# A Network Perspective on LDA

Cindy Cook
Network Analysis Final Project
5 May 2016

## 1 Introduction

With an increase in the amount of digitized text, topic modeling has become an important area of research. One of the most popular methods for topic modeling is Latent Dirichlet Allocation (LDA) first described in [4]. LDA, a probabilistic hierarchical generative model, uses the Bayesian framework to discover hidden structure, or *topics*, within a set of documents referred to as a corpus. Given a document, the distribution of latent topics is found through estimation of the posterior distribution. However, this is very challenging and computationally demanding since the posterior distribution of the latent variables given a document is intractable. A heavy development of fast and accurate approximation methods applied to LDA have been developed providing a user with several estimation options. A few of these options include collapsed Gibbs sampling (CGS) [8], collapsed variational Bayesian inference [16], maximum likelihood estimation [10], and maximum a posteriori estimation [7]. In [4], a variational EM (VEM) algorithm is applied to solve the problem. Although the development of these methods have allowed researchers to obtain topics, a major problem lies in the fact that using different methods can often lead to conflicting results.

There have been several studies published comparing estimation procedures for LDA including [13], [2], [16], and [15]. Most comparison studies focus on a single corpus and evaluate different estimation methods using precision/recall statistics. In general, these papers offer good advice, but do not address the main problem of identifying inaccurate results. More recently, deeper theoretical problems with LDA have been highlighted by Lancichinetti in [11]. The paper "High-Reproducibility and High-Accuracy Method for Automated Topic Classification" shows that the likelihood of the true generating model can be smaller then the likelihood of alternative models in the LDA framework [11]. After demonstrating their theoretical concerns, the authors tweak the LDA algorithm introducing a process they call TopicMapping, which leads to more robust results. TopicMapping involves five stages. The first is to clean the corpus, which involves stemming and removing of stop words. The second stage is to project the term frequency matrix onto the word space. In the third stage, edges are filtered according to a Poisson distribution. The fourth stage applies InfoMap clustering to obtain starting values for each topic distribution over the words. Finally, LDA is run with these initial starting values in the fifth step. This process is meant to decrease the size of the parameter space, which aids in the estimation procedure. TopicMapping is tested on both simulated and real world corpora. Although TopicMapping appears to solve the basic problems involved when applying LDA, I will show that it fails to produce both accurate and reproducible results across all types of corpora.

In step two of TopicMapping, the author's treat the term frequency matrix as a bipartite network and use the projected one mode network to inform the LDA algorithm. It is within this step that TopicMapping is making improvements over basic LDA. Hence it warrants further study into the creation and resulting structure of these networks. In all of the simulations used in [11], no topic shares any words, which immediately creates a perfectly clustered network. This set-up allows LDA to begin at the correct parameter values. It is not too surprising that, "we find that if the topics are not too heterogeneously distributed, the asymmetric LDA algorithm converges after only a few iterations" (pg.6, [11]). The two real world corpora are such that the words found in documents from one topic are not likely to appear in documents of another topic, again leading to clusters that are mostly correct. However, most corpora do not follow this basic pattern. The likelihood of topics sharing words is higher in general then what was considered in [**?**]. The network itself becomes an important part of the TopicMapping algorithm and should be studied further from a network perspective.

I hypothesis that basic differences in the modularity of the projected word network of a corpus determines whether or not TopicMapping will outperform LDA. In particular, I hypothesis that there are distinctive underlying features of each corpus' projected word network, and each network will be partially explained by homophily of the InfoMap clusters. The second section of this paper goes through a replication study done to ensure that the results of TopicMapping, published in [11], are accurate. The replication is also performed to make sure that I have the proper code and algorithms in place to apply their methods to other datasets. The third section introduces a new data set and explains how the networks used in this paper were obtained. The fourth section applies TopicMapping to the new data and obtains neither accurate nor reproducible results. It then compares both the new and original corpora using descriptive statistics. The fifth section uses ERGMs to estimate the homophily within the two networks. The sixth and final section concludes the paper with a discussion of the results.

# 2 Replicate the Study

The paper, "High-Reproducibility and High-Accuracy Method for Automated Topic Classification," consists of four main parts [11]. The first points out a major limitation to the LDA algorithm developed by Blei as a topic model [4]. The second describes a new method called TopicMapping developed to overcome theoretical drawbacks of LDA. The third consists of comparing the accuracy and reproducibility of TopicMapping as compared to LDA and pLSA [3] on synthetic data. The final aspect of the paper, applies TopicMapping to two real world datasets. I have been able to successfully reproduce the accuracy and reproducibility results for the comparisons between TopicMapping and randomly seeded LDA in both the simulated studies and one of the real world applications. The results of my replication can be found in Appendix A. All data and scripts can be found on github: here.

# 3 The Data

There are two main 'real world' datasets used to test the TopicMapping algorithm: WoS and Wikipedia. The WoS corpus contains $28,838$ documents from the web of science, where each document contains the title and abstract of a paper published in one of six top journals from different disciplines. After preprocessing the data, there are $106,143$ unique words and over $8.7 \times 10^6$ edges. The Wikipedia dataset contains over $4 \times 10^6$ nodes with over $800 \times 10^6$ edges. Due to the size of the Wikipedia data set, I was unable to receive this information from the authors of [11]. Therefore, I focused my replication and extension on the WoS dataset only. Although the WoS corpus was easily sent to me, it is still considered 'large.' In R, the adjacency matrix is too large to be stored.

Due to the size of the WoS dataset, I randomly selected 10 documents in each subject to create a working dataset for a total of 60 documents. This dataset is referred to as the Science corpus. The second corpus I will be considering was obtained from Indeed.com. Each document contains the past work experience from resumes within five different fields: teaching, graphic design, architecture, nursing, and accounting. The corpus contains 50 documents. I will refer to this corpus as the Indeed corpus. The chosen resumes were selected by first using Indeed.com's location filter to set it at State College, PA. The top ten resumes to appear in the search results were then selected.

## 3.1 Creating the Networks

After obtaining documents for both corpora, a network of words is created in R by first using the *tm* package to obtain the term frequency matrix. Then a stemmer is applied, and basic English stop words are removed. The stemmer and stop words applied are the same as used in [11]. The *bipartite* package is then used to project this term frequency (i.e. the adjacency matrix for a two-mode network) onto a one-mode network of words weighted by the number of times a pair of words appears together across all documents. Lastly, the filtering process laid out in [12] is applied.

| | Science | | Indeed | |
|---|---|---|---|---|
| | Nodes | Edges | Nodes | Edges |
| original corpus | 2189 | | 1717 | |
| cleaned corpus | 1995 | | 1092 | |
| projected network | 1995 | 121440 | 1092 | 50534 |
| filtered network | 1995 | 121039 | 1092 | 50186 |

Table 1: Shows the size of each network through the pre-processing steps as described above.

