Construction of Index of Expectations using Topic Models

Geeta Garg

February 18, 2019

This document contains a description of the methodology used to quantify newspaper text to create an index of interest rate expectations of the private sector using topic models. The first section provides a brief description of the data used and the data pre-processing techniques used in cleaning the data. This section also provides a brief overview of the errors that might still be present in the final data used in topic modeling. The section 2 provides the details of different topic models - Latent Dirichlet Allocation (LDA) using Variational Expectation Maximization (VEM) and Gibbs sampling and the Non-Negative Matrix Factorization (NMF) method - used in creating a measure of expectations from the newspaper articles. Section 3 compares the results from these methods.

1 Data

The data consists of the newspaper articles from two newspapers - the New York Times and the Wall Street Journal starting January 01, 1975 until December 31, 1981. These articles have been searched from the ProQuest database using keyword "Volcker" and amount to a total of 1399 articles in number. The Figure 1 below presents the raw counts of news articles by month containing the keyword "Volcker". The reason why article search was done using only one keyword is because the size of corpus gets reduced by 400 articles if I include either 'inflation' or 'interest rate' as additional keywords with 'Volcker'. In addition, the article search using additional words may exclude articles that uses the language short term rate, fed funds rate etc instead of interest rate since ProQuest will search only those articles that have the following combination of words "Volcker and Inflation" or "Volcker and interest rate".

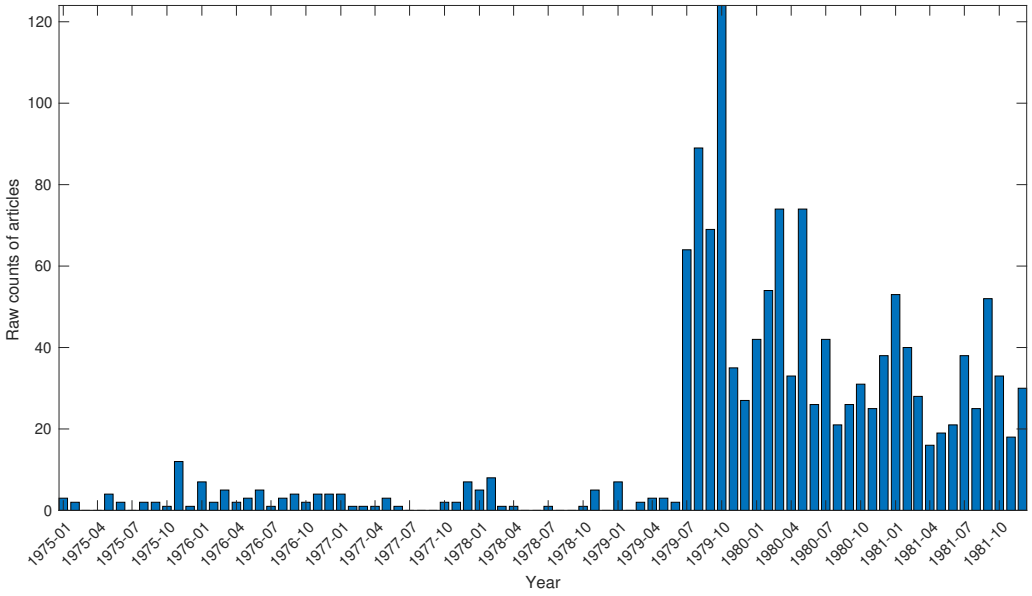


Figure 1: Frequency Counts of Articles with Keyword "Volcker"

The Figure 1 shows that there was a slight increase in the number of articles in the late 1970s due to Volcker's appointment as the president of the Federal Reserve Bank of New York in August, 1975. Then the number of articles increased substantially starting late July 1979 when the announcement of Volcker taking over as the chairman of the Federal Reserve Board was made and continued to stay high afterwards. Figure 2 and 3 below present similar counts for other chairmen of the Federal Reserve. These articles were searched using keywords 'Alan' and 'Greenspan' for Alan Greenspan, 'Bernanke' for Ben Bernanke, 'Janet' and 'Yellen' for Janet Yellen and 'Jerome' and 'Powell' for Jerome Powell.

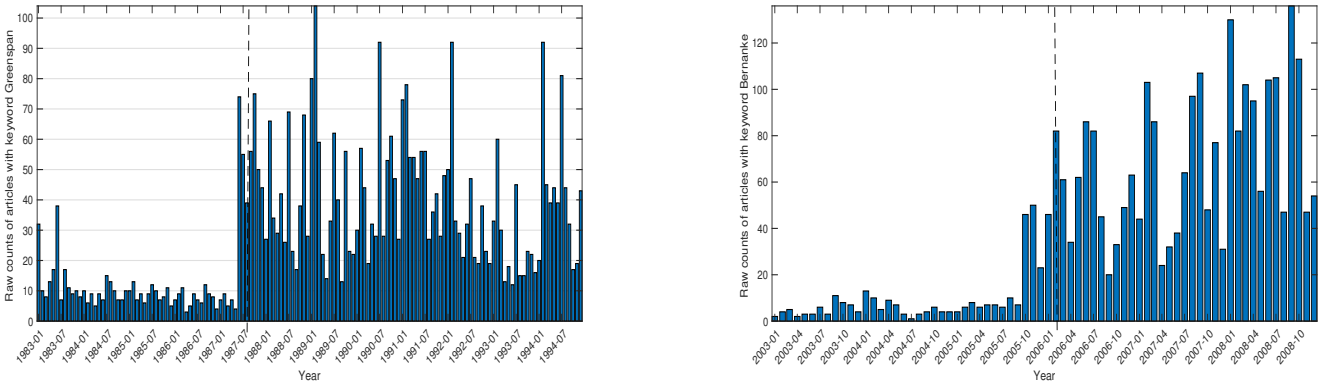


Figure 2: Frequency Counts of Articles with Keyword- 'alan' and 'greenspan' (left) & 'bernanke' (right)

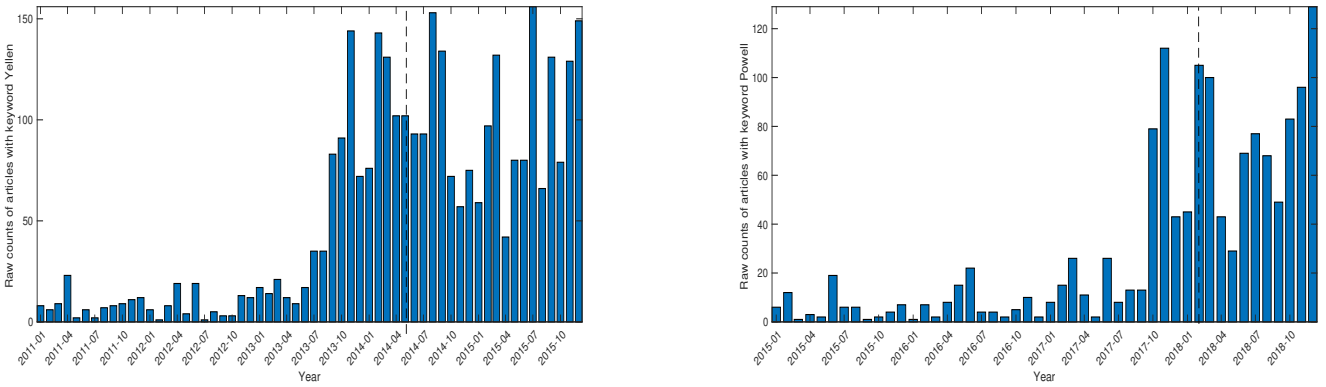


Figure 3: Frequency Counts of Articles with Keyword- 'janet' and 'yellen' (left) & 'jerome' and 'powell' (right)

The black line represents the date of appointment of the Federal Reserve chairmen. The trend looks more or less similar for Greenspan as that of Volcker, however, for Bernanke, Yellen and Powell, the number of articles increased much before the date of their appointment. This could also be due to the fact that during the times of appointment of these candidates, the overall number of newspaper articles also increased with the share of online articles being a huge fraction of the overall articles published by a newspaper. Also, as far as Volcker is concerned, his appointment was very sudden after the departure of former chairman William Miller amidst rising inflation in addition to Volcker not being the most preferred choice of President Carter. Since there was hardly any news about his appointment as the chairman of the Fed, the frequency of articles is very low before his appointment.

1.1 Preprocessing of Data

Prior to estimation, the raw data is preprocessed in several steps to bring it into usable form. The purpose is to reduce the vocabulary (in the corpus of documents) to a set of terms that are most likely to reveal the underlying content of interest and further facilitate the estimation of more semantically meaningful topics. The steps are as follows:

1. **Converting pdf articles to text form:** The newspaper articles from ProQuest are the scanned images that are saved in pdf format. These images are noisy due to poor scanning quality and one of the challenges is to convert these images to text format (which is how a software reads data) without losing information to spelling errors. However, due to limited ability OCR scanners and image pre-processing tools, large amount of spelling errors as still present in the final data. Python's OCR software has been know to be very competitive in terms of providing good accuracy, however, there

are other paid softwares also available that can offer even better accuracy. In addition, while converting the pdfs to text, the sequence of the sentence changes in the conversion process (in a lot of articles) which leads to errors in reading hyphenated words which can lead to a loss of important words from the articles. I used a spell-checker (SymSpellCompound) that fixes some of the spelling errors, however, it is difficult for spell-checkers to be 100% accurate due to the limited size of their dictionaries. In addition, they can also add further noise by mis-correcting certain words based on its dictionary. Table 1 provides an overview of the kind of errors that might be present in the final data. The other spell-checkers that might be more accurate than the one I used, however, they are extremely slow and may take days if the corpus is big.

2. **Removal of Stopwords and Punctuation**: The second step of preprocessing is to remove common stopwords like "the" and "of" that appear frequently in all texts and add little or no information in identifying latent topics. In addition to default stopwords, many other stopwords that frequently appeared in the final topics but added little information were also removed based on judgement. The text was further cleaned by removing punctuation and other characters.

3. **Other commonly used techniques avoided:** There are a lot of other commonly used techniques such as stemming and lemmatization that removes suffixes, like ing, ly, s, etc. by a simple rule-based approach and converts the word into its root word. For instance, the words inflation, inflationary, inflate will be reduced to inflate by using these methods. These can be helpful if the objective is to capture the overall tone of the articles. I tried these techniques as well, however, these further added noise by converting words like "Fed" to root word "feed" and therefore I avoided them. [1]

In addition, Blei and Lafferty (2009) suggested that ranking the words (after all the pre-processing steps) using term frequency-inverse document frequency (tf-idf) which is a measure of informativeness that punishes both rare and frequent words can also

[1]Note: It is possible that there are still a lot of errors (spelling errors, words miscorrected by spellchecker, hyphenated or compound words that are at the end of a sentence and could not be joined due to change in the sentence sequence during the conversion process) present in the text, however, it is difficult to capture all of them without manually going through the articles. Also, I could not separate the headlines of the articles from the body of the articles. I tried separating the first 4 sentences which could potentially be headlines, however, as I mentioned above the OCR jumbles up sentences during the conversion of pdf images to text thus sometimes mixing the headlines with the body of the text. Thus, when I run the topic modeling algorithms, they are run on the entire article including the headlines which also includes the names of the authors. This can sometimes increase the frequency of a word in an article, for instance, compare an article in which inflation is mentioned in both the headline as well as the body of the article vs other articles where inflation is not mentioned in the headlines.

be helpful in reducing the corpus to very important words where

$$\text{tf-idf weight} = tf \text{ (Term Frequency)} * idf \text{ (Inverse Document Frequency)}$$
$$\text{where } tf(t) = \frac{\text{Number of times term } t \text{ appears in a document}}{\text{Total number of terms in the document}}$$
$$\text{and } idf(t) = \log \frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}}$$

The rarest and the most common words end up getting a very low tf-idf weights and therefore words with scores below a chosen threshold are generally dropped from the corpus. I also tried this step but it ends up removing many words that are important for identifying the topics. If the corpus size is very big, this technique can be helpful and choosing a very low threshold for dropping words didn't help much given the size of my corpus. Other techniques involve dropping words that are repeated below a certain frequency.

Once the text is pre-processed, the final corpus of documents is obtained. The table 1 presents a part of the final data obtained after the pre-processing stage.

Table 1: Final Corpus of Documents

Index	Content	Date	Newspaper
1	credit markets bond prices score strong gains	sep 20 1979	new york times
2	carter inflation crisis policy review carter inflation crisis	feb 26 1980	new york times
3	profitability pennsylvania corp badly hurt fed credit basil	nov 30 1979	wall street journal
4	fed seems sensitive interest rates carter scored monetary	may 2 1975	new york times
5	whats news business finance world wide wall street	jan 28 1981	wall street journal

The table 1 above presents 5 articles from the final corpus. To highlight some of the errors still present, the content of article 3 mentions the word "basil" which could originally be "basel" (which seems more appropriate given the context), however, it seems to have been mis-corrected by the spell-checker. In addition, the content of the article 5 mentions "whats news business finance world wide wall street" where the part of the sentence "whats news business finance" is the standard headline of a Wall Street Journal article (its a specific WSJ article with the same headline everyday), however, if these words are removed as stopwords, then the words "business" and "finance" will be removed from the entire corpus even when

these words are important for an article in identifying its topic. On the other hand, including these words can increase the frequency of these words in an article making them important when they may not be.

2 Methodology

Topic models are probabilistic models of text used to uncover the hidden thematic structure in a collection of documents (Blei, 2012). The main idea in a topic model is that there are a set of topics that describe the collection and each document exhibits those topics with different degrees. As a probabilistic model, the topics and how they relate to the documents are hidden structure and the main computational problem is to infer this hidden structure from an observed collection. This section discusses the three main topic modeling techniques that are unsupervised learning approaches to clustering documents, to discover topics based on their contents. The use of two different approaches allow comparison of results to ensure consistency in the extracted topics. These approaches are LDA (Latent Dirichlet Analysis) using VEM and Gibbs sampling, and NMF (Non-negative Matrix factorization).

The basic idea behind all these methods is to find hidden semantic structures in the articles which is a form of dimensionality reduction. Say for example there is a corpus of D documents $\{d \in 1, 2, ..., D\}$ with a vocabulary size of M words $\{W = w \in 1, 2, ..., M\}$ (M unique words determined from the entire corpus). This gives us a document-term matrix in which every i, j cell gives the frequency count of word w_j in document d_i. The objective of both these methods is to convert this Document-Term Matrix into two lower dimensional matrices M_1 and M_2 where M_1 is a $(D * K)$ document-topics matrix and M_2 is a $(K * M)$ topic-terms matrix where n is the number of documents, K is the number of topics (latent factors) and M is the vocabulary size. The document-topic matrix M_1 thus obtained helps in classifying documents in a corpus under broad topics. While LDA uses probabilistic methods, NMF uses principle-component analysis to achieve this dimensionality reduction.

2.1 LDA Statistical Model

Latent Dirichlet Allocation (LDA) is a generative probabilistic model that assumes documents in a corpus are generated from a mixture of topics (latent factors that we are trying to extract). All documents contain a particular set of topics, but the proportion of each topic in each document is different. These topics then generate words based on their probability distribution. In order to understand this generative process, let's look at some of the broad underlying assumptions and the notation (Hoffman, Blei, Wang and Paisley, 2013) used in

this process:

- Observations are words, organized into documents. The n_{th} word in the d_{th} document is w_{dn}. Each word is an element in a fixed vocabulary of V terms (This vocabulary generally consists of the most meaningful words derived from the entire corpus that help in identifying a topic from the other).
- The topic β_k is a distribution over the vocabulary. Each topic is a point on the $V-1$ (term/vocabulary) simplex, a positive vector of length V that sums to one. The w_{th} entry in the k_{th} topic is β_{kw}. In LDA there are K topics (the number of topics is assumed to be fixed).
- Each document in the collection is associated with a vector of topic proportions θ_d, which is a distribution over topics. θ_d is a point on $K-1$ (topic) simplex. The k_{th} entry of the topic proportion vector θ_d is θ_{dk}.
- Each word in each document is assumed to have been drawn from a single topic. The topic assignment z_{dn} indexes the topic from which w_{dn} is drawn.

The geometric interpretation of LDA is depicted in the figure 4 below:

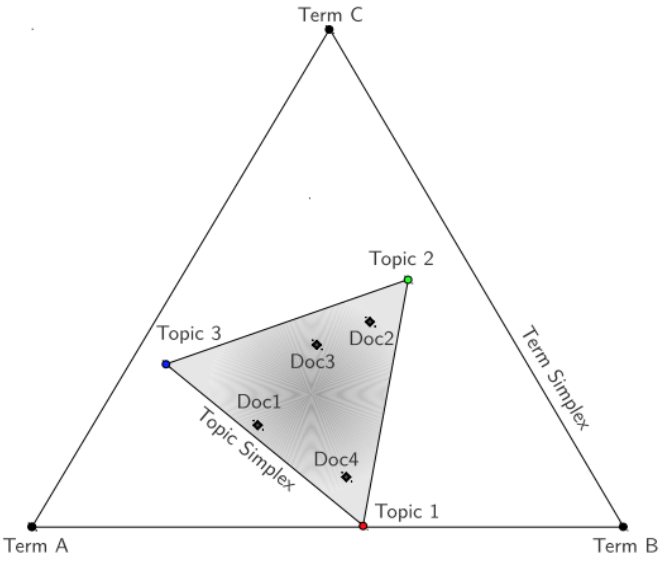


Figure 4: Geometric interpretation of Lda (Blei et al., 2003)

The outermost triangle is a $V-1$ simplex of the vocabulary and the inner triangle is a $K-1$ simplex of topics. The documents are contained in the innermost simplex (The vocabulary is assumed to be fixed in the beginning as well but we derive it from the corpus we are

using. A lot of words in the corpus do not go into analysis because they are not very useful in identifying topics. This is the reason why the vocabulary simplex is the outermost.) The generative process of a document (or how a document is generated) is described as follows (these assumptions will be clear in figure 6 and 7):

1. Draw topics $\beta_k \sim \text{Dirichlet}(\eta, ...\eta)$ for $k \in \{1, ..., K\}$
2. For each document $d \in \{1, ..., D\}$:
   (a) Draw a topic proportion $\theta \sim \text{Dirichlet}(\alpha, ..., \alpha)$.
   (b) For each word $w \in \{1, ..., N\}$:
      i Draw a topic assignment $z_{dn} \sim \text{Multinomial}(\theta_d)$, $Z_{dn} \in \{1, ..., K\}$
      ii Draw word $w_{dn} \sim \text{Multinomial}(\beta_{z_{dn}})$, $w_{dn} \in \{1, ..., V\}$

Figure 5 below present a graphical representation (plate model) of this generative process. The nodes are the random variables, edges indicate dependence. The shaded nodes are observed and the plates indicate replicated variables.

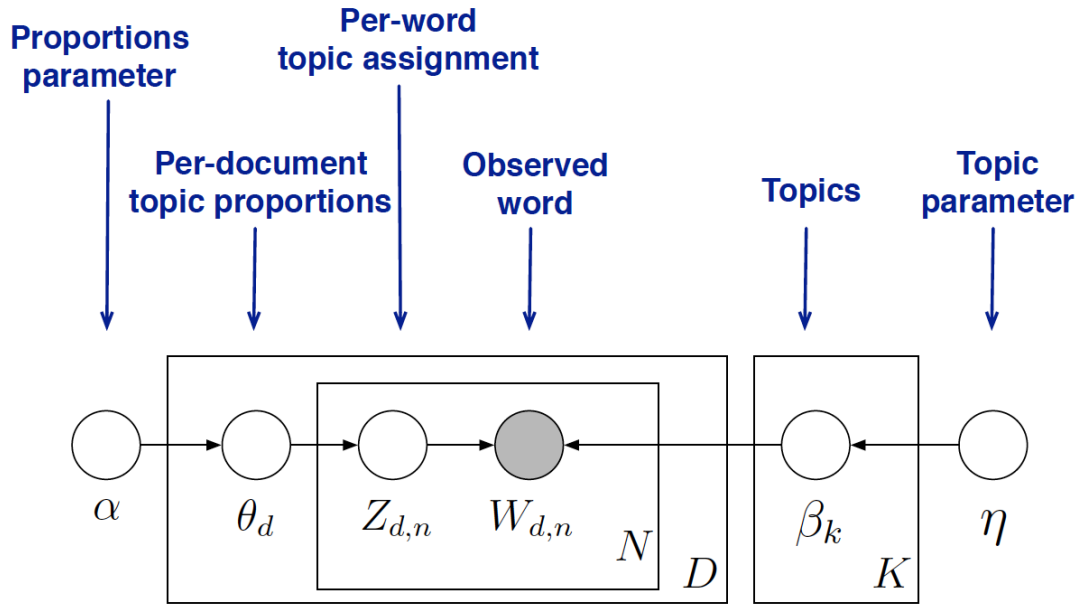


Figure 5: Hierarchical Bayesian generalization of LDA (Blei, Ng, and Jordan, 2001; 2003)

The joint distribution (of observed and hidden variables) corresponding to the process displayed in the figure 5 above can be represented as follows:

$$p(\theta, \beta, z, w|\alpha, \eta) = \prod_{i=1}^{K} p(\beta_i|\eta) \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{dn}|\beta_{1:K}, z_{d,n}) \right) \quad (1)$$

The following figure further explains this generative process:

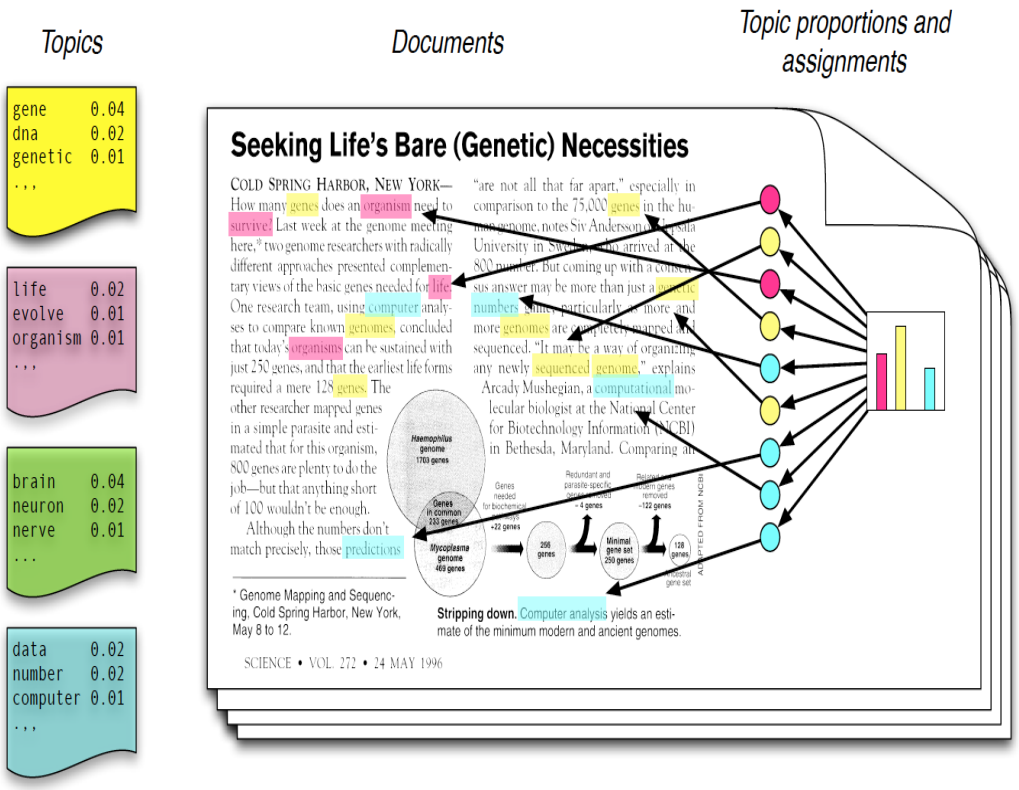


Figure 6: Generative process (Blei, 2012)

The distribution on the right is the distribution of a document (d) over a fixed number of topics (K) (assume this to be given or fixed in the beginning). On the left, there are topics each of which is a distribution over a fixed vocabulary (V) and in the middle we have a corpus of documents. The generative process displays how a document is generated. Assuming there are N_d words in this document d, for each word $w_{d,n}$, draw a topic from the topic proportions, find its topic number $z_{d,n}$ which are represented as different colored coins in the figure. Then draw a word from the topic (which has the same color as that of the coin) which will become the first word of the document. This process is repeated until all N_d words are generated which gives us the document d. Similarly, other documents can also be created (using the mixture of topics already assumed in the beginning). This figure exhibits that each document is a mixture of K different topics (seen as different colors - yellow, pink,

green...) and that each topic is a mixture of fixed vocabulary (the vocabulary here is list of fixed words - gene, genetic, dna, life, evolve, organism,....). When estimating LDA model, vocabulary is derived from the corpus of documents available at hand and these are generally the words that help identify a document from the other.

