Latent Dirichlet Allocation Project

Student: Siyi Gao, sg7nx@virginia.edu

1 Introduction

My project focused on Latent Dirichlet Allocation (LDA), a powerful generative model for topic modeling introduced by Blei et al. (2003). The task of topic modeling involves uncovering hidden themes or topics within a collection of documents. An intuitive example of this task is a librarian organizing books in a library without any prior knowledge of their subjects. By analyzing the co-occurrence of words in the text, the librarian would need to group the books into categories such as "science," "history," or "fiction" and estimate the proportion of each category represented in each book. Topic modeling has a wide range of applications, such as opinion mining to analyze customer reviews and recommending articles to customers based on their reading interests and associated topics. Despite its usefulness, the task presents significant challenges due to the unstructured nature of text data and the high dimensionality of word distributions.

LDA solves this challenge by assuming that each document is a mixture of topics and each topic is a distribution over words. To explain this intuitively, imagine a writing machine that generates news articles. This machine has three controls: one to set the number of words in each document, one to select the topic for the article, and one containing a bag of words associated with each topic. For instance, if we want the machine to write an article about the environment, we provide the desired number of words, choose the "environment" topic, and let the machine pick words from the corresponding bag of words to write the article.

The document generation process under the LDA model is quite intuitive. However, the challenge is that we only have a collection of articles generated by this hypothetical machine without any knowledge of the settings used during their creation. We do not know which topics were selected, how the topics were mixed, or which words were associated with each topic. LDA addresses this challenge by reverse-engineering the generation process: it infers the hidden topics, the proportions of these topics within each document, and the word distributions for each topic based on the observed words in the documents.

In the original paper by Blei et al. (2003), the parameters for LDA were estimated using the variational EM algorithm. In addition to the variational method presented in the original work, collapsed Gibbs sampling, introduced by Griffiths and Steyvers (2004), is another widely used approach for parameter estimation in LDA. Based on my studies and preliminary experiments, I found that collapsed Gibbs sampling offers several advantages: it is more intuitive, easier to implement, and generally faster to execute. For these reasons, I chose to incorporate collapsed Gibbs sampling into my project and compare its performance with the variational EM method.

The goal of my project was to learn and implement 2 algorithms for estimating LDA parameters—variational EM and collapsed Gibbs sampling. After implementation, I conducted 2 studies: one focused on topic modeling to compare the generalization performance of the models trained using these algorithms, and the other on document classification to explore the use of LDA for feature selection and dimensionality reduction.

2 Theoretical Background and Estimation Algorithms

2.1 LDA Model

The Latent Dirichlet Allocation (LDA) model was initially introduced by Blei et al. (2003). LDA is a generative probabilistic model designed for managing collections of discrete data. In the context of topic modeling, the core concept of LDA is that documents are represented as random mixtures of latent topics, with each topic defined by a specific distribution over words.

The document generating process is described as follows, according to Blei et al. (2003):

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\boldsymbol{\theta})$.
   (b) Choose a word w_n from $p(w_n|z_n, \boldsymbol{\beta})$, a multinomial probability conditioned on the topic z_n.

Some assumptions are made for this model. First, the dimensionality k of the Dirichlet distribution, representing the number of topics, is assumed to be known and fixed. Second, the word probabilities are parameterized by a $k \times V$ matrix $\boldsymbol{\beta}$, where $\beta_{ij} = p(w_j = 1|z_i = 1)$. This matrix is treated as a fixed quantity to be estimated. Third, words are generated by topics according to fixed conditional distributions, and these topics ($\{z_1, z_2, \ldots, z_n\}$) are assumed to be infinitely exchangeable within a document. Forth, the number of words N in each document is assumed to be independent of the other data-generating variables. Hence, N is considered an ancillary variable and is not included in the estimation process.

Given the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the joint distribution of a topic mixture $\boldsymbol{\theta}$, a set of topics $\boldsymbol{z}$, and a set of words w is given by:

$$p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) \prod_{n=1}^{N} p(z_n \mid \boldsymbol{\theta}) p(w_n \mid z_n, \boldsymbol{\beta}). \tag{1}$$

Where $p(z_n|\boldsymbol{\theta})$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating over $\boldsymbol{\theta}$ and summing over $\boldsymbol{z}$, the marginal distribution of a document is given by:

$$p(\boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int \left(\prod_{n=1}^{N} \sum_{z_n} p(w_n \mid z_n, \boldsymbol{\beta}) p(z_n \mid \boldsymbol{\theta}) \right) p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) \, d\boldsymbol{\theta}. \tag{2}$$

By taking the product of the marginal probabilities of single documents, the probability of a corpus is given by:

$$p(\boldsymbol{D} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d=1}^{M} \int p(\theta_d \mid \boldsymbol{\alpha}) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \boldsymbol{\beta}) \right) d\theta_d. \tag{3}$$

The probability of a sequence of words $\boldsymbol{w}$ and topics $\boldsymbol{z}$ is:

$$p(\boldsymbol{w}, \boldsymbol{z}) = \int p(\boldsymbol{\theta}) \left(\prod_{n=1}^{N} p(z_n \mid \boldsymbol{\theta}) p(w_n \mid z_n) \right) d\boldsymbol{\theta}. \tag{4}$$

Another probabilistic model for LDA, introduced by Griffiths and Steyvers (2004), describes the document generation process under the assumptions of LDA with a Dirichlet prior as follows:

1. For $k = 1, \ldots, K$, generate the topic-specific word distribution from a Dirichlet prior with hyperparameter $\boldsymbol{\beta}$:
   (a) $\phi^{(k)} \sim Dir(\boldsymbol{\beta})$.
2. For $d = 1, \ldots, D$, choose a topic distribution $\theta^{(d)}$ from a Dirichlet prior with hyperparameter $\boldsymbol{\alpha}$.
   (a) $\theta^{(d)} \sim Dir(\alpha)$.
   (b) For each word w_i in the document d:

i. $w_i \sim Multinomial(\phi^{(z_i)})$.
ii. $z_i \sim Multinomial(\theta^{(d)})$.