Plotting the networks using the Fruchterman-Reingold layout in NodeXL we obtain the following images:

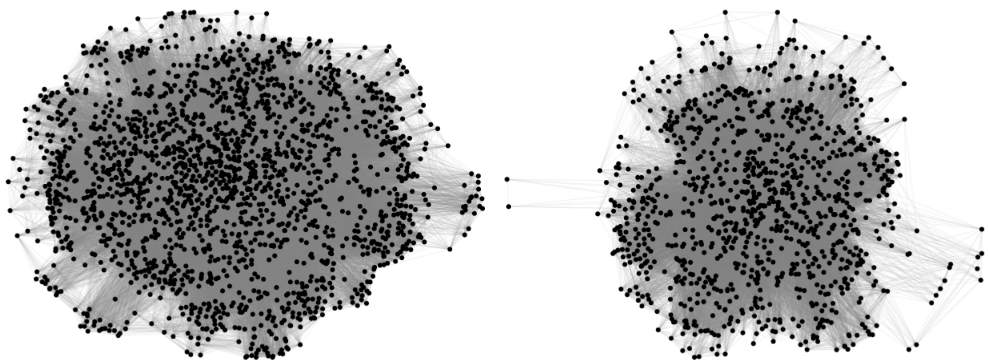


Figure 1: Science word network appears on the left; Indeed word network appears on the right

Not too much can be stated by the visual graphs alone.

# 4 Comparison

Using the code provided to me, I ran TopicMapping on both Science and Indeed 300 times: 100 times with $K = 3$, 100 times with $K = 5$ or 6 for Indeed and Science, respectively, and 100 times with $K = 10$. I also ran the LDA function in R with the VEM method for estimation and different random starting seeds on both Science and Indeed 300 times: 100 times with $K = 3$, 100 times with $K = 5$ or 6, and 100 times with $K = 10$. Note that for Science and Indeed the correct number of topics is 5 or 6, respectively. To compare results, I calculate the accuracy and reproducibility as defined in [12], and obtain the following results:

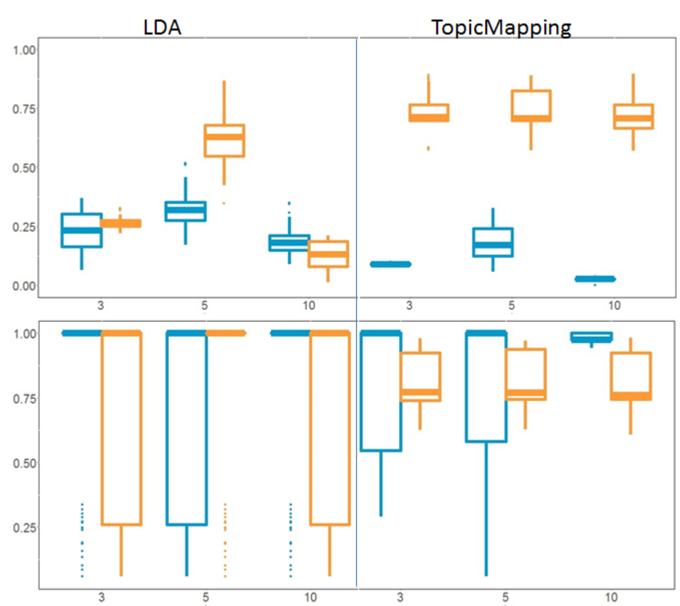



Figure 2: Plots on the top show boxplots of the accuracy for 100 runs on Science(orange) and Indeed (blue). Plots on the bottom show the reproducibility for the 100 runs.

We can see that TopicMapping performed with approximately 75% accuracy and 80% reproducibility for the Science corpus. However, the accuracy is around 10% and the reproducibility is extremely variable for the Indeed corpus. We should also notice that TopicMapping outperforms LDA only on the Science corpus, but fails to produce better results for the Indeed corpus. I hypothesize that TopicMapping is highly dependent on the clustering done through InfoMap. If InfoMap does not produce accurate results, it can lead the estimation procedure of LDA into the wrong part of the parameter space resulting in completely inaccurate results. The major difference between the two corpora lies in the fact that words used in titles and abstracts of journal articles do not readily appear across different disciplines. Whereas words used in resumes to describe past work experience cross fields more easily. In fact, we can see from the following plot, the proportion of words from our two cleaned corpus's that cross these boundaries:

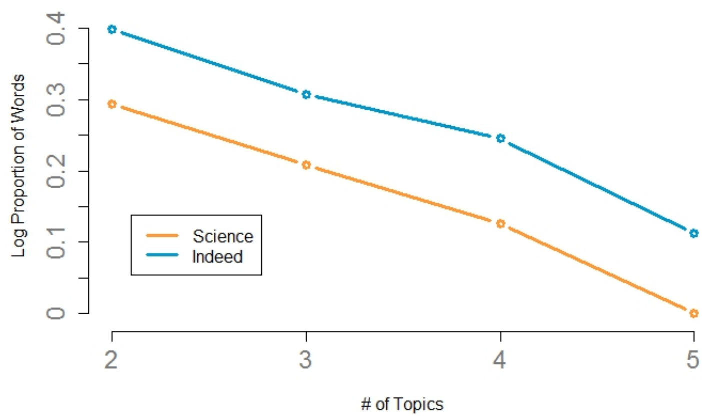
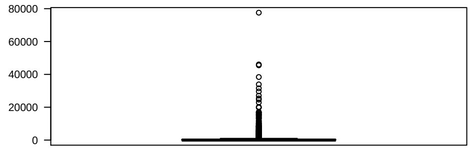



Figure 3: The proportion of unique words in each corpus that belong to at least 1, 2, 3, 4, and 5 topics for both the Science (orange) and Indeed (blue) are shown above.

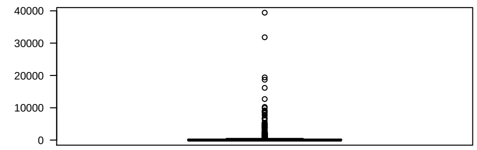
We can clearly see that close to zero percent of the vocabulary, or unique words, in the Science corpus is shared across all six topics; whereas, close to 20% of the vocabulary in the Indeed corpus is shared across all job categories.

## 4.1 Descriptive Statistics

To begin analysis on these networks, I use the *igraph* package in R to measure the betweenness centrality for each node, which will allow me to analyze structural holes and bridges. Words that are highly used within each document and common across all documents will have high betweenness scores. From Figure 3. (B), I assume that words in the Indeed corpus will have smaller variation across betweenness scores as compared to the scores for words in the Science corpus. Plotting the box-plots, we see:




(a) Science




(b) Indeed

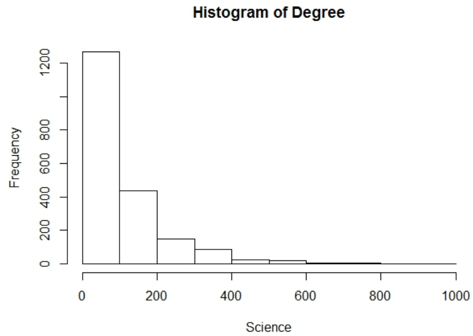
Figure 4: Boxplots of the betweenness centrality scores for each corpus.