In reality however, we only observe the documents or words in the documents. The other structure - document-topic proportions θ_d, topic assignment of words z_{dn} and the topic-word distribution β_k are hidden variables. The figure below represents this:

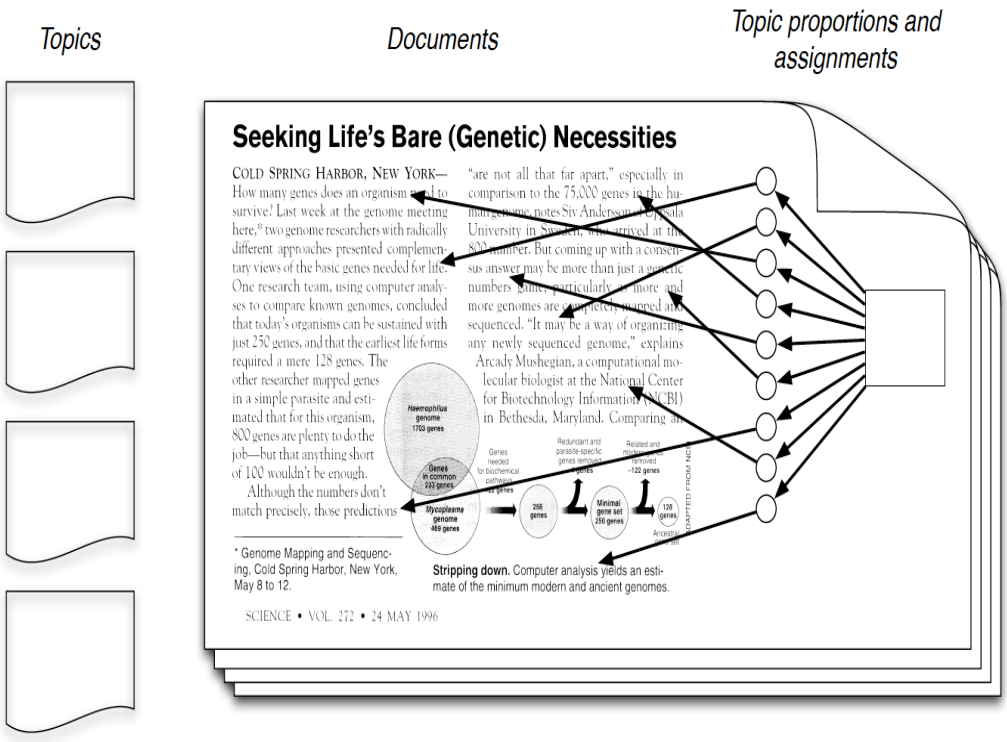


Figure 7: Generative process (Blei, 2012)

In this figure we only observe words in a document whereas the topic proportions and assignments (right of the figure above) and the topics (left of the figure above) are empty or hidden. The goal of the LDA is to observe the words in the documents (only observables in this model) in a corpus and then infer the hidden/latent variables (topic assignments of each word in a document, topic-word distribution and document-topic distributions), which is essentially the reverse problem given the generative process described above as the underlying assumption of LDA. The NMF also strives to achieve the same goal but it does not assume the generative process. It's purely an algebraic model. The following approximate

posterior inference algorithms are heavily used to estimate LDA:

- Mean field variational methods (Blei et al., 2001, 2003)
- Expectation propagation (Minka and Lafferty, 2002)
- Collapsed Gibbs sampling (Griffith and Steyvers, 2002)
- Collapsed variational inference (Teh et al., 2006)
- Online variational inference (Hoffman et al., 2010)

The section below will discuss the two heavily used methods to estimate LDA model and present the results from both methods - Mean field variational methods - MFVI (Blei et al., 2001, 2003) and Collapsed Gibbs sampling (Griffith and Steyvers, 2002) and the Non-Negative Matrix factorization method. The other methods discussed above are a variation of these methods. There are many other variations besides these. The method - Online variational inference (Hoffman et al., 2010) is almost same as Mean field variational methods (MFVI) but it was constructed to handle very large datasets and it also allows a constantly incoming stream of documents as and when more data is generated (for example if more articles are published online) which is not possible with MFVI which assumes the size of the corpus is fixed before estimation. Because the sizes of the text corpora are always growing, online variational inference (also called stochastic Variational Inference (SVI)) allows learning in an online fashion, that is, treat each document as it is arriving as sampled uniformly at random from a set of all possible documents, and perform inference using just this one document (or small chunk of articles in a mini-batch set up since the estimates based on just one document can be very noisy). The MFVI, however, is estimated on the entire corpus at once (also called the batch learning method) whereas online stochastic optimization is estimated on small chunks of a very large and constantly updated corpus. SVI also claims to produce more efficient estimates. I'll also include the results from this method in the appendix.

The key inference problem that we need to solve in order to use LDA is that of computing the posterior distribution of the hidden/latent variables (θ, β, z) given the a document (w) and the global fixed parameters α, η (integrating the observed words out in the joint distribution described in equation (1) above):

$$p(\theta, \beta, z, |w, \alpha, \eta) = \frac{p(\theta, \beta, z, w|\alpha, \eta)}{\int_\beta \int_\theta \sum_z p(\theta, \beta, z, w)} \tag{2}$$

where the denominator is

$$
\begin{aligned}
p(w|\alpha,\eta) &= \int_{\beta}\int_{\theta}\sum_{z} p(\theta,\beta,z,w)d\theta d\beta \\
&= \int_{\beta} p(\beta|\eta)\int_{\theta} P(\theta|\alpha)\sum_{z} p(z|\theta)p(w|z,\beta)d\theta d\beta
\end{aligned}
\tag{3}
$$

which is intractable due to the coupling between θ and β in the summation over the latent topic assignments - z. Thus, we need to use some approximate inference methods to compute the posterior distribution over the latent variables. The next section discusses different methods that have been applied to approximate this posterior distribution, each with its trade-offs.

3 Methods

3.1 Variational Inference for LDA

Blei, Ng and Jordan (2003) developed an inference method based on variational inference that simplifies the problem by approximating the posterior distribution (proxy distribution) of interest by a more tractable, conditionally independent variational distribution. They use a mean field approximation (also called Mean Field Variational Inference (MFVI) [2]) where they consider a family of distributions Q, over the hidden variables, that are fully factorized and are indexed by a set of free parameters that can be tuned or optimized to find the member of the family that is closest to the posterior of interest - p (Closeness is measured with Kullback-Leibler divergence). The resulting distribution is called the variational distribution that is used for approximation - $q(\theta_{1:D}, z_{1:D,1:N}, \beta_{1:K})$ which can then be used as a substitute for the true posterior. The following graphical representation represents the variational distribution that is used to approximate the posterior:

[2]MFVI is form of variational inference that uses a family where each hidden variable is independent which solves the coupling problem that we observe in the true posterior.

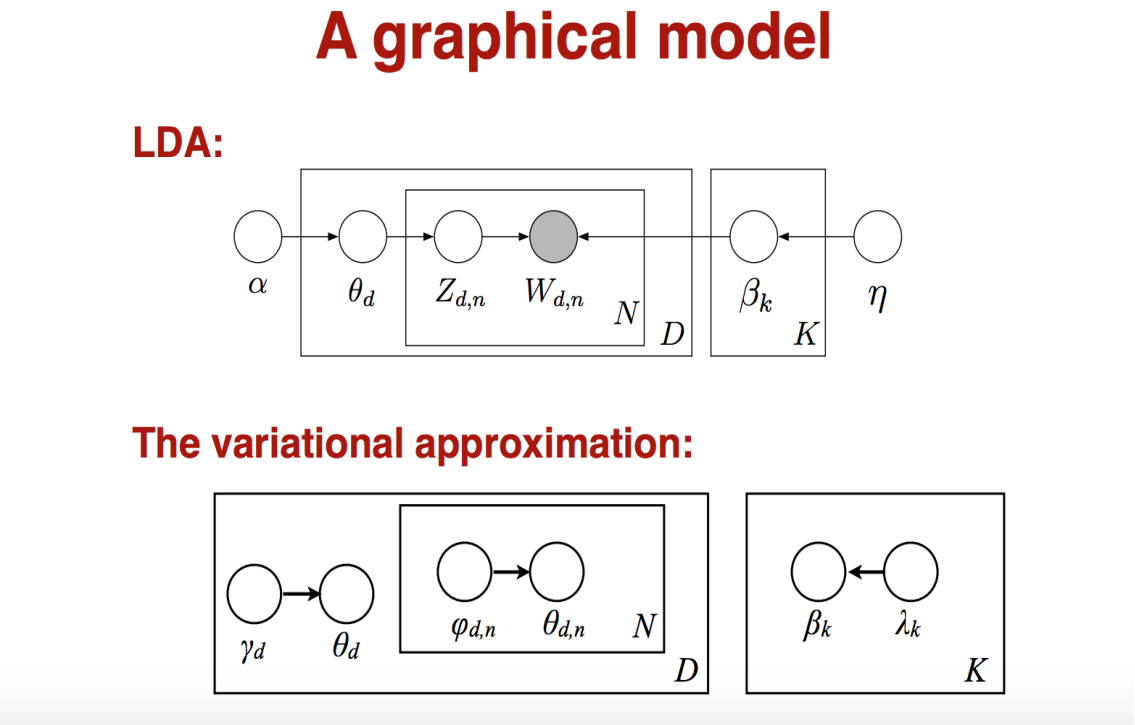


Figure 8: Variational Distribution as an approximation of True Posterior of LDA (Blei et al.)

Note: There is a correction in the above figure. The θ_{dn} in the N plate is in fact z_{dn} - the topic assignment of a word.

The variational approximation in the above figure modifies the original problem in which some of the edges and nodes are removed. The problem in estimating the true posterior was the coupling between θ and β that arises due to the edges between θ, z and w. By dropping these edges and endowing the resulting simplified graphical model with free variational parameters ϕ, γ and λ, we obtain a family of distributions over the latent variables. The variational distribution $q(\theta, z, \beta)$ in which each hidden variable is independent and governed by its own variational parameter ϕ, γ, λ can be represented as a fully factored variational distribution:

$$q(\theta, z, \beta) = \prod_{k=1}^{K} q(\beta_k|\lambda_k) \prod_{d=1}^{D} \left(q(\theta_d|\gamma_d) \prod_{n=1}^{N} q(z_{dn}|\phi_{dn}) \right) \tag{4}$$

The global parameters λ govern the global variables β_k (document-topic distribution) which are independent of the documents. The local parameters $\phi_{d,n}$ govern the local variables

in the n_{th} word context in the d_{th} document (or the topic assignment of the n_{th} word in the d_{th} document - z_{dn}) and the other local parameter γ_d governs the d_{th} document-topic distribution θ_d (document context). The local refers to the document specific context or the context of a word in a document here. Global refers to the topic context (topic distribution over the vocabulary) which is already assumed in the beginning and does not depend on the documents. Variational inference minimizes the Kullback-Leibler (KL) divergence from the variational distribution to the posterior distribution. It maximizes the evidence lower bound (ELBO), a lower bound on the logarithm of the marginal probability of the observations $\log p(w)$. The ELBO is equal to the negative KL divergence up to an additive constant. The ELBO is derived by introducing a distribution over the hidden variables $q(\theta, z, \beta)$ and using Jensens inequality as follows:

$$\begin{aligned}
p(w|\alpha,\eta) &= \log \int\int \sum_z p(w,\beta,\theta,z|\alpha,\eta) d\theta d\beta \\
&= \log \int\int \sum_z p(w,\beta,\theta,z|\alpha,\eta) \frac{q(\theta,z,\beta)}{q(\theta,z,\beta)} d\theta d\beta \\
&\geq \int\int \sum_z q(\theta,z,\beta) \log p(w,\beta,\theta,z|\alpha,\eta) d\theta d\beta - \int\int \sum_z q(\theta,z,\beta) \log q(\theta,z,\beta) d\theta d\beta \\
&= E_q[\log p(w,\beta,\theta,z|\alpha,\eta)] - E_q[\log q(\beta,\theta,z)] \\
\rightarrow p(w|\alpha,\eta) &\geq E_q[\log p(w,\beta,\theta,z|\alpha,\eta)] - E_q[\log q(\beta,\theta,z)] \triangleq \mathcal{L}(q(\gamma,\phi,\lambda;\alpha,\eta))
\end{aligned} \tag{5}$$

which is the lower bound (ELBO) on the log likelihood for an arbitrary variational distribution $q(\theta, z, \beta)$. The $q(\theta, z, \beta)$ is restricted to be in a family that is tractable, one for which the expectations in the ELBO can be efficiently computed and then that member of the family is found that maximizes the ELBO. Finally, the optimized distribution is used as a proxy for the posterior. Solving this maximization problem is equivalent to finding the member of the family that is closest in KL divergence to the posterior:

$$\begin{aligned}
KL(q(\theta,z,\beta|\gamma,\phi,\lambda)||p(\theta,z,\beta|w,\alpha,\eta)) &= E_q[\log q(\beta,\theta,z)] - E_q[\log p(\beta,\theta,z|w,\alpha,\eta)] \\
&= E_q[\log q(\beta,\theta,z)] - E_q[\log p(\beta,\theta,z|\alpha,\eta)] + p(w|\alpha,\eta) \\
&= -\mathcal{L}(q(\gamma,\phi,\lambda;\alpha,\eta)) + constant(\text{does not depend on q})
\end{aligned} \tag{6}$$

This can again be written as:

$$p(w|\alpha,\eta) = \mathcal{L}(q(\gamma,\phi,\lambda;\alpha,\eta)) + KL(q(\theta,z,\beta|\gamma,\phi,\lambda)||p(\theta,z,\beta|w,\alpha,\eta)) \tag{7}$$

where $\mathcal{L}(q(\gamma, \phi, \lambda; \alpha, \eta))$ is the ELBO from (5) and $KL(q(\theta, z, \beta|\gamma, \phi, \lambda)||p(\theta, z, \beta|w, \alpha, \eta))$ is the KL divergence between the variational probability distribution and the true posterior probability. Thus maximizing the ELBO $\mathcal{L}(q(\gamma, \phi, \lambda; \alpha, \eta))$ with respect to ϕ, γ, λ is equivalent to minimizing the KL divergence. Define

$$\begin{aligned}\mathcal{L}(q(\gamma, \phi, \lambda; \alpha, \eta)) &= E_q[\log p(w, \beta, \theta, z|\alpha, \eta)] - E_q[\log q(\beta, \theta, z)] \\ &= E_q[\log p(\theta|\alpha)] + E_q[\log p(\beta|\eta)] + E_q[\log p(z|\theta)] + E_q[\log p(w|z, \beta)] \\ &\quad - E_q[\log q(\theta|\gamma)] - E_q[\log q(\beta|\lambda)] - E_q[\log q(z|\phi)]\end{aligned} \tag{8}$$

In the MFVI, the variational distributions of each variable - $q(\theta|\phi), q(\beta|\lambda)$ and $q(z|\gamma)$ are assumed to be in the same family as the the complete conditionals of the true model - $p(\theta|\alpha)$, $p(\beta|\eta)$ and $p(z|w, \theta, \beta, w)$ respectively, where the complete conditionals from the original model and the variational distributions are discussed below:

Since $\theta \sim Dirichlet(\alpha)$ (prior) and $w_{dn} \sim$ multinomial $(\beta_{z_{dn}})$ (likelihood), the complete conditional of θ (as a function of the parameters underlying true posterior distribution) can be written as:

$$p(\theta_d|z_d) = Dirichlet\left(\alpha + \sum_{n=1}^{N} z_{dn}\right) \tag{9}$$

where z_{dn} is an indicator vector, the k_{th} element of the parameter to this Dirichlet is the sum of hyperparameter α and the number of word assigned to topic k in document d. Therefore variational distribution of θ (as a function of variational parameters) is

$$q(\theta_d) \sim Dirichlet(\gamma_d) \tag{10}$$

where γ_d is a K-dimensional parameter. There is a different variational Dirichlet parameter for each topic allowing different documents to be associated with different topics in different proportions. Similarly, the complete conditional of topic-assignments z_{dn} is a multinomial since the prior is $z_{dn} \sim$ multinomial(θ_d) and the likelihood is $w_{dn} \sim$ multinomial $(\beta_{z_{dn}})$.

$$p(z_{dn} = k|\theta_d, \beta_{1:K}, w_{dn}) \propto \exp(\log \theta_{dk} + \log \beta_{k,w_{dn}}) \tag{11}$$

Therefore, its variational distribution is

$$q(z_{dn}) = \text{multinomial}(\phi_{dn}) \tag{12}$$

where the variational parameter ϕ_{dn} is a point on the $K-1$ simple such that each observed word is endowed with a different variational distribution for its topic assignment, allowing

different words to be associated with different topics. These are the local hidden variables (for each document). There complete conditionals only depend on the other variables in the local context (i.e. the document) and the global variables, they do not depend on variables of other documents. Finally the complete conditional for the topic β_k is also a Dirichlet (just like θ_d),

$$p(\beta_k|z, w) = Dirichlet\left(\eta + \sum_{d=1}^{D}\sum_{n=1}^{N} z_{dn}^{k} w_{dn}\right) \tag{13}$$

since z_{dn} is a multinomial distribution. The v^{th} (in the vocab.) element of the posterior to the Dirichlet for topic k is the sum of the hyperparameter η and the number of times the term v in the unique vocabulary (chosen from the entire corpus) was assigned to topic k. This is a global variable - its complete conditional depends on the words and topic assignments of the entire collection of the words in the corpus. The variational distribution for each topic k is a $V-1$ dimensional Dirichlet,

$$q(\beta_k) = Dirichlet(\lambda_k) \tag{14}$$

This information is summarized in the table below:

Table 2: The Hidden Variables, Complete Conditionals, Variational Parameters and Expected Sufficient Statistics

Var	Type	Conditional	Var. Param.	Relevant Expectations w.r.t q
z_{dn}	Multinomial	$\log\theta_{dk} + \log\beta_{k,w_d n}$	ϕ_{dn}	$E[z_{dn}] = \phi_{dnk}$
θ_d	Dirichlet	$\alpha + \sum_{n=1}^{N} z_{dn}$	γ_d	$E[\log\theta_{dk}] = \Psi(\gamma_{dk}) - \sum_{j=1}^{K}\Psi(\gamma_{dj})$
β_d	Dirichlet	$\eta + \sum_{d=1}^{D}\sum_{n=1}^{N} z_{dnk} w_{dn}$	λ_k	$E[\log\beta_{kw}] = \Psi(\lambda_{kw}) - \sum_{v=1}^{V}\Psi(\lambda_{kv})$

The fourth column of the table 2 above will be used in expanding the expectations in the ELBO equation (8) below. Using the second column of the above table and the information on the variational distribution of the hidden variables, the ELBO in equation (8) (reproduced below) can be expanded in terms of the model parameters α, β and the variational parameters γ, λ, ϕ as follows:

$$
\begin{aligned}
\mathcal{L}(\gamma, \phi, \lambda; \alpha, \eta) &= E_q[\log p(\theta|\alpha)] + E_q[\log p(\beta|\eta)] + E_q[\log p(z|\theta)] \\
&\quad + E_q[\log p(w|z,\beta)] - E_q[\log q(\theta|\gamma)] - E_q[\log q(\beta|\lambda)] - E_q[\log q(z|\phi)] \\
&= \underbrace{\sum_{d=1}^{D} E_q[\log p(\theta_d|\alpha)]}_{T1} + \underbrace{\sum_{k=1}^{K} E_q[\log p(\beta_k|\eta)]}_{T2} + \underbrace{\sum_{d=1}^{D}\sum_{n=1}^{N} E_q[\log p(z_{dn}|\theta_d)]}_{T3} \\
&\quad + \underbrace{\sum_{d=1}^{D}\sum_{n=1}^{N} E_q[\log p(w_{dn}|z_{dn}, \beta_{z_{dn}})]}_{T4} - \underbrace{\sum_{d=1}^{D} E_q[\log q(\theta_d|\gamma_d)]}_{T5} - \underbrace{\sum_{k=1}^{K} E_q[\log q(\beta_k|\lambda_k)]}_{T7} \\
&\quad - \underbrace{\sum_{d=1}^{D}\sum_{n=1}^{N} E_q[\log q(z_{dn}|\phi_{dn})]}_{T7}
\end{aligned}
\tag{15}
$$

For a Dirichlet distribution, since it belongs to exponential family and distributions in this family can be written as $p(\theta|\eta) = h(\theta)\exp\{\eta^T t(\theta) - a(\eta)\}$ where η is the natural parameter, $t(\theta)$ is the sufficient statistic, $h(\theta)$ is the underlying measure, and $a(\eta)$ is the log normalizer and the derivatives of log normalizer are the moments of sufficient statistic, that is, $E_p[t(\theta)] = \partial\alpha/\partial\eta^T$. Using these facts, we can find an analytic expression for the expectation of the logarithm of θ (first expectation in the ELBO above):

$$
E[\ln\theta] = \Psi(\alpha_k) - \Psi\left(\sum_{k=1}^{K}\alpha_k\right) \tag{16}
$$

where $\psi(.)$ is digamma function such that $\frac{d}{dx} ln\Gamma(x)$ and similarly

$$
E[\ln p(\theta|\alpha)] = \ln\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\ln\Gamma(\alpha_k) + \sum_{k=1}^{K}\left(\alpha_k - 1\right)\Psi(\alpha_k) - \Psi\left(\sum_{k=1}^{K}\alpha_k\right) \tag{17}
$$

Using the above identities we simplify each of the terms in the lower-bound of the likelihood:

$$
\begin{aligned}
\mathcal{L}(\gamma,\phi,\lambda;\alpha,\eta) = & \underbrace{\sum_{d=1}^{D}\left[\log\Gamma(\sum_{k=1}^{K}\alpha_k) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) + \sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_i) - \Psi(\sum_{k'=1}^{K}\gamma_k^{'})\right)\right]}_{T1} \\
& \underbrace{\sum_{k=1}^{K}\left[\log\Gamma(\sum_{v=1}^{V}\eta_v) - \sum_{v=1}^{V}\log\Gamma(\eta_v) + \sum_{v=1}^{V}(\eta_v-1)\Psi(\lambda_{zv}) - \Psi(\sum_{v'=1}^{V}\lambda_{zv'})\right]}_{T2} \\
& + \underbrace{\sum_{d=1}^{D}\left[\sum_{n=1}^{N}\sum_{i=1}^{K}1(z_{dn=k})\left(\Psi(\gamma_{dk}) - \Psi\left(\sum_{k'=1}^{K}\gamma_{dk'}\right)\right)\right]}_{T3} \\
& + \underbrace{\sum_{d=1}^{D}\left[\sum_{n=1}^{N}\sum_{k=1}^{K}E_q\left[1(z_{dn=k})1(w_{dn=v})\ln\beta_{kv}\right]\right]}_{T4} \\
& \underbrace{-\sum_{d=1}^{D}\left[\log\Gamma(\sum_{k=1}^{K}\gamma_{dk}) + \sum_{k=1}^{K}\log\Gamma(\gamma_{dk}) - \sum_{k=1}^{K}(\gamma_{dk}-1)\Psi(\gamma_{dk}) - \Psi(\sum_{k'=1}^{K}\gamma_k^{'})\right]}_{T5} \\
& \underbrace{-\sum_{k=1}^{K}\left[\log\Gamma(\sum_{v=1}^{V}\lambda_v) + \sum_{v=1}^{V}\log\Gamma(\lambda_{kv}) - \sum_{v=1}^{V}(\lambda_{kv}-1)\Psi(\lambda_v) - \Psi(\sum_{v'=1}^{V}\lambda_{kv'})\right]}_{T6} \\
& - \underbrace{\sum_{d=1}^{D}\left[\sum_{n=1}^{N}\sum_{i=1}^{k}E_q\left[1(z_{dn}=k)\log(\phi_{dnk})\right]\right]}_{T7}
\end{aligned}
\tag{18}
$$

Further simplifying:

$$
\begin{aligned}
\mathcal{L}(\gamma,\phi,\lambda;\alpha,\eta) = & \underbrace{\sum_{d=1}^{D}\left[\log\Gamma(\sum_{k=1}^{K}\alpha_k) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) + \sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_i) - \Psi(\sum_{k'=1}^{K}\gamma_k^{'})\right)\right]}_{T1} \\
& \underbrace{\sum_{k=1}^{K}\left[\log\Gamma(\sum_{v=1}^{V}\eta_v) - \sum_{v=1}^{V}\log\Gamma(\eta_v) + \sum_{v=1}^{V}(\eta_v-1)\Psi(\lambda_{zv}) - \Psi(\sum_{v'=1}^{V}\lambda_{zv'})\right]}_{T2} \\
& + \underbrace{\sum_{d=1}^{D}\left[\sum_{n=1}^{N}\sum_{i=1}^{K}\phi_{dnk}\left(\Psi(\gamma_{dk}) - \Psi\left(\sum_{k'=1}^{K}\gamma_{dk'}\right)\right)\right]}_{T3} \\
& + \underbrace{\sum_{d=1}^{D}\left[\sum_{n=1}^{N}\sum_{k=1}^{K}\phi_{dnk}\Psi(\lambda_{k,w_{dn}} - \Psi(\sum_{v'}^{V}\lambda_{kv'})\right]}_{T4} \\
& \underbrace{-\sum_{d=1}^{D}\left[\log\Gamma(\sum_{k=1}^{K}\gamma_{dk}) + \sum_{k=1}^{K}\log\Gamma(\gamma_{dk}) - \sum_{k=1}^{K}(\gamma_{dk}-1)\Psi(\gamma_{dk}) - \Psi(\sum_{k'=1}^{K}\gamma_k^{'})\right]}_{T5} \\
& \underbrace{-\sum_{k=1}^{K}\left[\log\Gamma(\sum_{v=1}^{V}\lambda_v) + \sum_{v=1}^{V}\log\Gamma(\lambda_{kv}) - \sum_{v=1}^{V}(\lambda_{kv}-1)\Psi(\lambda_v) - \Psi(\sum_{v'=1}^{V}\lambda_{kv'})\right]}_{T6} \\
& \underbrace{-\sum_{d=1}^{D}\left[\sum_{n=1}^{N}\sum_{i=1}^{k}\phi_{dnk}\log(\phi_{dnk})\right]}_{T7}
\end{aligned}
\tag{19}
$$

So, we finally have

$$
\mathcal{L}(\gamma,\phi,\lambda;\alpha,\eta) = \sum_{d=1}^{D}\log(p(w_d|\alpha,\eta) \tag{20}
$$

That is, the marginal log likelihood of the entire corpus is the sum of marginal log likelihoods of the individual documents. Since the log likelihood cannot be computed tractably, we can maximize the ELBO (obtained in 18) which provides a tractable lower bound on the log likelihood. Thus approximate empirical Bayes estimates for the LDA model can be found via an alternating variational Expectations Maximization (VEM) procedure that maximizes the lower bound with respect to the variational parameters γ, λ, ϕ and then for

fixed values of the variational parameters, maximizes the lower bound with respect to the model hyperparameters α, η. The derivation yields the following iterative algorithm:

1. (E-step): For each document, find the optimizing values of the local variational parameters $(\gamma_d^*, \phi_d^*, d \in D)$. That is:

$$(\gamma_d^*, \phi_d^*) = argmax\ \mathcal{L}(\gamma, \phi, \lambda; \alpha, \eta) \tag{21}$$

The maximization of ELBO with respect to the local variational parameters provides the following updates for γ_d^* and ϕ_d^*:

$$\phi_{dnk} \propto \exp E_q[\log \theta_{dk}] + E_q[\log \beta_k] \tag{22}$$

where the expressions for $E_q[\log \theta_{dk}]$ and $E_q[\log \beta_k]$ are given in the last column of table 2 above [3]. The maximization with respect to the other two parameters γ, λ is unconstrained. Similarly the update for the other document specific local variable γ is:

$$\gamma_{dk} = \alpha_k + \sum_{w=1}^{W} n_{dw} \phi_{dwk} \tag{23}$$

where γ is a variational parameter that corresponds to θ the document-topic distribution and the update for γ looks very similar to the conditional posterior of θ (both have Dirichlet distribution). The n_{dw} is the number of times word w appears in a document and ϕ_{dwk} is the probability of the word w to appear in topic k in document d or the probability weight of topic k corresponding to the word w in document d (obtained above). The γ_{dk} essentially gives the weight of each topic in each document depending on the number of words that correspond to each topic in a document and this information comes from ϕ which is the topic assignment of each word in each document (computed above). Once the topic assignment of each word in each document in the entire corpus is obtained, we move to the maximization step where we compute the update for the corpus specific global variable λ_k which corresponds to β_k, that is, the topic-terms distribution.