The generative process described above can be summarized as:

$$P(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = P(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) P(\boldsymbol{\phi} \mid \boldsymbol{\beta}) P(\boldsymbol{z} \mid \boldsymbol{\theta}) P(\boldsymbol{w} \mid \boldsymbol{\phi}^{\boldsymbol{z}}). \tag{5}$$

Based on this model, given K topics and a vocabulary containing W unique words, the probability of the i-th word in a specific document is:

$$P(w_i) = \sum_{k=1}^{K} P(w_i \mid z_i = k) P(z_i = k). \tag{6}$$

Here, z_i is a latent variable representing the topic from which the i-th word is drawn and $P(w_i|z_i = k)$ is the probability of selecting the word w_i given that the topic is k. The term $P(z_i = k)$ is the probability of topic k being chosen to generate the i-th word in the current document. This probability varies across different documents.

Assuming there are D documents containing K topics and a global vocabulary of W unique words, $P(\boldsymbol{w}|\boldsymbol{z})$ can be interpreted as a set of K multinomial distributions $\boldsymbol{\phi} = \{\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(K)}\}$, where each $\phi^{(k)}$ is a multinomial distribution over the W words. Specifically, $\phi^{(k)}$ describes the word distribution for topic k, such that:

- $\phi^{(k)} = (P(w_1|z = k), P(w_2|z = k), \ldots, P(w_W|z = k))$.
- $\phi_i^{(k)} = P(w_i|z = k)$.
- $\sum_{i=1}^{W} \phi_i^{(k)} = 1$.

When representing $\phi^{(k)}$ in matrix form, each $\phi^{(k)}$ is a 1-dimensional vector of length W. With K such vectors, the matrix $\boldsymbol{\phi}$ has dimensions $K \times W$.

$P(\boldsymbol{z})$ can be interpreted as a set of D multinomial distributions over the K topics, such that for a document d, the probability of selecting topic k is given by $P(z = k) = \theta_k^{(d)}$.

Here,

- $\boldsymbol{\theta^{(d)}} = \{\theta_1^{(d)}, \theta_2^{(d)}, \ldots, \theta_K^{(d)}\}$.
- $\sum_{i=1}^{K} \theta_k^{(d)} = 1$.
- $\theta^{(d)}$ is a 1-dimensional vector of length K, representing the topic mixture for document d.
- With D such vectors, the matrix $\boldsymbol{\theta}$ has dimensions $D \times K$.

2.2 Inference and Parameter Estimation

In the previous section, I presented two probabilistic models for LDA. In this section, I discussed the inference and estimation methods for both models. In the subsection Inference and Variational EM Algorithm, I detailed the inference and estimation method for the LDA model proposed by Blei et al. (2003). In the subsection Inference and Collapsed Gibbs Sampling, I described the inference method for the LDA model proposed by Griffiths and Steyvers (2004).

2.2.1 Inference and Variational EM Algorithm

The inference goal for the LDA model is to compute the posterior distribution of the latent topic structure, specifically $p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$.

$$p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\mathbf{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})}. \tag{7}$$

$\boldsymbol{\theta}$ is the the document-topic distribution, $\boldsymbol{z}$ is the topic assignments for each word, $\boldsymbol{w}$ is the observed set of words in the documents, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the hyperparameters. However, this distribution cannot be computed directly. Therefore, the authors introduced a method called variational inference. The core idea of variational inference, as described by Blei and Jordan (2003), is to use Jensen's inequality to derive an adjustable lower bound on the log-likelihood. This method approximates complex distributions by selecting from a family of simpler, tractable distributions. The family of distributions used in variational inference can be characterized as follows:

$$q(\boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{\phi}, \boldsymbol{\gamma}) = q(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) \prod_{n=1}^{N} q(z_n \mid \phi_n). \tag{8}$$

where,

- the Dirichlet parameter $\boldsymbol{\gamma}$ and the multinomial parameters $(\phi_1, \ldots, \phi_N)$ are the free variational parameters.
- $q(\boldsymbol{\theta}|\boldsymbol{\gamma})$ is the variational distribution for $\boldsymbol{\theta}$, the document-topic distribution, which is modeled as a Dirichlet distribution parameterized by $\boldsymbol{\gamma}$.
- $q(z_n|\phi_n)$ is the variational distribution for z_n, the topic assignment for the n-th word, which is modeled as a multinomial distribution parameterized by ϕ_n.