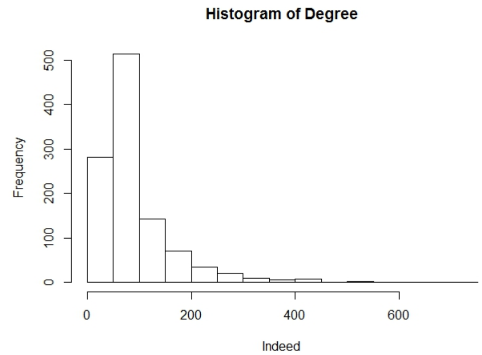
Words with the highest betweenness scores for Science are as follows: result (123287.95), use (67756.22), model (67159.62), data (60749.09), suggest (56224.20), and studi (55947.48). Words with the highest betweenness scores for Indeed are as follows: present (39422.6), pa (31815.2), respons (19401.64), includ (18680.75), work (16158.91), and design (12703.92). As predicted, these words are used often in each document no matter what discipline or field the document is in. This fact would be obvious for any projected word network. There is only a slight difference in the distribution of betweenness scores. We find that for the Science corpus only 680 words in the network, or about (34%), have a betweenness centrality greater than one, while 1829, or about (92%), have a betweenness score greater than 0. For the Indeed corpus. For the Indeed corpus, 404 words, or about (37%) of the words, have a betweenness score greater then 1, while 1061, or about (98%) of the words, have a betweenness score greater than 0. The variation of these scores is 4598699 and 13205460 for Indeed and Science, respectively. I would expect this difference to be more pronounced given the TopicMapping results.

Next, I will find the transitivity score for each network. Transitivity can be used as a clustering coefficient for networks. Thus, I hypothesis that this value is positive for both networks, but higher for the Science corpus than the Indeed corpus. However, it turns out that this measure it similar for both corpora: 0.4181239 for the Science corpus and 0.4279717 for the Indeed corpus. Taking a closer look, we see the following triad censuses:

| | 003 | 102 | 201 | 300 |
|---|---|---|---|---|
| Science | 83.40% | 15.21% | 1.12% | 0.27% |
| Indeed | 77.78% | 19.67% | 2.04% | 0.51% |

Since both networks are undirected, there are only four possible triads, and the distribution of these is very similar in both Science and Indeed. In fact, it turns out that several descriptive statistics are very similar for both networks including density $(0.06085374, 0.08424908)$, degree assortitivity, $(-0.06119152, -0.12577)$, and even their degree distributions:




as well as their modularity as applied to several different clustering algorithms all performed in R using the *igraph* package:

| | Modularity | |
|---|---|---|
| Clustering Type | Science | Indeed |
| InfoMap | 0.3884 | 0.3881 |
| Fast-Greedy | 0.3398 | 0.3358 |
| Eigenvalue | 0.3624 | 0.3616 |
| Louvain | 0.3936 | 0.3904 |
| Walktrap | 0.3257 | 0.3431 |

TopicMapping uses the InfoMap clustering algorithm, which yields 41 different clusters for the Science corpus and 24 for the Indeed corpus. Plotting each network with nodes colored according to the InfoMap cluster ids and clustering each node according to which topic it appears in most often, we obtain the following:

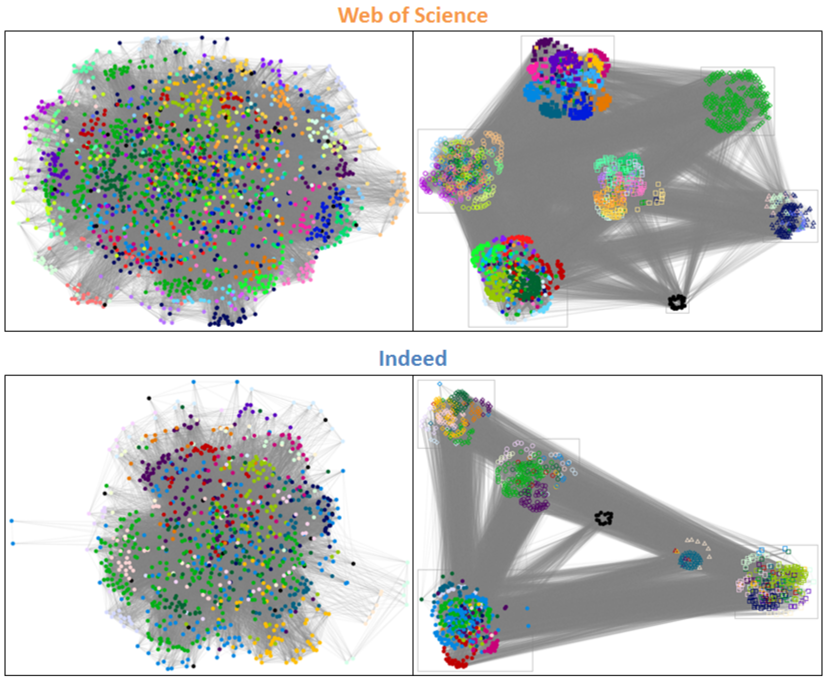



Figure 5: Science is plotted on the top, while Indeed is plotted on the bottom. The network colored according to InfoMap clusters is on the left, while words are clustered and are assigned symbols according to the topic they appear in the most often through the corpus.

In both networks, we can see a small cluster of black filled in circles, which are words that appear equally in all topics and are not assigned a clustering. Removing these nodes, there are six and five clusters, one for each topic in Science and Indeed, respectively. Recalling that the cluster ids assigned ot each word from InfoMap is the node color, we see that InfoMap perfectly assigns one topic for the Science corpus as shown by the green open circles. InfoMap does not perfectly assign cluster ids to any other topic in either Science or Indeed. These are the only results we can conclude from this visual analysis.

# 5 ERGMS

I hypothesize that projected word networks which do not cluster well will not perform well in terms of TopicMapping. Since there do not appear to be any differences in the word networks from a structural stand-point, I explore the extent of homophily in each network as quantified using ERGMS. For each network, I first use the InfoMap clustering algorithm in the *igraph* package in R to obtain membership ids for each word. I then fit an ergm with the nodematch term to test for homophily in each network. I include both edges and isolates as controls. Although there are several clustering algorithms to choose from, I focus only on InfoMap because it is used in the TopicMapping algorithm. The coefficients were obtained using MLE. The MCMC diagnostic plots as well as goodness of fit plots can be found in Appendix B. I obtain the following Maximum Pseudolikelihood Results:

| | Science | | | | Indeed | | | |
|---|---|---|---|---|---|---|---|---|
| | Estimate | Std. Error | MCMC% | p-value | Estimate | Std. Error | MCMC% | p-value |
| edges | -3.3591 | 0.0040 | 0 | $< 0.0001$ | -2.9925 | 0.0063 | 0 | $< 0.0001$ |
| isolate | -Inf | 0 | 0 | $< 0.0001$ | -Inf | 0 | 0 | $< 0.0001$ |
| homophily | 3.3696 | 0.0077 | 1 | $< 0.0001$ | 3.3616 | 0.0122 | 0 | $< 0.0001$ |

First note that the coefficient for the isolates term is negative infinity since there are no isolates in either network. I included the term because, by definition, there should never be any isolate in a word projection network. Next we should notice that both of the homophily terms are significant, positive, and approximately 3.36. There is a slight decrease in the term for Indeed, but not as much as I expected.

