Genre vs year: analysis and evaluation of the Netflix winter 2001-2002 dataset

Alan Kreher[1,2,3], Chantal van de Luijtgaarden[1,2,3], Dion Verschuren[1,2,3], Dylan van Wonderen[1,2,3], Suzanne Ekhart[4], & Tawab Ghorbandi[1,2,3]

[1] Jheronimus Academy of Data Science

[2] Eindhoven University of Technology

[3] Tilburg University

[4] Utrecht University

Genre vs year: analysis and evaluation of the Netflix winter 2001-2002 dataset

Contents

Executive Summary

In this research a light was shed on the question what the effects of genre on the likelihood of having two movies of the same genre watched is and how much liking the same movies would influence disliking the same movies (and vice versa).

Using a dataset with all movie reviews of the streaming site Netflix in the winter of 2001/2002 extended with own collected data about movie genres and release year, these questions were answered with a network perspective. Using QAP and ERGM the network of movie ratings by users are analysed.

It was found that homophily in both genre and the decade a movie came out were good predictors of an individual liking both movies. It was found that liking a movies was a better predictor for disliking movies compared to the effect the other way around.

Introduction

Main topic. In this paper, we are going to study the presence of social networks within a movie streaming platform. We are focusing on the structure of links among a group of social players, which consist of users watching and rating movies on Netflix.

Users of Netflix's movie recommendation algorithms are frequently given specific questions about their interests for certain items (which they provide by liking or disliking them, for example). These choices are then immediately integrated into the underlying learning system for future suggestions. If a recommender system starts promoting unwanted products after incorporating new preferences, the user may try to steer the system in the future by correcting it or supplying alternate preference information.

Importance. It is important to study the presence of these socials networks because this could potentially improve the recommender engine that this currently in place. For example, if you know that a user is likely to like a movie that other users with the same "liking profile" also like, you can recommend that movie to the user. When these connections are studied thoroughly, you could have a high probability that the recommendation is successful. This could have a large impact on a movie streaming platform.

Existing studies. In this paper, we will be looking into the Netflix Price Dataset. In 2006 Netflix decided to start a competition with a grand prize of 1 million US dollars. The goal of the competition was to create a collaborative filtering algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films. In order to win you had to at least improve on Netflix's own algorithm by 10%. During the competition a lot of literature has emerged about the dataset and the competition. (Bell & Koren, 2007; Narayanan & Shmatikov, 2006; Takács, Pilászy, Németh, & Tikk, 2008)

However, no papers or any other literature can be found on network analysis on this dataset. We like to fill this gap in the literature by analysing the network and network structure that arises from the subset of the data that we will use. (Guillory & Bilmes, 2011) developed an active movie recommendation system for Netflix. They found that a recommender system should not constantly ask questions to a user, because those reduces the user's mental image of how the recommendation system learns, prompting some participants to "lose track of what they were teaching." According to Amershi, Cakmak, Knox, and Kulesza (2014), this was because users are not always eager to act as simple oracles (repeatedly telling the recommendation system whether they like something or not). This is interesting to take into account for our research, because this would mean that a social network within a movie recommendation system can never be fully exposed.

Research Questions and hypotheses. In this dataset, we can easily see connections between users and movies, but not between just the users or just the movies. At least not, when we do not include one or the other. That is why we have defined the following research questions:

- RQ1: What are the effects of the genre on the likelihood of having both movies watched by the same user?
- RQ2: How much does liking the same movies influence disliking the same movies, and vice versa? (In a network of frequent reviewers)

In order to be able to answer these research questions, we have made set up the following hypothesis accordingly:

- Research Question 1:
  - Comparable movies, in terms of either genre and/or year, are more often watched by the same user than uncomparable movies.
  - Movies of genre horror are generally rated higher than other movie genres.
- Research Question 2:
  - Users that like the same movies to a certain degree, are likely to dislike the same movies to that same degree
  - Users that dislike the same movies to a certain degree, are likely to like the same movies to that same degree

Methodology

Dataset. During this study, the data that was shared by Netflix during the Netflix Prize open competition is used. The competition was about developing the best algorithm to predict user ratings for content on Netflix. The contest was started in order to improve their recommender system.

The data consisted of movies and account holders on Netflix who rated the movies on a 5 point scale. Also, the date the rating that was given and the year of the movie release are included in the dataset. Data was collected between October 1998 and December 2005 and the data consists of all ratings that were given during this period (Netflix, 2019). The initial dataset contains about 24 million ratings from over 480k users on almost 18k movies. (Takács, Pilászy, Németh, & Tikk, 2008)

For the movies, the dataset only contains the title of the movie and the year it was released. This was too little information about the movies for our project. Therefore, we decided to look for an additional data source to enrich the dataset. We used Amazon Prime movie genres to add genres to the dataset for further analysis. This data was collected on our own by looking up every movie separately in the Amazon Prime database, and retrieving the genre.

The large size of the dataset made it difficult to work with and made running analysis models infeasable. Therefore, we decided to only use a selection of the ratings. The dataset has ratings from October 1998 to December 2005. For this analysis, only the data from December 2001, Januari 2002, and februari 2002 are considered. We chose this winter because this is the first time where users started to rate a good amount of movies while the number of ratings still did not completely explode yet.

To answer our two research questions we had to transform our data further in two different ways. For the first research question, the network is defined as movies for the nodes and the movies have an edge between them if there is an user that likes both movies. Liking a movie is defined as a user giving a 4 or 5 rating for a movie. Of course, modelling the network this way means that movies that are highly rated are always connect. Therefore, the network only included movies that have between 20 and 50 ratings. That means that our analysis only concerns niche movies with a small number of ratings. Moreover, there is a lower threshold too to make sure that we only include movies that have at least a few connections. All in all, the network is just under 500 nodes and the attributes for the nodes are the year of release of the movie and the main movie genre that comes from Amazon Prime.

In order to answer research questions about liking behaviour of customers on streaming platforms (such as Netflix) this dataset provides us with a great opportunity. The dataset contains a unusual amount of real, user generated data. Datasets with similar types of data usually contain a lot less data and the large dataset is thus very useful in order to answer research questions about liking behaviour. (Takács, Pilászy, Németh, & Tikk, 2008)

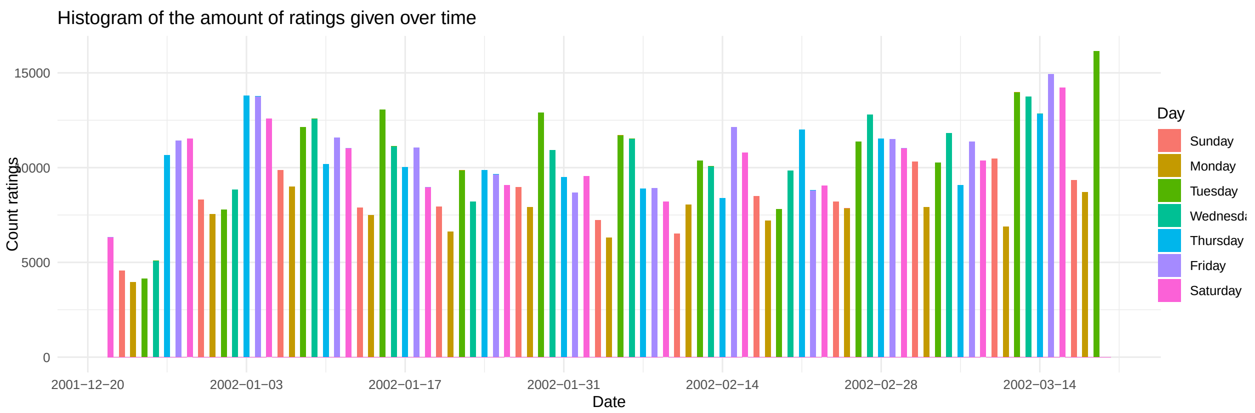
As stated in the initial distribution of the data, Netflix can and will not guarantee the correctness of the data. As no perfect documentation of the data collection exists, this can cause inaccuracies in the results. Also, Netflix uses algorithms that determine what users see and this effect can influence the results of this research. We could interpret the effect of the algorithm as a effect caused by user behaviour.

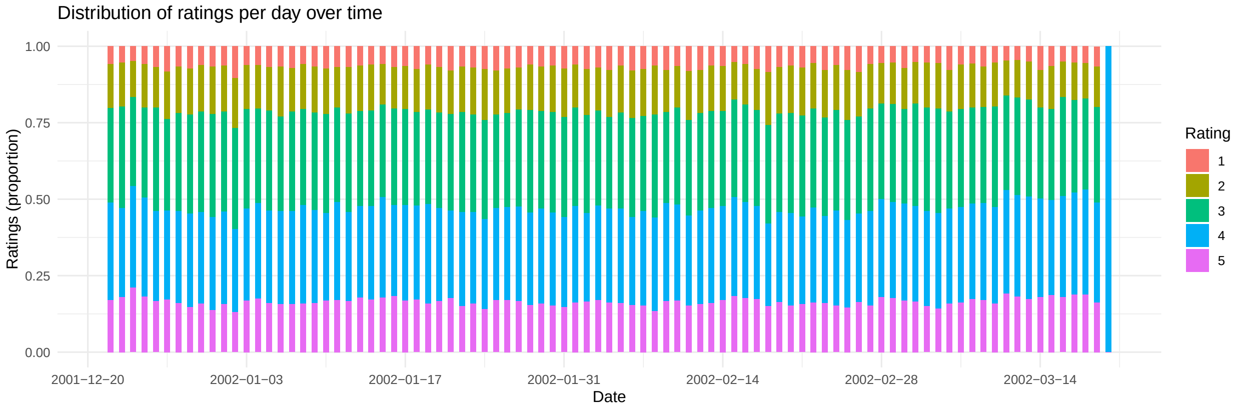
Descriptives.

