games are naturally rater higher than othes, and certain players are more generous in their ratings than others. The global residual will be the result of removing these baises. Note that this process by itself can be used as a prediction technique, but more importantly, the residuals it generates can be used as the input to other prediction strategies to improve overall performance (as seen in the section where we evaluate the performance of the different strategies). The process for finding these residuals is identical to the method described by [that paper that i used from the netflix part].

7.2 Model Fitting

The model fitting needs to be autonomous, so we need to algorithmically proceed through the model fitting. We'll proceed as follows:

1. Update α
 - (a) Update α_1 - parameter for game means
 - (b) Update α_2 - parameter for player means
2. Update k for k-means

This follows the natural flow of data through the model, so this is the best method for finding parameter estimates.