[3] The maximization of ELBO with respect to ϕ is a constrained maximization since $\sum_{k=1}^{K} \phi_{dnk} = 1$. Since ϕ is a variational parameter corresponding to z which is topic assignment, the sum of probability of a word appearing in all the topics in a document must sum to 1.

2. (M-step): Now we update the Dirichlet for each topic (in the corpus):

$$\lambda_{kv} = \eta_v + \sum_{d=1}^{D} n_{dw}\phi_{dwk} \tag{24}$$

where $\sum_{d=1}^{D} n_{dw}\phi_{dwk}$ is the sum of the number of times word w appears in the entire corpus times the probability weight of topic k corresponding to word w in the entire corpus.

In the M-step, we can also find updates of the model hyperparameters α and η by maximizing the ELBO with respect to these parameters. The first derivative of the log likelihood with respect to α_k is

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = D\bigg(\Psi\bigg(\sum_{k'=1}^{K} \alpha_k^{'}\bigg) - \Psi(\alpha_k)\bigg) + \sum_{d=1}^{D}\bigg(\Psi(\gamma_{dk}) - \Psi\bigg(\sum_{k'=1}^{K} \alpha_k^{'}\bigg)\bigg) \tag{25}$$

Since this derivative depends on $k^{'}$ where $k \neq k^{'}$, an iterative linear time Newton-Raphson algorithm is used to find the maximum value of α. The Newton-Raphson optimization technique finds a stationary point of a function by iterating $\alpha_{new} = \alpha_{old} - H(\alpha_{old})^{-1}g(\alpha_{old})$ where $H(\alpha)$ and $g(\alpha)$ are the Hessian matrix and the gradient respectively at point α where Hessian is of the form:

$$\frac{\partial \mathcal{L}}{\partial \alpha_k \alpha_j} = \delta(k,j)D\Psi'(\alpha_k) - \Psi'(\sum_{j=1}^{k} \alpha_j) \tag{26}$$

The estimate of η can be similarly found. The updates of these hyperparameters depends on a specific problem. A lot of existing papers fix these parameters at 0.01 or at $1/K$ where K is the total number of topics instead of tuning these parameters. Several other values are also used depending on the type of corpus and information sought from the corpus. A lot of these papers analyze the topics generated from these topic models and if the topics don't make sense, there parameters are accordingly tuned.

Again, lets recall the MFVI of the true posterior:

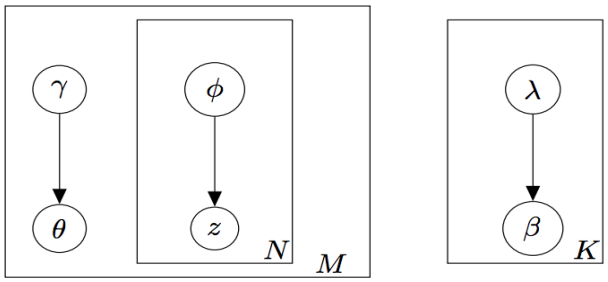


Figure 9: Plate model for Mean Field Approximation to LDA

The algorithm for batch MFVI (based on figure 9 above) can be written as follows :

1. Initialize λ randomly (global param.)
2. Repeat:
3. For a single document d:
4. Repeat:
5. Update ϕ_{dwk} using equation 21 for $w \in 1, ..., W, k \in 1, ..., K$ (local param.)
6. Update γ_d using equation 22 (local param.)
7. until local parameters ϕ_{dn} and γ_d converge:
8. For $k \in 1, ..., K$, update the global variational parameters: λ_k using equation 23.
9. Until the ELBO converges.

One of the drawbacks of this algorithm is as D (the number of documents) increases, inference becomes increasingly inefficient because step 8 requires processing the entire dataset before updating λ_k in each iteration which is computationally expensive and inefficient. One of the variations of this algorithm which is also called the Stochastic variational inference (SVI) that draws a single document uniformly from the corpus in each iteration and in order to update the global parameter λ, it replicates the update from a single document D times instead of waiting to process the entire corpus before updating λ. The algorithm for SVI can be written as follows :

1. Initialize λ randomly (global param.)
2. for t = 1, . . . , T
3. Sample a document d uniformly from the dataset.

4. Repeat:

5. Update ϕ_{dwk} using equation 21 for $w \in 1, ..., W,\ k \in 1, ..., K$ (local param.)

6. Update γ_d using equation 22 (local param.)

7. until local parameters ϕ_{dn} and γ_d converge:

8. For $k \in 1, ..., K$, update the global variational parameters: λ_k using the following equation which is a slight variation of equation 23.

$$\hat{\lambda_k} = \eta_k + D \sum_{w=1}^{W} n_{dw} \phi_{dwk} \tag{27}$$

where this step assumes that document d is replicated D times since λ is a global parameter (based on entire corpus)

9. Set $\lambda = (1 - \rho_t)\lambda + \rho_t \hat{\lambda}$

10. Until the ELBO converges.

11. The update of hyperparameters (α, η) is slightly different in case of SVI

$$\alpha_{new} = \alpha_{old} - \rho_t H(\alpha_{old})^{-1} g(\alpha_{old}) \tag{28}$$

where the update for λ is a weighted sum of its previous value and the proposed estimate. $\rho_t = (\tau_0 + t)^{-\kappa}$ is the adaptive learning rate (weight on new λ), $\kappa \in (0.5, 1]$ controls the rate at which old values of λ are forgotten and τ_0 slows the early iterations of the algorithm. Another variation of this algorithm is that instead of sampling one document from the corpus, a chunk of documents (S) can be sampled from the entire corpus which produces slightly better estimates since the parameters will be based on a chunk of documents instead of just one document in each iteration. This model is estimated using Python's Gensim library which is widely used for this LDA-VEM models. We'll discuss the results and some of the issues faced in the results section. The next section discusses the model selection methods for the LDA-VEM models.

3.1.1 Model Selection Methods for LDA-VEM Models

Perplexity is the most typical evaluation of LDA models. This provides a convenient measure to judge how good a given topic model is. These is a model selection method and is generally used in selecting the hyperparameters (in case these are fixed and not estimated in the model)

such a α, η and number of topics - K. Perplexity is a measurement of how well a probability distribution or probability model predicts a sample. It is often used to evaluate the models on held-out (test data) data and is equivalent to the geometric mean of the inverse marginal probability of each word in the held-out set of documents (test data or unobserved dataset). Better models have lower perplexity (or higher likelihood) suggesting lower model uncertainty about the unobserved document.

$$perplexity(w^{test}, \lambda, \alpha) = \exp\left(- \frac{\sum_{d=1}^{D} \log p(w_d^{test}|\alpha, \eta)}{\sum_{d=1}^{D} \sum_{n=1}^{N} w_{dn}^{test}} \right)$$

where w_d^{test} denotes the vector of word for the d^{th} document and w_{dn} is the n^{th} word in document d. Since we cannot directly compute $\log p(w_d^{test}|\alpha, \eta)$, we use a lower bound on perplexity as a proxy (similar to what we do in MFVI algorithm above):

$$perplexity(w^{test}, \lambda, \alpha) \leq \exp\left\{ - \left(\sum_{d=1}^{D} E_q[\log p(w_d^{test}\theta_d, z_d|\alpha, \eta)] - E_q[\log q(\theta_d, z_d)] \right) \left(\sum_{d=1}^{D} \sum_{n=1}^{N} w_{dn}^{test} \right) \right\}$$

The per document parameters γ_d and ϕ_d for the variational distributions $q(\theta_d)$ and $q(z_d)$ are fit using the E step in the algorithms given above. The topics λ are fit to training set of documents and are then held fixed while fitting the model on the test (unobserved or held-out) data. This experiment can be helpful in tuning the hyperparameters such as α and η and the number of topics K (if there is little prior information available about their optimal values (especially for K which is generally assumed to be fixed because it cannot be estimated from the model unlike α and η)).

The overall idea is to divide the corpus into training and test set of documents and then using the training set the model is estimated topics are obtained (including the top words in a topic). Then using this estimated model and the topics obtained, the trained model is then fitted on the test set of documents and is then checked for how well the trained model predicts the test data or if it can correctly label an article (based on its content) under the appropriate topic. The size of the corpus in this case will be an issue since we only have 1399 documents. A large corpus of documents is generally needed to train such machine learning based models. More details on this are included in the results section.

3.2 Gibbs Sampling for LDA

Another general approach for posterior inference with intractable distributions is to appeal to Markov chain Monte Carlo methods. Here, the goal is to produce samples from a distri-

bution that is hard to sample from directly. This is done using the 'topicmodels' package in R written by Grun and Hornik (2011) which includes interfaces to two algorithms for fitting topic models: the variational expectation-maximization (MFVI) algorithm provided by David M. Blei and co-authors (2003) and an algorithm using Gibbs sampling by Xuan-Hieu Phan, Nguyen and Horiguchi (2008). This package is mostly used to implement Gibbs sampling despite the fact that several other topic models are included in this library. The accuracy of other topic models (apart from Gibbs sampling) in this package is still debatable, although, the Gibbs sampling method also experiences convergence issues but that is mostly the problem of the method and not necessary an issue with the package.

In Gibbs sampling for LDA, latent variables in the graphical model are sampled iteratively given the rest based on the conditional distribution. A more commonly applied approach is the collapsed Gibbs sampling, where we do not have to sample all the latent parameters involved, as θ and β can be integrated out (or collapsed). From above, we know that the document-topic proportions $\theta_d \sim Dirichlet(\alpha)$ and topic assignments of the words in a document $z_{dn} \sim Multinomial(\theta_d)$, where Dirichlet is a conjugate prior for Multinomial, the posterior distribution of $\theta_d \sim Dirichlet(\alpha_k + n_{dk})$ where n_{dk} is the number of times document d use topic k. This implies:

$$\theta_{dk} = \frac{n_{dk} + \alpha_k}{\sum_{k=1}^{K} n_{dk} + \alpha_k} \tag{29}$$

The exact same holds for β_k which is topic-terms distribution where $\beta_k \sim Dirichlet(\eta_{w_{dn}})$ and $w_{dn} \sim Multinomial(\beta_{z_{dn}})$, the posterior distribution of β is $\beta_{kw} \sim Dirichlet(\eta_w + n_{kw})$ where n_{kw} is the number of times word w is assigned to topic k (over all documents in the corpus). Therefore,

$$\beta_{kw} = \frac{n_{kw} + \eta_w}{\sum_{v=1}^{V} n_{kv} + \eta_v} \tag{30}$$

where V is the vocabulary developed from the entire corpus and v is a word in the vocabulary. One thing to notice here is that the estimates of both θ and β above depend only on the topic assignments of the words z. Once we know the topic assignment of each word in all the documents in the corpus z_{dn}, we can easily get the document-topic distribution θ_{dk} and the topic-word distribution β_{kw}. Therefore, we can only focus on inferring the latent variable z and the other latent variables can be computed directly from z which is essentially what collapsed Gibbs sampling does. It allows building a Gibbs sampler only over z by

integrating out the θ and β from the joint distribution over the observed and hidden variables.

Once again, the joint distribution (of observed and hidden variables (reproduced from equation (1) above)) corresponding to the process displayed in the figure 4 above can be represented as follows:

$$p(\theta, \beta, z, w|\alpha, \eta) = \prod_{k=1}^{K} p(\beta_k|\eta) \prod_{d=1}^{D} p(\theta_d|\alpha) \Big(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{dn}|\beta_{1:K}, z_{d,n}) \Big) \tag{31}$$

Integrating out θ and β, we get,

$$p(z, w|\alpha, \eta) \int_{\beta} \int_{\theta} \underbrace{\prod_{k=1}^{K} p(\beta_k|\eta)}_{Dirichlet(\eta)} \underbrace{\prod_{d=1}^{D} p(\theta_d|\alpha)}_{Dirichlet(\alpha)} \Big(\prod_{n=1}^{N} \underbrace{p(z_{d,n}|\theta_d)}_{multinomial(\theta_d)} \underbrace{p(w_{dn}|\beta_{1:K}, z_{d,n})}_{multinomial(\beta_{z_{dn}})} \Big) d\theta d\beta \tag{32}$$

Solving this, we get,

$$p(z, w|\alpha, \eta) \propto \prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma(n_{dk} + \alpha_k)}{\Gamma(\sum_{k=1}^{K}(n_{dk} + \alpha_k))} \times \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(n_{kv} + \eta_v)}{\Gamma(\sum_{k=1}^{K}(n_{kv} + \eta_v))} \tag{33}$$

Again, n_{dk} is the number of times topic k has been chosen for words in document d and n_{kv} is the number of times topic k has been chosen for word v in the corpus. Rewriting this by separating z into z_{dn} - topic assignment of n^{th} word in document d and the topic assignments of all other words z_{-dn} as follows:

$$p(z_{dn} = k, z_{-dn}, w|\alpha, \eta) \propto \prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma(n_{dk} + \alpha_k)}{\Gamma(\sum_{k=1}^{K}(n_{dk} + \alpha_k))} \times \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(n_{kv} + \eta_v)}{\Gamma(\sum_{v=1}^{V}(n_{kv} + \eta_v))} \tag{34}$$

where z_{-dn} is the topic assignment of all the words excluding the topic assignment of the n^{th} word in document d. Now, collecting only those terms that depend on the specific position dn (word n in document d) that we are sampling for:

$$p(z_{dn} = k, z_{-dn}, w|\alpha, \eta) \propto \frac{\prod_{k=1}^{K} \Gamma(n_{dk} + \alpha_k)}{\Gamma(\sum_{k=1}^{K}(n_{dk} + \alpha_k))} \times \prod_{k=1}^{K} \frac{\Gamma(n_{kv} + \eta_v)}{\Gamma(\sum_{v=1}^{V}(n_{kv} + \eta_v))} \tag{35}$$

Let,

$$n_{dk}^{-dn} = \begin{cases} n_{dk} - 1 & \text{if } z_{dn} = k \\ n_{dk}, & \text{otherwise} \end{cases} \tag{36}$$

This means that if the topic assignment of n^{th} word in document d is k, that is if $z_{dn} = k$, then reduce n_{dk}^{-dn}, which is the number of times topic k in document d excluding the current topic assignment z_{dn}, by 1. Otherwise, the counts will not change. Similarly let,

$$n_{kv}^{-dn} = \begin{cases} n_{kv} - 1 & \text{if } z_{dn} = k \text{ and } w_{dn} = v \\ n_{kv}, & \text{otherwise} \end{cases} \tag{37}$$

That is if the n^{th} word in document d w_{dn} is the v^{th} word in the vocabulary and its topic assignment is k, then the number of times word v in the vocabulary (over the entire corpus) will be assigned to topic k excluding the current assignment, n_{kv}^{-dn} will be reduced by 1. Otherwise, the counts will not change.

Thus, using (33) and (34) on the right hand side of (32), we get

$$\begin{aligned} p(z_{dn} = k, z_{-dn}, w|\alpha, \eta) \propto & \frac{\prod_{i=1;i\neq k}^{K} \Gamma(n_{di}^{-dn} + \alpha_i)\Gamma(n_{dk}^{dn} + \alpha_k)}{\Gamma(\sum_{i=1}^{K}(n_{di} + \alpha_i))} \times \\ & \prod_{i=1;i\neq k}^{K} \frac{\Gamma(n_{i,wdn}^{-dn} + \eta_{w_{dn}})}{\Gamma(\sum_{v=1}^{V}(n_{iv}^{-dn} + \eta_v))} \frac{\Gamma(n_{k,wdn}^{dn} + \eta_{w_{dn}})}{\Gamma(\sum_{v=1}^{V}(n_{kv}^{dn} + \eta_v))} \end{aligned} \tag{38}$$

Now, substituting in the values of $n_{dk}^{dn} = n_{dk}$ and $(n_{kv}^{dn} = n_{kv})$ from equation (33) and (34) in (35) and using the fact that $\Gamma(1 + x) = x\Gamma(x)$, we get,

$$\begin{aligned} p(z_{dn} = k, z_{-dn}, w|\alpha, \eta) \propto & \frac{\prod_{i=1;i\neq k}^{K} \Gamma(n_{di}^{-dn} + \alpha_i)\Gamma(1 + n_{dk}^{-dn} + \alpha_k)}{\Gamma(1 + \sum_{i=1}^{K}(n_{di}^{-dn} + \alpha_i))} \times \\ & \prod_{i=1;i\neq k}^{K} \frac{\Gamma(n_{i,wdn}^{-dn} + \eta_{w_{dn}})}{\Gamma(\sum_{v=1}^{V}(1 + n_{iv}^{-dn} + \eta_v))} \frac{\Gamma(1 + n_{k,wdn}^{dn} + \eta_{w_{dn}})}{\Gamma(1 + (\sum_{v=1}^{V} n_{kv}^{-dn} + \eta_v))} \\ = & \frac{\prod_{i=1;i\neq k}^{K} \Gamma(n_{di}^{-dn} + \alpha_i)\Gamma(n_{dk}^{-dn} + \alpha_k)(n_{dk}^{-dn} + \alpha_k)}{\Gamma(\sum_{i=1}^{K}(n_{di}^{-dn} + \alpha_i))(\sum_{i=1}^{K}(n_{di}^{-dn} + \alpha_i))} \times \\ & \prod_{i=1}^{K} \frac{\Gamma(n_{i,wdn}^{-dn} + \eta_{w_{dn}})}{\Gamma(\sum_{v=1}^{V}(n_{iv}^{-dn} + \eta_v))} \frac{\Gamma(n_{k,wdn}^{-dn} + \eta_{w_{dn}})(n_{k,wdn}^{dn} + \eta_{w_{dn}})}{\Gamma(\sum_{v=1}^{V} n_{kv}^{-dn} + \eta_v)(\sum_{v=1}^{V} n_{kv}^{-dn} + \eta_v))} \end{aligned} \tag{39}$$

Simplifying, we get,

$$p(z_{dn}=k,z_{-dn},w|\alpha,\eta) \propto \frac{\prod_{i=1}^{K}\Gamma(n_{di}^{-dn}+\alpha_i)(n_{dk}^{-dn}+\alpha_k)}{\Gamma(\sum_{i=1}^{K}(n_{di}^{-dn}+\alpha_i))(\sum_{i=1}^{K}(n_{di}^{-dn}+\alpha_i))}\times \prod_{i=1;i\neq k}^{K}\frac{\Gamma(n_{i,wdn}^{-dn}+\eta_{w_{dn}})}{\Gamma(\sum_{v=1}^{V}(n_{iv}^{-dn}+\eta_v))}\frac{(n_{k,wdn}^{-dn}+\eta_{w_{dn}})}{(\sum_{v=1}^{V}n_{kv}^{-dn}+\eta_v))} \tag{40}$$

Now, collecting only those terms that depend on the current assignment $z_{dn}=k$, we get the following sampling probability:

$$p(z_{dn}=k|z_{-dn},w,\alpha,\eta) \propto \frac{(n_{dk}^{-dn}+\alpha_k)}{(\sum_{i=1}^{K}(n_{di}^{-dn}+\alpha_i))}\times\frac{(n_{k,wdn}^{-dn}+\eta_{w_{dn}})}{(\sum_{v=1}^{V}n_{kv}^{-dn}+\eta_v))} \tag{41}$$

Given the above conditional posterior distribution of the current topic assignment of a word in a document given the topic assignment of all other words in a document, we can implement the Gibbs sampling algorithm as follows:

1. Initialize the topic assignments of all the words in all the documents in the corpus.
2. These topic assignments gives us initial but incorrect counts of the number of times a topic is seen in a document n_{dk} and the number of times a word in the vocabulary is assigned a topic k n_{kv}.
3. Assuming that these topic assignment are correct,
4. for each word n in document d,
5. reassign a new topic to w_{dn} or resample each z_{dn} - topic assignment of word w_{dn}
6. subtract one from the entries corresponding to document d, word n and old topic assignment of word w_{dn} from n_{dk} and n_{kv}.
7. Assign that topic k to word w_{dn} at which conditional posterior is maximized.
8. Update the counts n_{dk} and n_{kw} by adding one to the counts that correspond to the entries associated with the new assignment of z_{dn}
9. Return final n_{dk}, n_{kw}

   Once these counts are obtained, we can get the estimates of θ and β using (26) and (27) above.

3.2.1 Topic Selection for Gibbs LDA

Perplexity is also used for model selection in Gibbs LDA except that the computing the likelihood is more straightforward in this case. If the model is fitted using Gibbs sampling the likelihood is determined for the perplexity using

$$perplexity(w^{test}) = \exp\left(-\frac{1}{N} \log p(w_d^{test}\right)$$

where w_d^{test} denotes the vector of word for the d^{th} document and w_{dn} is the n^{th} word in document d and

$$\log(p(w^{test})) = \sum_{d=1}^{D}\sum_{n=1}^{N} w_{dn}^{test} \log\left[\sum_{k=1}^{K} \theta_{dk}\beta_{kw}\right]$$

In this experiment, the hyperparameters α and η and the number of topics K are fixed as in the original model. Griffith and Steyvers (2004) suggest a value of $\frac{50}{K}$ for α and 0.1 for η. So the values of α and K (any other value can also be assumed) is related in their model while θ_{dk} and β_{kw} are estimated from the model. Any other value for these parameters can also be chosen.

4 Estimation and Results

There are two libraries used in the estimation of these topic models. The Python's Gensim library estimates the batch and the online version of the Mean field Variational Inference LDA model using the Variational Expectation Maximization algorithm (VEM) discussed above. The other one is R's topicmodels library which is widely used for estimating Gibbs sampling LDA model. This also allows estimation of MFVI-LDA using the VEM algorithm but it doesn't appear to be very accurate. I mainly focus on the results from Python Gensim's LDA-VEM (batch LDA) and R topicmodels' Gibbs sampling. In addition I'll also present the topics from Python Gensim's LDA-VEM (online stochastic LDA) and R 'topicmodels' LDA-VEM for comparison purposes. The next section discusses the results. The results presented are based on the full corpus of 1399 documents (since its a small corpus and leaving out documents will not give distinct topics), however, for perplexity, I have divided the corpus into training and test data in order to assess the predictive accuracy of the models.

4.1 Results of MFVI LDA Topic Model

The following parameters were fed into the Gensim LDA to estimate the model using the full corpus (batch LDA) using the Mean Field Variational Inference (MFVI):

```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
        id2word=id2word, ## (Document-term matrix)
        num_topics=5, ## (No. of topics - fixed)
        random_state=100, # (seed)
        update_every=0, ## (0 for batch LDA (full corpus passed
                        #in batch LDA))
        #chunksize=, ## (chunksize is required for online LDA)
        passes=10, ## (repeated training to improve the model
        #(higher the better))
        eval_every=10, ## (for perplexity - calcuated every
        #10 documents (lower the better))
        alpha='auto', ## ((hyperparameter for theta (doc-topic dist.)
        #learn from the model (preferred choice in literature)
        # offers different alpha for different topics (can be
        # fixed to symmetric alpha - same value for all topics))
        eta = 0.1, #(hyperparameter for beta (topic-terms dist.)
        #fix this to some number. Learning from the model makes
        #convergence difficult.Although changing the value of eta
        #from 0.1 to 0.01 to 0.2 doesn't really change the output.
        #It changes the probability of some words but the overall
        #composition of words in a topic remains more or less same.)
        iterations=200, ## (maximum no. of iterations)
        gamma_threshold = 0.00001, ## (to achieve convergence of ELBO
        #to approximate the true posterior)
        per_word_topics=True) ## (print topics)
```

In this batch LDA, the whole corpus is passed into the model at once and the model was trained 10 times (see the parameter *passes* = 10 in the above vode) to produce sensible estimates (10 may be less for a large corpus but should be sufficient for a small corpus). There is not enough information available on the relationship between optimal number of training and corpus size). This model assumes that hyperparameter for β (also understood as 'concentration parameter' or a regularizer) is fixed at 0.1 (see $\eta = 0.1$ above) which controls the prior concentration of words in a topic. A small value for η makes it harder for words to appear in a topic. Although, changing the value of η to other values close to 0.1 doesn't really change the overall topics except for changing the probability of some words which also implies that Gensim offers stable results. Allowing the model to estimate the

value of η produces incorrect topics (some online discussions also argue that the value of η should not be estimated from the model in the Python Gensim library). The value of the hyperparameter of θ - document-topic distribution$-\alpha$ is, however, learned from the model. I allow for different topics to have different values of α ($\alpha = `auto'$ in the above code) instead of keeping them symmetric across topics. This can also be changed to symmetric α which allows for same value for all the topics (The topicmodels library in R forces symmetric α by default even when it is estimated from the model).[4].

In order to decide the number of topics that should be chosen for this model, I estimated perplexity for this model for different number of topics (which is evaluated every 10 documents (see the parameter $evalevery = 10$ in the code above. This can be changed to a smaller number as well but the computation time increases much more as this number becomes smaller.). Perplexity is monotonically decreasing in the likelihood of the test data, and is algebraically equivalent to the inverse of the geometric mean per-word likelihood. A lower perplexity score indicates better generalization performance (Blei et al., 2003).

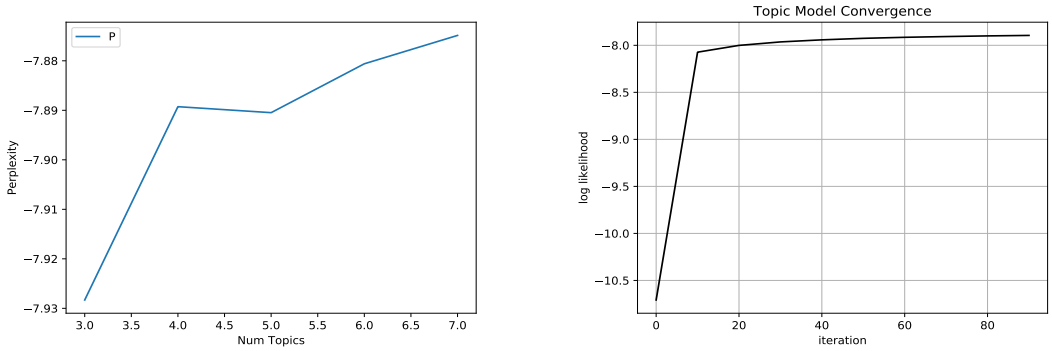


Figure 10: Perplexity Estimates for different Models (k=3,...,7) (left) & Approximate Log-likelihood (or ELBO) for the Model with k=5 (right)

The figure 10 (left) above shows that perplexity is the lowest for when the number of topics is 3. It increases when $K = 4$, slightly decreases at $K = 5$ and then continues to increase. I chose k=5 because fixing k at 3 combines topics 1 & 5 and some words from topics 2,3 & 4 and creates one more topic which is difficult to label. Again, a large corpus may provide more information about the relevant number of topics. Given $K = 5$, the figure 10 (right) provides the convergence of the log likelihood or the approximate Evidence Lower Bound

[4]A complete grid search over different values parameters and different iterations might make more sense as is done in most of this literature to find out the relevant parameters for these models. However, such grid search can be computationally very expensive and may take days to complete.

(ELBO - $(\ln(p(w|\alpha,\eta)) \sim E_q[\log p(w,\beta,\theta,z|\alpha,\eta)] - E_q[\log q(\beta,\theta,z)])$) of the variational distribution which approximates the log likelihood of the true model (The log-likelihood shows almost similar convergence for $K = 3, 4$).

I also divided the data into training corpus of size 1200 documents and a testing corpus of 199 documents to assess the optimal number of topics. The results do not appear to be correct. The figure 11 below provides estimates of perplexity for different number of topics for training and testing data.

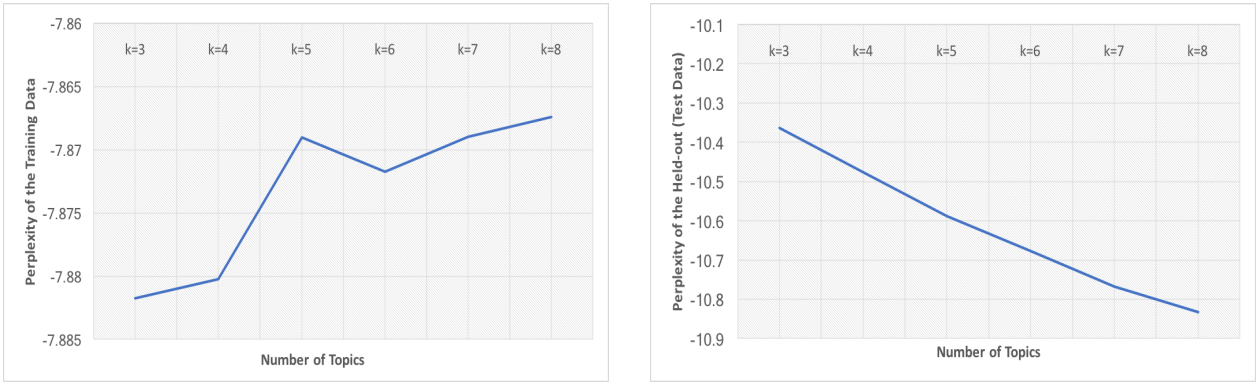


Figure 11: Perplexity Estimates for Training Corpus (k=3,...,8) (left) & Perplexity Estimates for Testing Corpus (k=3,...,8) (right)

From figure (11) left, the estimates of perplexity over training data seems to show that $K \leq 4$ might be more appropriate whereas the testing data (figure 11 (right)) shows that an even larger number of topics might be admissible (since the perplexity is continuously declining) which does not make much sense for only 199 documents. This reflects the issues with the predictive capacity of the model [6]. A larger corpus would be more helpful in analyzing the accuracy of these results. Training of a model on such a small corpus of only 1200 documents may not yield good estimates.

For now, assuming $K = 5$, the estimates of optimized α (which controls the sparsity of topics in the document distribution θ) for each of the topics as estimated by the model are in the table 3 below:

[6] Highlighting the problems using perplexity as a method for model selection, Zhao, Perkins, Liu, Ding and Zou (2015) argue that since the log-likelihood of the LDA model is non-convex, different initial parameters in approximate algorithms, such as Laplace approximation, variational approximation and MCMC, will lead to distinct local maximums. With different random seeds in MCMC or different initial parameters in variational inference approach, the approximate optimizing solutions to LDA may converge to a different local optimal point for the same dataset.

Table 3: Estimates of Hyperparameter α of the Document-Topic Distribution (θ)

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
0.0387	0.1091	0.04346	0.11706	0.05767

All the topics have a very low value of α which indicates a very high probability density at the corners/edges of the $K-1$ topic simplex in the figure 4 above, that is, each document involves very few topics. This should allow easy classification of documents under topics (The value of α estimated using R topicmodels library provides an estimate of 0.08 for all the topics (since it estimates a symmetric α) which is not too off from the estimates in the table 3). I have also estimated this model using a very high value of α (=10 (because Griffith and Steywers (2004) suggest a value of $\alpha = \frac{50}{K}$ which is equal to 5 in our case (although their suggestion is specific to Gibbs sampling model))) and also a large value of η that allows a document to be associated with a large number of topics. The resulting topics from this model made little sense. Some of the topics were indistinguishable from each other because of excessive repetition of similar words. The small size of corpus in this case could be an issue.

The results presented below are different from those presented in the last writeup, although, there is no major change in terms of the topics-term distribution or document-topic distribution. Based on this LDA model, the table 4 below presents five topics with top 50 words in order of their probability of occurrence in a topic:

Table 4: Top 50 Keywords in 5 Topics in LDA MFVI (batch LDA) Model

Index	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	stock	tax	bank	rate	dollar
2	market	inflation	president	bank	gold
3	price	president	chairman	fed	company
4	gold	economic	volcker	money	market
5	share	cut	hunt	reserve	price
6	volume	carter	federal	market	chryslar
7	inflation	reagen	reserve	federal	rate
8	commodity	federal	silver	credit	bank
9	rate	policy	fed	volcker	currency
10	industrial	administration	board	supply	million
11	average	budget	banker	growth	board
12	president	rate	foreign	inflation	west
13	exchange	volcker	year	policy	official
14	issue	price	house	monetary	german
15	oil	billion	market	fund	exchange
16	dow	chairman	american	price	federal
17	analyst	house	international	bond	oil
18	rose	government	loan	term	ounce
19	economic	business	company	current	volcker
20	point	economy	monetary	economy	foreign
21	silver	program	committee	increase	billion
22	high	reverse	treasury	billion	loan
23	standard	spending	secretary	economy	mark
24	fell	congress	banking	increase	american
25	company	increase	member	billion	current
26	recession	current	current	economist	european
27	american	official	future	deposit	world
28	active	million	million	committee	president
29	bank	year	world	control	chairman
30	federal	money	executive	high	international
31	investor	plan	chief	target	reserve
32	board	oil	country	president	inflation
33	cent	recession	business	economic	plan
34	money	fed	state	account	government
35	chairman	monetary	rate	account	meeting
36	government	fiscal	vice	short	london
37	current	deficit	exchange	saving	country
38	future	american	solomon	recession	trading
39	reserve	white	financial	business	miller
40	volcker	energy	central	month	treasury
41	metal	control	carter	million	sale
42	trader	secretary	government	prime	dealer
43	jones	supply	national	security	fell
44	decline	cost	official	higher	economic
45	policy	board	brother	member	franc
46	declining	nation	staff	banking	high
47	block	committee	policy	banker	central
48	dollar	high	problem	institution	monetary
49	traded	growth	position	long	carter
50	carter	world	meeting	change	germany

The topic 2, 3 and 4 appear to be topics of our interest as the keywords in these topics relate to Volcker, short term interest rates, inflation, appointment to the federal reserve board, oil

prices etc. The figure below presents a graphical representation of these 5 topics. Topics 2, 3 and 4 together constitute about 79% (in the last write up these topics constituted about 66% articles of the total corpus) of the total documents present in the entire corpus. All the topics are different from each other except for a slight overlap between topic 2 and 5. However, when I estimate the online (stochastic) Variational Inference model with 5 topics, the topics appear to overlap each other a lot more than batch variational inference model (see figure 31).

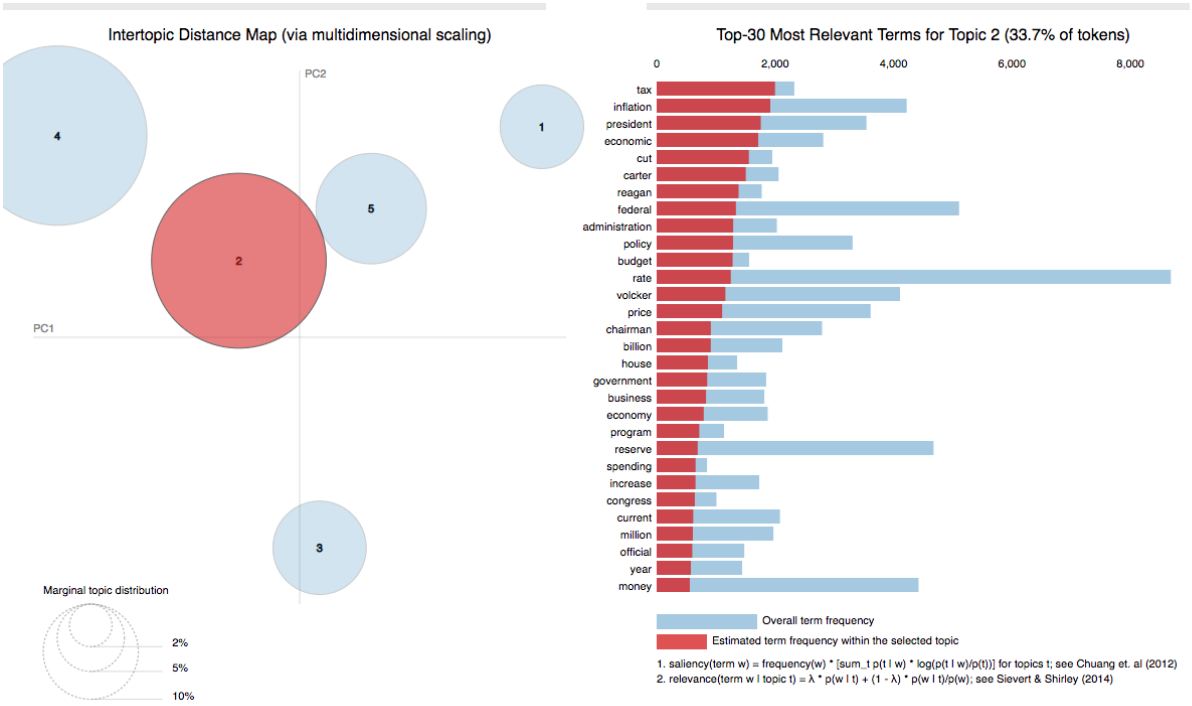


Figure 12: LDA Model Topics

Using the documents covered under topics 2,3 and 4, an index of a measure of expectations regarding Volcker's appointment and expectations of interest rates can be created following Baker, Bloom and Davis (BBD, 2016). The steps to create the index based on BBD (2016) are as follows:

- Scale the raw counts by the total number of articles in the same newspaper and month.
- Standardize each monthly newspaper-level series to unit standard deviation from 1975 to 1981 and then average across 2 papers by month.
- Finally, normalize the 2-paper series to a mean of 100 from 1997 to 1981.

The figure 13 below presents the index created from the documents that belong to topics 2, 3 and 4 in the LDA model along with the aggregate index (index of raw counts without using any topic model created using steps in BBD) created from all the articles collected from the newspapers following the steps mentioned above.

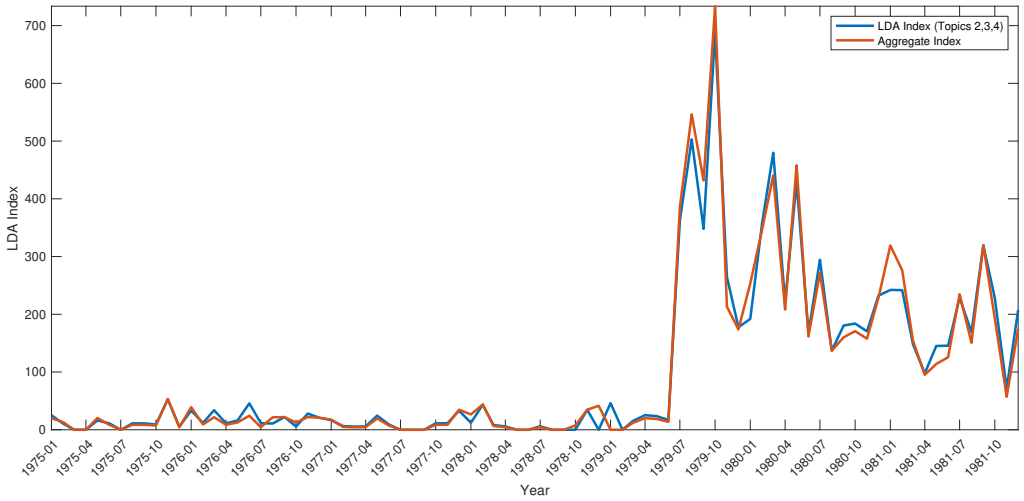


Figure 13: Monthly LDA Index (Topics 2,3, and 4) and Aggregate Index (based on all topics)

This figure shows that there is not much difference between the index created using topics 2,3, and 4 of the LDA model and the aggregate index given that these 3 topics together constitute a large fraction (about 79%) of the total number of documents/articles. While there are small movements in the index before June, 1979, it was in late July, 1979 when the announcements regarding the appointment of Volcker to the Fed were made that we see a massive jump in both indices. The figure 14 below presents the LDA index (topics 2,3, and 4) along with the fed funds rate and the annualized CPI inflation to compare the movements of the index with interest rate and inflation.

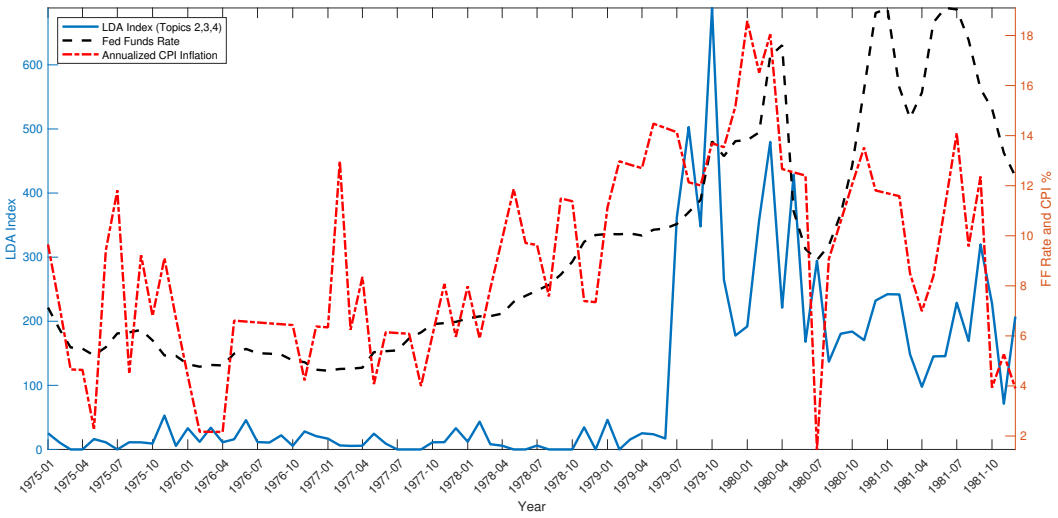


Figure 14: LDA Index (Topics 2,3, and 4) with Fed Funds Rate and Annualized CPI Inflation

There appears to be some correlation between the index and the fed funds rate and little correlation between the index and CPI inflation after late 1979 and the correlation is positive. It's hard to say if the resulting index can act as a measure of interest rate or inflation expectations. A longer time series along with further processing of text to include bigrams (compound words) might allow this index to capture more information.

I have also plotted LDA index for topic 5 with some of the keywords - "dollar, gold, price, market, exchange, currency, ounce, west, german, trading, oil, exchange " which appears to be a topic about international exchange rates, commodities and trade along with fed funds rate and CPI inflation to see if this index performs better in capturing information about either inflation or interest rates. This topic (topic 5) constitutes about 13.5% of the total number of news articles. Figure 15 below presents keywords of topic 5 along with percentage of documents captured by topic 5.

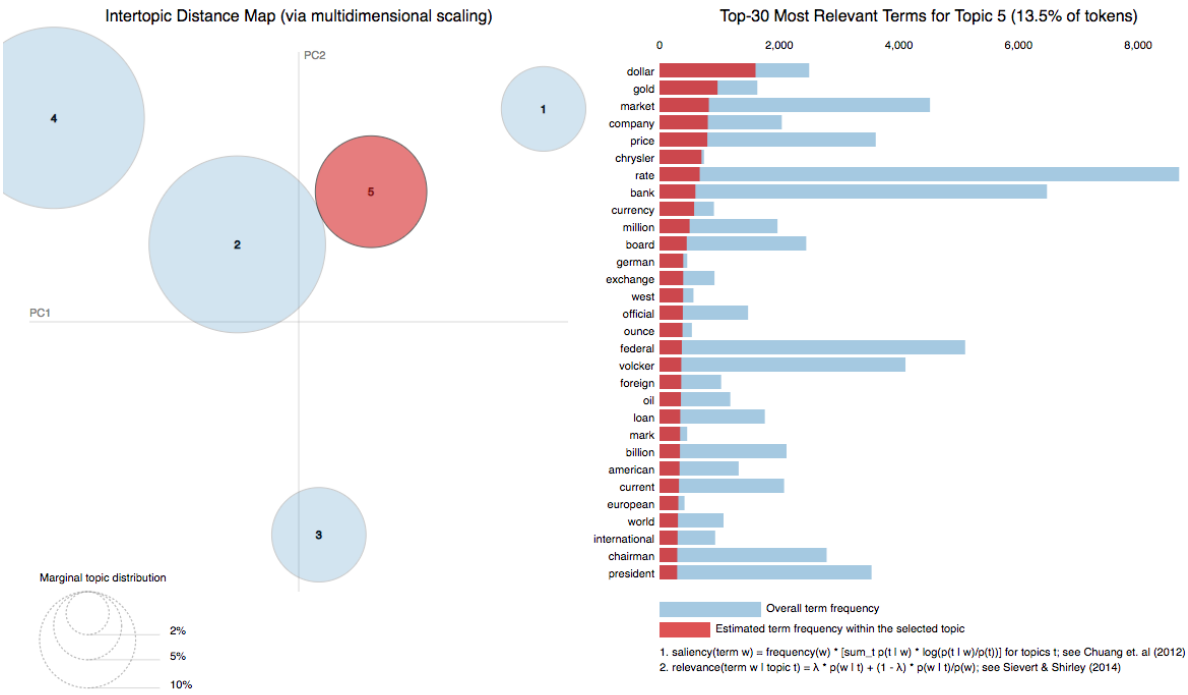


Figure 15: LDA Model Topic (Topic 5)

Figure 16 (top panel) plots LDA index based on topic 5 against LDA index based on topics 2,3 and 4 (reproduced from figure 13 above) and the bottom panel plots LDA index for topic 5 along with annualized CPI inflation and fed funds rate.