It is important before analyzing these results further to take a close look at the goodness of fit statistics. Referring to Appendix B, we can clearly see that neither ergm fits well. After several attempts to obtain a better fit by including terms for different degree properties, edge weights, and several other options, none of the more complex models converged with either MLE or MPLE. Since neither model fits well, it is important to verify these results. I will first apply QAP regression, where the response is the weighted projected word network, and the predictor is the binary network indicating if two nodes are in the same cluster. I use the netlm function in the sna package in R. I obtain the following results:

| | Science | | | | Indeed | | | |
|---|---|---|---|---|---|---|---|---|
| | Estimate | $\Pr(\leq b)$ | $\Pr(\geq b)$ | $\Pr(\geq \lvert b \rvert)$ | Estimate | $\Pr(\leq b)$ | $\Pr(\geq b)$ | $\Pr(\geq \lvert b \rvert)$ |
| intercept | 0.0463 | 1 | 0 | 0 | 0.0621 | 1 | 0 | 0 |
| Homophily | 0.9537 | 1 | 0 | 0 | 0.9379 | 1 | 0 | 0 |
| Adj. $R^2$ | 0.2406 | | | | 0.245 | | | |

Note that for the binary predictor, a one represents that two words are in the same cluster. As we can see, the two models are extremely similar, which is expected after seeing the ergm output. The coefficients have the same sign, but are lower. The models do have the same interpretation: a new additional node with a cluster id will attach to other nodes with the same clustering id with a coefficient of about 2.9 on average and will attach to a node with different clustering ids with a coefficient of about 0.05 regardless of corpus. These results suggest that there is positive homophily in both networks at a similar rate, which is contrary to what I hypothesized.

As a second precaution, I will perform a conditional uniform graph test to make sure that the clustering coefficient is in fact significantly different for both models as compared to what would be expected based on each network's respective dyad distribution. I use the rguman function in the sna package to simulate 100 random networks conditional on the dyad distribution for each corpus. I then obtain the clustering ids from the InfoMap algorithm and calculate the modularity for each simulated network. Comparing these results, it is easy to see that for all of the simulated networks, the modularity is negative and close to 0. Where as we saw earlier, the modularity for both Science and Indeed was close to 0.39.

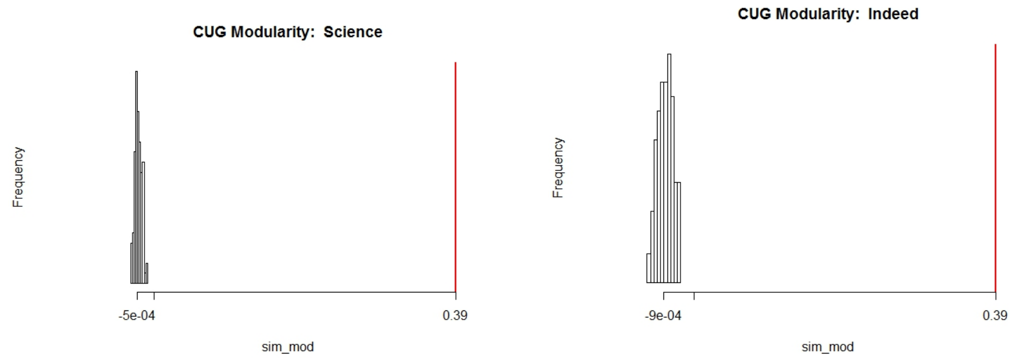



Figure 6: Science is plotted on the left, while Indeed is plotted on the right. The red line indicates the actual modularity score, while the simulated modularity scores are plotted as a histogram.

I also create a CUG test for the transitivity of each network. Here, we see that the transitivity is about 0.42 for both Science and Indeed, but the mean for the simulated values is about 0.75. Both networks have lower than expected transitivity as compared to all 100 runs.

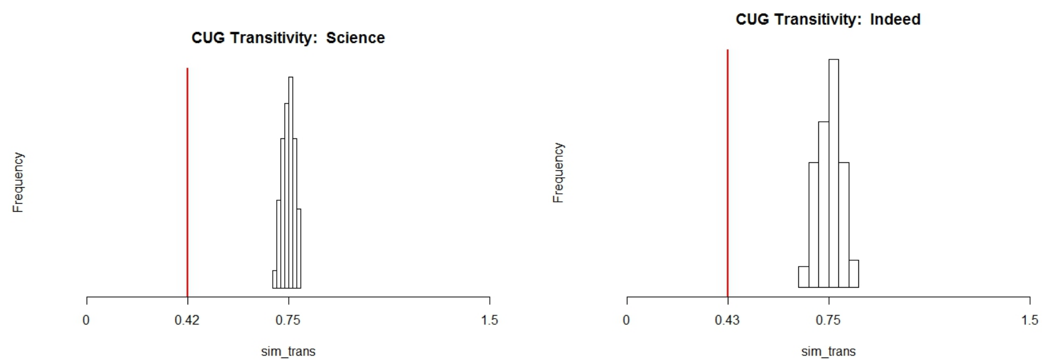



Figure 7: Science is plotted on the left, while Indeed is plotted on the right. The red line indicates the actual transitivity score, while the simulated transitivity scores are plotted as a histogram.

# 6 Conclusions

In [11], TopicMapping is motivated by the inaccuracy and unreproducible results of classical LDA as defined in [4]. The algorithm is shown to outperform both LDA and pLSA through simulations and two real world datasets. The author's results are reproducible. However, I have shown that applying the TopicMapping algorithm to other real world datasets, collections of resumes, does not produce results that outperform LDA. I first hypothesis that there are distinctive underlying features of each corpus' projected word network. After obtaining little differences in descriptive statistics, I further my hypothesis to state that each network would be partially explained by homophily between the InfoMap clusters and that Science would be explained more so than Indeed.

After fitting ERGMS with QAP and CUG robustness checks, I found that homophily did in fact help explain the netowrks, but that it did so equally in both Science and Indeed.

These results could be impacted by several factors. The first is that although the ERGMs were not degenerate and the MCMC values converged, the fit was not the best, refer to Appendix B. All of the variation can not be explained by homophily alone. There is most likely many confounding variables that I have not considered. By including other variables such as transitivity or density, the effect of homophily could be impacted. In particular, the addition could impact homophily differently in each model. A second concern involves the filtering process. Although I do not discus the filtering method used in [12] to obtain the final word networks, this step in the TopicMapping algorithm is crucial. The method used is refered to as an, 'agent degree conditioned thresholds' method in [14]. The method introduced in [14] called the, 'stochastic degree sequence model,' (SDSM) in theory would lead to better results, which may help the robustness of TopicMapping.