The inference task can then be transformed into an optimization task, with the goal of finding $(\boldsymbol{\gamma}^*, \boldsymbol{\phi}^*)$ that minimize the Kullback-Leibler (KL) divergence between the variational distribution and the target distribution. The optimization problem is formed as:

$$(\boldsymbol{\gamma}^*, \boldsymbol{\phi}^*) = \arg\min_{(\boldsymbol{\gamma}, \boldsymbol{\phi})} D\big(q(\theta, \mathbf{z} \mid \boldsymbol{\gamma}, \boldsymbol{\phi}) \parallel p(\theta, \mathbf{z} \mid \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})\big). \tag{9}$$

According to the derivation in the original paper, applying Jensen's inequality yields a lower bound on the log-likelihood with respect to the defined variational distribution:

$$\log p(\boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) \geqslant \mathbb{E}_q\big[\log p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})\big] - \mathbb{E}_q\big[\log q(\boldsymbol{\theta}, \boldsymbol{z})\big] := L(\boldsymbol{\gamma}, \boldsymbol{\phi} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{10}$$

The difference between the right-hand side and the left-hand side in Equation 10 is the KL divergence between the variational posterior probability and the true posterior probability. Therefore, the formula can be rewritten as:

$$\log p(\boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = L(\boldsymbol{\gamma}, \boldsymbol{\phi} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) + D\big(q(\theta, \mathbf{z} \mid \boldsymbol{\gamma}, \boldsymbol{\phi}) \parallel p(\theta, \mathbf{z} \mid \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})\big), \tag{11}$$

This shows that maximizing the lower bound $L(\boldsymbol{\gamma}, \boldsymbol{\phi} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$ is equivalent to minimizing the KL divergence between the variational posterior probability and the true posterior probability. Hence, the optimization goal in Equation 9 can be replaced by maximizing $L(\boldsymbol{\gamma}, \boldsymbol{\phi} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})$.

By taking derivatives with respect to $L(\boldsymbol{\gamma}, \boldsymbol{\phi} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})$ and setting the derivatives to zero, the following pair of update equations are derived:

$$\phi_{d,n,k} = \beta_{k,w_{dn}} \cdot \exp(\Psi(\gamma_{d,k}) - \Psi(\sum_{k}^{K} \gamma_{(d,k)})) \tag{12}$$

$$\gamma_d = \alpha + \sum_{n}^{N} \phi_{d,n,\cdot} \tag{13}$$

$$\beta_{k,w} = \sum_{d}^{D} \sum_{n}^{N} \phi_{d,n,k} * W_{d,n}^{w} \tag{14}$$

Where,

- Ψ is the first derivative of the logarithm of the Gamma function, which can be computed using Taylor approximations.
- $\phi_{d,n,k}$ is the variational distribution to approximate topic-word distribution for each document.
- $\beta_{k,w}$ is the hyperparameter for topic-word distribution.
- $\gamma_{d,k}$ is the variational distribution to approximate document-topic distribution.
- α is the hyperparameter for document-topic distribution.
- $W_{d,n}^{w}$ is the word counts for each word in each document.

The variational EM algorithm for estimating LDA is as follows:

1. Repeat until convergence:
   (a) (E-step) Update $(\boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\phi}^{(t+1)}) = \arg\max_{(\boldsymbol{\gamma},\boldsymbol{\phi})} L(\boldsymbol{\gamma}, \boldsymbol{\phi} | \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)})$
   (b) (M-step) Update $(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}) = \arg\max_{(\boldsymbol{\alpha},\boldsymbol{\beta})} L(\boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\phi}^{(t+1)} | \boldsymbol{\alpha}, \boldsymbol{\beta})$

For $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$, the updating equations are shown in Equation 12, 13 and 14. For updating $\boldsymbol{\alpha}$, I followed the same method as described in Blei et al. (2003), using a linear-time Newton-Raphson algorithm. To summarize, the variational EM algorithm for solving LDA is as follows:

Algorithm 1: Variational EM Algorithm

Input : Corpus of documents with words

Output: α, β, γ, ϕ

begin

  Initialize:

    W: Number of unique words in the corpus

    K: Number of topics

    N: Number of words in each document

    D: Number of documents in the corpus

    α: Hyperparameter for document-topic distribution

    $\beta_{k,w}$: Hyperparameter for topic-word distribution

    $\phi_{d,n,k}$: Variational distribution to approximate topic-word distribution for each document

    $\gamma_{d,k}$: Variational distribution to approximate document-topic distribution

  repeat

    `/* E-Step */`

    for $d = 1 \to D$ **do**

      for $n = 1 \to N$ **do**

        for $k = 1 \to K$ **do**

          $\phi_{d,n,k}^{(t+1)} = \beta_{k,w_{dn}} \cdot \exp(\Psi(\gamma_{d,k}^{(t)}) - \Psi(\sum_k^K \gamma_{(d,k)}^{(t)}))$

        Normalize $\phi_{d,n,\cdot}^{(t+1)}$ to sum to 1.

      $\gamma_d^{(t+1)} = \alpha^{(t)} + \sum_n^N \phi_{d,n,\cdot}^{(t+1)}$

    `/* M-Step */`

    Update α using Newton-Raphson algorithm based on optimized γ values;

    for $k = 1 \to K$ **do**

      for $w = 1 \to W$ **do**

        $\beta_{k,w}^{(t+1)} = \sum_d^D \sum_n^N \phi_{d,n,k}^{(t+1)} * \text{word\_count}_{d,n}^w$

    Normalize β

  until *convergence of lower bound for joint log likelihood*;

return α, β, γ, ϕ

2.2.2 Inference and Collapsed Gibbs Sampling

For the model of LDA proposed by Griffiths and Steyvers (2004), the goal is to solve:

$$P(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z} \mid \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{P(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})}{P(\boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})}. \tag{15}$$

However, the distribution is intractable to compute directly. One solving method is to use Gibbs sampling. Gibbs sampling (Geman and Geman (1984)) is a Markov Chain Monte Carlo (MCMC) algorithm used to approximate the joint distribution of random variables when direct sampling is difficult. It works by iteratively sampling each variable from its conditional distribution, given the current values of the other variables. Over time, the sequence of samples generated by Gibbs sampling converges to the target joint distribution (Geman and Geman (1984)).