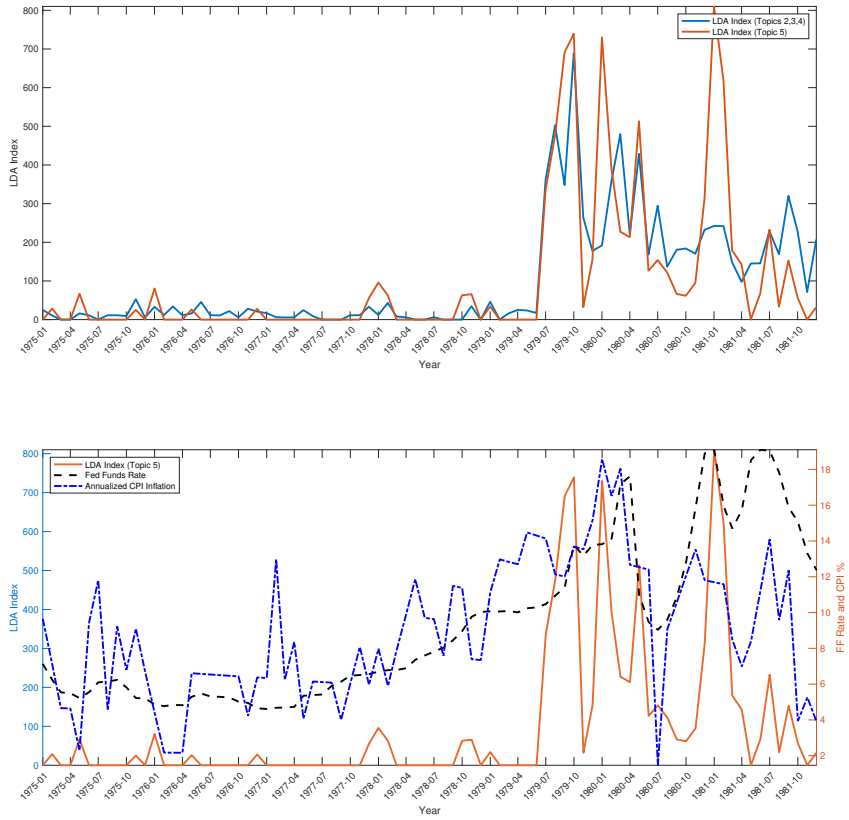


Figure 16: LDA Index (Topics 1,3,4) and LDA Index (Topic 5) (top), LDA Index (Topic 5) and Interest Rate and CPI Inflation (bottom)

The top panel in the figure above shows that there seems to be wide variation in the movements of both indices. The spikes in the topic 5 index are relatively much larger both before and after February, 1979. In the lower panel, it appears that the LDA index based on topic 5 does relatively better (than the index based on topics 2, 3 and 4) in tracking the movement of inflation but not interest rates. One of the reasons we see a slight correlation between the index based on topic 5 and fed funds rate and inflation is because Volcker tied his inflation policy to dollar in order to deal with the high inflation and the falling value of dollar during the late 1970s. There are many news articles where it is evident that given the concern of European countries' about the US inflation, they supported Volcker's policies that were in favor of stabilizing the value dollar and controlling inflation. The case in point are the following excerpts from several news articles from around that time:

CURRENCY MARKETS Dollar Surges and Gold Dips on Volcker Naming: Development Hailed (NYT, Jul 26, 1979) The dollar rose sharply on foreign exchange markets at home and abroad yesterday following President Carter's nomination of Paul A. Volcker as chairman of the Federal Reserve Board. The price of gold fell, retreating from its record high of $306.25 an ounce. Gold prices usually ease when the American currency gains.Gold dropped $2,50 an ounce in Zurich and London, closing at $303.625 and $303.75, respectively. In New York, the price of gold fell $1.90 to $303.35. The nomination of Mr. Volcker, president of the Federal Reserve Bank in New York, was greeted warmly in financial circles, with analysts predicting that he would follow a policy strongly in support of the dollar.

Businessmen Praise Selection of Volcker, Citing Reputation as Defender of Dollar (WSJ, Jul 26, 1979)
The selection of Paul Volcker to head the Federal Reserve Board appears to be among the more popular actions President Carter has taken in. quite a while, at least among business executives. Word of the nomination yesterday brought swift and almost unanimous praise from the business community here and abroad. What seemed to please these officials most is Mr. Volckers international Standing as a defender of the sagging U.S. dollar, an immediate concern of most of those commenting on the selection. In Zurich, Hans J. Mast, chief economist of Swiss Credit Bank, called the choice "the most constructive of the new appointments" to come out of the Carter administration Shake-up. "For once, I am positive," the banker declared. "Mr. Volcker's unanimously appreciated on the monetary scene as an able financial expert and a rigorous man who can be expected to take good care of the dollar," a French central bank official said.

Defending The Dollar: A Gamble (NYT, Aug 17, 1979)
The Federal Reserves aggressive moves in the last two days to raise short-term interest rates make clear that international considerations, and specifically the defense of the dollar, are now influencing American economic policy to a degree unparalleled in the post-Economic war period. The shift, Analysis which began last November when the Federal Reserve significantly tightened monetary policy as part of a rescue package for a sinking dollar, represents a major political risk on the part of the Carter Administration. The President and his new chairman of the Federal Reserve, Paul A. Volcker, are gambling that tighter money will not only strengthen foreign confidence in the dollar, but will also slow down this countrys galloping inflation.

Europeans Pin Hopes On Volcker: Cite Old Tie to Giscard and Schmidt (NYT, Jul 30, 1979)
Paul A. Volcker, the prospective new Federal Reserve Board chairman, whose confirmation hearings open in Congress tomorrow, is expected to emerge as the trusted economic liaison between Americas Euro- pean allies and a United States Administration for which they have often expressed contempt. The remarkable speed and enthusiasm with which European Governments have welcomed Mr- Volckers nomination, officials here say, testifies to the pivotal role they see him playing in fostering good economic relations between Europe and the United States during the difficult months. that now seem to lie ahead.

The figure 17 below presents the index based topic 1 which appears to be about stock and commodity markets (and constitute about 8% documents in the entire corpus) against the LDA index based on topics 2, 3 and 4 (once again reproduced from figure 13 above). The overall behavior of two indices is very similar except the topic 1 index is relatively more volatile.

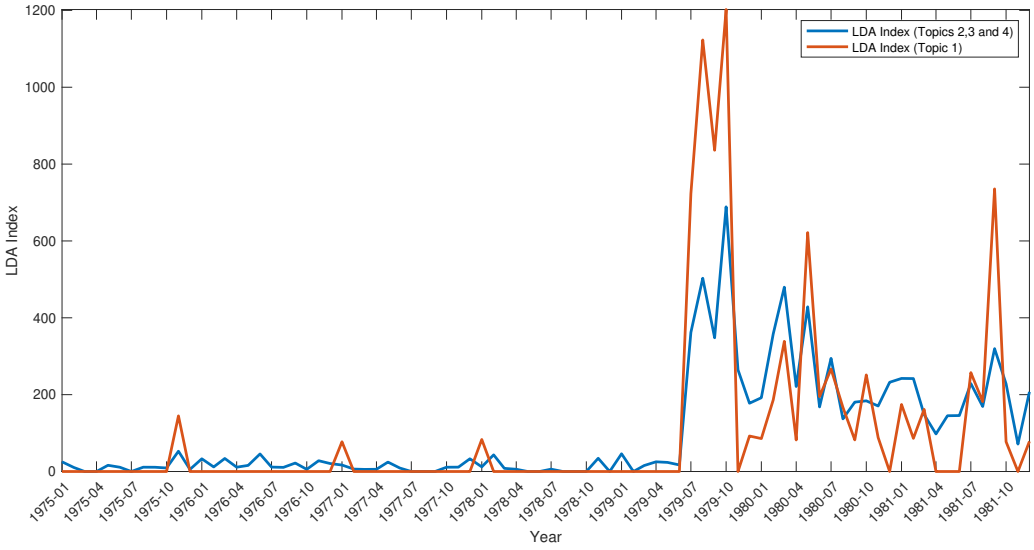


Figure 17: LDA Topic Model Index based on Topic 2 and Topics 2, 3 and 4

I also compared these indices with the ones that are created using the probabilities of topics appearing in a document. One of the reasons for using weights/probabilities of topics in documents instead of the number of counts (which is what I used to compute the indices presented above) was that under the counts method, I separated documents into different topics based on the highest probability of a topic appearing in a document which by defaults make the probability of other topics 0 and assigns a probability of 1 to the topic with the highest probability (it becomes a binary probability case). However, all the topics can appear in every document with a positive probability (although may not be true for documents). Using probabilities instead of counts allows me to use all the probabilities rather than just the topic with highest probability. However, I did not find major difference in the indices based on these 2 methods, although, the indices based on the probabilities were relatively volatile which is expected because we have data for more dates if we use probabilities (on some dates a topic can have some probability of appearing in a document but it may not be highest. In the case of counts, this data point will be discarded but in case of probability weights, this will be one more data point).

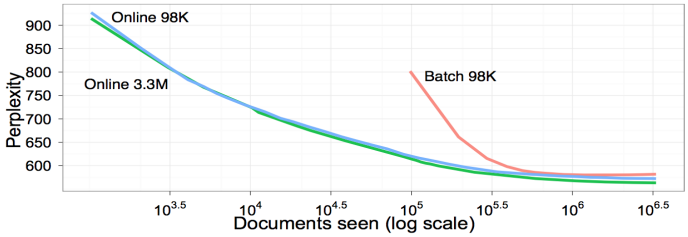
Overall, based on above results, none of the indices seem to be a good indicator of either interest rates or inflation based on the LDA model. A longer time series of the data might offer better results. Blei and Lafferty (2006) introduced the dynamic topic models (which is an extension to static LDA model) to analyze the time evolution of topics in large document collections. Hansen, McMahon, and Prat. (2017) used this dynamic LDA model to construct a measure of FOMC communication and to what topics individuals allocate attention in FOMC meetings. By dividing their data at meeting-speaker level, they computed the distribution over topics discussed in FOMC meetings and then constructed a time series measure of attention devoted to each topic. This method can also be tried to check if our measure of expectations based on dynamic LDA model tracks interest rates or inflation better than static LDA model.

Issues with LDA-VEM in Python Gensim: I have looked into the detailed code of the original paper by Blei et al. (2003) and Hoffman et al. (2012) who are the creators of LDA topic modeling techniques and use variational inference method to estimate it. I matched their code with the source code of the Python Gensim's LDA-VEM package and both the codes are more or less similar to each other. A lot of functions in Python Gensim's LDA-VEM are directly picked up from the code in the original papers (original code is written in C++). Therefore, as far as methodology of estimation is concerned, I don't think the reliability of Python's Gensim should be a problem. There seems to be a great approval for

this library on many blogs and discussion forums like stackoverflow.

However, there are clearly some issues with the predictive capacity of the LDA-VEM estimated in Python Gensim as seen in figure 11 above in the perplexity estimates of the training and test data. This indicates misrepresentation of the words of the test documents by the trained topics and therefore uncertainty about the fitted model.

The following figure from Hoffman et al. (2012) compares different methods - batch LDA with online (stochastic LDA) in terms of perplexity. The batch LDA is estimated on a Wikipedia corpus of 98000 articles whereas online LDA allows an increasing size of an already very large corpus. As a function of number of documents, the perplexity decreases much faster for online LDA. This figure also indicates that the size of corpus can be important in increasing the accuracy of the model.



Documents analyzed	2048	4096	8192	12288	16384	32768	49152	65536
Top eight words	systems road made service announced national west language	systems health communication service billion language care road	service systems health companies market communication company billion	service systems companies business company billion health industry	service companies systems business company industry market billion	business service companies industry company management systems services	business service companies industry services company management public	business industry service companies services company management public

Figure 1: Top: Perplexity on held-out Wikipedia documents as a function of number of documents analyzed, i.e., the number of E steps. Online VB run on 3.3 million unique Wikipedia articles is compared with online VB run on 98,000 Wikipedia articles and with the batch algorithm run on the same 98,000 articles. The online algorithms converge much faster than the batch algorithm does. Bottom: Evolution of a topic about business as online LDA sees more and more documents.

Figure 18: Perplexity on Held-out Documents

In order to further check the predictive capacity of the LDA-VEM, I manually checked some of the articles in the held-out dataset and compared the fitted topics on this data with the content of the articles in this dataset. Some of the topics didn't seem to match the content of the documents/articles (I think sometimes the problem with classification of documents under different topics is that if a word in a document is repeated once (even if that word doesn't indicate the overall topic of the article) but if that word occurs in a topic with a very high probability, the model sometimes classifies the article under that topic (in which the word occurs in the top few words with very high probability) even if the overall tone of the article is not based on that word. However, this is just an observation and I could

be missing something here. This could possibly be due to the very small size of the corpus. Training of models in machine learning models generally requires huge datasets as in the case of Hoffman et al. (2012) [5].

This MFVI (LDA-VEM discussed above) optimization is solved with a coordinate ascent algorithm, iterating between re-analyzing every data point in the data set and re-estimating its hidden structure. This is inefficient for large data sets because it requires a full pass through the data at each iteration. Hoffman et al. (2013) derived a more efficient algorithm by using stochastic optimization which iterates between subsampling the data and adjusting the hidden structure based only on the subsample which is claimed to be much more efficient than traditional variational inference and is called stochastic variational inference. The results for online Stochastic Mean Field Variation LDA are included in the Appendix.

I also compare the topics from LDA-VEM estimated using Python Gensim with LDA-VEM estimated using R topicmodels and the topics from both the models appear broadly consistent with each other (although the R 'topicmodels' for Gensim is not very stable as the topic-term distributions fluctuate a lot on re-estimation of the model). The code for estimating VEM using R topicmodels package is as follows:

```
k <- 5 ## no. of topics
SEED <- 2010 #3 fix the seed
## Parameters
VEM_control <- list(estimate.alpha = TRUE, alpha = 50/k, ## estimate the
#value of alpha with 50/k as initial value (as in Griffith....(2004))
estimate.beta = TRUE, ## estimate beta from the model
#(topic-terms distribution)
verbose = 0, prefix = "/Users/vemresults",
save = 0, keep = 1, ## save all the results
```

[5] The problem mainly comes from the bag of words approach of LDA that completely ignores the sequence of words. There is another modeling technique LDA-word2vec which is based on word embeddings in which every word is represented by its context (a vector of neighboring words) which takes into account the context in which a word is written. This is based on deep learning techniques. I haven't tried the LDA-word2vec but I tried to calculate the probabilities of a word to appear in a certain context using word2vec and then I manually checked the context to make sure if it was correct. The algorithm seems to make good predictions. For example, when I fed the word 'volcker', it gives me the probability of a word such as 'inflation' or 'chairman' to appear in its neighborhood. Other general predictions such as for example, if the algorithm learns that $man \leftrightarrow woman$ and $king \leftrightarrow queen$ from a corpus, then it can predict $king - man + woman \rightarrow queen$. But these algorithms appear to be a work in progress in the context of LDA topic modeling.

```
seed = as.integer(Sys.time()), nstart = 1, best = TRUE,
var = list(iter.max = 500, tol = 10^-6),#arguments control how convergence
# for variational inference step
em = list(iter.max = 1000, tol = 10^-4), #arguments control how convergence
# for EM algorithm (tol ensures convergence of likelihood)
initialize = "random")

## Estimate the model
VEM <- LDA(mycorpus_dtm, k = k, control = VEM_control)
```

The estimates of perplexity for different number of topics are presented in the figure below:

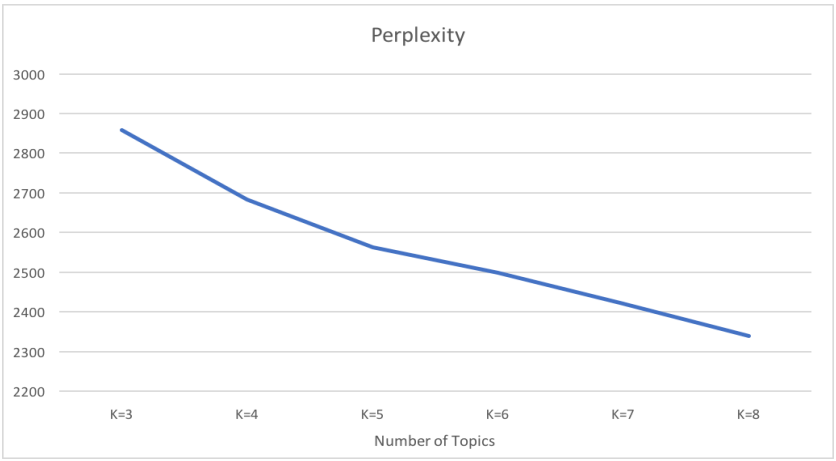


Figure 19: Estimated Perplexity in LDA-VEM using R 'topicmodels'

The perplexity is declining even for $K \geq 5$ which does not make sense for a corpus this small (This is similar to what we saw above with Python's Gensim). The topics cannot be distinguished from each other if a large number of topics are used for small number of documents. Therefore, perplexity in this case does not seem to be very informative. The estimated value of α is 0.08 which is not too off from the estimates of α by LDA-VEM in python Gensim. However, there doesn't seem to be any problem in the convergence of the likelihood if the number of topics K is fixed at 5.

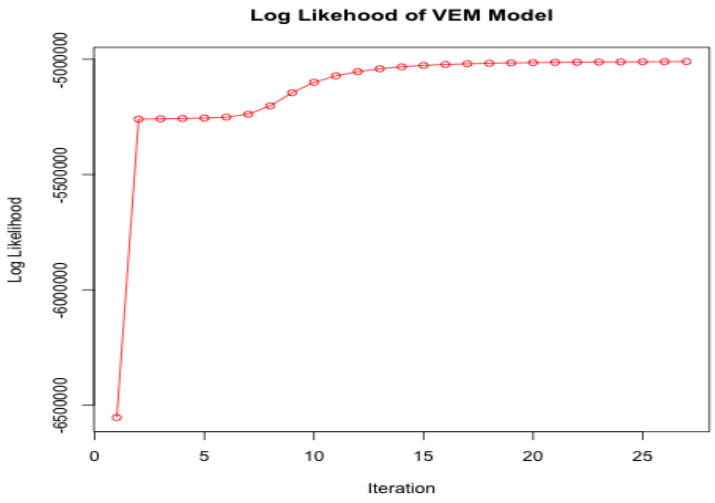


Figure 20: Estimated Likelihood in LDA-VEM using R 'topicmodels' (K=5)

The table 5 below compares the topics from both models (I have fixed the number of topics at 5). The broad topics appear to be similar in both models with distribution of topics over vocabulary being different in both models.

Table 5: Comparing Topics from LDA Python Gensim (Left 5 columns) with R Topicmodels Package (Right 5 columns)

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
stock	tax	bank	rate	dollar	dollar	fed	president	market	inflation
market	inflation	president	bank	gold	gold	money	million	rates	economic
price	president	chairman	fed	company	bank	carter	chrysler	rate	tax
gold	economic	volcker	money	market	price	federal	company	interest	budget
share	cut	hunt	reserve	price	silver	banks	state	prices	reagen
volume	carter	federal	market	chryslar	exchange	reserve	house	million	policy
inflation	reagen	reserve	federal	rate	international	budget	federal	markets	administration
commodity	federal	silver	credit	bank	foreign	interest	carter	bonds	volcker
rate	policy	fed	volcker	currency	volcker	rates	government	billion	federal
industrial	administration	board	supply	million	world	bank	chairman	stock	president
average	budget	banker	growth	board	monetary	rate	billion	credit	economy
president	rate	foreign	inflation	west	rates	volcker	reagen	federal	carter
exchange	volcker	year	policy	official	market	credit	board	treasury	recession
issue	price	house	monetary	german	markets	monetary	business	company	cuts
oil	billion	market	fund	exchange	currency	supply	department	money	cut
dow	chairman	american	price	federal	dollars	growth	american	rose	spending
analyst	house	international	bond	oil	inflation	policy	officials	bond	interest
rose	government	loan	term	ounce	west	board	city	term	chairman
economic	business	company	current	volcker	ounce	chairman	years	stocks	money
point	economy	monetary	economy	foreign	american	funds	secretary	price	price
silver	program	committee	increase	billion	countries	market	foreign	average	growth
high	reverse	treasury	billion	loan	central	committee	senate	volume	billion
standard	spending	secretary	economy	mark	german	reserves	plan	month	rate
fell	congress	banking	increase	american	reserve	inflation	loan	trading	business
company	increase	member	billion	current	president	feds	soviet	shares	government
recession	current	current	economist	european	economic	banking	request	analysts	fiscal
american	official	future	deposit	world	prices	president	congress	fell	monetary
active	million	million	committee	president	oil	deposits	administration	high	supply
bank	year	world	control	chairman	european	savings	union	higher	reserve
federal	money	executive	high	international	chairman	system	world	issues	credit
investor	plan	chief	target	reserve	trading	intitutions	oil	securities	economy
board	oil	country	president	inflation	london	treasury	white	oil	congress
cent	recession	business	economic	plan	federal	financial	national	decline	years
money	fed	state	account	government	banks	accounts	people	reserve	program
chairman	monetary	rate	account	meeting	currencies	term	industry	short	high
government	fiscal	vice	short	london	treasury	reuqest	energy	points	house
current	deficit	exchange	saving	country	hunts	staff	companies	banks	prices
future	american	solomon	recession	trading	commodity	central	made	banks	controls
reserve	white	financial	business	miller	system	loan	plans	quarter	unemployment
volcker	energy	central	month	treasury	interest	economy	former	investors	secretary
metal	control	carter	million	sale	meeting	business	committee	industrial	policies
trader	secretary	government	prime	dealer	bankers	control	bank	companies	increase
jones	supply	national	security	fell	hunt	high	court	prime	wages
decline	cost	official	higher	economic	trade	targets	sales	inflation	advisers
policy	board	brother	member	franc	germany	short	days	funds	miller
declining	nation	staff	banking	high	money	increase	expected	dow	deficit
block	committee	policy	banker	central	mark	commercial	bank	reported	treasury
dollar	high	problem	institution	monetary	swiss	members	court	yield	white
traded	growth	position	long	carter	pound	discount	sales	fed	increases
carter	world	meeting	change	germany	officials	prime	days	volcker	began

The table 6 below presents the topic comparison in both models:

Table 6: Comparison between the Topics of LDA-VEM Python Gensim and R topicmodels

Python Gensim	R topicmodels	Topics
Topic 1	Topic 4	stock markets, commodities
Topic 2	Topic 5	government, fiscal, monetary policy
Topic 3	Topic 2 and 3	Industry and Monetary policy
Topic 4	Topic 2 and 3	Industry and Monetary policy
Topic 5	Topic 1	Currencies, exchange rates

Possible Issues with R 'topicmodels' package for LDA-VEM: The topic-terms distribution appears to fluctuate a lot if the model is re-estimated using LDA-VEM in R 'topicmodels' which indicates that this package may not yield stable estimates. On the other hand re-estimating LDA-VEM in Python 'Gensim' package delivers stable results. Although, I haven't found too many criticisms of R 'topicmodels' package except for small discussions on stackoverflow where they discourage this package to estimate LDA using VEM. I searched for the source code for this package, however, I could not find much information on it. So, I am not sure how well the code is adapted to the original LDA-VEM paper. R 'topicmodels', however, is widely used for LDA-Gibbs sampling model, the results from which I'll present in the next section.

4.2 Results of Gibbs LDA Topic Model

This section presents the results from LDA - Gibbs sampling estimated using R 'topicmodels' library. This is one of the most widely used package for LDA-Gibbs in addition to another 'lda' package in R by Chang (2015) which is also mentioned in some of the presentations by the creators of LDA model (Blei et al.), however, the documentation of this package is incomplete, therefore, 'topicmodels' package models because of relatively detailed documentation is preferred over 'lda' package in R. The parameters of the model are described in the R code below:

```
k <- 5 ## no. of topics
## Parameters
gibbs_control = list( alpha = 0.1, ##alpha cannot be
#estimated in this model
estimate.beta = TRUE, ## (beta is the topic-terms
```

```
#distribution) estimate beta because
verbose = 25, prefix = "/Users/gibbsresults",
save = 0, keep = 50, ## save all the results
seed = as.integer(Sys.time()), nstart = 1, best = TRUE,
delta = 0.1, ##(delta is the hyperparameter of beta
#(doc-topic) dist)
burnin = 1000, thin = 100, iter = 4000)

## Implement Gibbs sampling
Gibbs <- LDA(mycorpus_dtm, k = k, method = "Gibbs", control = gibbs_control)
```

The hyperparameter α cannot be estimated in this model, so it is fixed at 0.1 (to allow comparison with the results from LDA-VEM obtained above) and it is the same value across topics.

The following figures presents the word clouds (since it was easy to make word clouds in R) with top 50 words in each of the five topics obtained from the LDA-Gibbs model. The words in larger fonts are the words that appear with higher probability than the small words and words in same color are the words that appear with similar probabilities.