# References


[1] Anonymous. (2012), "Two R Packages for Topic Modeling, lda and topicmodels?" *Cross Validated*: http://stats.stackexchange.com/questions/24441/two-r-packages-for-topic-modeling-lda-and-topicmodels.

[2] Asuncion, A., Welling, M., Smyth, P., and Teh, Y. (2009), "On Smoothing and Inference for Topic Models." *UAI*: 27-34.

[3] BHoffman, T. (1999), "Probabilistic Latent Semantic Indexing." *Proccedings of the Twenty-Second Annual International SIGIR Conferance.*

[4] Blei, D., Ng, A., and Jordan, M. (2003), "Latent Dirichlet Allocation." *Journal of Machine Learning Research*: 3 993-1022.

[5] Blei, D. (2004),"LDA-C."

[6] Chang, J. (2015), "Package 'lda'." *CRAN.*

[7] Chien, J., and Wu, M. (2008), "Adaptive Bayesian Semantic Analysis." *Audio, Speech, and Language Processing*, IEEE Transactions on: 16(1), 198-207.

[8] Griffiths, L. and Steyvers, M. (2004), "Finding Scientific Topics." *PNAS*: 1(Suppl 1), 5228-5235.

[9] Grun, B. and Hornik, K. (2015), "Package 'topicmodels'." *CRAN.*

[10] Hofmann, T. (2001), "Unsupervised Learning by probabilistic Latent Semantic Analysis." *Machine Learning*: 42(1), 177-196.

[11] Lancichinetti, A., Sirer, M., Wang, J., Acuna, D., Kording, K., and Amaral, Luis. (2015), "High-Reproducibility and High-Accuracy Method for Automated Topic Classification." *Physical Review X*: 5, 0011007, 2160-3308.

[12] Lancichinetti, A., Sirer, M., Wang, J., Acuna, D., Kording, K., and Amaral, Luis. (2015), "High-Reproducibility and High-Accuracy Method for Automated Topic Classification: Supplemental Material." *Physical Review X*: 5, 0011007, 2160-3308.

[13] Mukherjee, I. and Blei, D. (2009), "Relative Performance Grantees for Approximate Inference in Latent Dirichlet Allocation." *NIPS*: 21, 11291136.

[14] Neal, Z. (2014), "The backbone of bipartite projections: Inferring relationships fromco-authorship, co-sponsorship, co-attendance and other co-behaviors." *Social Networks*: 39, 84-97.

[15] Speh, J., Muhic, A., and Rupnik, J. (2013), "Parameter Estimation for Latent Dirichlet Allocation." *Artificial Intelligence Laboratory*: Ljubljana, Slovenia.

[16] Teh, Y. W., Newman, D., and Welling, M. (2007), "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation."*NIPS*: 3, 1353-1360.

# A Replication Study

## A.1 LDA Limitations

The paper [11] begins with a simple thought experiment. Given a Corpus consisting of documents in three distinct languages, say English, Spanish, and French, let all words in each language be unique and each document contain only one language. Ideally, a topic model would produce three distinct topics, one for each of the unique languages. However, this is not the case with LDA. The authors of [11] show the likelihood of the generative model (i.e. placing all English documents in one topic, all Spanish documents in another, and all French documents in the last topic) is not always maximum. In fact they describe an alternative model, which theoretically obtains a higher likelihood. The alternative model is defined by separating one language into two dialect topics and combining the other two languages into one topic. In fact the authors prove in the supplemental material that, "it is possible to find an extremely large number of alternative models (with the same number of topics) which overfit some topics and underfit some others but have a better likelihood then the true generative model" [12]. These results are visualized in Fig. 2 (pg. 3,[11]). Part a illustrates the differences in the generative and alternative models considered. Part b and c show the theoretical limits on the log likelihood for both generative and alternative models when considering a vocabulary of 60 unique words, where 20 are in either $E, S$, or $F$. The corpus contains 1000 documents, where each document contains 10 words chosen uniformly from one language. Specifically, they show that the for these particular parameter values, if the fraction of $E$ documents in the corpus is greater than 0.936, then the likelihood of the alternative model become greater than the likelihood for the generative model.

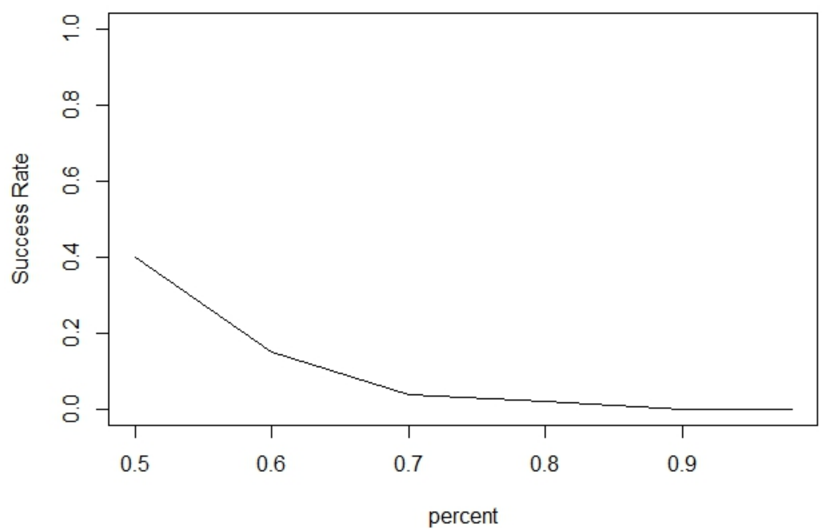
In order to replicate part d of Fig. 2, I will first create a synthetic language corpus and then apply the LDA algorithm as programed in the *topicmodels* package in R. We can determine the success rate as shown in Fig. 2 (pg. 3 [11]) part d. Here, I also consider the average cosine similarity between each document's topic distribution and the true topic distribution. Following the process in [11], I first create a list of twenty words in English $E$, Spanish $S$, and French $F$. I then create a corpus of 1000 documents, where the number of $E, S$, and $F$ documents follow the following equation:

$$100 = pE + \frac{(100-p)}{2}S + \frac{(100-p)}{2}F,$$

for $p = 0.5, 0.6, 0.7, 0.8, 0.9, 0.96$. Note that each of the twenty words of a particular language has an equal probability of being sampled for each document with replacement, and each document consists of exactly ten words. I then apply LDA using the Variational EM Approach, which was used in [11]. I run the algorithm 100 times calculating the average cosine similarity and success rate with each run to obtain the following results:

| $p$ | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 0.96 |
|---|---|---|---|---|---|---|
| Min. | 0.5763 | 0.5593 | 0.5318 | 0.5780 | 0.5774 | 0.5765 |
| 1st Qu. | 0.5844 | 0.5821 | 0.5802 | 0.5800 | 0.5791 | 0.5774 |
| Median | 1.0000 | 0.5871 | 0.5834 | 0.5822 | 0.5802 | 0.5777 |
| Mean | 0.8252 | 0.7240 | 0.6481 | 0.5919 | 0.5803 | 0.5776 |
| 3rd Qu. | 1.0000 | 1.0000 | 0.5913 | 0.5852 | 0.5809 | 0.5779 |
| Max. | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.5858 | 0.5783 |