Looking at the average Rating of the users in the data, the average rating they gave was about a $3,36$.

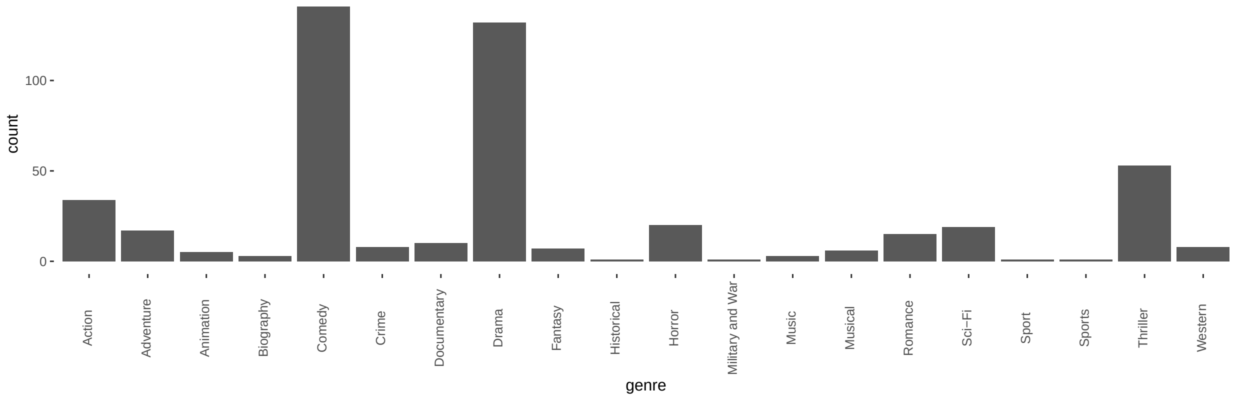
	mean	sd	min	max	n
Rating	3.362346	1.114827	1	5	864610

Looking at the histogram of the amound of ratings given on a certain date we can see that a peak in ratings exists always in the middle of the week (around wednesday/thursday). Looking at the distribution of the ratings over time we don't see great differences. Over time, people give consistent ratings.


Histogram of the amount of ratings given over time


Distribution of ratings per day over time

The genres in the data were determined using Amazone Prime. The first genre that Amazone Prime displayed is taken as the genre of the movie. As the genres that were displayed in Amazone Prime were not on alfabethic order, we assume that the first genre given is the main genre. 20 genres were found in the Netflix dataset. Most of the movies had *comedy* or *drama* as genre.



Data analysis (Research Rationale).

QAP.

In general, QAP is very interesting, when you have two or more networks and want to compare those. In addition, we also choose QAP regression, since we thought it would be interesting to use this model, based on its permutation of actually observed graphs and therefore staying closer to the nature of our networks.

There are three different QAP models from which one had to be chosen for our data and research question. The first is the simple QAP test where you measure the association between two networks. Furthermore, there is a QAP linear model, which can be used on a valued dependent network and one or more explanatory networks. Finally, there is the QAP logistic model, which differs from the linear model, in terms of a binary-valued dependent network.

Our weights in our network are not binary so the logistic model is not an option for us. Even though, we only have two networks we decided to go for the linear model since we would like to see if we could "explain" the first network with the second and vice versa.

As QAP is the only method (to our knowledge) that is able to compare two networks to each other, there aren't any other methods available to address the second research question.

ERGM.

Many metrics like density and centrality are used to describe the structural characteristics of an observed network. These metrics, on the other hand, describe the observed network, which is just one of many possible alternative networks. The structural characteristics of this group of alternative networks may or may not be similar. A statistical model must consider the set of all possible alternative networks weighted in their similarity to an observed network to credit a statistical inference about the processes that influence the formation of network structures.

Network data, on the other hand, violates the independence and identical distribution assumptions of statistical models like linear regression because they are inherently relational. Alternative statistical models must reflect the uncertainty associated with a given observation, allow inference about the relative frequency of theoretically important network substructures, eliminate ambiguities in the influence of confounding processes, efficiently represent complex structures, and link local-level processes with global-level properties. For example, degree-preserving randomization is a method of considering an observed network in terms of multiple alternative networks.

Exponential Random Graph models (ERGM) is a powerful technique for conducting statistical inference on network data. ERGM is a good choice as a network modelling framework for cases where the outcome of interest is in the presence or absence of edges. That makes it a suitable method for our research question, because this question is focused to explain the structure of a network of movies with a specific genre and decade year.

Results

QAP Model.

Constructing data and the model.

During preprocessing of the data, an interesting artifact was encountered; When users were removed from the dataset that either liked or disliked a movie, it was clearly shown that it occurred much more often that user gave a 4/5 star rating to a movie than a 1/2 star rating. This is interesting to keep in mind, as this might pose a potential bias in the dataset.

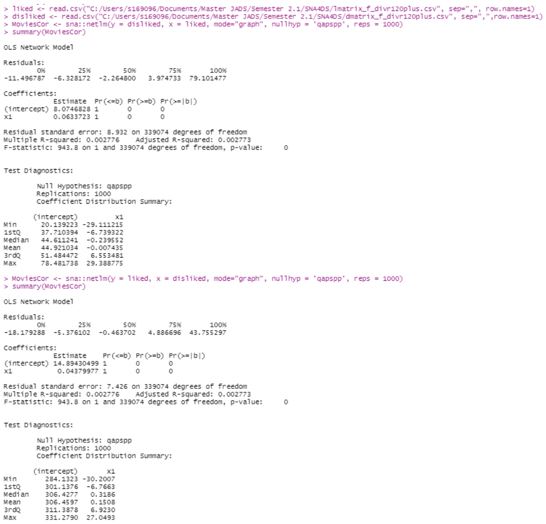
To give a weight to the edges in terms of importance, we calculate the weight according to a formula: $weight = roundup((\frac{no.ofmutuallikedmovies}{max(no.oflikedmovies)}) * 100)$

Results.

We have generated results for different (filtered) versions of the dataset. The results are explained and shown below.

More than 120 reviews.

In order to get a feeling of the QAP regression model, and to get initial results in a reasonable amount of time, we first did the analysis on users who had 120 reviews in total(like and dislike) or more. The results are shown in the figure below.



```
> liked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/lmatrix_f_divr120plus.csv", sep=",", row.names=1)
> disliked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/dmatrix_f_divr120plus.csv", sep=",",row.names=1)
> MoviesCor <- sna::netlm(y = disliked, x = liked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor)

OLS Network Model

Residuals:
        0%         25%         50%         75%        100% 
-11.496787  -6.328172  -2.264800   3.974733  79.101477 

Coefficients:
            Estimate  Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 8.0746828 1       0       0        
x1          0.0633723 1       0       0        

Residual standard error: 8.932 on 339074 degrees of freedom
Multiple R-squared: 0.002776    Adjusted R-squared: 0.002773 
F-statistic: 943.8 on 1 and 339074 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)         x1
Min      20.139223 -29.111215
1stQ     37.710394  -6.739322
Median   44.611241  -0.239552
Mean     44.921034  -0.007435
3rdQ     51.484472   6.553481
Max      78.481738  29.388775

> MoviesCor <- sna::netlm(y = liked, x = disliked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor)

OLS Network Model

Residuals:
        0%         25%         50%         75%        100% 
-18.179288  -5.376102  -0.463702   4.886696  43.755297 

Coefficients:
            Estimate    Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 14.89430499 1       0       0        
x1           0.04379977 1       0       0        

Residual standard error: 7.426 on 339074 degrees of freedom
Multiple R-squared: 0.002776    Adjusted R-squared: 0.002773 
F-statistic: 943.8 on 1 and 339074 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)       x1
Min       284.1323 -30.2007
1stQ      301.1376  -6.7663
Median    306.4277   0.3186
Mean      306.4597   0.1508
3rdQ      311.3878   6.9230
Max       331.2790  27.0493
```

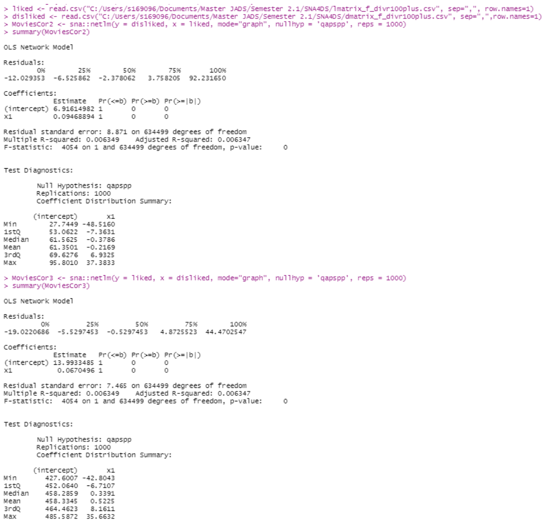
Figure 1

As can be seen from the figure, the QAP regression model explains only 6% of the variation in the like/dislike interaction network for movies when looking at liked movie ratings. When looking at disliked movie ratings, it becomes even less: 4%.