For example, if directly sampling $\boldsymbol{x}$ from its joint distribution $p(\boldsymbol{x}) = p(x_1, \ldots, x_n)$ is impossible, but the conditional distributions $p(x_i|\boldsymbol{x_{-i}})$ are available, the Gibbs sampling procedure, as described by Gilks et al. (1995), updates each variable iteratively as follows:

1. Randomly initiate $x_1, \ldots, x_n$.

2. For iterations $t = 1, \ldots, T$:

   (a) $x_1^{(t+1)} \sim P(x_1 | x_2^{(t)}, \ldots, x_n^{(t)})$

   (b) $\ldots$

   (c) $x_n^{(t+1)} \sim P(x_n | x_1^{(t+1)}, \ldots, x_{n-1}^{(t+1)})$

To apply this algorithm to the LDA model, the full conditional distribution $P(z_i = k | \boldsymbol{z_{-i}}, \boldsymbol{w})$ is needed. By Griffiths and Steyvers (2004), marginalizing $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the posterior probability $P(z_i = k | \boldsymbol{z_{-i}}, \boldsymbol{w})$ for each document d can be approximate by:

$$P(z_i = k \mid \boldsymbol{z_{-i}}, \boldsymbol{w}) \propto \frac{n_{-i,k}^{(w_i)} + \boldsymbol{\beta}}{n_{-i,k}^{(\cdot)} + W\boldsymbol{\beta}} \cdot \frac{n_{-i,k}^{(d)} + \boldsymbol{\alpha}}{n_{-i,\cdot}^{(d)} + K\boldsymbol{\alpha}}. \tag{16}$$

Where, $n_{-i,k}^{(w_i)}$ is the counts of word w_i assigned to topic k, excluding the counts of w_i word in document d. $n_{-i,k}^{(\cdot)}$ is the total number of words assigned to topic k across all documents, excluding the counts of current word in document d. $n_{-i,k}^{(d)}$ is the number of words in document d that are assigned to topic k, excluding the counts of current word. $n_{-i,\cdot}^{(d)}$ is the total number of words in document d across all topics, excluding the counts of current word.

After getting a set of samples from posterior distribution $P(\boldsymbol{z}|\boldsymbol{w})$, the $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be computed as:

$$\hat{\phi}_k^{(w)} = \frac{n_k^{(w)} + \boldsymbol{\beta}}{n_k^{(\cdot)} + W\boldsymbol{\beta}}. \tag{17}$$

$$\hat{\theta}_k^{(d)} = \frac{n_k^{(d)} + \boldsymbol{\alpha}}{n_\cdot^{(d)} + K\boldsymbol{\alpha}}. \tag{18}$$

Where $\hat{\phi}_k^{(w)}$ is the estimated probability of word w from topic k. $n_k^{(w)}$ is the counts of word w in topic k and $n_k^{(\cdot)}$ is the counts of all words in topic k. $\hat{\theta}_k^{(d)}$ is the estimated probability of assigning topic k to document d, where $n_k^{(d)}$ is the counts of words assigned to topic k in document d and $n_\cdot^{(d)}$ is the counts of all words in document d.

My implementation pseudocode is as follows:

Algorithm 2: LDA Collapsed Gibbs Sampling

Input : Corpus of documents with words
K: Number of topics
α: Dirichlet prior for document-topic distribution
β: Dirichlet prior for topic-word distribution
n_iter: Number of iterations

Output: Topic assignments $\mathbf{z}$, Counts $C_{d,k}$, $C_{k,w}$, C_k

begin
  Initialize:
    W: Number of unique words in the corpus
    K: Number of topics
    N: Number of words in each document
    D: Number of documents in the corpus
    $C_{kw}[k, w]$: Number of times word w assigned to topic k
    $C_{dk}[d, k]$: Counts of words in document d assigned to topic k
    $C_k[k]$: Counts of words assigned to topic k
    Randomly assign initial topics $\mathbf{z}$ to words;
  foreach *iteration* **do**
    for $d = 1 \rightarrow D$ **do**
      for $i = 1 \rightarrow N$ **do**
        $\texttt{word} \leftarrow \mathbf{w}[i]$;
        $\texttt{topic} \leftarrow \mathbf{z}[i]$;
        $C_{d,k}[d, k] -= 1$;
        $C_{k,w}[k, w] -= 1$;
        $C_k[k] -= 1$;
        for $k = 1 \rightarrow K$ **do**
          $p(z = k \mid \cdot) = \frac{(C_{d,k}[d,k]+\alpha)*(C_{k,w}[k,w]+\beta)}{(\sum(C_{d,k}[d,:])+K*\alpha)*(C_k[k]+W*\beta)}$;
        $\texttt{topic} \leftarrow$ sample from $p(z \mid \cdot)$;
        $\mathbf{z}[i] \leftarrow \texttt{topic}$;
        $C_{d,k}[d, \texttt{topic}] += 1$;
        $C_{k,w}[\texttt{topic}, w] += 1$;
        $C_k[\texttt{topic}] += 1$;
return $\mathbf{z}, C_{dk}, C_{kw}, C_k$;

3 Applications and Empirical Results

In this section, I presented the empirical evaluations of the collapsed Gibbs sampling and variational EM algorithms in LDA across 2 problem domains: topic modeling and document classification. I chose these domains for 2 primary reasons.