We Notice that the minimum cosine similarity stays relatively constant across percent levels. This is due to the fact that in all runs, there are only three options of outcomes: First, LDA correctly identifies the generative model resulting in a cosine similarity measure of about 0.99; Second, LDA identifies the alternative model resulting in a cosine similarity of about 0.73; and Third, LDA assigns a uniform topic distribution for each document resulting in a cosine similarity of about 0.55. Although LDA applies a uniform topic distribution each of the corpuses at least once, we notice that when the percent level is 0.9 and 0.96 the algorithm applies a uniform topic distribution in all 100 runs. Also, notice that the average cosine similarity decreases as the percent of English documents increases as expected. The following plot shows the success rate as the percentage of runs that correctly identifies the generative model:




We now notice that the success rate is lower than the paper suggests, but there are many reasons for this. First, we are not analyzing the same corpus. Second, LDA uses Variational EM methods to fit the topic distribution, which is a stochastic process. Even though the overall success rate is lower, the trend stays the same. LDA tends to produce the correct generative model when the percentage of English documents is lower. In this example, when the percentage of English documents is greater than 0.9, LDA never produces the generative model.

## A.2 TopicMapping

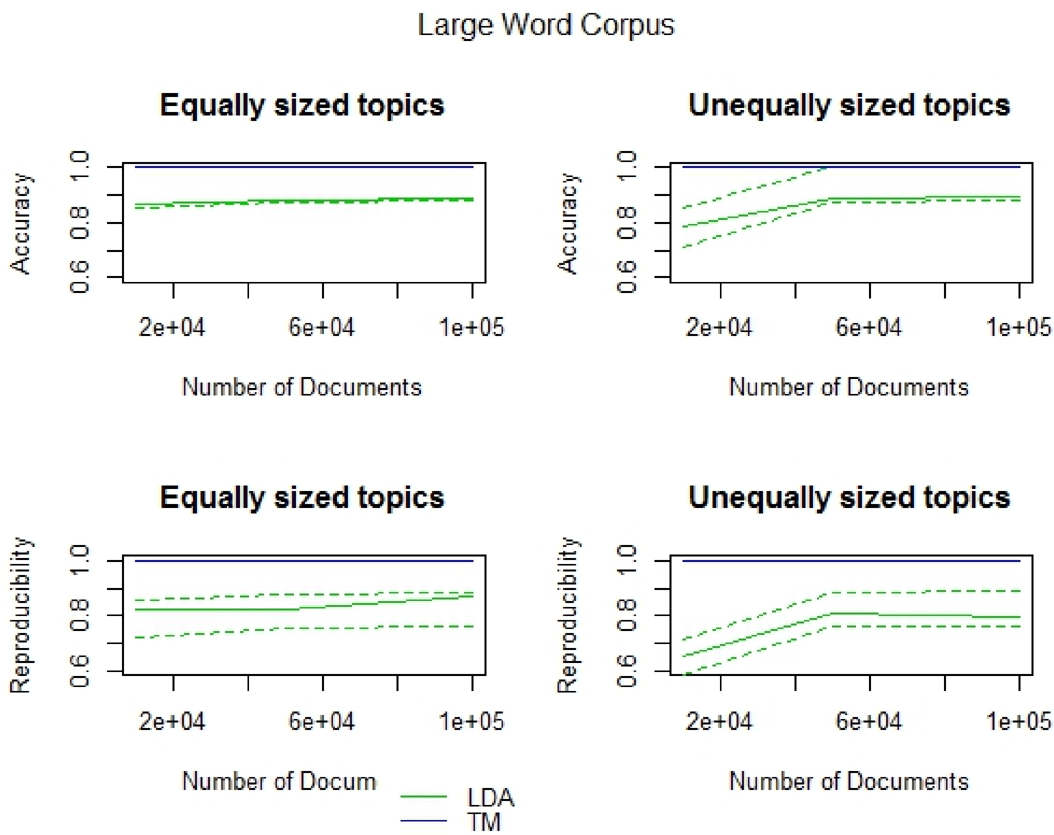
The second part of the paper introduces a new algorithm for topic modeling called TopicMapping. The algorithm is described in [11], with more details in [12]. In order to reproduce the article's main empirical results, I need to run their algorithm. I contacted the author's to not only obtain their original data, but their code if possible. I received code from one of the authors. The code is written mostly in C++, but also in Python. After several attempts, I was able to successfully run their code on a mac, but have run into the following error on a pc: "InfoMap did not compile. Please contact me: arg.lanci@gmail.com ." I have sent an email asking for help, but have yet to hear back. I was also able to code the algorithm in R. However, my code is neither efficient nor fast. For the following results, I ran everything I could on my pc, but decided to simply run the TopicMapping algorithm on a mac, save the results to a flash drive, and upload them to my pc for analysis.

## A.3 Comparison of LDA to TopicMapping

There are two types of major synthetic data analyzed in the paper. The first type of data is similar to that in Section 2.1, with corpuses of differing sizes contain ten languages in different proportions. The other type of data consists of corpuses constructed using the generative process defined in [4], where the number of documents is 1000, the number of words in each document is 50, the number of topics is set to 20, and the hyper-parameters are set at $\alpha = 0.001, 0.004, 0.016, 0.064$. They also vary the percentage of words in the corpus that are considered generic, or equally likely across topics. Along with considering cases where each topic is equally distributed across documents, they consider cases where four topics make up 15% and the last 16 topics make up 2.5%. To simplify the amount of computing, I will focus on six corpuses of the first type: Three corpuses have ten equally distributed languages of size $1000, 5000, 10000$, and the other three have two languages making up 15% each of the corpus and the remaining eight languages making up 8.75% of size $1000, 5000, 10000$. I will also analyze twelve corpuses of the second type: Six corpuses have $\alpha = 0.001$ with generic proportion $0.2, 0.5, 0.8$, and either equal topic distribution or not; the last six have $\alpha = 0.064$ with generic proportion $0.2, 0.5, 0.8$, and either equal topic distribution or not.