More than 100 reviews.

Next, we decreased the filter threshold for the number of ratings that users have given in the dataset. This increased the amount of datapoints in the dataset. So, we had a dataset with users who had 100 reviews in total (like and dislike) or more, and ran the analysis. The results are shown in the figure below.

As can be seen from the figure, the QAP regression model explains only 9% of the variation in the like/dislike interaction network for movies when looking at liked movie ratings. When looking at disliked movie ratings, it becomes even less: 6%.



```
> liked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/lmatrix_f_divr100plus.csv", sep=",", row.names=1)
> disliked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/dmatrix_f_divr100plus.csv", sep=",",row.names=1)
> MoviesCor2 <- sna::netlm(y = disliked, x = liked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor2)

OLS Network Model

Residuals:
        0%        25%        50%        75%       100% 
-12.029353  -6.525862  -2.378062   3.758205  92.231650 

Coefficients:
            Estimate   Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 6.91614982 1       0       0        
x1          0.09468894 1       0       0        

Residual standard error: 8.871 on 634499 degrees of freedom
Multiple R-squared: 0.006349    Adjusted R-squared: 0.006347 
F-statistic:  4054 on 1 and 634499 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)       x1
Min        27.7449 -48.5160
1stQ       53.0622  -7.3631
Median     61.5625  -0.3786
Mean       61.3501  -0.2169
3rdQ       69.6276   6.9325
Max        95.8010  37.3833

> MoviesCor3 <- sna::netlm(y = liked, x = disliked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor3)

OLS Network Model

Residuals:
         0%         25%         50%         75%        100% 
-19.0220686  -5.5297453  -0.5297453   4.8725523  44.4702547 

Coefficients:
            Estimate   Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 13.9933485 1       0       0        
x1           0.0670496 1       0       0        

Residual standard error: 7.465 on 634499 degrees of freedom
Multiple R-squared: 0.006349    Adjusted R-squared: 0.006347 
F-statistic:  4054 on 1 and 634499 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)       x1
Min       427.6007 -42.8043
1stQ      452.0640  -6.7107
Median    458.2859   0.3391
Mean      458.3345   0.5225
3rdQ      464.4623   8.1611
Max       485.5872  35.6632
```

Figure 2

More than 80 reviews.

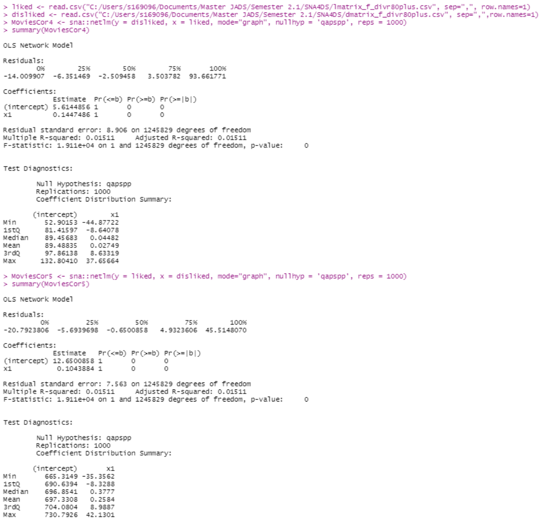
Because of the improved results in the last section, we decreased the filter threshold for the number of ratings that users have given in the dataset even more. This again increased the amount of datapoints in the dataset, and we did the analysis on users who had 80 reviews in total (like and dislike) or more. The results are shown in the figure below.

As can be seen from the figure, the QAP regression model explains 14% of the variation in the like/dislike interaction network for movies when looking at liked movie ratings. When looking at disliked movie ratings, it again becomes a bit less: 10,4%.

More than 60 reviews.

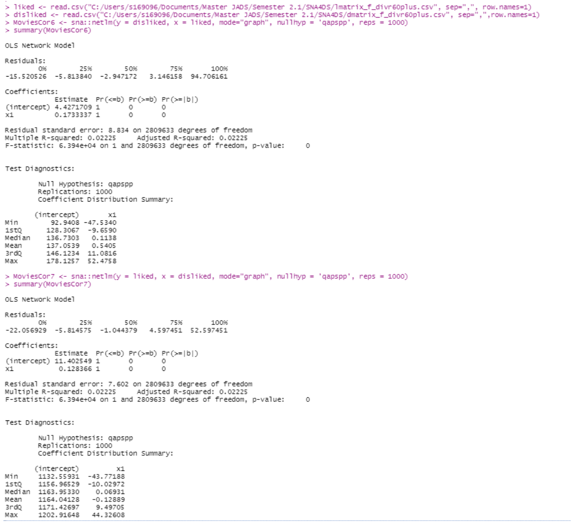
For the last time, we decreased the filter threshold for the number of ratings that users have given in the dataset. This also increased the amount of datapoints in the dataset, and we ran the analysis on users who had 80 reviews in total (like and dislike) or more. The results are shown in the figure below.

As can be seen from the figure, the QAP regression model explains 17% of the variation in the like/dislike interaction network for movies when looking at liked movie ratings. When looking at disliked movie ratings, the model can explain 13% of the



```
> liked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/lmatrix_f_divr80plus.csv", sep=",", row.names=1)
> disliked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/dmatrix_f_divr80plus.csv", sep=",",row.names=1)
> MoviesCor4 <- sna::netlm(y = disliked, x = liked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor4)

OLS Network Model

Residuals:
        0%        25%        50%        75%       100% 
-14.009907  -6.351469  -2.509458   3.503782  93.661771 

Coefficients:
            Estimate  Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 5.6144856 1       0       0        
x1          0.1447486 1       0       0        

Residual standard error: 8.906 on 1245829 degrees of freedom
Multiple R-squared: 0.01511     Adjusted R-squared: 0.01511 
F-statistic: 1.911e+04 on 1 and 1245829 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)        x1
Min       52.90153 -44.87722
1stQ      81.41597  -8.64078
Median    89.45683   0.04482
Mean      89.48835   0.02749
3rdQ      97.86138   8.63319
Max      132.80410  37.65664

> MoviesCor5 <- sna::netlm(y = liked, x = disliked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor5)

OLS Network Model

Residuals:
         0%         25%         50%         75%        100% 
-20.7923806  -5.6939698  -0.6500858   4.9323606  45.5148070 

Coefficients:
            Estimate   Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 12.6500858 1       0       0        
x1           0.1043884 1       0       0        

Residual standard error: 7.563 on 1245829 degrees of freedom
Multiple R-squared: 0.01511     Adjusted R-squared: 0.01511 
F-statistic: 1.911e+04 on 1 and 1245829 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)       x1
Min       665.3149 -35.3562
1stQ      690.6394  -8.3288
Median    696.8541   0.3777
Mean      697.3308   0.2584
3rdQ      704.0804   8.9887
Max       730.7926  42.1301
```

Figure 3



```
> liked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/lmatrix_f_divr60plus.csv", sep=",", row.names=1)
> disliked <- read.csv("C:/Users/s169096/Documents/Master JADS/Semester 2.1/SNA4DS/dmatrix_f_divr60plus.csv", sep=",",row.names=1)
> MoviesCor6 <- sna::netlm(y = disliked, x = liked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor6)

OLS Network Model

Residuals:
        0%        25%        50%        75%       100% 
-15.520526  -5.813840  -2.947172   3.146158  94.706161 

Coefficients:
            Estimate  Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 4.4271709 1       0       0        
x1          0.1733337 1       0       0        

Residual standard error: 8.834 on 2809633 degrees of freedom
Multiple R-squared: 0.02225     Adjusted R-squared: 0.02225 
F-statistic: 6.394e+04 on 1 and 2809633 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)       x1
Min        92.9408 -47.5340
1stQ      128.3067  -9.6590
Median    136.7303   0.1138
Mean      137.0539   0.5405
3rdQ      146.1234  11.0816
Max       178.1257  52.4758

> MoviesCor7 <- sna::netlm(y = liked, x = disliked, mode="graph", nullhyp = 'qapspp', reps = 1000)
> summary(MoviesCor7)

OLS Network Model

Residuals:
        0%        25%        50%        75%       100% 
-22.056929  -5.814575  -1.044379   4.597451  52.597451 

Coefficients:
            Estimate  Pr(<=b) Pr(>=b) Pr(>=|b|)
(intercept) 11.402549 1       0       0        
x1           0.128366 1       0       0        

Residual standard error: 7.602 on 2809633 degrees of freedom
Multiple R-squared: 0.02225     Adjusted R-squared: 0.02225 
F-statistic: 6.394e+04 on 1 and 2809633 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qapspp 
        Replications: 1000 
        Coefficient Distribution Summary:

       (intercept)         x1
Min     1132.55931  -43.77188
1stQ    1156.96529  -10.02972
Median  1163.95330    0.06931
Mean    1164.04128   -0.12889
3rdQ    1171.42697    9.49705
Max     1202.91648   44.32608
```

Figure 4

variation. This finally gave us the feeling that we are building somewhat meaningful models.