First, they have broad and impactful real-world applications. For example, topic modeling is widely used for mining customer reviews, identifying trends in research publications, and analyzing social media content. Document classification is essential for tasks such as sentiment analysis and news categorization. LDA can serve as a foundational model for topic modeling and can also be employed as a dimensionality reduction technique to enhance document classification.

Second, the purpose of this section was to evaluate the generalization performance of the 2 models trained by 2 algorithms and to assess the efficiency of LDA as a dimensionality reduction method. In the topic modeling study, I focused on comparing the generalization performance and training efficiency (measured by training

time) of the 2 algorithms. In the document classification study, I compared the prediction accuracy of an SVM classifier with 2 different inputs: raw word features and features derived from LDA with collapsed Gibbs sampling. This comparison aimed to assess how much discriminatory information was retained or lost when using LDA to reduce the dimensionality of document representations.

3.1 Data Preprocessing Description

I used the BBC news article dataset from Kaggle for my project [1]. The dataset covered 5 topics: politics, sports, technology, business, and entertainment.

For the topic modeling study, I split the dataset into two parts, allocating 80% of the data to the training set and 20% to the testing set. The training set consisted of 1,702 documents of text data and the testing set consisted of 425 documents. For the document classification study, I varied the training set proportion from 5% to 75% to evaluate the performance of SVM classifier with 2 types of inputs under different sizes of training data.

The preprocessing steps for both studies were consistent, ensuring uniform data preparation across the analyses. The entire project was implemented using Python.

To prepare the text data for topic modeling, I used a structured data preprocessing pipeline. This pipeline included text cleaning and tokenization, lemmatization and stopwords removal. Below is a step-by-step description of this process:

1. Clean and tokenize text.
   (a) Convert all text data to lowercase format.
   (b) Use NLTK's word_tokenize function to split the text into individual words (tokens).
2. Lemmatize tokens using NLTK's WordNetLemmatizer. This step is to convert each token to its base or root form, also known as its lemma. The purpose is to ensure that variations of a word (such as "eating," "ate," and "eats") are treated as the same term ("eat"), which can improve the quality of analysis and help the model focus on the core meaning of words.
3. Drop stopwords using NLTK's stopwords list. Stopwords are commonly used words (like "you", "and", "me") that carry little meaning and can add noise to text data. In addition to the stopwords from NLTK, I reviewed the news dataset and removed additional commonly used non-informative words in my dataset, such as "first," "news," "day," and others. This step helps further reduce noise and focus on more meaningful content.

3.2 Implementation Verification and Concept Illustration

Before conducting the 2 studies, I first verified the correctness of my implementations for the 2 algorithms (variational EM and collapsed Gibbs sampling) by using 2 example cases. To further illustrate the concept of LDA model, I applied the collapsed Gibbs sampling method to a sample document and colored the words according to their potential topics.

The 2 algorithms, variational EM and collapsed Gibbs sampling, were trained on the training set, which were 80% of the total dataset. The same training dataset was used consistently throughout the subsequent topic modeling study to ensure comparability. For both methods, the number of topics was set to 5, matching the dataset's 5 labels. This alignment made it easier to check if the training results aligned with the expected topics. Below are the detailed initialization steps for variational EM.

[1] Dataset available at: https://www.kaggle.com/datasets/jacopoferretti/bbc-articles-dataset

- α: The starting point for α was initialized as a vector of size K (the number of topics) by sampling from a Gamma distribution with a shape parameter of 100 and a scale parameter of 0.01.
- β: β was initialized as a matrix where each row corresponds to a topic. Each row was sampled from a Dirichlet distribution with parameters set to a vector of ones of size W (the vocabulary size), creating a uniform prior distribution over words for each topic.
- γ: γ was initialized as a matrix of size (D, K), where D is the number of documents. Each entry in γ was derived from the initial α values, scaled by dividing the number of words in each document evenly across the K topics.
- ϕ: ϕ was initialized as a matrix with each element set to $1/K$, ensuring an equal probability distribution across topics at the start.

In the following studies, I followed the same initialization approach and steps whenever implementing the variational EM algorithm to ensure consistency and comparability.

When I first ran the variational EM algorithm with the number of topics set to 5, I found it to be quite slow without parallelization. For instance, each iteration (including both the E-step and M-step) took about 150 seconds. To prevent an excessively long runtime, I used parallelization and set the stopping criteria to 0.01 or a maximum of 100 iterations. With parallelization, the runtime per iteration reduced to about 45 seconds. The entire training process took about 1 hour and 18 minutes, reaching the 100-iteration limit before convergence.

To assess convergence, I plotted the loss function (the right-hand side of Equation 10) over iterations, as shown in Figure 1. The plot indicated that the loss function nearly converged. Additionally, I presented the training results by displaying the top 10 most frequent words for each topic in Table 1. Although the model did not fully converge, the results showed that it identified 5 distinct topics. However, there is a noticeable gap between the identified topics and the dataset's actual labels. This discrepancy may be attributed to the unsupervised nature of LDA, where the algorithm infers topics based purely on word co-occurrence patterns, without prior knowledge of the dataset's predefined labels.

"business"	"politics"	"sport"	"entertainment & tech"	"law"
market	labour	people	people	people
company	party	win	game	company
price	election	england	music	court
firm	government	player	film	case
growth	blair	world	show	law
economy	minister	time	technology	right
share	tory	play	tv	time
sale	film	second	mobile	make
bank	brown	best	phone	many
rate	people	back	use	want

Table 1: Top 10 Most Frequent Word in Each Topic: variational EM

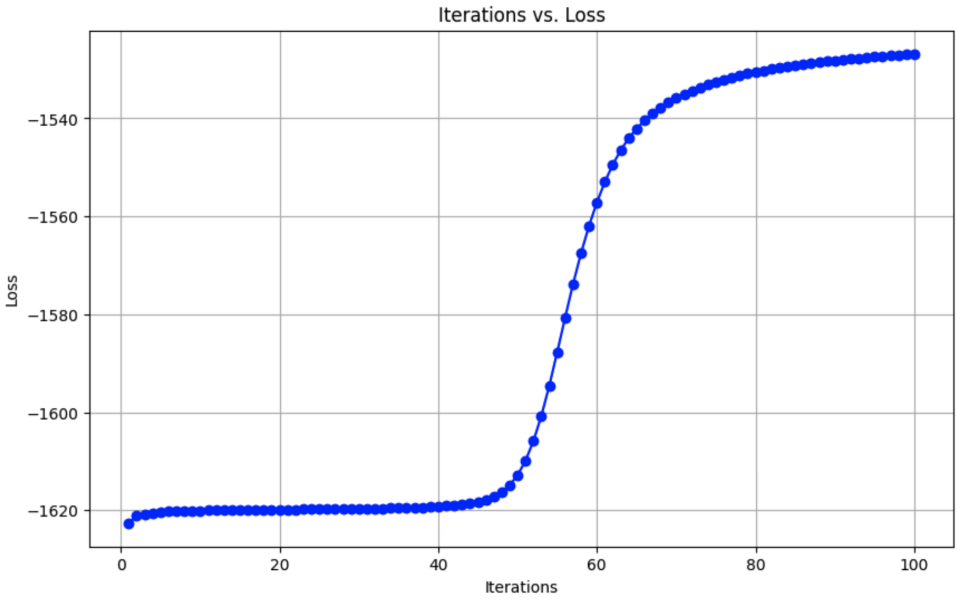

Figure 1: Loss Value per Iteration

For the collapsed Gibbs sampling method, I ran 500 iterations with $\alpha = 0.1$, $\beta = 0.1$. To present the training results, I displayed the top 10 most frequent words in Table 2. The listed words indicated that the model successfully identifies 5 hidden topics, which align well with the dataset's labels. Additionally, I found that collapsed Gibbs sampling is significantly faster than variational EM, with each iteration taking approximately 1.6 seconds. The entire training process was completed in about 16 minutes.

"business"	"politics"	"sport"	"entertainment"	"tech"
company	government	game	film	people
market	people	win	best	game
firm	labour	england	award	technology
bank	party	player	show	service
economy	election	world	star	phone
price	minister	club	music	mobile
share	blair	time	actor	user
sale	plan	play	band	net
growth	tory	back	number	computer
business	brown	team	director	make

Table 2: Top 10 Most Frequent Word in Each Topic: Collapsed Gibbs Sampling

glazer makes new man utd approach malcolm glazer has made a fresh approach to buy manchester united , which could lead to a bid valuing the premiership club at £800m . the us tycoon , who has been wooing the club for the last 12 months , has approached the united board with `` detailed proposals '' , it has confirmed . mr glazer , who owns the tampa bay buccaneers team , hopes this will lead to a formal bid being accepted . his new offer is expected to contain substantially less debt . mr glazer has already had one takeover attempt turned down by the red devils and responded by using his 28.1 % shareholding to vote off three board members last november . man united had turned down the bid because it was based on a high level of borrowing . but newspapers have speculated recently that the tycoon had gained the support of leading banks to come up with a stronger and less debt-laden bid . last week , however , mr glazer issued a statement to the stock exchange distancing himself from a new bid . meanwhile , united 's chief executive david gill said in december that talks would not resume unless glazer came up with `` definitive proposals '' . now the board has confirmed that the us bidder is back , with a statement issued on sunday reading : `` the board can confirm it has now received a detailed proposal subject to various preconditions which may form the basis of an offer . `` a further announcement will be made in due course . '' to succeed malcolm glazer will still need the approval of major shareholders john magnier and jp mcmanus , who own 28.9 % of the club . but the irish duo have cut off talks with glazer over the proposed sale of their stake and have so far made no comment on his latest approach . united fans have reacted with anger at the announcement . they have vehemently opposed any proposed takeover by glazer since he first showed interest in the club in september 2003 and after sunday 's announcement they vowed to fight on . `` we will fight tooth and nail to stop him whatever his offer says . we do not want him or anybody else taking over united , '' said mark longden of the independent manchester united supporters ' association . `` the campaign against this proposed takeover will continue as it has done since glazer first showed interest in the club . ''

Figure 2: Example Text to Show LDA Thought

To provide a clearer understanding of the concept of LDA and its working mechanism, I selected an example document from the training set. In this example, each word in the document was assigned to a specific topic based on the latent factors inferred by the LDA model using the collapsed Gibbs sampling method. I color-coded the words in the document, with each color representing a different topic from which the word was likely generated. I also applied the same color scheme to the topic names displayed in the Table 2 that list the top most frequent words for each topic.

This visualization effectively showed how a document is generated as a mixture of topics, with each topic contributing a specific proportion of words under the LDA model. By looking into this example, one can better understand LDA's generative process, where documents are modeled as random mixtures of latent topics, and each topic is characterized by a distinct probability distribution over words.

3.3 Topic Modeling

After verifying the correctness of the algorithm implementations, I conducted the topic modeling study to compare the generalization performance of the 2 algorithms. I designed experiments by keeping the training and testing datasets fixed while varying the number of topics among 2, 5, 10, 20, 50, and 100. For the variational EM algorithm, the stopping criterion was set to a convergence threshold of 0.01 or a maximum of 100 iterations. Although the algorithm might not fully converge within 100 iterations, this choice aimed to balance training accuracy and runtime. The initialization steps followed the same approach as previously described. For the collapsed Gibbs sampling method, the model was trained for a fixed 500 iterations with $\alpha = 0.1$ and $\beta = 0.1$.

In the topic modeling study, the documents in the corpora are treated as unlabeled. Therefore, following the approach outlined by Blei et al. (2003), I used perplexity as a criterion to evaluate the performance of the models. For a test set of M documents, the perplexity is defined as:

$$perplexity(D_{test}) = exp\{-\frac{\sum_{d=1}^{D}\sum_{n=1}^{N_d} log\ p(w_{d,n})}{\sum_{d=1}^{D} N_d}\}$$

Where, N_d is the number of words in each document, D is the number of documents, $p(w_{d,n})$ is the likelihood of the n-th word in document d given the model.

The resulting perplexity scores for both methods across the different topic numbers are shown in Figure 3. Due to the long training time of the variational EM method, I only trained the model for cases with 2, 5, and 10 topics. From the plot, it can be seen that for collapsed Gibbs sampling, the perplexity score reaches its lowest point when the number of topics is 50. In the cases with 5 and 10 topics, variational EM demonstrated better performance than collapsed Gibbs sampling, as indicated by its lower perplexity scores, even though the model had not yet reached the convergence criterion.

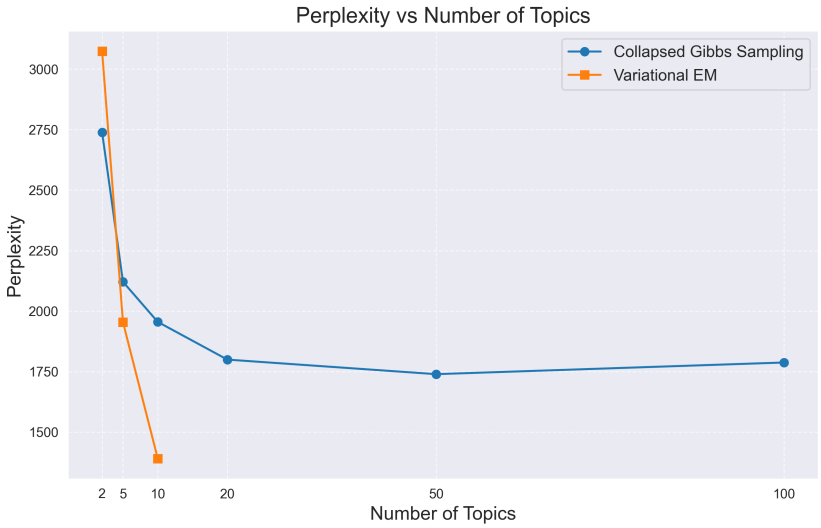

Figure 3: Perplexity Scores

I also summarized the training times for both methods in Table 3. While the perplexity scores for variational EM were lower, its significantly longer training time poses a limitation when considering the method for practical applications, especially when the corpus is large or number of topics is large.

Number of Topics	**2**	**5**	**10**	**20**	**50**	**100**
Training Time (collpased Gibbs sampling)	13:45	13:57	14:04	14:23	15:12	16:39
Training Time (variational EM)	40:45	1:18:21	2:19:24			

Table 3: Training Time for Both Algorithms

3.4 Document Classification

In the document classification study, my objective was to classify each document into 1 of 5 categories based on their news labels. A key challenge in document classification is selecting appropriate feature representations. A straightforward approach is to represent individual words as features, which provides a rich but extremely large feature set (Joachims (1999)). Another approach is to use LDA model as a dimensionality reduction technique to create more compact feature representations. In this section, I explored 2 different inputs for the SVM model in the text classification task—raw word counts and topic distributions generated using collapsed Gibbs sampling in LDA. I varied the proportion of the training set to 5%, 10%, 25%, 40%, 50%, and 75% while fixing the number of topics at 10 to compare the prediction accuracy of the SVM using 2 different inputs. This comparison provided insights into how much discriminatory information is lost when reducing the document representation using the LDA model.

The following table provides a detailed comparison between the raw word input and the LDA input for the SVM model:

Input Type	**Raw Word Input**	**LDA Input**
Dimension	(1595, 8000)	(1595, 10)
Explanation	- 1595: Number of documents in the training set. - 8000: Number of features (word counts). - Top 8000 most frequent words are selected as input features. - Each element represents the counts for each word in each document.	- 1595: Number of documents in the training set. - 10: Number of topics. - Each element represents the probability of the document being generated from a specific topic.

Table 4: Comparison Between the Raw Word Input And the LDA Input (Using 75% Training Set As Example).

I followed the workflow below to perform SVM classification using LDA-generated inputs.

