Computational Techniques in Latent Variable Network Models

Kelly

November 30, 2018

1 Introduction

Game of Thrones is a popular HBO TV series adapted from George R.R. Martin's best-selling book series *'A Song of Ice and Fire'*. The medieval fantasy epic describes the stories of two powerful families - kings and queens, knights and renegades, liars and honest men - playing a deadly game for control of the Seven Kingdoms of Westeros, and to sit atop the Iron Throne. Conspiracy and deception, power and exile, blood and tear ran through the plot, sewing together characters with various backgrounds: from royals to peasants, from icy zombies to dragons. As the heart tucking plot develops with each book release, the readers start to wonder where the storyline would lead towards: some may argue by the set up of the theme where the battle of ice and fire shall never die out; others proposed a different view on the analysis of the plot – using the data extracted from the book to investigate the relations between characters.

After discovering some data on the verses exchanged between the characters from the third book, we start to wonder if we can extract information out of it – in particular, making inference to identify the character being "good" or "bad". With the questions in mind, we found the paper written by Krivitsky, Handcock, Raftery, and Hoff on *Representing Clustering in Social Networks with Latent Cluster Random Effects Models*. The relationships between characters from the book naturally arises as a network, each person being a node, and the lines they exchange being the vertex. (We probably want to cooperate Kelly's writing about section 2 here). Once we have established a network based on the characters from the novel, we fit it with a latent network model proposed by Hoff in his paper in Section 4, and analyse the data by using two different numerical schemes: one is through Expectation- Maximization where we treat the of the characters as latent; another is through Markov-Chain-Monte-Carlo where we blah blah blah (I am not perfectly sure what to fill in here. Help plz!)

(Should we say more about... comparing the results from the previous two methods?)

2 Data

Due to its global fame, *Game of Thrones* has been studied in many different contexts, especially in network analysis. Therefore, there are many readily available datasets. In our project, we use the dataset from Beveridge and Shan's [**beveridge2016network**] article which contains information about characters' interactions in the third book of the series. In this case, an *interaction* occurs if the characters' names appear within fifteen words of one another. This could mean that the characters interacted with each other, conversed with each other, or they were mentioned together by another ends. There is also a column that contains the number of times each pair interacts with one another. Using this dataset, we constructed a weighted network using the number of interactions as weights. Here, the nodes represent the characters and the edges represent the interactions. We use an adjacency matrix, A, to represent the network, where the $a_{i,j}$ element represents the number of times the characters interacted with each other. Note that this means if $a_{i.j} = 0$, there are no recorded interactions between character i and j, based on how *interactions* is defined. Although the original dataset is intended as a directed network, we treat it as an undirected network in order to simplify our models.

Our network G contains $N_V(G) = 107$ nodes and $N_E(G) = 352$ edges which means it is quite sparse since it only contains approximately 6.20% of 5,671 ($\binom{N_V(G)}{2}$) possible edges. Figure –include figure– shows the network described. In order to account for the sparsity of our network, we consider a subnetwork which only contains pairs of characters with at least 75 interactions (maybe even 100?). We chose a cutoff of 75 interactions because we want to focus our analysis on only the main characters. Looking at the distribution of the weighted degree (include pic?), we see that 65.42% of the characters had fewer than 75 interactions. Therefore, it makes sense to use this cutoff to limit our analysis to only the main characters. By doing so, our new network G' contains $N_V(G') = 35$ nodes and $N_E(G') = 140$ edges. Here, we see that the network now contains 23.53% of 595 possible edges which shows an improvement from before. –should we mention that this is not great but we cant limit it too much?– Our analysis that follows will be done on this subnetwork G'. Figure –include figure– shows this subnetwork G', and indeed, only the main characters are still represented in the network.

3 Models

3.1 Latent Network Models

1. Introduction of network data
2. Community detection in networks
3. Latent variable models
4. Inclusion of observed covariates

3.2 Model Formulation (in our context)

$$logitP(Y_ij = 1) = ...$$

1. latent variables

4 Results

1. Analysis
2. Results

5 Conclusion

6 References

Appendix

A Estimation Maximization Code

B Markov Chain Monte Carlo Code

C Figures Code