We even wanted to go a step further and decrease the filter threshold for the number of ratings that users have given to 40. But this would yield a dataset that was too large to analyse, as it costs 25+ hours to run an analysis on it.

Findings in relation to hypothesis.

When relating our findings to our aforementioned hypothesis, we see that it is possible to build models that somewhat can explain the variance in the network in a significant way. However, we can clearly see that more data is not always giving better explanations.

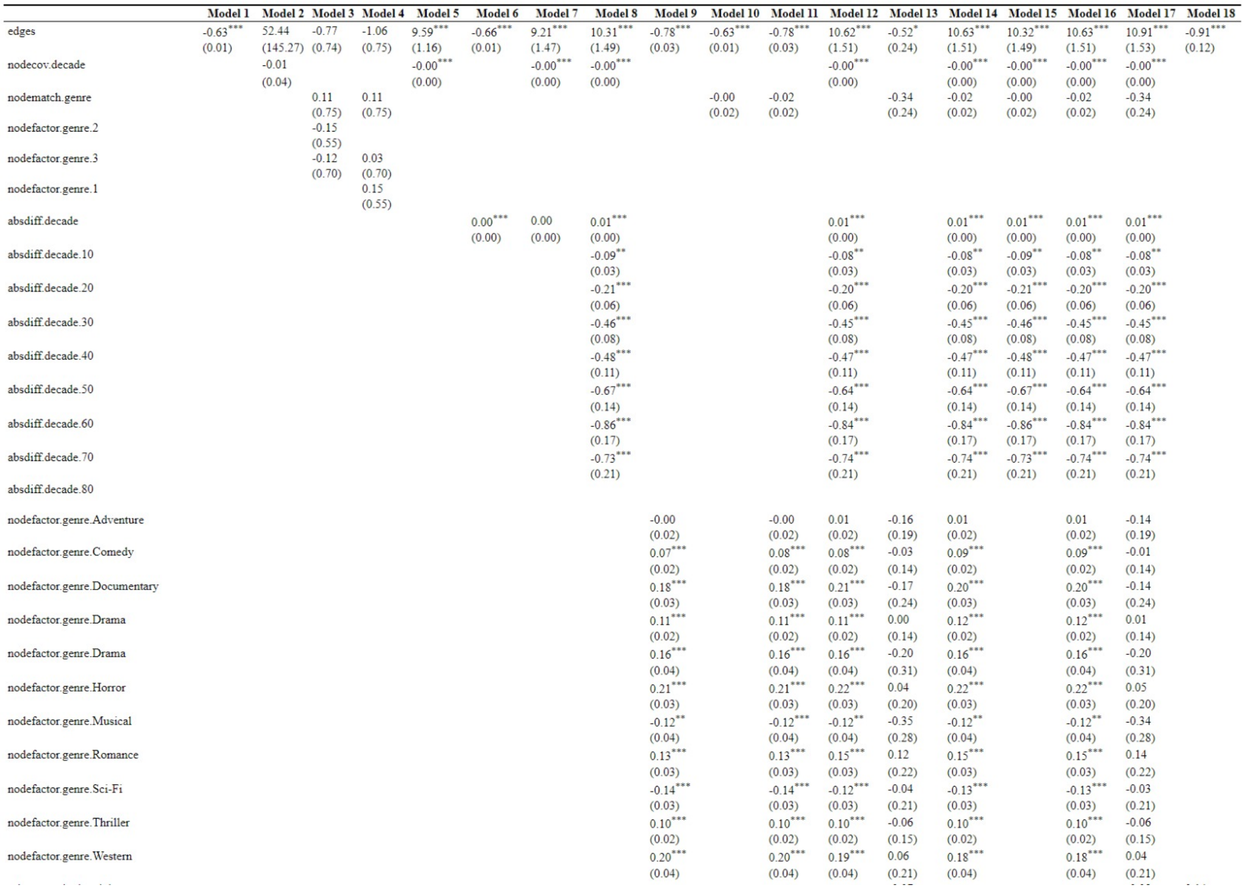
Also, it very interesting to see that liking movies says more about which movies you might dislike, than vice versa. An explanation for this might be that we saw in the data that there were more positive ratings than negative ratings, which could explain that disliking defines a more specific taste that is better detectable.

ERGM. In order to answer research question 1, an ERGM model has been used. The network created consists of movies as nodes. Two movies get an undirected edge if at least one user exists that has watched and liked both movies. An important assumption that is made is that a user liked a movie if the user has given a rating of at least four out of five. Next, in order to assess the influence of the release date, the decade of the release date is used. Furthermore, only the main genre of the movies have been used to assess the influence of two movies being watched by one user.

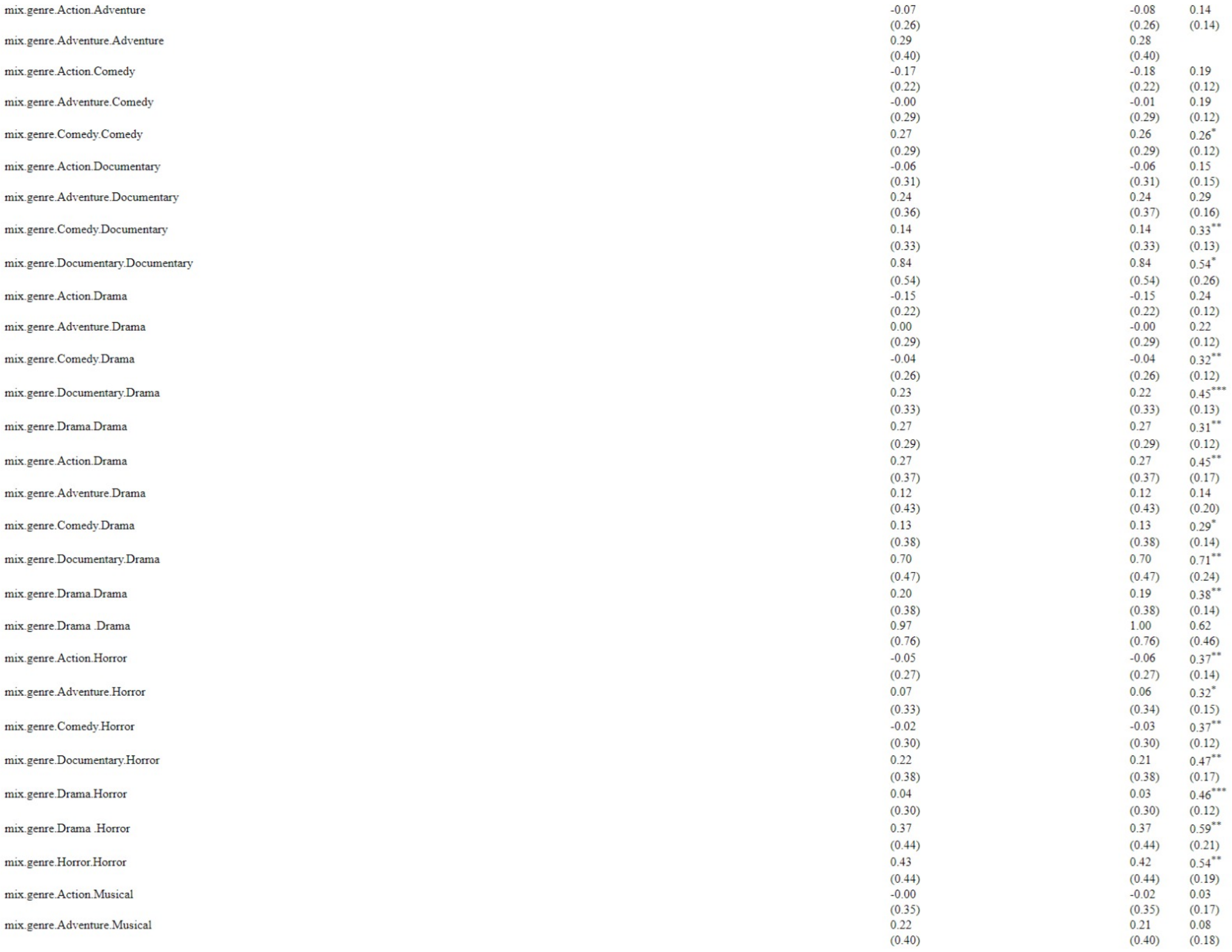
Homophily is hypothesized for genre and decade. Two movies with the same genre are more likely to have an edge and two movies from the same release decade are more likely to have an edge. Furthermore, the likeliness of an edge is even bigger when a movie has both the same genre and is from the same decade. To measure homophily the ERGM term Nodematch has been used.

ERGM Models.

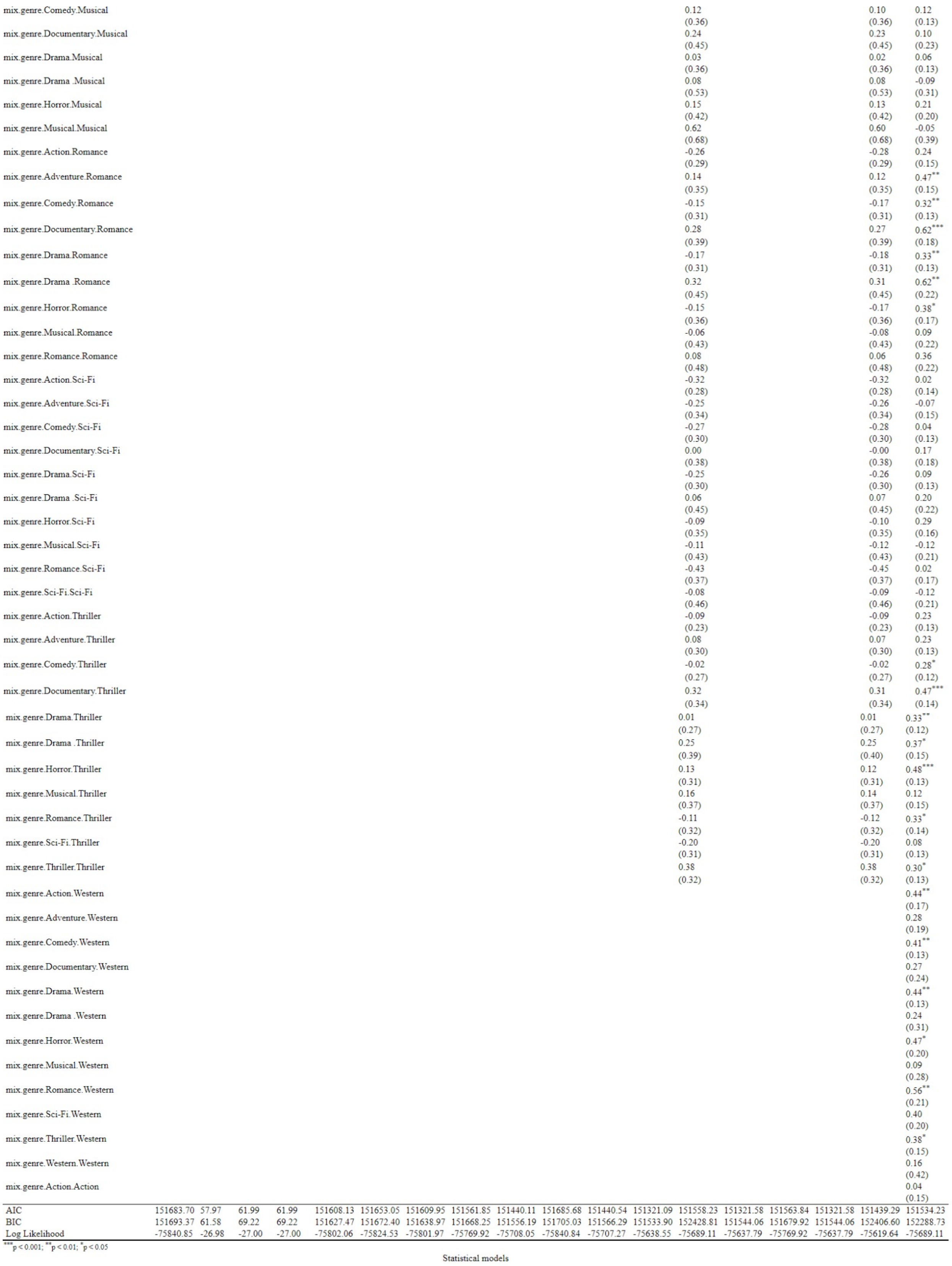
A total of 18 ERGM models have been constructed with 9 different ERGM terms. These terms are a selection of most popular nodal covariate terms. Such as Nodecov, Absdiff, Nodefactor, Nodematch and Nodemix. A mixture of ERGM terms were required due to the difference in categorical and continuous variables. I.e. decade year and genre. The results from all models can be seen in the figure below.



	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12	Model 13	Model 14	Model 15	Model 16	Model 17	Model 18
edges	-0.63*** (0.01)	52.44 (145.27)	-0.77 (0.74)	-1.06 (0.75)	9.59*** (1.16)	-0.66*** (0.01)	9.21*** (1.47)	10.31*** (1.49)	-0.78*** (0.03)	-0.63*** (0.01)	-0.78*** (0.03)	10.62*** (1.51)	-0.52* (0.24)	10.63*** (1.51)	10.32*** (1.49)	10.63*** (1.51)	10.91*** (1.53)	-0.91*** (0.12)
nodecov.decade		-0.01 (0.04)			-0.00*** (0.00)		-0.00*** (0.00)	-0.00*** (0.00)				-0.00*** (0.00)		-0.00*** (0.00)	-0.00*** (0.00)	-0.00*** (0.00)	-0.00*** (0.00)	
nodematch.genre			0.11 (0.75)	0.11 (0.75)						-0.00 (0.02)	-0.02 (0.02)		-0.34 (0.24)	-0.02 (0.02)	-0.00 (0.02)	-0.02 (0.02)	-0.34 (0.24)	
nodefactor.genre.2			-0.15 (0.55)															
nodefactor.genre.3			-0.12 (0.70)	0.03 (0.70)														
nodefactor.genre.1				0.15 (0.55)														
absdiff.decade						0.00*** (0.00)	0.00 (0.00)	0.01*** (0.00)				0.01*** (0.00)		0.01*** (0.00)	0.01*** (0.00)	0.01*** (0.00)	0.01*** (0.00)	
absdiff.decade.10								-0.09** (0.03)				-0.08** (0.03)		-0.08** (0.03)	-0.09** (0.03)	-0.08** (0.03)	-0.08** (0.03)	
absdiff.decade.20								-0.21*** (0.06)				-0.20*** (0.06)		-0.20*** (0.06)	-0.21*** (0.06)	-0.20*** (0.06)	-0.20*** (0.06)	
absdiff.decade.30								-0.46*** (0.08)				-0.45*** (0.08)		-0.45*** (0.08)	-0.46*** (0.08)	-0.45*** (0.08)	-0.45*** (0.08)	
absdiff.decade.40								-0.48*** (0.11)				-0.47*** (0.11)		-0.47*** (0.11)	-0.48*** (0.11)	-0.47*** (0.11)	-0.47*** (0.11)	
absdiff.decade.50								-0.67*** (0.14)				-0.64*** (0.14)		-0.64*** (0.14)	-0.67*** (0.14)	-0.64*** (0.14)	-0.64*** (0.14)	
absdiff.decade.60								-0.86*** (0.17)				-0.84*** (0.17)		-0.84*** (0.17)	-0.86*** (0.17)	-0.84*** (0.17)	-0.84*** (0.17)	
absdiff.decade.70								-0.73*** (0.21)				-0.74*** (0.21)		-0.74*** (0.21)	-0.73*** (0.21)	-0.74*** (0.21)	-0.74*** (0.21)	
absdiff.decade.80																		
nodefactor.genre.Adventure									-0.00 (0.02)		-0.00 (0.02)	0.01 (0.02)	-0.16 (0.19)	0.01 (0.02)		0.01 (0.02)	-0.14 (0.19)	
nodefactor.genre.Comedy									0.07*** (0.02)		0.08*** (0.02)	0.08*** (0.02)	-0.03 (0.14)	0.09*** (0.02)		0.09*** (0.02)	-0.01 (0.14)	
nodefactor.genre.Documentary									0.18*** (0.03)		0.18*** (0.03)	0.21*** (0.03)	-0.17 (0.24)	0.20*** (0.03)		0.20*** (0.03)	-0.14 (0.24)	
nodefactor.genre.Drama									0.11*** (0.02)		0.11*** (0.02)	0.11*** (0.02)	0.00 (0.14)	0.12*** (0.02)		0.12*** (0.02)	0.01 (0.14)	
nodefactor.genre.Drama									0.16*** (0.04)		0.16*** (0.04)	0.16*** (0.04)	-0.20 (0.31)	0.16*** (0.04)		0.16*** (0.04)	-0.20 (0.31)	
nodefactor.genre.Horror									0.21*** (0.03)		0.21*** (0.03)	0.22*** (0.03)	0.04 (0.20)	0.22*** (0.03)		0.22*** (0.03)	0.05 (0.20)	
nodefactor.genre.Musical									-0.12** (0.04)		-0.12*** (0.04)	-0.12** (0.04)	-0.35 (0.28)	-0.12** (0.04)		-0.12** (0.04)	-0.34 (0.28)	
nodefactor.genre.Romance									0.13*** (0.03)		0.13*** (0.03)	0.15*** (0.03)	0.12 (0.22)	0.15*** (0.03)		0.15*** (0.03)	0.14 (0.22)	
nodefactor.genre.Sci-Fi									-0.14*** (0.03)		-0.14*** (0.03)	-0.12*** (0.03)	-0.04 (0.21)	-0.13*** (0.03)		-0.13*** (0.03)	-0.03 (0.21)	
nodefactor.genre.Thriller									0.10*** (0.02)		0.10*** (0.02)	0.10*** (0.02)	-0.06 (0.15)	0.10*** (0.02)		0.10*** (0.02)	-0.06 (0.15)	
nodefactor.genre.Western									0.20*** (0.04)		0.20*** (0.04)	0.19*** (0.04)	0.06 (0.21)	0.18*** (0.04)		0.18*** (0.04)	0.04 (0.21)	



mix.genre.Action.Adventure	-0.07	-0.08	0.14
	(0.26)	(0.26)	(0.14)
mix.genre.Adventure.Adventure	0.29	0.28	
	(0.40)	(0.40)	
mix.genre.Action.Comedy	-0.17	-0.18	0.19
	(0.22)	(0.22)	(0.12)
mix.genre.Adventure.Comedy	-0.00	-0.01	0.19
	(0.29)	(0.29)	(0.12)
mix.genre.Comedy.Comedy	0.27	0.26	0.26*
	(0.29)	(0.29)	(0.12)
mix.genre.Action.Documentary	-0.06	-0.06	0.15
	(0.31)	(0.31)	(0.15)
mix.genre.Adventure.Documentary	0.24	0.24	0.29
	(0.36)	(0.37)	(0.16)
mix.genre.Comedy.Documentary	0.14	0.14	0.33**
	(0.33)	(0.33)	(0.13)
mix.genre.Documentary.Documentary	0.84	0.84	0.54*
	(0.54)	(0.54)	(0.26)
mix.genre.Action.Drama	-0.15	-0.15	0.24
	(0.22)	(0.22)	(0.12)
mix.genre.Adventure.Drama	0.00	-0.00	0.22
	(0.29)	(0.29)	(0.12)
mix.genre.Comedy.Drama	-0.04	-0.04	0.32**
	(0.26)	(0.26)	(0.12)
mix.genre.Documentary.Drama	0.23	0.22	0.45***
	(0.33)	(0.33)	(0.13)
mix.genre.Drama.Drama	0.27	0.27	0.31**
	(0.29)	(0.29)	(0.12)
mix.genre.Action.Drama	0.27	0.27	0.45**
	(0.37)	(0.37)	(0.17)
mix.genre.Adventure.Drama	0.12	0.12	0.14
	(0.43)	(0.43)	(0.20)
mix.genre.Comedy.Drama	0.13	0.13	0.29*
	(0.38)	(0.38)	(0.14)
mix.genre.Documentary.Drama	0.70	0.70	0.71**
	(0.47)	(0.47)	(0.24)
mix.genre.Drama.Drama	0.20	0.19	0.38**
	(0.38)	(0.38)	(0.14)
mix.genre.Drama .Drama	0.97	1.00	0.62
	(0.76)	(0.76)	(0.46)
mix.genre.Action.Horror	-0.05	-0.06	0.37**
	(0.27)	(0.27)	(0.14)
mix.genre.Adventure.Horror	0.07	0.06	0.32*
	(0.33)	(0.34)	(0.15)
mix.genre.Comedy.Horror	-0.02	-0.03	0.37**
	(0.30)	(0.30)	(0.12)
mix.genre.Documentary.Horror	0.22	0.21	0.47**
	(0.38)	(0.38)	(0.17)
mix.genre.Drama.Horror	0.04	0.03	0.46***
	(0.30)	(0.30)	(0.12)
mix.genre.Drama .Horror	0.37	0.37	0.59**
	(0.44)	(0.44)	(0.21)
mix.genre.Horror.Horror	0.43	0.42	0.54**
	(0.44)	(0.44)	(0.19)
mix.genre.Action.Musical	-0.00	-0.02	0.03
	(0.35)	(0.35)	(0.17)
mix.genre.Adventure.Musical	0.22	0.21	0.08
	(0.40)	(0.40)	(0.18)



mix.genre.Comedy.Musical													0.12 (0.36)				0.10 (0.36)	0.12 (0.13)
mix.genre.Documentary.Musical													0.24 (0.45)				0.23 (0.45)	0.10 (0.23)
mix.genre.Drama.Musical													0.03 (0.36)				0.02 (0.36)	0.06 (0.13)
mix.genre.Drama .Musical													0.08 (0.53)				0.08 (0.53)	-0.09 (0.31)
mix.genre.Horror.Musical													0.15 (0.42)				0.13 (0.42)	0.21 (0.20)
mix.genre.Musical.Musical													0.62 (0.68)				0.60 (0.68)	-0.05 (0.39)
mix.genre.Action.Romance													-0.26 (0.29)				-0.28 (0.29)	0.24 (0.15)
mix.genre.Adventure.Romance													0.14 (0.35)				0.12 (0.35)	0.47** (0.15)
mix.genre.Comedy.Romance													-0.15 (0.31)				-0.17 (0.31)	0.32** (0.13)
mix.genre.Documentary.Romance													0.28 (0.39)				0.27 (0.39)	0.62*** (0.18)
mix.genre.Drama.Romance													-0.17 (0.31)				-0.18 (0.31)	0.33** (0.13)
mix.genre.Drama .Romance													0.32 (0.45)				0.31 (0.45)	0.62** (0.22)
mix.genre.Horror.Romance													-0.15 (0.36)				-0.17 (0.36)	0.38* (0.17)
mix.genre.Musical.Romance													-0.06 (0.43)				-0.08 (0.43)	0.09 (0.22)
mix.genre.Romance.Romance													0.08 (0.48)				0.06 (0.48)	0.36 (0.22)
mix.genre.Action.Sci-Fi													-0.32 (0.28)				-0.32 (0.28)	0.02 (0.14)
mix.genre.Adventure.Sci-Fi													-0.25 (0.34)				-0.26 (0.34)	-0.07 (0.15)
mix.genre.Comedy.Sci-Fi													-0.27 (0.30)				-0.28 (0.30)	0.04 (0.13)
mix.genre.Documentary.Sci-Fi													0.00 (0.38)				-0.00 (0.38)	0.17 (0.18)
mix.genre.Drama.Sci-Fi													-0.25 (0.30)				-0.26 (0.30)	0.09 (0.13)
mix.genre.Drama .Sci-Fi													0.06 (0.45)				0.07 (0.45)	0.20 (0.22)
mix.genre.Horror.Sci-Fi													-0.09 (0.35)				-0.10 (0.35)	0.29 (0.16)
mix.genre.Musical.Sci-Fi													-0.11 (0.43)				-0.12 (0.43)	-0.12 (0.21)
mix.genre.Romance.Sci-Fi													-0.43 (0.37)				-0.45 (0.37)	0.02 (0.17)
mix.genre.Sci-Fi.Sci-Fi													-0.08 (0.46)				-0.09 (0.46)	-0.12 (0.21)
mix.genre.Action.Thriller													-0.09 (0.23)				-0.09 (0.23)	0.23 (0.13)
mix.genre.Adventure.Thriller													0.08 (0.30)				0.07 (0.30)	0.23 (0.13)
mix.genre.Comedy.Thriller													-0.02 (0.27)				-0.02 (0.27)	0.28* (0.12)
mix.genre.Documentary.Thriller													0.32 (0.34)				0.31 (0.34)	0.47*** (0.14)
mix.genre.Drama.Thriller													0.01 (0.27)				0.01 (0.27)	0.33** (0.12)
mix.genre.Drama .Thriller													0.25 (0.39)				0.25 (0.40)	0.37* (0.15)
mix.genre.Horror.Thriller													0.13 (0.31)				0.12 (0.31)	0.48*** (0.13)
mix.genre.Musical.Thriller													0.16 (0.37)				0.14 (0.37)	0.12 (0.15)
mix.genre.Romance.Thriller													-0.11 (0.32)				-0.12 (0.32)	0.33* (0.14)
mix.genre.Sci-Fi.Thriller													-0.20 (0.31)				-0.20 (0.31)	0.08 (0.13)
mix.genre.Thriller.Thriller													0.38 (0.32)				0.38 (0.32)	0.30* (0.13)
mix.genre.Action.Western																		0.44** (0.17)
mix.genre.Adventure.Western																		0.28 (0.19)
mix.genre.Comedy.Western																		0.41** (0.13)
mix.genre.Documentary.Western																		0.27 (0.24)
mix.genre.Drama.Western																		0.44** (0.13)
mix.genre.Drama .Western																		0.24 (0.31)
mix.genre.Horror.Western																		0.47* (0.20)
mix.genre.Musical.Western																		0.09 (0.28)
mix.genre.Romance.Western																		0.56** (0.21)
mix.genre.Sci-Fi.Western																		0.40 (0.20)
mix.genre.Thriller.Western																		0.38* (0.15)
mix.genre.Western.Western																		0.16 (0.42)
mix.genre.Action.Action																		0.04 (0.15)
AIC	151683.70	57.97	61.99	61.99	151608.13	151653.05	151609.95	151561.85	151440.11	151685.68	151440.54	151321.09	151558.23	151321.58	151563.84	151321.58	151439.29	151534.23
BIC	151693.37	61.58	69.22	69.22	151627.47	151672.40	151638.97	151668.25	151556.19	151705.03	151566.29	151533.90	152428.81	151544.06	151679.92	151544.06	152406.60	152288.73
Log Likelihood	-75840.85	-26.98	-27.00	-27.00	-75802.06	-75824.53	-75801.97	-75769.92	-75708.05	-75840.84	-75707.27	-75638.55	-75689.11	-75637.79	-75769.92	-75637.79	-75619.64	-75689.11

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

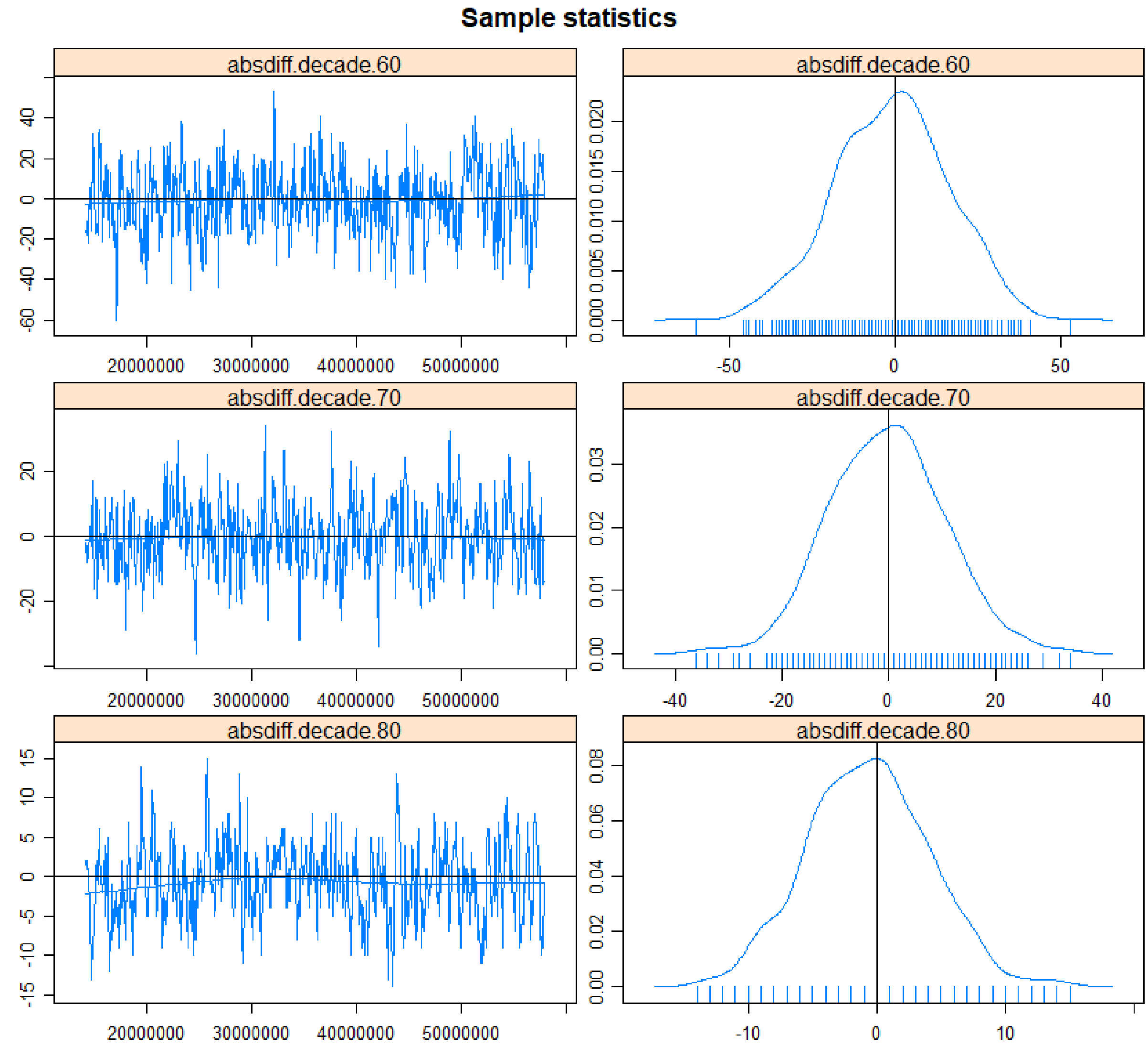
Statistical models

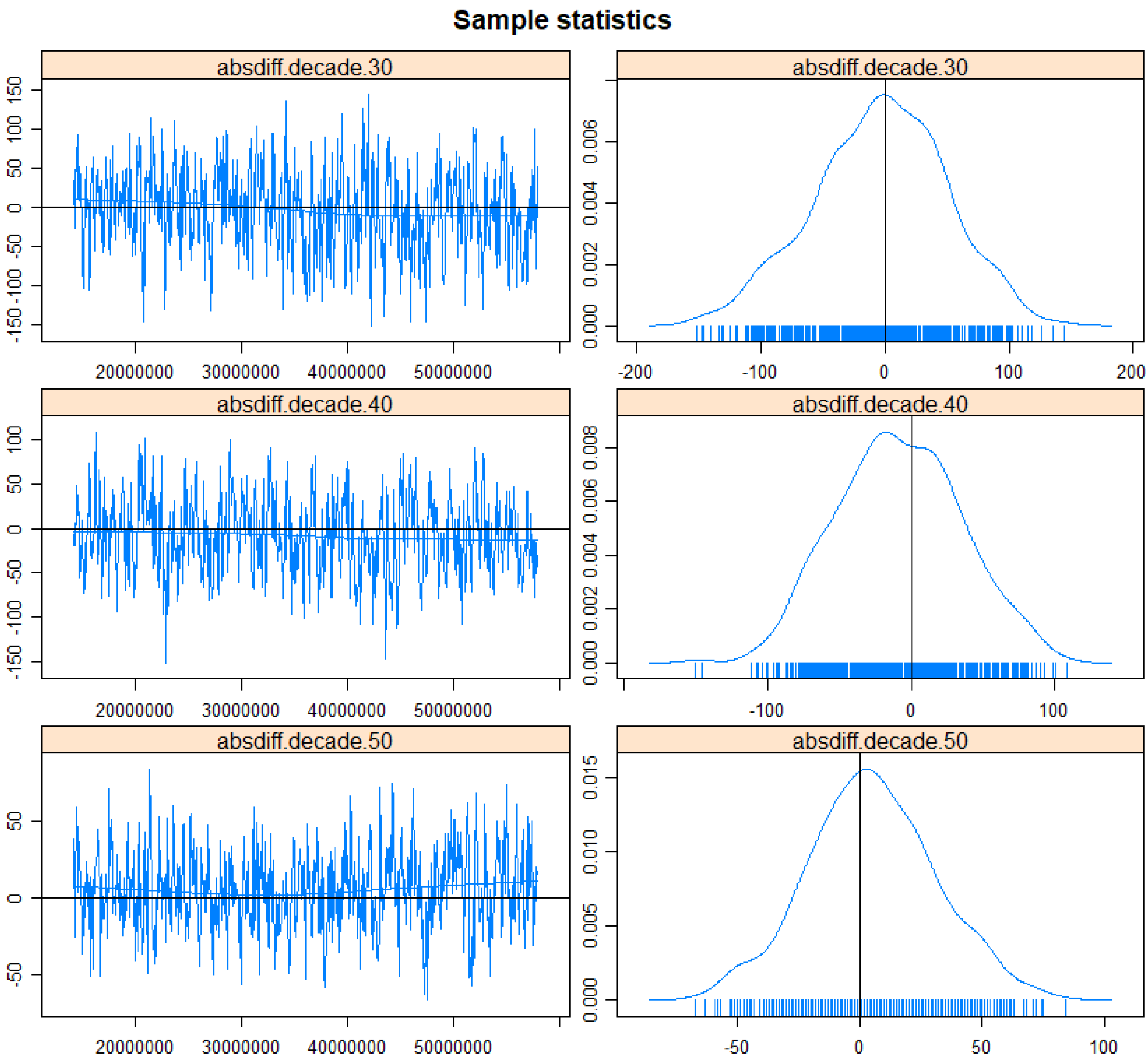
This model output can be found in directory “root/ergm_result/models.html.”

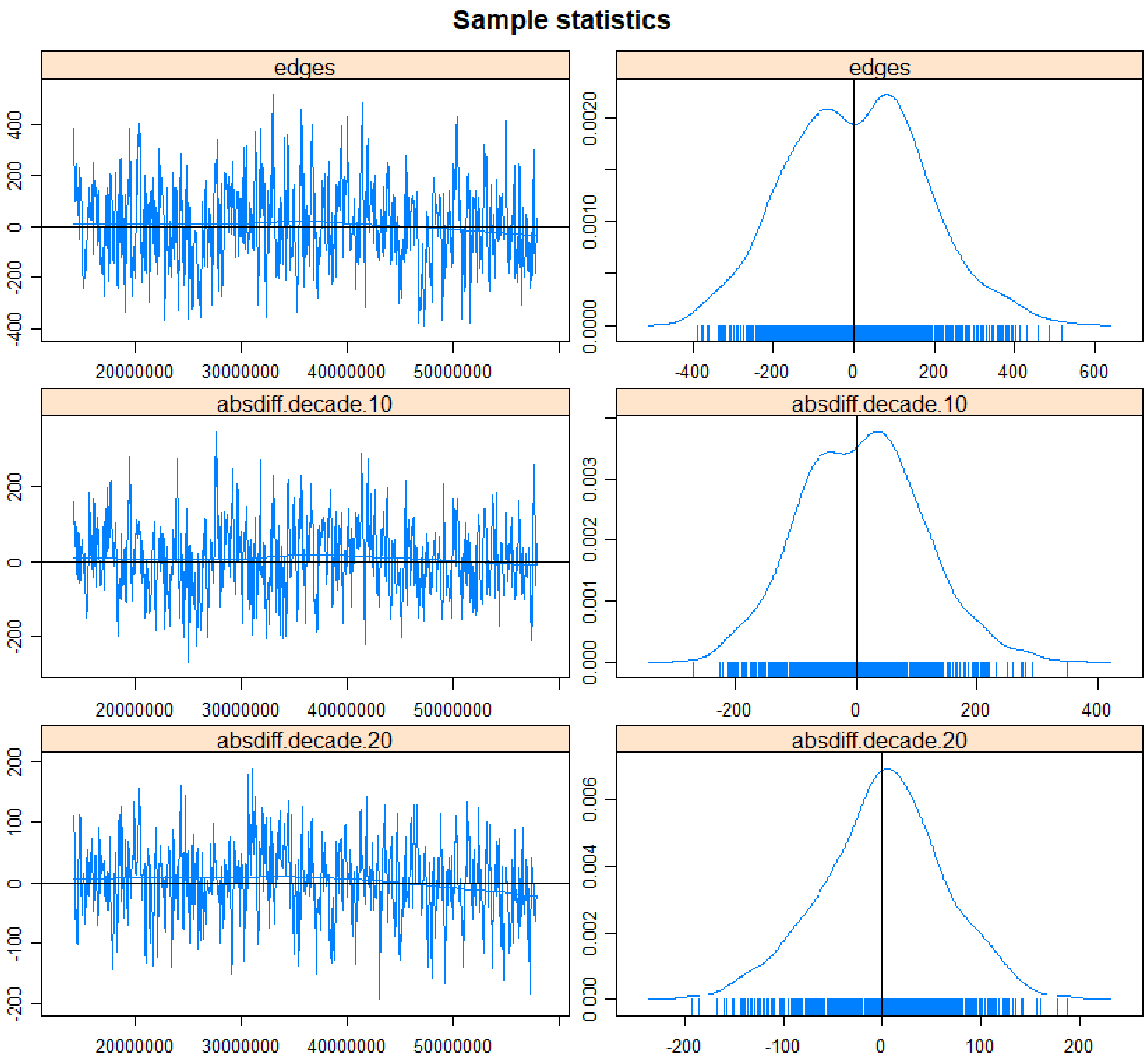
In order to find the best performing model, two model selection criteras have been used. The Akaike's Information Criteria (AIC) and Bayesian Information Criteria (BIC). The AIC tries to measure high dimensional reality, while BIC selects model among a class of parametric models with different numbers of parameters (mainly in the domain of underfitting). As can be seen from the figure, model 2, 3 and 4 yield the lowest AIC and BIC scores.

The best performing was then used as input for the MCMC diagnostics and ERGM Goodness Of Fit.

Estimate parameters using Markov Chains Monte Carlo simulations. A Markov chain is a sequence of random variables that depends upon the value taken by the previous variable. This method then generates a series of graphs that differs from each other by only one edge. The sequence of generated graphs, i.e. plots of Markov Chains Monte Carlo diagnostics are seen below.


Sample statistics
absdiff.decade.60
40
20
0
-20
-40
-60
20000000
30000000
40000000
50000000
absdiff.decade.60
0.000 0.005 0.010 0.015 0.020
-50
0
50
absdiff.decade.70
20
0
-20
20000000
30000000
40000000
50000000
absdiff.decade.70
0.00
0.01
0.02
0.03
-40
-20
0
20
40
absdiff.decade.80
15
10
5
0
-5
-10
-15
20000000
30000000
40000000
50000000
absdiff.decade.80
0.00
0.02
0.04
0.06
0.08
-10
0
10


Sample statistics
absdiff.decade.30
absdiff.decade.30
150
100
50
0
-50
-100
-150
20000000
30000000
40000000
50000000
0.000
0.002
0.004
0.006
-200
-100
0
100
200
absdiff.decade.40
absdiff.decade.40
100
50
0
-50
-100
-150
0.000
0.002
0.004
0.006
0.008
-100
0
100
absdiff.decade.50
absdiff.decade.50
50
0
-50
0.000
0.005
0.010
0.015
-50
0
50
100



Goodness of Fit.

Goodness of fit calculates p-values for geodesic distance, degree, and reachability summaries to diagnose the goodness-of-fit of ERGM models.

The output is composed of 4 Parts:

1. Goodness-of-fit for degree
2. Goodness-of-fit for edgewise shared partner
3. Goodness-of-fit for minimum geodesic distance

4. Goodness-of-fit for model statistics

As can be seen from the result all terms used within the model follow a moderately normal distribution centered around a mean of zero. The trace plots contain comparable results. These plots show that the distribution is well calibrated as the MCMC algorithm runs.

Odds Ratio & Probabilities.

An increase of one number of edges, decreased the log odds with -0.678. And it makes the odds of the effect 0.508 times larger. Odds ratio is below the 1 threshold, which means that there is a lower offs of forming edges in this network.

Additionally, for each decade year (10,20,30,40,50,60,70 and 80) the log odds increases with approximately 1.137. The odds of the effect increases accordingly to above the 1 threshold. Therefore it is concluded that having a movie made in a later year has a greater odds of multiple users rating the movie.

The best performing model has a 3.3% prob. of forming edges. However, decade 70 has 59% of forming an edge and decade 80, 78%. Which is consistent with the OR result.

Conclusion

QAP. For the QAP model on the Netflix dataset the following research question was answered, namely “How much does liking the same movies influence disliking the same movies, and vice versa?” For the second model, the users were the main focus instead of the movies.

Netflix could have benefited from the insights derived by the research for their recommender systems back then. Even though old data is used, this could be replicated for more recent data. The insights gained were that networks on more enthusiastic reviewers do not give a clearer picture of their movie's taste. In addition, liking movies says more about which movies are possibly disliked, than vice versa.

ERGM. It appeared that both genre and decade affected the existence of an edge between nodes, hence confirming genre and decade as predictors for movies being liked by an individual. Furthermore, it became clear that adding terms to an ERGM can significantly improve results. By combining ERGM terms like Nodefactor, Nodematch, Nodecov, Absdiff and Nodemix, the best result was obtained. Therefore, the hypothesis was confirmed. The results are valuable for movie streaming services. By confirming that users that liked a movie, will be likely to also like movies from the same decade or genre, recommender systems can be improved. The results provide new insights in users' taste of movies, which can be used by recommender systems to give better movie recommendations. For example, recommender systems could give recommendation towards movies, equal in decade or genre of previously liked movies.

What does remain an open problem? What remains an open problem would be to combine the movie genres in the user networks to give extra insights, connected to the first research question. Also, is that we assume that a frequent reviewer gives a rating to a movie after watching it. However, we don't have information on (re-)viewers that watch a movie but don't give a rating afterwards. This is difficult to overcome as we simply don't have data available on this matter.

Future work. For future research it could be beneficial to dive more into the literature and find out more about what is already known in order to give a stronger reasoning behind the explanatory variables used in the analysis. Also, it would be interesting to look into more recent data as streaming services are widely used nowadays and the variation of users is maybe larger now.

References

Amershi, S., Cakmak, M., Knox, W. B., & Kulesza, T. (2014). Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, *35*(4), 105–120.

Bell, R. M., & Koren, Y. (2007). Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, *9*(2), 75–79.

Guillory, A., & Bilmes, J. A. (2011). Simultaneous learning and covering with adversarial noise. In *ICML.*

Narayanan, A., & Shmatikov, V. (2006). How to break anonymity of the netflix prize dataset. *arXiv Preprint Cs/0610105.*

Netflix. (2019). Netflix prize data. *Kaggle.* Retrieved from https://www.kaggle.com/netflix-inc/netflix-prize-data

Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2008). Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM conference on recommender systems* (pp. 267–274).