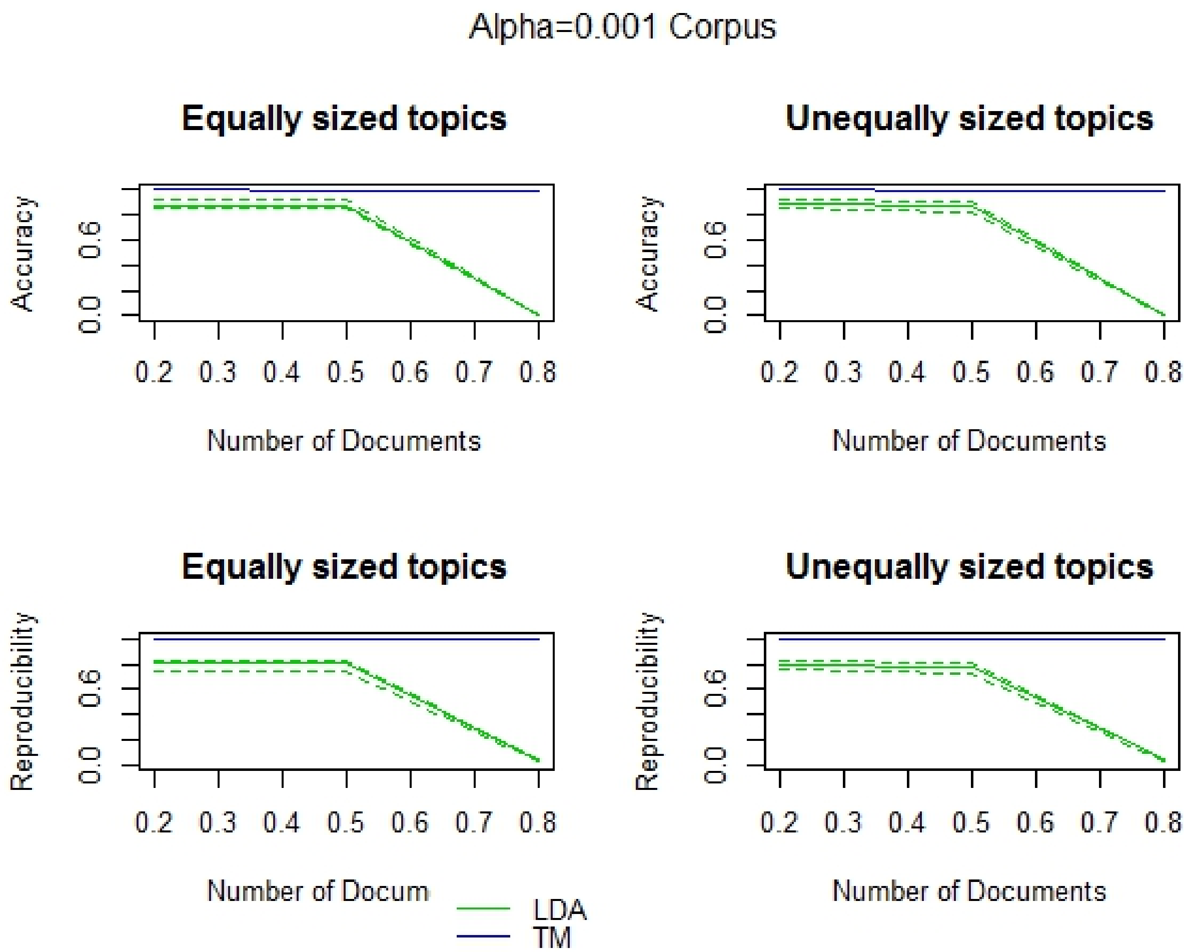
I was able to compile the code given to me by the authors, which simulate similar, but not the exact synthetic datasets. They did not save their original seed values. I was also able to successfully compile their program, which calculates the accuracy and reproducibility of a model. My aim is to reproduce the results as shown in Fig. 4 (pg.5, [11]) and Fig. 7 (pg.6, [11]) using a mac to run the TopicMapping algorithm and my pc to run LDA in R. The two other models that are considered by the authors are pLSA and LDA with seeded start values. Since the authors do not provide an explanation of what the start seeds are or how to find them, I ignore this model comparison. I also ignore the comparison to pLSA since this algorithm is not already programed in R nor do the authors provide code for it. The results for LDA and pLSA were similar to the results of LDA in all cases studies considered. Therefore, I will focus on reproducing the results for comparisons between TopicMapping and randomly seeded symmetric LDA only.

I begin with the language corpuses, where I obtained the following accuracy and reproducibility results:



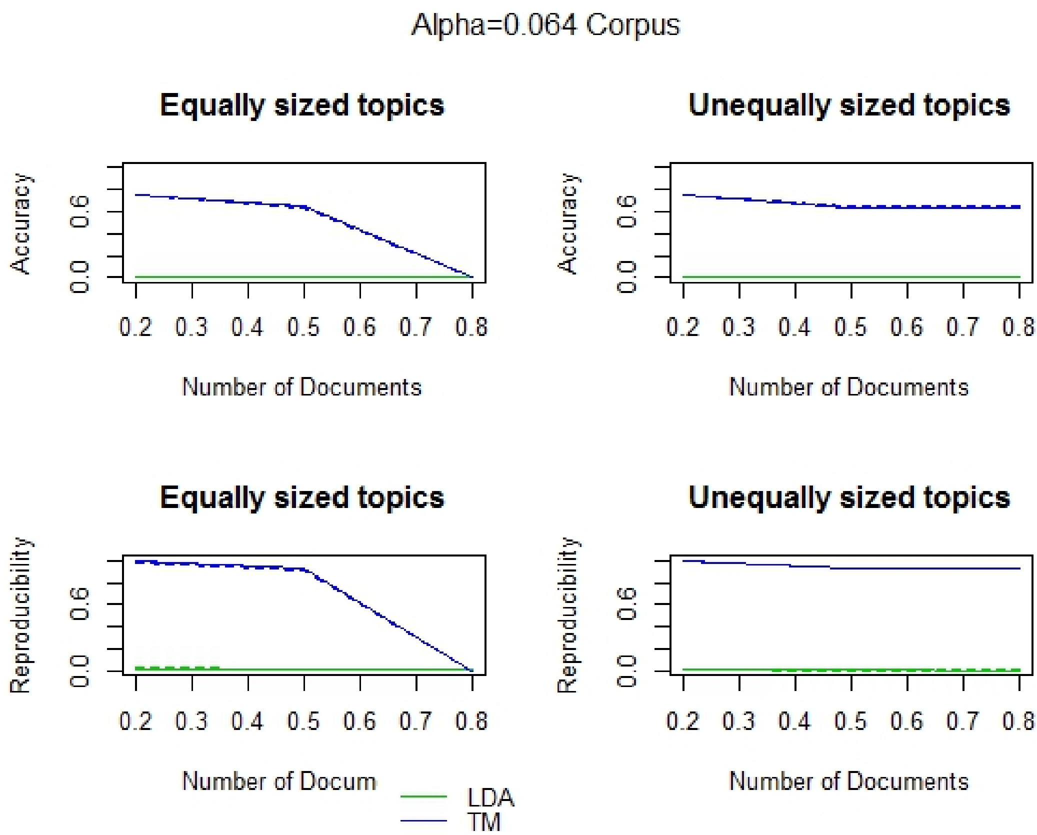

The plot above shows accuracy (top) and reproducibility (bottom) results for running LDA (green) and TopicMapping (blue) 100 times on both equal (left) and unequal (right) topic distributions. With both algorithms, the number of topics was fixed at 10. The median for the 100 runs is plotted, where dotted lines indicate the 25th and 75th percentiles for both algorithms as described in [11]. We can clearly see that the results follow the same patterns as shown in [11]. TopicMapping is performing at near perfect accuracy and reproducibility for both equally and unequally sized topics regardless of the number of documents in the corpus. LDA clearly performs with greater variance, and does slightly better as the number of documents increases. Yet, TopicMapping outperforms LDA in all cases.