- **Step1: Train the LDA Model Using Collapsed Gibbs Sampling.**
  - Train the LDA model on the training set using collapsed Gibbs sampling.
  - Obtain the following counts:
    - C_{dk}^{train}: Document-topic counts, representing the number of words in each document assigned to each topic.
    - C_{kw}^{train}: Topic-word counts, representing the number of times each word is assigned to each topic.
    - C_{k}^{train}: Counts of words assigned to topic k.
  - Compute the document-topic distributions $\hat{\theta}_k^{(d)}$ for the training set based on C_{dk}^{train} and $\alpha = 0.1$.
- **Step 2: Extract Test Features.**
  - For the test set, compute the document-topic distributions $\hat{\theta}_k^{(d)}$ for the testing set using the trained LDA model's C_{kw}^{train}, C_{k}^{train} and $\alpha = 0.1$.
  - Use these distributions as feature representations for the test documents.
- **Step 3: Train and Evaluate the SVM Model:**
  - Use the document-topic distributions $\hat{\theta}_k^{(d)}$ from the training set as input features for the SVM model.
  - Train the SVM on the training features $\hat{\theta}_k^{(d)}$ and their corresponding labels.
  - Evaluate the SVM's performance on the test features $\hat{\theta}_k^{(d)}$ and their associated labels.

For the SVM model with raw word input, I used the same preprocessed, clean text as that used for the LDA model and used CountVectorizer from the Python sklearn package to transform it into a matrix form, which was then used as input for training the SVM model.

For both methods, the SVM model was implemented using the Python sklearn package.

Figure 4 showed classification results for both types of inputs. The prediction accuracies were shown in Table 5. From the results, it can be seen that using LDA-based features leads to only a minimal reduction in classification performance compared to raw word input. Overall, both input types achieve similar prediction accuracy, while the LDA model significantly reduces the input dimensionality. Notably, when the training set proportion is small, such as 5% or 10%, the LDA-based input outperforms the raw word input. These findings suggest that LDA can serve as an effective and efficient filtering method for feature selection in text classification tasks, especially when the amount of training data is limited.

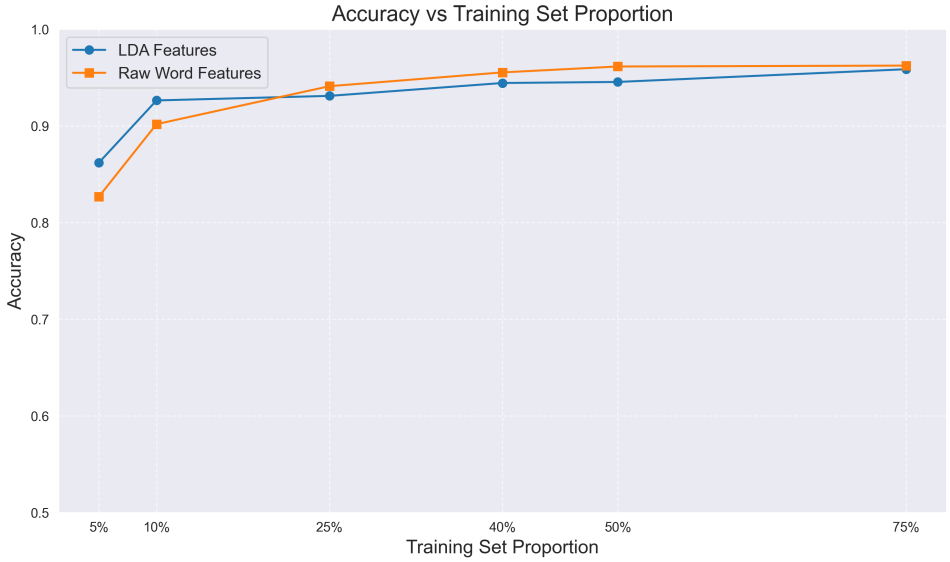

Figure 4: Classification Results

Training Set Proportion	**5%**	**10%**	**25%**	**40%**	**50%**	**75%**
Prediction Accuracy (LDA Input)	86.20%	92.64%	93.11%	94.44%	94.55%	95.86%
Prediction Accuracy (Raw Word Input)	82.68%	90.18%	94.11%	95.53%	96.15%	96.24%

Table 5: Prediction Accuracies for Both Types of Input

4 Conclusion

This project focused on studying the LDA model and its parameter estimation methods, variational EM and collapsed Gibbs sampling. Two studies were conducted to explore the applications of the LDA model in topic modeling and document classification. The key findings from this project are as follows:

1. Optimization Methods: The variational EM algorithm provides the advantage of a calculable convergence rate, which is not available in collapsed Gibbs sampling due to the lack of formal convergence criteria. However, during my implementation, I found that the variational EM algorithm is computationally expensive. For example, with the number of topics set to 5 and a training corpus containing 1702 documents and 21,927 unique words in the vocabulary, variational EM required approximately 1 hour and 18 minutes for 100 iterations without reaching the convergence criterion of 0.01. In contrast, collapsed Gibbs sampling completed 500 iterations in just 16 minutes, demonstrating its superior computational efficiency.

2. Generalization Performance: The generalization performance of both methods was similar, indicating that either method can be effectively used depending on computational constraints and requirements.

3. Feature Selection and Dimensionality Reduction: In the document classification study, the LDA model demonstrated its potential as an efficient dimensionality reduction method for feature selection in text classification tasks. This was especially evident when the training data was limited, as LDA-based features achieved comparable classification performance to raw word features while significantly reducing input dimensionality.

In summary, this project offered a practical exploration of the computational trade-offs and applications of the LDA model and its optimization methods. The findings support the use of LDA as a tool for both topic modeling and text classification tasks. If the variational EM method is chosen, implementing parallelization or using advanced data engineering and coding techniques is recommended to improve runtime efficiency.

References

Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Geman, S. and Geman, D. (1984). Geman, d.: Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. ieee trans. pattern anal. mach. intell. pami-6(6), 721-741. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1995). *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics. Taylor & Francis.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235.

Joachims, T. (1999). Making large scale svm learning practical. *Advances in Kernel Methods: Upport Vector Machines*.