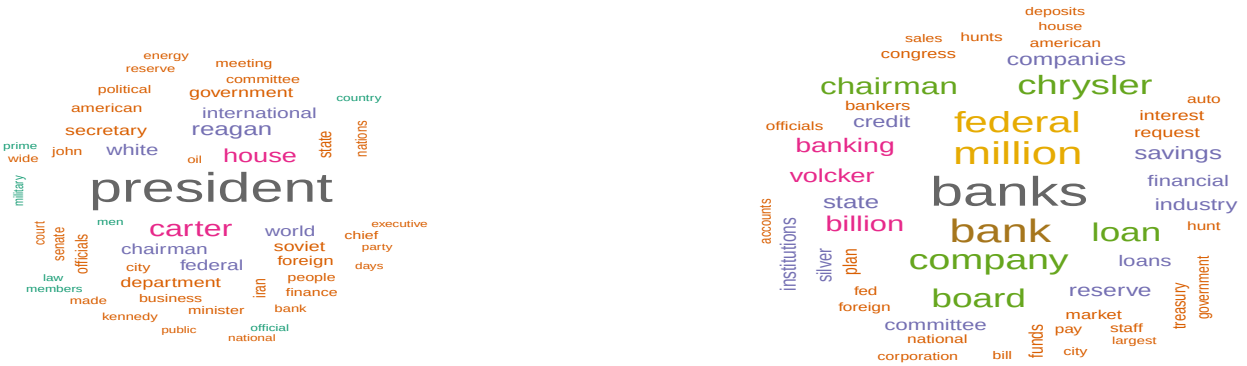


Figure 21: Topic 1 (left) - Fiscal policy and Defense & Topic 2 (right) - Monetary Policy and Industry

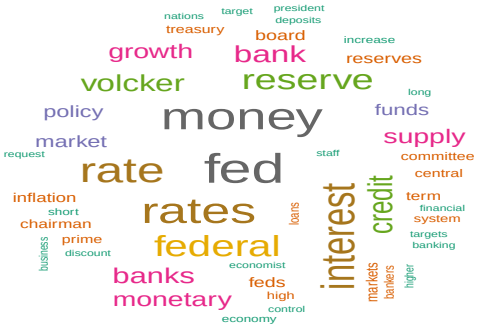


Figure 22: Topic 3 (left) - Fiscal Policy & Topic 4 (right) - Volcker and Monetary Policy

Figure 23: Topic 5 (Combines Topic 1 on stocks and 5 on currencies and exchange rates in LDA-VEM (python)

In topic 1, no word clearly dominates in terms of the probabilities that can be helpful in

labeling this topic but it appears to be related to government and defense policies. All other topics have words that can help them identify from the other. Broadly, the topics estimated by the Gibbs sampling are also not too off from the topics estimated by the LDA-VEM in Python Gensim. The overall comparison of topics between the two models are presented in the table below:

Table 7: Comparison between the Topics of LDA-VEM and LDA-Gibbs Models

VEM	Gibbs	Topics
Topic 1	Topic 5	Stock markets, commodities
Topic 2	Topic 1 and 3	Government, fiscal policy
Topic 3	Topic 2	Monetary policy and industry
Topic 4	Topic 4	Monetary policy
Topic 5	Topic 5	Currencies

In order to assess the predictive accuracy of Gibbs sampling, I once again divided the dataset into a training corpus of 1200 documents and a testing corpus of 199 documents. After estimating the model on the training dataset, I fitted it onto the training dataset and computed perplexity. The figure below provides the estimates of perplexity.

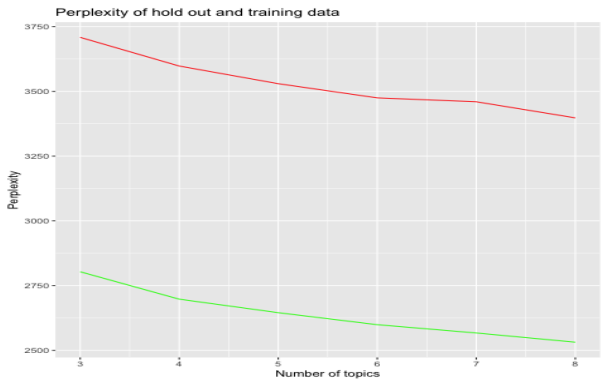


Figure 24: Estimates of Perplexity from LDA-Gibbs

Note: The red line is perplexity estimate of testing set and the green line is for the training set.

The same problem exists in the estimation of Gibbs sampling model as in the LDA-VEM. The perplexity doesn't appear to converge at any topic and continues to decline for higher topics. The likelihood of the model doesn't show convergence as well and this has been cited to be the problem with Gibbs sampling method. However, this method is widely used in the topic modeling literature in addition to LDA-VEM and its variants. I could not create

the indices based on Gibbs sampling because I had manually changed a lot of things while preparing the corpus for Python due to the problems with the OCR scanned (which were a lot) and I am not very sure how to make similar changes in R. I tried making those changes in R but I could not do it successfully. So I could not bring the dates of the articles in the correct format in R. Since I only had the documents without the dates, I could not create time series indices based on the topics estimated from the Gibbs sampling. The next section presents the results from NMF which are same as presented in the earlier writeup. The LDA methods seem to be widely used in the topic modeling literature because of their probabilistic basis despite the fact the NMF also produces quite accurate results. So I did not extend the results in the next section.

4.3 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is a linear-algeabreic model, that factors high-dimensional vectors into a low-dimensionality representation. Similar to Principal component analysis (PCA), NMF takes advantage of the fact that the vectors are non-negative. By factoring them into the lower-dimensional form, NMF forces the coefficients to also be non-negative.

Suppose a nonnegative matrix $X \in \mathbb{R}^{D*M}$ is given. When the desired lower dimension is k, the goal of NMF is to find the two matrices $W \in \mathbb{R}^{D*K}$ and $H \in \mathbb{R}^{K*M}$ having only nonnegative entries and $K < min\{D, M\}$ such that

$$X \approx WH$$

where $X_{d,m}$ (document-term matrix) contains the number of times word m appears in document d, $W_{d,k}$ comprises basis vectors that that captures the topics (clusters) discovered from the document or how many times document d contains topic k and $H_{k,m}$ (coefficient matrix) contains the membership weights for the words in each topic. The matrices W and H are found by solving an optimization problem defined with the Frobenius norm[2] (a distance measure between two given matrices) to minimize the following objective function:

$$\min_{W \geq 0, H \geq 0} f(W, H) = ||X - WH||_F^2$$

[2]Other optimization methods such as multiplicative update solver are also available but the Frobenius norm seems to perform better

There are different methods available to optimize the objective function and I use block coordinate descent (BCD) framework which is also called the alternating nonnegative least squares method. It divides the variables into several disjoint subgroups and iteratively minimizes the objective function with respect to the variables of each subgroup at a time. A natural partitioning of the variables is the two blocks representing W and H, respectively. That is to say, we take turns solving

$$W \leftarrow \min_{W \geq 0} ||W^T H^T - X^T||_F^2$$
$$H \leftarrow \min_{H \geq 0} ||WH - X||_F^2$$

Once either H or W is initialized, these equations can be solved iteratively.

4.4 Result of NMF Topic Model (This section is same as earlier

The table 8 below presents the topics generated from the NMF topic model along with the top 50 words in each topic.

Table 8: Top 50 Keywords in 5 Topics in NMF Model

Index	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	rate	stock	tax	dollar	chryslar
2	fed	bond	cut	gold	bank
3	money	market	reagan	ounce	loan
4	bank	share	inflation	currency	company
5	reserve	volume	budget	franc	million
6	supply	price	economic	price	hunt
7	growth	million	administration	german	board
8	credit	yield	fed	mark	skate
9	federal	company	policy	yen	banking
10	monetary	issue	president	london	president
11	market	dow	spending	pound	guarantee
12	policy	average	deficit	west	silver
13	fund	rate	recession	market	official
14	target	oil	fiscal	swiss	plan
15	inflation	rose	economy	trading	banker
16	volcker	trading	program	dealer	federal
17	prime	investor	volcker	exchange	intitution
18	deposit	industrial	price	zurich	thrift
19	committee	fell	adviser	rate	billion
20	term	billion	billion	trader	foreign
21	economist	active	house	french	chairman
22	discount	jones	congress	european	house
23	account	reported	government	silver	saving
24	control	note	reduction	silver	deposit
25	loan	traded	wage	cent	miller
26	short	treasury	miller	foreign	agency
27	range	decline	secretary	commodity	reserve
28	increase	sale	federal	closed	committee
29	economy	bill	increase	late	member
30	board	analyst	economist	japanese	union
31	bond	quarter	monetary	tokyo	industry
32	recession	security	business	europe	proposal
33	member	earnings	unemployment	canadian	carter
34	central	block	white	world	senate
35	checking	climbed	growth	fell	city
36	banker	loser	chairman	central	international
37	treasury	utility	rate	delivery	government
38	borrowing	composite	rate	frankfurt	american
39	change	gainer	official	metal	law
40	saving	nassau	council	bullion	secretary
41	chairman	month	income	paris	auto
42	high	sold	republican	rose	car
43	month	advancing	plan	american	source
44	higher	rally	energy	international	meeting
45	measure	debenture	problem	country	department
46	commercial	term	restraint	oil	sale
47	analyst	rated	stockman	germany	acquisition
48	decline	high	suppy	buying	business
49	demand	high	political	compared	country
50	meeting	offering	productivity	monetary	corporation

The topics generated by the NMF model are quite similar to the ones generated by the LDA model. A quick comparison shows following similarities between the topics of LDA and

NMF models:

Table 9: Comparison between the Topics of LDA and NMF Models

LDA	NMF	Topics
Topic 1	Topic 3	Monetary
Topic 2	Topic 2	Commodities, stock market
Topic 3	Topic 5	Monetary policy and industry
Topic 4	Topic 1	Monetary policy
Topic 5	Topic 4	Currencies

The first row shows that topic 1 in LDA model which is about monetary and fiscal policy is similar to topic 3 in the NMF mode;. Similarly, topics 2, 4 and 5 in LDA model are similar to topics 2, 1 and 4 in NMF model respectively. Topic 3 in LDA model, however, resembles little with topic 5 in NMF model. I am going to use topics 1 and 3 in the NMF model to construct an index that measures expectations of interest rates or inflation since these topics have the keywords such as inflation, short, term, rates etc. These 2 topics constitute about 48% articles of the total number of articles in the corpus.

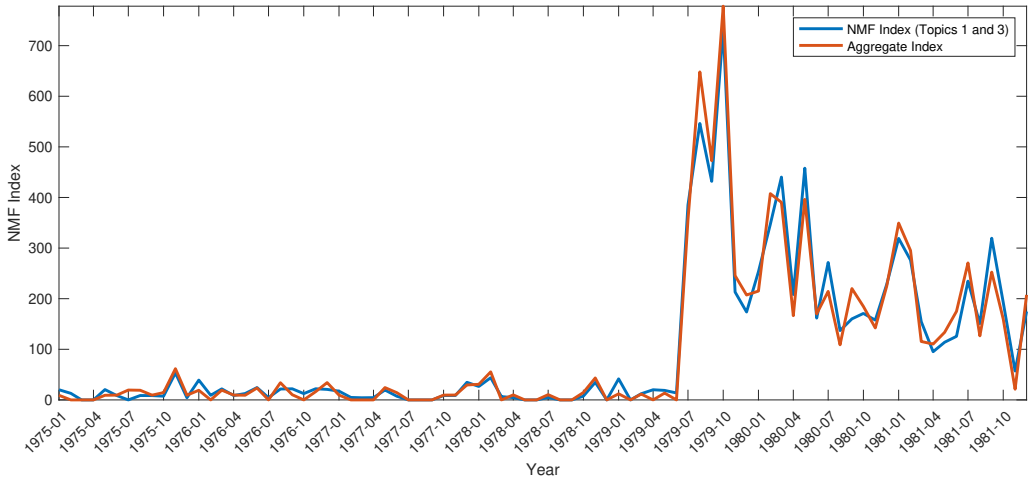


Figure 25: NMF Topic Model Index (Topics 1 and 3) and Aggregate Index

The figure 25 above shows the NMF topic model index based on topics 1 and 3 plotted against the aggregate index. The two indices almost coincide with each other. Thus, there seems to be no additional information captured by the index based on NMF model in comparison to the raw counts of the articles (without using any topic model). The figure 26 below plots NMF index based on topics 1 and 3 against the similar LDA index based on

topics 1,3, and 4 . Again, it seems that the information captured by the indices based on both topic models is more or less similar. These methods might lead to different results in a bigger corpus with articles related to varying topics. There are evidence where NMF model provides better results, however, given the probabilistic based methodology of LDA is used more often than NMF.

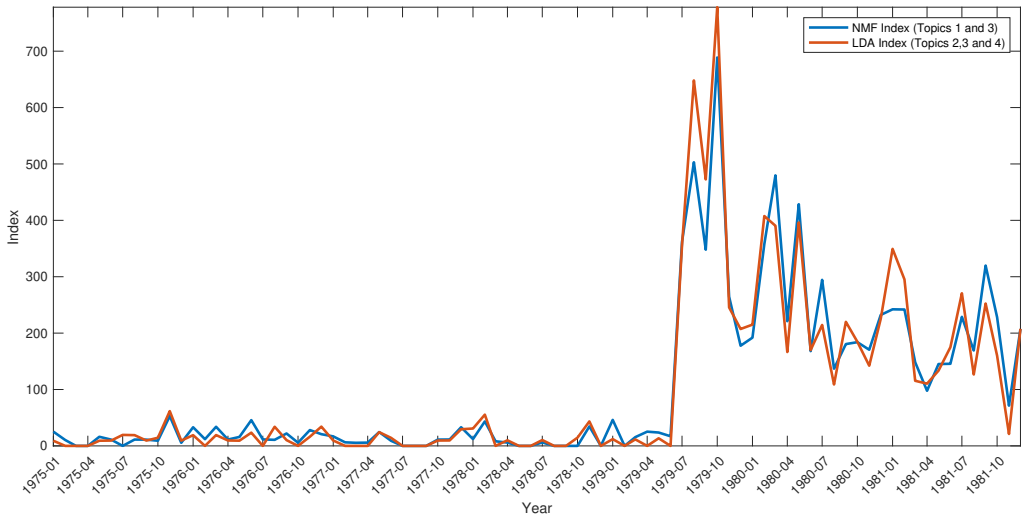


Figure 26: NMF Index (Topics 1 and 3) and LDA Index (Topics 2,3, and 4

The figure 27 below (top panel) plots the NMF index based on topic 4 which is the topic about currencies against the LDA topic 5 index which is also based on currencies. The middle panel plots NMF index based on topic 5 against fed funds rate and annualized CPI inflation. In order to allow comparison of the LDA and NMF indices based on currencies topic, the bottom panel in this figure reproduces the lower panel of figure 16 above that plots LDA topic 5 index against fed funds rate and annualized CPI inflation.

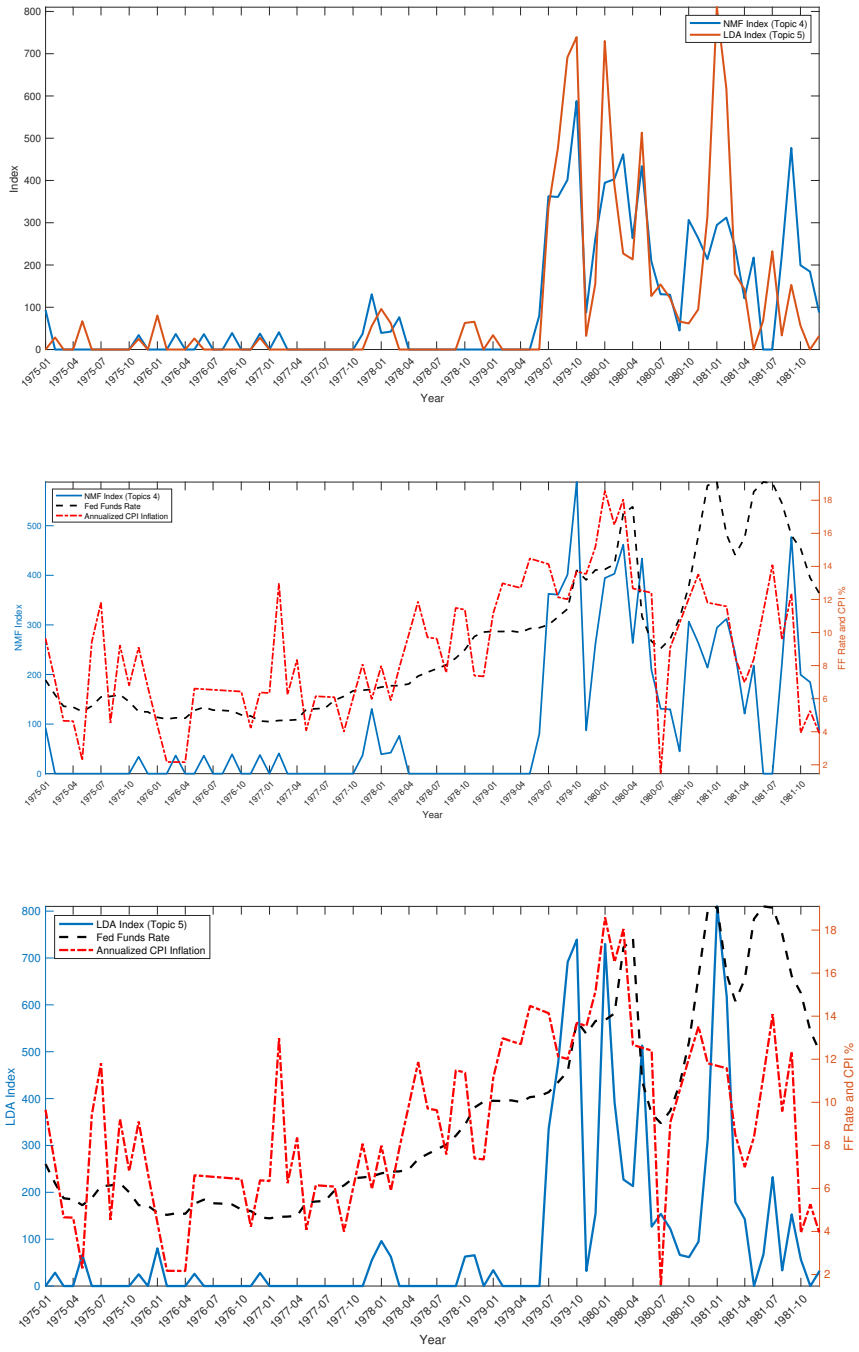


Figure 27: NMF Index (Topic 4) and LDA Index (Topic 5) (top), NMF Index (Topic 4) and Interest Rate and CPI Inflation (middle) LDA Index (Topic 5) and Interest Rate and CPI Inflation (bottom)

The middle and bottom panel of figure 27 above show that the NMF index based on currencies topic performs relatively better than LDA index based on the same topic in tracking CPI inflation. However, it is still far from being a good measure of inflation expectations. Based on the analysis above, it is possible that the topics that include information about currencies,

trade etc might capture more information about inflation or interest rate expectations.

5 Conclusion

Overall, none of the methods seem to be highly accurate and the problem appears to be more with the size of the corpus. As far as methodologies are concerned, both LDA-VEM and Gibbs are heavily used methods in the topic modeling literature. At this stage, I am not sure which of these methods will provide accurate results unless a corpus with a big enough size is used for estimation.

References

[1] Blei, D.M., 2012. Probabilistic topic models. Communications of the ACM, 55(4), pp.77-84.

[2] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3, no. Jan (2003): 993-1022.

[3] Blei, David M., and John D. Lafferty. "Dynamic topic models." In Proceedings of the 23rd international conference on Machine learning, pp. 113-120. ACM, 2006.

[4] Hansen, Stephen, Michael McMahon, and Andrea Prat. "Transparency and deliberation within the FOMC: a computational linguistics approach." The Quarterly Journal of Economics 133, no. 2 (2017): 801-870.

[5] Hoffman, M., Bach, F.R. and Blei, D.M., 2010. Online learning for latent dirichlet allocation. In advances in neural information processing systems (pp. 856-864).

[6] Hoffman, M.D., Blei, D.M., Wang, C. and Paisley, J., 2013. Stochastic variational inference. The Journal of Machine Learning Research, 14(1), pp.1303-1347.

[7] Hornik, K. and Grn, B., 2011. topicmodels: An R package for fitting topic models. Journal of Statistical Software, 40(13), pp.1-30.

[8] Kuang, Da, Jaegul Choo, and Haesun Park. "Nonnegative matrix factorization for interactive topic modeling and document clustering." In Partitional Clustering Algorithms, pp. 215-243. Springer, Cham, 2015.

[9] Baker, Scott R., Nicholas Bloom, and Steven J. Davis. "Measuring economic policy uncertainty." The Quarterly Journal of Economics 131, no. 4 (2016): 1593-1636.

[10] http://www.rpubs.com/MNidhi/NumberoftopicsLDA

[11] https://www.r-bloggers.com/cross-validation-of-topic-modelling/

[12] https://radimrehurek.com/gensim/models/ldamodel.html#gensim.models.ldamodel.LdaModel.save

Appendix 1

5.1 Online Stochastic MFVI (LDA)

The following parameters were fed into the Gensim online LDA to estimate the model using the chunks of the full corpus (online LDA) using the Stochastic Mean Field Variational Inference (MFVI):

```
lda_model = gensim.models.ldamodel.LdaModel
            (corpus=corpus,
            id2word=id2word,
            num_topics=4, ## No. of topics - fixed after checking
            ## model perplexity for different topics
            random_state=100, # seed
            update_every=1, ## updated once every chunksize
            #(chunksize < corpus size)
            chunksize=10, ## chunksize (instead of full corpus)
            #online LDA
            passes=1, ## No. of passes at same corpus - in online LDA
            ## training the model only once is sufficient
            eval_every=1, ## for perplexity - calcuated every document (lower
            alpha= 'auto', ## learn from the model
            eta = 0.1, ## fix this to some number.
             #Learning from the model makes convergence difficult
            iterations=200, ## maximum no. of iterations
            gamma_threshold = 0.00001,
            decay=0.7, ## #tau_0 - should be between 0.5 to 1 for convergence
             # changing this to 0.9 barely changed the topics
            #(weight on learning of previous lambda (global param.)
            offset=1.0, ## kappa in online LDA
            per_word_topics=True)
```

Only difference from batch LDA discussed above is that now the chunksize is 10 that with each iteration, the data is updated in chunks of 10 documents instead of the full corpus as in the batch LDA and since its a small chunksize, I am now evaluation perplexity every 1 document (eval every= 1) which should produce better results instead of 10 documents

in batch LDA (it is too slow to evaluate perplexity every one document in batch LDA). However, this model is trained only once instead of 10 times as batch LDA. The Python Gensim documentation argues that given that the estimation is done over small chunks of the corpus (which itself is efficient), the training of the model does not need to be repeated. The additional parameters are offset (same as κ) and decay (same as τ_0) which help in determining the learning weight ρ_t assigned to the global parameter λ the variational parameter of the β - topic-terms distribution. The values assigned to these parameters is based on the stochastic variational inference literature. A grid search can also be helpful here in obtaining the correct or more sensible values for these parameters.

The figure 28 below shows that perplexity is the lowest for when $K = 4$ in this model (the estimates of perplexity are different than obtained in the earlier models) and then it increases if the number of topics increase and then again starts declining when the number of topics increase above 7. In this model, I choose $K = 4$ to see if the results are better than obtained in other models.

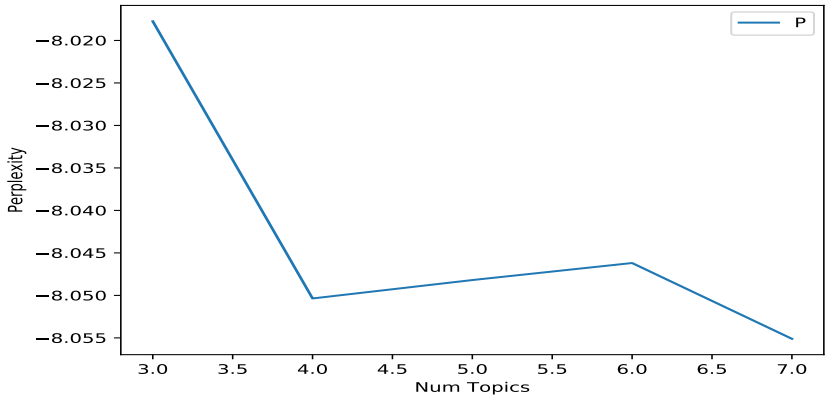


Figure 28: Online LDA Model Perplexity Estimates for Different Number of Topics

The table 10 below presents the estimates of the hyperparameter α that corresponds to the document-topic distribution θ_d. The estimates (in row 1 for 4 topic online LDA) are much higher relative to the estimates for the batch LDA for all the topics. Even if I fix the number of topics to 5, the estimates of α (in row 2) are still very high.

Table 10: Estimates of Hyperparameter α of the Document-Topic Distribution (θ)

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1.768	1.99	1.302	2.074	-
1.218	2.34	1.03	2.12	1.83

The optimized value of ρ_t - the weight on the new value of the global parameter λ (the underlying variational parameter of the topic-terms distribution), given the parameters $\kappa = 0.7$ (decay parameter which controls the rate at which old values of λ are forgotten) and $\tau_0 = 1$ (slows down the rate of early iterations of the algorithm (I couldn't get a clear interpretation of this parameter)), is 0.031456. That is, about 3% learning weight is assigned to the new value of λ during every iteration. I also changed the value of κ from 0.7 to 0.9 that barely changed the results but made the convergence relatively worse. I also tried some variations of the parameter $\kappa = 64$ and 1024 (some of the values tried in Hofffman et al. (2012)), however, the results make little sense at very high values of κ.

The figure 29 below presents the estimates of variational bound (ELBO) that approximates the log likelihood of the true model. The online LDA uses 1400 iterations as opposed to only 80 iterations in batch LDA. In addition, the convergence is a bit choppy in online LDA as opposed to smooth convergence in case of batch LDA. The reason could again be the small size of the corpus since online LDA is more suited to handling very large corpus.

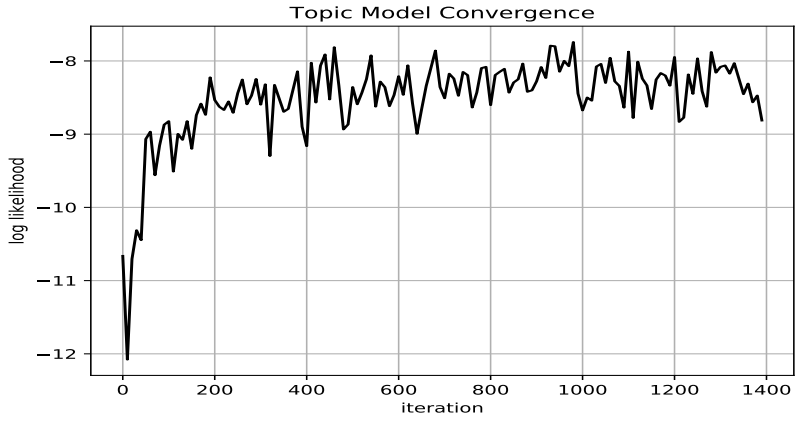


Figure 29: Log Likelihood Estimates for Online LDA (or ELBO Estimates

The figure below presents the graphic view of topics for the online LDA model. Each of the 4 topics seem to constitute a large fraction of the total documents which was not the case with the batch LDA model where some topics constituted a much small fraction of the

documents in the entire corpus relative to others. In addition, the topics are far apart from each other which shows that each topic is very different from the other which was again not the case with batch LDA where topics 2 and 5 slightly overlapped indicating the presence of common words. I re-estimated this model for $K = 5$ and 3 topics showed much higher overlap (see figure 31) which could be an indication that less number of topics might be more suitable for this corpus which is also evidenced by the similarity of some of the topics.

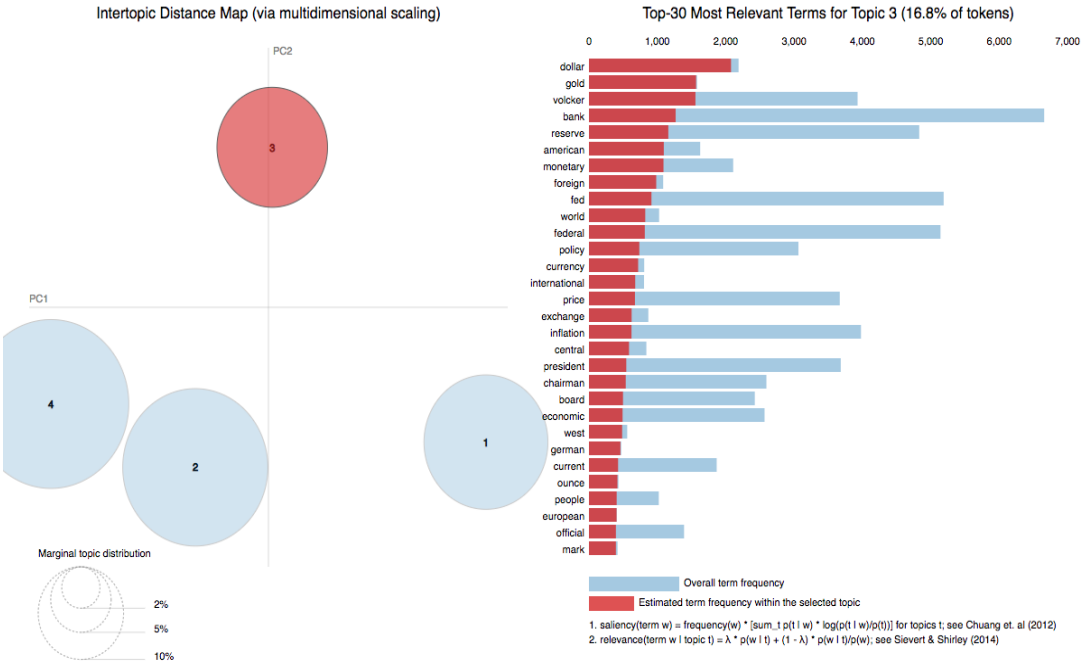


Figure 30: Online LDA Model Topics

The table 22 below presents the topics generated from the NMF topic model along with the top 50 words in each topic. The topic 1 is about the government and defense policies. Topic 2 is about fiscal policy. Topic 3 is about currencies, commodities and international exchange rates. And topic 5 is about monetary policy. Once again, the topics are broadly similar to the ones obtained in the batch LDA model.

Table 11: Topics Estimated by Stochastic LDA-VEM Model

Index	Topic 1	Topic 2	Topic 3	Topic 4
1	reagen	tax	dollar	rate
2	inflation	president	gold	bank
3	president	carter	volcker	market
4	chairman	cut	bank	fed
5	economic	administration	reserve	money
6	policy	government	american	reserve
7	committee	budget	monetary	federal
8	industry	company	fed	credit
9	growth	house	foreign	price
10	support	inflation	world	supply
11	economy	federal	federal	volcker
12	recession	economic	policy	fund
13	proposal	program	currency	growth
14	money	congress	international	bond
15	institution	million	price	loan
16	soviet	oil	exchange	stock
17	secretary	billion	inflation	term
18	future	country	central	board
19	federal	plan	president	company
20	leader	business	chairman	high
21	republican	board	board	increase
22	going	state	economic	treasury
23	ford	official	west	million
24	area	chrysler	german	deposit
25	power	volcker	current	higher
26	finance	price	ounce	month
27	able	year	people	inflation
28	restraint	energy	european	decline
29	billion	spending	official	short
30	saving	policy	mark	banker
31	free	white	commodity	trading
32	democrat	cost	control	current
33	defense	increase	rate	billion
34	account	miller	business	analyst
35	auto	member	meeting	issue
36	london	chairman	trade	economist
37	thrift	department	talk	policy
38	fight	people	economy	security
39	association	fiscal	nation	monetary
40	monetarist	control	burn	rise
41	period	nation	governor	rose
42	continue	wage	future	average
43	monetary	deficit	role	prime
44	regulation	senate	financial	business
45	asset	problem	question	target
46	william	group	secretary	economy
47	possible	current	national	share
48	wide	loan	job	level
49	treasury	car	political	lower
50	meeting	city	europe	change

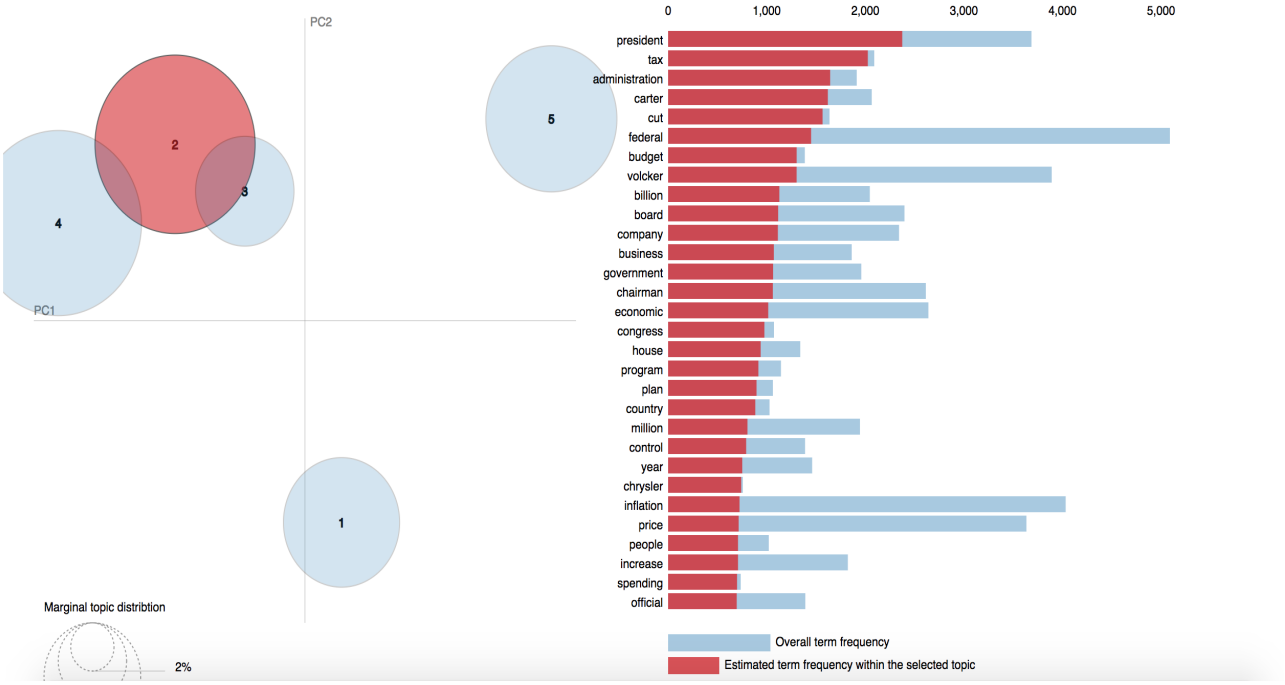


Figure 31: Online Variational Inference Model with 5 topics

Appendix 2

5.2 Additional Notes on Gibbs Sampling

This section contains further results from model selection methods for Gibbs sampling LDA model (using 'topicmodels' library in R) in continuation to section 4.2. The objective is to find the optimal number of topics and the value of hyperparameter α.

This is based on a 10-fold cross validation to evaluate the performance of the models. First, the data set is split into a randomly sampled set of test dataset with the remaining data as training data. This process is repeated 10 times thus producing 10 different datasets each containing a randomly chosen test and training dataset. And then for each of these folds (fold of a test and training dataset), perplexity is calculated for different number of topics ($K = 2, 3, ..., 10$). The following figure presents the estimates of perplexity estimated by fitting the trained model (that is using the topics-term distribution obtained from the trained model) onto the test dataset for each of the topics in each fold. The value of α is allowed to be estimated in this procedure, however, the estimated value of α for Gibbs sampling in R 'topicmodels' library is the default $\alpha = \frac{50}{K}$ as in Griffith and Stywers (2004). Therefore, for the model in which $K = 2$, $\alpha = 10$.

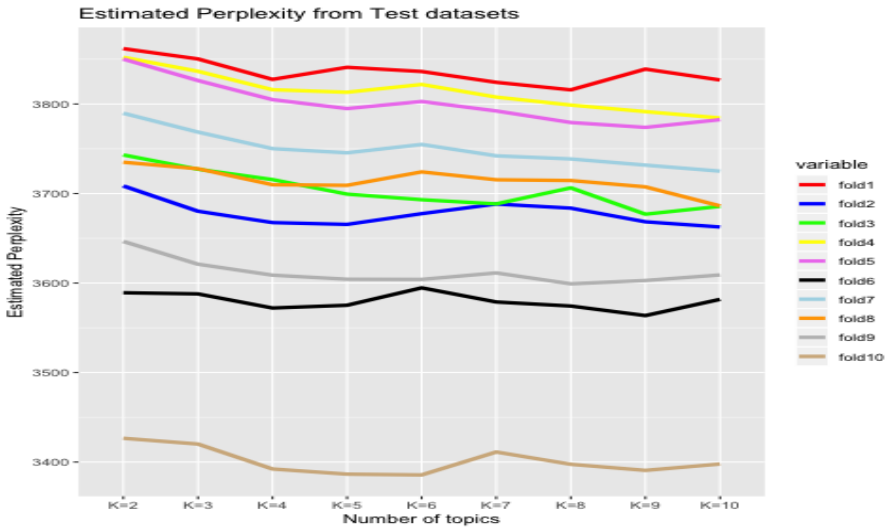


Figure 32: Estimated Perplexity from LDA Gibbs with $\alpha = \frac{50}{K}$

The above figure shows that for 3 out of 10 folds estimated perplexity declines until $K = 4$, while for 5 out of 10 folds, it declines until $K = 5$ and until $K = 6$ for the rest. On the basis of this result, both $K = 4$ or 5 could be the appropriate number of topics. I estimated the perplexity based on the full corpus for both $K = 4$ and 5 and it is less for $K = 6$.

I also attempted this 10-fold cross validation for Gibbs LDA keeping the value of α fixed at 0.1 which is much smaller than $\alpha = 10$ assumed in the previous cross-validation and following are the perplexities estimated from this model:

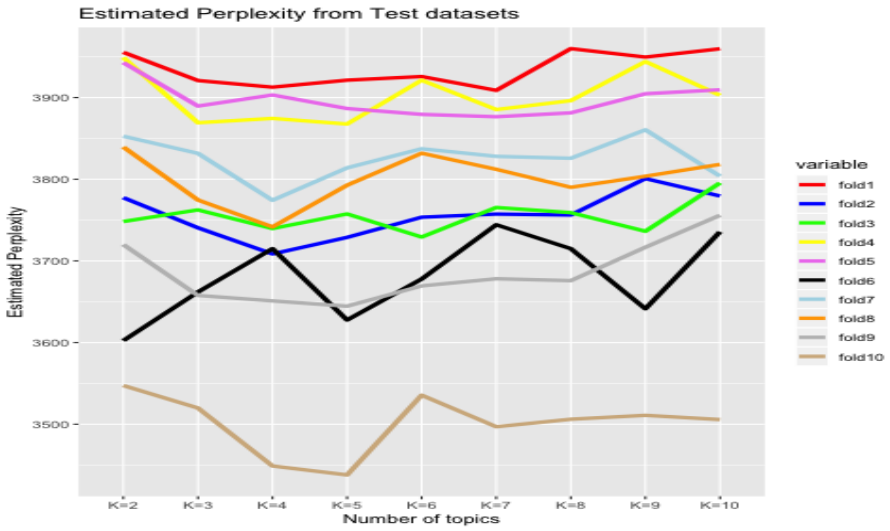


Figure 33: Estimated Perplexity from LDA Gibbs with $\alpha = 0.1$

The figure 33 above shows that for 4 out of 10 folds, the perplexity declines until $K = 4$, for 2 folds, the perplexity declines until $k = 5$ and for the remaining, the optimal choice of K appears to be 2 or 3. This model confirms that the optimal choice of K should be certainly less than 5 and given that for maximum folds/samples, perplexities decline until $K = 4$, it seems to be appropriate choice for the number of topics for the given corpus.

In order to analyze if the topics from the model with $K = 5$ and $\alpha = 10$ are different from the topics from the model with $K = 5$ and $\alpha = 0.1$ are very different from each other, I estimated the both LDA-Gibbs models and compared their resulting topic-terms distribution with each other. The top 20 matching terms between the topics from both the models where the topics are matched based on the Hellinger distance between the term distributions of the topics (using inbuilt package in R). The Hellinger distance is used to quantify the similarity between two probability distributions. The estimates of Hellinger distance are as follows:

Table 12: Hellinger distance between Topics of LDA-Gibbs with estimated α vs fixed α

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Topic 1	0.27	0.73	0.70	0.73	0.72
Topic 2	0.77	0.299	0.75	0.73	0.77
Topic 3	0.65	0.68	0.284	0.71	0.75
Topic 4	0.78	0.71	0.73	0.26	0.75
Topic 5	0.74	0.77	0.76	0.76	0.25

The above table shows that the topics on the diagonal are the closest to each other since the Hellinger distance between these topics is the smallest.The top 20 best matched terms of the two models are given in the following table:

Table 13: Comparing Topics from LDA Python Gensim (Left 5 columns) with R Topicmodels Package (Right 5 columns)

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
market	fed	house	inflation	president
gold	money	president	economic	chairman
prices	federal	carter	tax	dollar
million	rate	state	economy	volcker
markets	interest	billion	policy	reserve
price	banks	reagan	budget	bank
dollar	rates	chryslar	administration	foreign
stock	reserve	million	recession	international
bonds	credit	company	volcker	world
rose	bank	federal	rates	american
trading	volcker	government	growth	monetary
company	supply	congress	reagan	system
silver	funds	officials	cut	treasury
oil	board	plan	monetary	central
billion	growth	senate	spending	secretary
bond	markets	department	cuts	board
average	reserves	city	business	people
rates	monetary	chairman	price	miller
fell	feds	company	economists	countries
issues	committee	board	supply	country

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
market	fed	million	inflation	president
prices	banks	president	economic	bank
gold	money	house	policy	dollar
million	federal	chrysler	volcker	international
markets	rate	reagan	rates	volcker
dollar	interest	carter	tax	foreign
price	rates	billion	economy	world
trading	bank	company	money	american
rates	reserve	state	interest	reserve
stocks	credit	government	federal	chairman
bonds	volcker	federal	monetary	monetary
silver	funds	plan	administration	gold
rose	market	officials	budget	central
fell	supply	congress	recession	countries
interest	board	senate	growth	exchange
stocks	reserves	department	rate	system
volume	growth	administration	reagan	federal
shares	chairman	chairman	fed	board
oil	loans	business	supply	treasury
company	board	board	price	bankers

The above table shows the most similar words between the topics of both the models. These are also the words that have the highest probability of occurring in these topics and therefore it can be concluded that overall the topics are almost similar. This seems to imply that the variation in the value of hyperparameter α does not seem to have much impact on the terms that lie in a topic. The distribution of terms in a topic could however be different. The comparison of document-topic distribution under both models is also presented in the figure below.

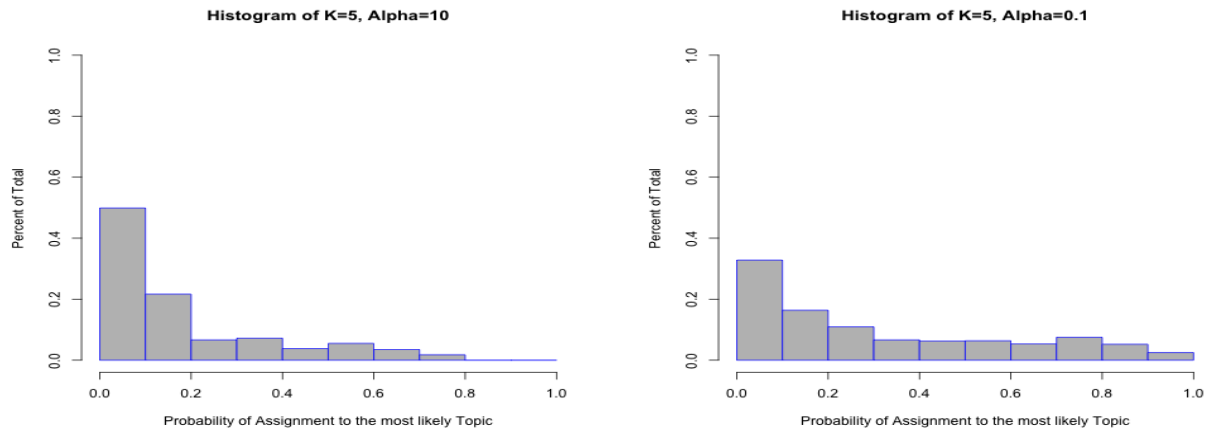


Figure 34: Histogram of the Probabilities of Assignment to the most likely Topic for all documents for model with $K = 5$ and $\alpha = 10$ (Left) and for Model with $K = 5$ and $\alpha = 0.1$ (Right)

In general, lower is the α, higher is the percentage of documents which are assigned to one single topic with a high probability. However, figure 34 shows that the association of documents to one topic is higher for the model with $K = 5$ and $\alpha = 0.1$ as expected since there are some documents that have a very high probability of appearing in one topic, however, the number of such topics is very small. Overall the two histograms are very similar to each other implying that the value of α does not affect the document-topic distributions in LDA-Gibbs model.

The table below compares the

Table 14: Comparison between the Topics of LDA-VEM and LDA-Gibbs Models

VEM	% of Documents	Gibbs	% of Documents	Topics
Topic 1	7.7%	Topic 5	17.8%	Stock markets, commodities
Topic 2	33.7 %	Topic 1 and 3	Topic 1 - 23.3 % and Topic 3 - 18.7%	Government, fiscal policy
Topic 3	9.6%	Topic 2	22%	Monetary policy and industry
Topic 4	35.5 %	Topic 4	18.2 %	Monetary policy
Topic 5	13.5 %	Topic 5	17.8%	Currencies

Similarly, the estimates of perplexity for LDA-VEM model (using R 'topicmodels') have also been estimated. There are two variations to this model. The first model estimates the value of α taking $\alpha = \frac{50}{K}$ as the initial value and the value of α is fixed for the other model

and is equal to $\alpha = \frac{50}{K}$. For the model in which the value of α, the estimates of both α and perplexity are presented in the following figure:

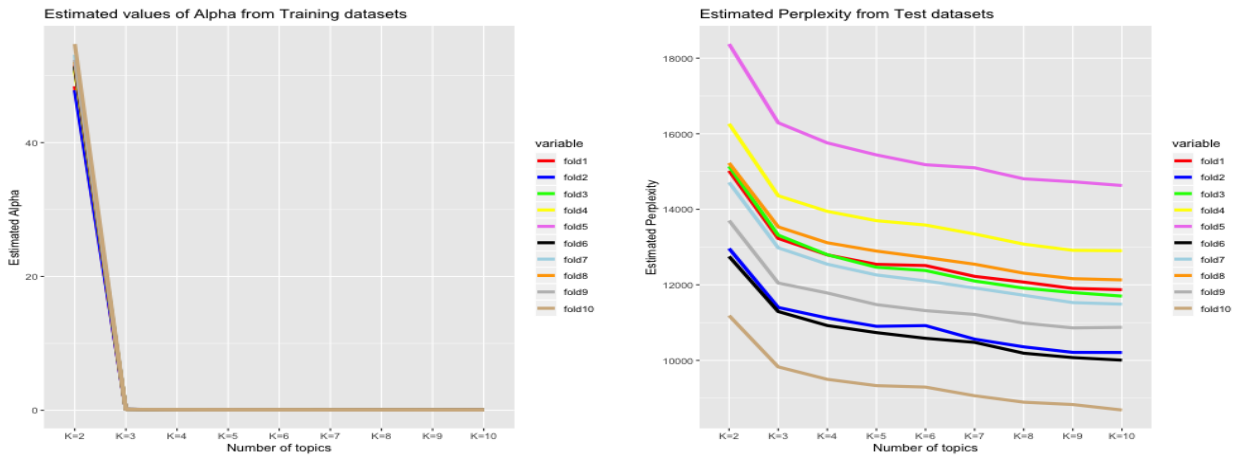


Figure 35: Estimates of α (left) and perplexity (right) for LDA-VEM

The figure 33 (left) shows that for all the models, the estimated value of $\alpha \sim 0.08$ for $K \geq 3$ which is much smaller than the default $\alpha = \frac{50}{K}$. The figure 33 (right) shows the estimates of perplexity for this model. For some folds, the optimal number of topics is between 5 and 6 whereas it is continuously declining for the others even when $K \geq 6$. The perplexity based on LDA-VEM when $\alpha = \frac{50}{K}$ are presented in the figure below

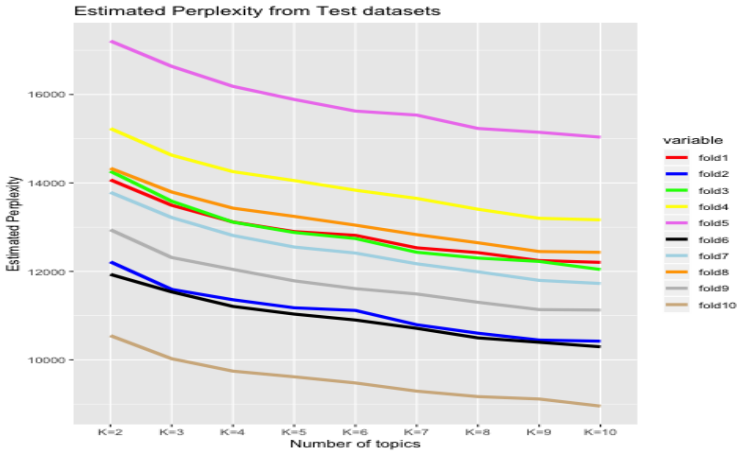


Figure 36: Estimated Perplexity from LDA-VEM when α is fixed

The estimates of perplexity are very similar for this model as that of the above model in which values of α are estimated. It appears that while perplexity can give some idea about approximate number of topics, the best method is to select the number of topics based on manually checking the topics assigned by the trained model to the test data for different

models. The model that assigns the best topics to the test data in line with the content of the articles in the dataset should be the best model.

Appendix 3

5.3 Literature Review

This section discusses some of the papers that I came across from the machine learning and economics literature that uses different methods to create a time series out of textual data.

1. Lucca and Trebbi (2009) used automated scoring techniques to measure the content of the central bank communication that attempt to extract information about future policy rate actions based on information from the internet and news sources. The first method they use is Google Semantic Orientation (GSO) score that relies on estimating the systematic co-occurence of concepts retrieved from the universe of webpages. The application they propose in their paper is to assign to each sentence (of FOMC statement) an objective and automated score that is able to capture the semantic orientation of the statement, or one of its parts, along a metric that includes words that are associated with positive rate changes - hawkish, tighten, hike, raise, increase, boost and words that are associated with negative changes in policy rates movement - dovish, ease, cut, lower, decrease, loose.

They begin by defining a measure of association between concepts. If the meaning of a string of text x can be commonly interpreted as hawkish, then x and the word "hawkish" should show a degree of positive statistical dependence in a sufficiently large corpus of text. In other words, the string x and the word hawkish should appear in a corpus of the text with a joint frequency, $Pr(x\&hawkish)$, which is greater than if the two strings were statistically independent concepts, in which case the joint frequency would be equal to the product of the marginals, or $Pr(x)Pr(hawkish)$. The Pointwise Mutual Information (PMI) is a central concept in information theory, and it is derived from the joint entropy of two random variables. The PMI between the string of text x and the word hawkish is defined as:

$$PMI(\text{x, hawkish}) = log\left(\frac{Pr(\text{x, hawkish})}{Pr(\text{x})Pr(\text{hawkish})}\right)$$

A measure of the relative degree of association between the string x and the word "dovish" can be computed accordingly hence obtaining the degree of "dovishness" that can be attributed to x. The theoretical score of semantic orientation (SO) of string x can therefore obtained as:

$$SO(\text{x}) = PMI(\text{x, hawkish}) - PMI(\text{x, dovish})$$

The Internet represents a very large corpus of text from which it is possible to obtain empirical frequencies of each string of text in a statement, and the words "hawkish" and "dovish". Since it is unfeasible to directly compute the joint frequencies of co-occurrence in the actual population of webpages (at least without devoting huge computing resources), they empirically implement the information retrieval (IR) process through hits counts on the search engine Google. The feasible estimator of the semantic orientation score obtained by information retrieval on Google using the "hawkish - dovish" word pair (and other pairs) is:

$$GSO(\text{x}) = log\left(\frac{hits(\text{x \& hawkish}) * \text{hits(dovish)}}{hits(\text{x \& dovish}) * \text{hits(hawkish)}}\right)$$

They implement this scheme on each sentence of an FOMC statement by creating chunks of five words of each statement and after obtaining the GSO scores for each search unit x, they average GSO(x) over all x in the statement to obtain a score for FOMC statement t at time t. Then they approximate the unexpected change in the content of the statement at date t as the difference between the semantic orientation score at meeting t and the score at meeting $t-1$. They found that GSO appears reasonable as a measure of communication, and leads the policy rate by about two quarters and the correlation of the GSO score and the fourth Eurodollar futures implied rate is about 40% if they use just hawkish-dovish antonymy and 80% if they use all the antonymy pairs related to positive and negative rate changes.

The GSO score relies on the Internet as the corpus of text on which the joint frequencies that form the score are estimated which allows access the text of the corpus of webpages indirectly through Google searches and in turn a rather limited control over which specific texts the search is run over, the specific time periods of reference, or the relevance of the matches obtained from the search engine. In addition, Google does not publicly disclose the algorithm it uses to calculate and approximate the count of hits of a given search. Due to these issues they created another index based on the discussions of FOMC announcements from newspaper, magazine, newswires and newsletters that are included in the Dow Jones Factiva database, a leading provider of business and financial news.

The original documents range from very short pieces of newswire information to long newspaper articles and commentaries. To implement the score they searched all sources available worldwide in English, for articles with headlines involving the words "Federal Reserve",

"Fed" "FOMC", around times of FOMC meetings and record all the sentences in the database that match this criterion. They select all these articles on a 3-days window around the FOMC meeting starting on the day before, and ending on the day after, the announcement allowing them to focus on information as pertinent as possible to each given policy announcement. They compute the Factiva semantic orientation score for statement t as:

$$FSO_t = \log \frac{\sum_{s \in T_t} I[s, R, P]}{\sum_{s \in T_t} I[s, R, N]}$$

where T_t is the set of sentences in news articles around release date t and hence pertaining to statement t, I[s, R, P] is an indicator function that takes value 1 if sentence s contains a relevant word from R = { Rates, Policy, Policies, Statement, Announcement, Fed, FOMC, Federal Reserve} and a word that denotes positive interest rate change and 0 otherwise. Similarly I[s, R, N] gets a value 1 if statement s contains a relevant word and a word that denotes a negative rate change. The unexpected changes in the stance of the FOMC announcement at t as the difference between the Factiva semantic orientation score based on news released before and after the announcement is computed as

$$\triangle FSO_t = FSO_t^+ - FSO_t^-$$

where FSO_t^+ is FSO score based on set of sentences in news articles a day after the announcement date t and FSO_t^- is FSO score based on set of sentences in news articles a day before the announcement date t. One thing they also pay attention to in constructing FSO is that $FSOt^+$ does not include any instances of matches in the past tense for verbs, thus avoiding discussions of the most recent or past policy action (at t) and thus focusing on future policy moves (and similarly $FSOt^-$ for only discussions of past, rather than the immediately forthcoming action at t). In addition, they also refine their measures by excluding from the set of joint matches direct negations of the words included in the list of antonymies (for example, not hawkish) and include direct negations of the opposite (for example, not dovish for hawkish). They found that the Factiva automated score FSO constructed with this algorithm leads the policy rate by more than two quarters, with movements in levels that track the rate implied by the fourth Eurodollar futures contract fairly accurately (a correlation of about 40 percent if they use just hawkish-dovish antonymy and 80% if they use all the antonymy pairs related to positive and negative rate changes).

2. Boukus and Rosenberg (2006) also analyze the information content of FOMC minutes from 1987-2005. They apply a statistical methodology known as Latent Semantic Analysis (LSA) to decompose each minutes release into its characteristic themes and show that these themes are correlated with current and future economic conditions. This is very similar to the Non-negative matrix factorization (NMF) method that I also used in my analysis above. LSA uses Singular Value Decomposition (SVD) to perform a low rank approximation on term-document matrix, thereby bringing out the semantic connectedness present among the documents of the corpus. One of the pitfalls of using SVD is that the truncated matrix will have negative components, which is not natural for interpreting the textual representation. NMF presented in the analysis above addresses this issue by generating nonnegative parts-based representation as the low rank approximation for performing LSA. Like NMF, LSA also takes term-frequency-inverse-document-frequency matrix (tf-idf) X, described above, as an input and factors it into

$$X = USV^T$$

where the columns of U and V represent the orthonormal eigenvectors of XX^T and X^TX respectively. S is a diagonal matrix of singular values. Its elements are equal to the non-negative square roots of the eigenvalue ordered by decreasing magnitude. The U matrix relates terms to topics (topic-term matrix as described above) the V matrix relates documents to topics (documnent-term matrix as described above). The S matrix contains singular values arranged in descending order in which the magnitude of the i_th singular value indicates the importance of the i_th topic in explaining differences across documents. Their empirical analysis is based on the FOMC meeting minutes over the period from January 1987 to December 2005. Figure 37 below displays the ten most important terms, according to term contributions to themes/topics (U matrix entries), for each of the first 5 topics in their paper.

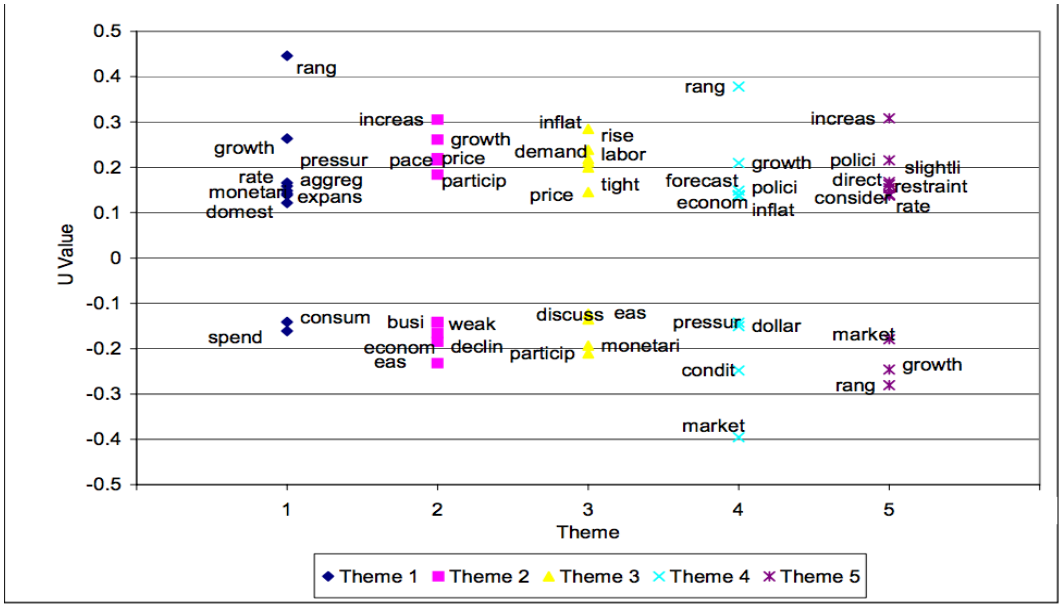


Figure 37: Characteristic terms for Topics 1 through 5 (Boukus and Rosenberg (2006))

They find that theme 1 include spend, consum, domest, expans, pressur, and growth, which suggests a relationship to *consumer spending and sentiment, and perhaps economic expansion.* Theme 2 highlights *the general macroeconomic conditions* with the following important terms: econom, weak, busi, ease, declin, price, growth, and increas. Theme 2 exhibits a *cyclical pattern, with a tendency to decline during periods of recession, further supporting a connection to the business cycle.* Predominant terms for Theme 3 include inflat, rise, demand, labor, tight, and price. The presence of tight, rise, demand, and inflat *imply a growing inflationary pressure coupled with contractionary monetary policy.* Theme 4 emphasizes *concern over the dollar and foreign exchange market*, as revealed by the terms dollar, market, inflat, condit, and pressur. Finally, Theme 5 appears to reflect the *monetary policy stance* with words such as polici, slightli, restraint, and consider.

Using the rows of the V matrix (document-topic matrix), they construct a time series of the evolution of topics over time. Each element of V matrix defines the contribution of topic i to document j and if documents are arranged chronologically, one row of V defines the time series of loadings for a particular topic. Using this time series as a measure of topic visibility in FOMC statements (*similar to what I did in my analysis as well. I used the total counts of articles in a particular topic in a month normalized by the total number of articles that month for each newspaper as a measure of the topic visibility. I also used the probabilities and the results were similar except the indicator based on the probabilities was relatively smooth.*). They found that treasury market responds to certain topics/themes

in the FOMC statements (at the time they are released) in addition to level of monetary policy uncertainty and the prevailing economic outlook. There results suggest that market participants can extract a complex, multifaceted signal from the minutes of FOMC meetings.

3. Hansen, McMohan and Tong (2018) use the publication of the Bank of Englands Inflation Report (IR) to study the market impact of such communication. They find compelling evidence for a long-run information effect driven more by narrative than by quantitative forecasts (given in the inflation report) and this effect mainly operates through term-premium channel. They use 2 methods to create a time series based on the text in the inflation report.

In the first step, they take the numerical information from the IR forecast, together with the VIX uncertainty index, and use it to purge the effects of numerical data on the asset price news. Then they determine whether there is information contained within the text shocks is relevant for explaining the asset price news residual.

Bigram frequency: In their first method, they use adjacent two-term phrasesalso called bigramssuch as 'slow growth' and 'strong growth' and then count the frequency of each bigram across each Report. This yields a 70*22,211 document-term matrix whose $(t, v)_{th}$ element is the count of bigram v in the Inflation Report released on day t. The resulting time series is the frequency counts of these bigrams in each report. However, this is a high-dimensional object in the sense that the number of dimensions of variation (i.e. the number of bigrams) across reports exceeds by an order of magnitude the number of reports in their sample. To solve the dimensionality issue, they first fit a LASSO (with four nominal short and long rates derived from UK bond prices as dependent variables such as 1 year spot rate, 3 year forward rate, 5 year forward rate and 5-year, 5-year forward rates) selecting less than 60 regressors for each dependent variable. They find that bigrams do have some explanatory power for interest rates. However, one of the drawbacks of LASSO is that it screens relevant variables so that their inclusion is guaranteed in the selected set (at least asympototically) but does not necessarily select the true model in the sense that noise variables are also guaranteed to be in the selected set.

LDA: In their second method, they use Latent Dirichlet Allocation (LDA) which is also the method I have used in my analysis. They extract 30 topics from the IRs. In order to create a time series that measures communication in the IRs, they use the information contained in the document-topic matrix (θ_t) in my analysis that contains the contribution of a topic in the inflation report published at time t. They also use another regressor $\delta_t = \theta_t - \theta_{t-1}$ that

measures the change in the topic coverage over time. Once again they use LASSO to select M most important topics that have predictive power for short and long term bond rates. They find that the relative contribution of text variables, therefore, is weakest for one-year spot rates, stronger for three-year rates, and strongest for five-year and five-year ahead, five-year rates (these last two rates are essentially indistinguishable). For the longest maturities, including around five topic controls is already enough to capture as much variation in market rates as all of the forecast variables whereas signals in the quantitative forecasts in the inflation reports explain is greatest variation in one-year spot rate.

There are some papers from the machine learning literature that discuss different methods use to extract information from the textual data.

3. Blei and lafferty (2006) developed dynamic LDA that analyzes the time evolution of topics in in a sequentially organized corpus of documents (sequenced along time dimension). LDA (static) assumes that words are exchangeable within each document, i.e., their order does not affect their probability under the model. LDA further assumes that documents are exchangeable within the corpus, and, for many corpora, this assumption is inappropriate, for example, scholarly journals, email, news articles, and search query logs all reflect evolving content. I believe that newspaper articles and FOMC statements also reflect content evolving over time although this would be true over a sufficiently long span of time. In the DTM, the corpus of documents is divided by time slice, e.g., by year or month and then the documents in each slice are modeled with a K-component topic model, where the topics associated with slice t evolve from the topics associated with slice $t-1$. DTM is nothing but LDA in each time slice except topics in time slice t depends on the same topic in time slice $t-1$. The algorithm for dynamic LDA is different from static LDA given the time dependency. Since the topics are evolving over time, the dynamic LDA uses state space models on the natural parameters of the multinomial distributions that represent the topics. The posterior inference over the latent topics is approximated using variational approximations (also used for static LDA) based on Kalman filters and nonparametric wavelet regression. Dynamic LDA is known to demonstrate greater predictive accuracy when compared with static topic models.

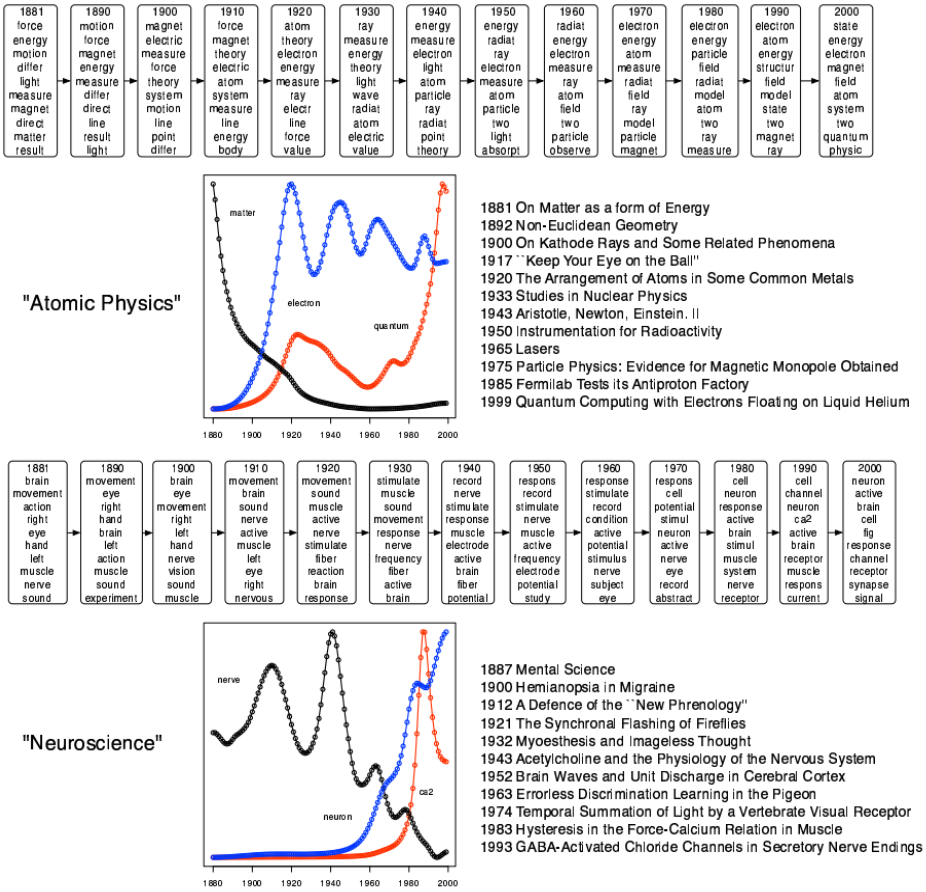


Figure 38: Example from the posterior analysis of a 20-topic dynamic model estimated from the Science corpus in Blei and Lafferty (2006). For two topics, they illustrate: (a) the top ten words from the inferred posterior distribution at ten year lags (b) the posterior estimate of the frequency as a function of year of several words from the same two topics (c) example articles throughout the collection which exhibit these topics. Note that the plots are scaled to give an idea of the shape of the trajectory of the words' posterior probability (i.e., comparisons across

The figure 38 above gives the time evolution of topics and the use of certain words over articles under a topic change over time, for example, the use of word "matter" declined over time whereas the use of word "electron" picked up around 1900s and the word "quantum increased a little around early 1900s, declined in the mid-1900s, and then increased a lot in the late 1900s.

4. Manel and Moreira (2016) created a text-based measure of uncertainty starting from the co-movement between the front-page coverage of the Wall Street Journal and options implied volatility (VIX) over the years 1890 until 2009. They call this measure news implied volatility (NVIX). Using this indicator, they find that in US postwar data, periods when NVIX is high are followed by periods of above average stock returns, even after controlling

for contemporaneous and forward-looking measures of stock market volatility. In addition, news coverage related to wars and government policy explains most of the time variation in risk premia our measure identifies. Their underlying assumption is that the choice of words by the business press provides a good and stable reflection of the concerns of the average investor.

Their news data set includes the title and abstract of all front-page articles of the Wall Street Journal from July 1889 to December 2009.4 We focus on front-page titles and abstracts to make the data collection feasible. They "manually" edited and corrected these articles after applying OCR to convert them from pdf to text, which improves their earlier sample reliability. They omit titles that appear daily. Titles and abstracts are separately broken into one- and two-word ngrams (either individual or compound words) using a standard text analysis package that replaces highly frequent words (stopwords) with an underscore and removes n-grams containing digits. Aggregating the n-grams to monthly frequency and normalizing this frequency by the total number of n-grams each month, they get a very high dimensional time series object. Eventually, they want to run a regression on the test data (for which the data for VIX published by CBOE is available)

$$v_t = w_0 + w.x_t + \nu_t, t = 1, ..., T$$

where v_t is VIX, w is K vector of regression coefficients and x_t is a frequency count of n-grams in month t. However, this regression cannot be estimated using least squares given the number of regressors is huge. So they overcome this problem using support vector regression (which turns the problem into a convex optimization problem and uses kernel functions to solve the regression), an estimation procedure shown to perform well for short samples with an extremely large feature space K.

5. Another paper by Nguyen, Shirai and Velcin (2015) built a model to predict stock price movement using the sentiment from social media. Unlike the usual approaches that consider only the overall moods or sentiments, their method incorporates the sentiments of the specific topics of the company into the stock prediction model. The feature that they use to predict stock price movement is called the "topic-sentiment" (extracting topics and sentiments simultaneously) that recognizes what topics are discussed in social media and how people feel about these specific topics of the company (product, service, dividend and so on).

To get the sentiment information of the stocks, they collected messages from18 message boards of the 18 stocks from Yahoo Finance Message Board for a period of one year (from July 23, 2012 to July 19, 2013). On the message boards, users usually discuss company news, prediction about stock going up or down, facts, comments (usually negative) about specific company executives or company events. In 15.6% messages in this dataset, when users posted messages on these message boards, they annotated each message as one of the following sentiment tags: Strong Buy, Buy, Hold, Sell and Strong Sell. There first time series consists of just these 15.6% messages with annotated sentiment by the users and discard the other messages. The purpose of this method is that how mood annotated by human can be used to predict the stock. For each transaction date t, the percentage of each class (Strong Buy, Buy, Hold, Sell, and Strong Sell) was calculated. The percentage of a class is the number of messages having sentiments as that class label divided by the number of messages in the current transaction date t and then they integrate this time series into the prediction model.

To extract the sentiments form the remaining 84.4% of the messages without the explicit sentiments they built a classification model using Support Vector Machine classification model that uses a linear kernel and was trained from the messages with annotated sentiments on the training dataset. Then it was used to classify the remaining messages into five classes (Strong Buy, Buy, Hold, Sell, and Strong Sell). In addition, they also use LDA (as above) to discover these hidden topics and for each transaction date t they built a time series using the probability of each topic (defined as the average of the probabilities of that topic in the messages belonging to that transaction date).

The final method they use is the JST-based method (joint sentiment-topic). The underlying idea is that when people post the message on the social media to express their opinion for a given stock, they tend to talk their opinions for a certain topic such as profit and dividend. Based on pairs of topic-sentiment, they would think that the future price of that stock goes up or down. To extract pairs of topic-sentiment, they used two kinds of models - latent topic based model and the JST model.

They consider each message as a mixture of hidden topics and sentiments. The JST model (it is a variation of LDA) was used to extract topics and sentiments simultaneously. In LDA model, there is only one document specific topic distribution for each document.Each document in JST is associated with S sentiment labels. Each of sentiment labels is associated with a document specific topic distribution with the same number of topics. A word in the

document is drawn from distribution over the words defined by the topic and sentiment label. Next, the joint probability of each pair of topic and sentiment is calculated for each message. After that, for each transaction date t, the joint probability of each topic-sentiment pair is defined as the average of the joint probabilities of that in the messages belonging to that transaction date.

The other model is the Aspect-based sentiment that considers that mixtures of topics and sentiments as " not hidden" in the messages as considered in the previous model (that considers them as latent). Each message is represented as a list of topics and their corresponding sentiment values. In our proposed method, the topic is the consecutive nouns in the sentence. For example, the message "The profit will go up." contains the topic "profit" and a positive sentiment "up" for that topic. They extract the consecutive nouns as the topics in the sentence. To eliminate rare topics, topics occurring less than 10 times are removed from the list of the topics. Next, based on the topic list, they extracted their sentiment values in each sentence. For each sentence, opinion words are identified based on the list of opinions from SentiWordNet (lexical resource for opinion mining). SentiWordNet assigns each word three sentiment scores: positivity, objectivity and negativity. They combined scores of positivity and negativity into a single opinion value. The closer between the topic phrase and the opinion word, the higher affection of that opinion on the topic phrase. Therefore, the sentiment value of a topic phrase in a sentence is the summation of overall opinion values divided by their distance to that topic. For each message, the sentiment value of each topic is defined as the average of the sentiment scores of that topic in the sentences. Finally, for each transaction date t, the sentiment value for each topic is defined as the average of the sentiment values of that topic in the messages belonging to that transaction date. In addition to the sentiment values of the topics, the importance of the topics for each transaction date were also considered. Intuitively, some topics have more impact on the prediction than others. If a topic was discussed in many messages, it might be an important topic in the given transaction date and the importance of a topic i in a transaction date t was calculated as the number of messages containing the current topic i in the transaction date t divided by the number of messages in that transaction date.

They found that the Aspect-based sentiment method outperformed over 2.54%, 2.14% and 2.87% on average accuracy compared to Sentiment classification, LDA-based method and JST-based method, respectively. The LDA-based and JST-based method seem to be not successful in this experiment. The limitation of these methods is that we have to specify the number of hidden topics in LDA and the number of hidden topics and sentiments in JST

which is not the case with aspect-based sentiment model.

6. Lastly, Das, Zaheer and Dyer (2015) proposed a Gaussian LDA for topic models with word embeddings that replace LDA's parameterization of "topics" as categorical distributions over opaque word types with multivariate Gaussian distributions on the embedding space. One of the major benefits of this model is that it aims to derive topics from text that are semantically coherent which is ignored in the LDA since LDA considers a document as a bag of words and ignores the semantic coherence between words. Their approach replaces the opaque word types usually modeled in LDA with continuous space embeddings of these words, which are generated as draws from a multivariate Gaussian. Word embeddings have been shown to capture lexico-semantic regularities in language: words with similar syntactic and semantic properties are found to be close to each other in the embedding space. Instead of considering documents as collection of individual words, their method considers each document as collection of word embedding wherein each word is a vector that contains the words that are contextually related to that word (generally words that come before and after the main word). This method takes into account that word in a context rather than the word alone which is expected to produce semantically coherent results. To perform inference, they introduce a fast collapsed Gibbs sampling algorithm based on Cholesky decompositions of covariance matrices of the posterior predictive distributions. This approach is relatively new however. Qualitatively, they find that Gaussian LDA infers different (but still very sensible) topics relative to standard LDA. Quantitatively, our technique outperforms existing models at dealing with words in held-out (test) documents. This is one problem I faced in my LDA analysis above. I found that the predictive accuracy of LDA on my corpus (it was a small corpus however) was not very accurate.