The next results we will look at are those obtained from running both LDA and TopicMapping on the second set of simulated datasets when the hyper-parameters are set to 0.001:




The figure above shows accuracy (top) and reproducibility (bottom) results for running LDA (green) and TopicMapping (blue) 100 times on both equal (left) and unequal (right) topic distributions. With both algorithms, the number of topics was fixed at 20 and the hyper-parameters are set at 0.001. Again, the median for 100 runs is plotted, where dotted lines indicate the 25th and 75th percentiles for both algorithms. We can clearly see that the results follow the same patterns as shown in [11]. TopicMapping is performing at near perfect accuracy and reproducibility for both equally and unequally sized topics regardless of the number of documents in the corpus. LDA clearly performs with greater variance, and does slightly better as the percentage of generic words decreases. Yet, TopicMapping outperforms LDA in all cases.

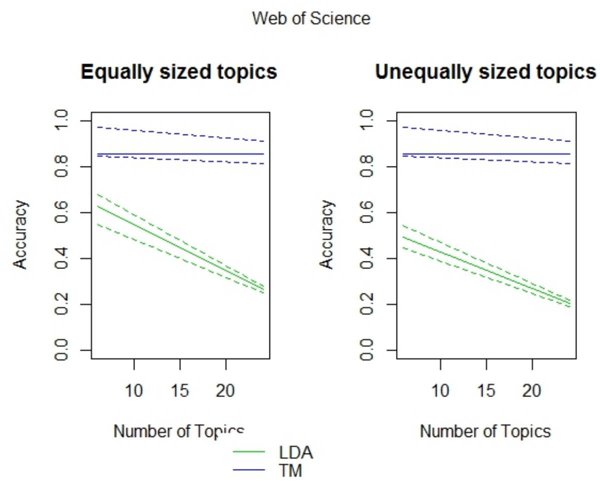
Lastly, we will look at the results obtained from running both LDA and TopicMapping on the second set of simulated datasets when the hyper-parameters are set to 0.064:




The figure above shows accuracy (top) and reproducibility (bottom) results for running LDA (green) and TopicMapping (blue) 100 times on both equal (left) and unequal (right) topic distributions. With both algorithms, the number of topics was fixed at 20 and the hyper-parameters are set at 0.064. Again, the median for 100 runs is plotted, where dotted lines indicate the 25th and 75th percentiles for both algorithms. We can clearly see that the results follow similar patterns as shown in [11]. Both LDA and TopicMapping perform uniformly across differing percentages of generic words. TopicMapping with 0.7 accuracy and 0.8 reproducibility outperforms LDA, which performs poorly. Although LDA consistently performs poorly with unequally sized topics, TopicMapping's accuracy and reproducibility drop as the percentage of generic words increases, which is exactly what we see in [11].

## A.4 Applications

I received emails back from two of the authors providing me with the corpus for the first application dataset: *Web of Science* or WoS. Both authors stated that the second application dataset, Wikipedia, was too large to send. They stated that even if they provided me access to it through some means, I would be unable to successfully reproduce their results in a semester. I will focus on reproducing the results for the first application only. The WoS Corpus contains documents consisting of the titles and abstracts for 23838 journal articles published in six different top journals for the following fields: astronomy, biology, economics, geology, mathematics, and psychology. I was able to reproduce the results on Fig. 8 (pg. 7, [11]):




I have plotted the median accuracy (left) and reproducibility (right) from the 100 runs for both TopicMapping (blue) and LDA (green) with dotted lines representing the 25th and 75th percentiles. The results were found when the number of topics was set at 6 the correct number and at 24 as in [11]. We notice that with both cases the correct number of topics and incorrect number, TopicMapping does well in both accuracy and reproducibility. However, LDA performs better when the correct number of topics specified. Again, TopicMapping outperforms LDA in all cases.

# B ERGM GOF

From the MCMC diagnostics plots below, we can see that the MCMC estimates converged:

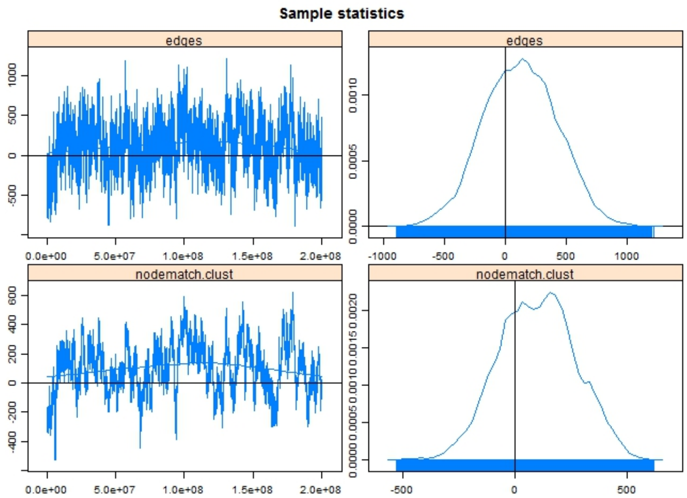



Figure 8: MCMC diagonostic plots for Science.

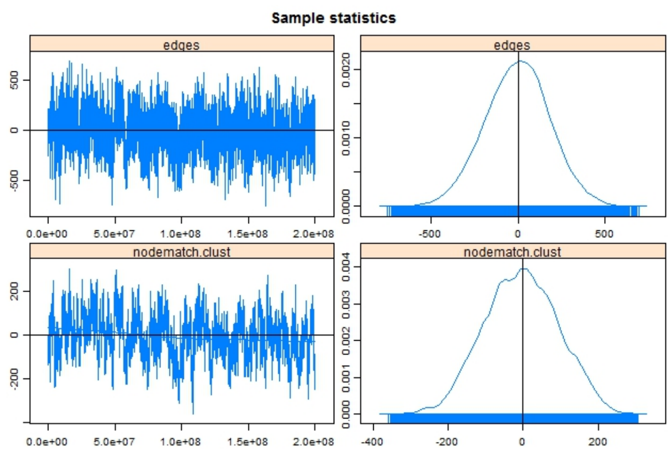



Figure 9: MCMC diagonostic plots for Indeed.

The Geweke statistics for the Science and Indeed corpus were all insignificant at the 0.05 level with an overall p-value of 0.1125 and 0.04224, respectively. The quantiles can be seen below:

| | 2.5% | 25% | 50% | 75% | 97.5% |
|---|---|---|---|---|---|
| Science | | | | | |
| edges | -387 | -154 | -24 | 106 | 363 |
| nodematch.clust | -247 | -103 | -22 | 63 | 225 |
| Indeed | | | | | |
| edges | -376 | -131 | -2 | 121 | 359 |
| nodematch.clust | -191 | -73 | -5 | 64 | 183 |

We can clearly see that both ergms are not degenerate. However in the following goodness of fit plots, we can see how complex these networks are. Neither ergm fits the data well:

The following show the first part of each image blown up. We can now clearly see that neither ergm model fits well: