towardsdatascience.com /topic-modeling-with-latent-dirichlet-allocation-e7ff75290f8

Topic Modeling with Latent Dirichlet Allocation - Towards Data Science

Haaya Naushan ⋮ 26-33 minutes ⋮ 12/3/2020

A practical exploration of the Natural Language Processing technique of Latent Dirichlet Allocation and its application to the task of topic modeling.

tds

Topic modeling is a form of unsupervised machine learning that allows for efficient processing of large collections of data, while preserving the statistical relationships that are useful for tasks such as classification or summarization. The goal of topic modeling is to uncover latent variables that govern the semantics of a document, these latent variables representing abstract topics. Currently, the most popular technique for topic modeling is Latent Dirichlet Allocation (LDA), and this model can be used effectively on a variety of document types such as collections of news articles, policy documents, social media posts or tweets.

This article will necessarily and briefly mention precursive topic modeling techniques, such as Latent Semantic Indexing (LSI, also referred to interchangeably as Latent Semantic Analysis/LSA) and probabilistic Latent Semantic Indexing (pLSI). The main focus will be a discussion of the LDA model, with an emphasis on understanding the role of hyperparameters and the challenge of inference. Next, I offer a practical introduction to implementation, covering dataset requirements, fine-tuning, and evaluation. Lastly, I conclude with a discussion of the limitations of the LDA model.

Foundation of modern Topic modeling

A brief history and a theoretical understanding of the foundational techniques that preceded LDA will explain the importance of the improvements made by this modern technique. Back in the 1980's the field of information retrieval produced a text representation scheme called term frequency–inverse document frequency (tf-idf) for applying a numerical statistic to a word that would be representative of its importance within a document (Salton and McGill, 1983).

Tf-idf vectorization is where a normalized term frequency (count of the number of occurrences of a term within a document) is compared to the normalized inverse document frequency (count of the number of occurrences of a term within a corpus) on a log scale. This results in a term-by-document weight matrix that is unfortunately very sparse, noisy and redundant. The process to calculate the tf-idf vector for any word in a document can be represented by the equation below.

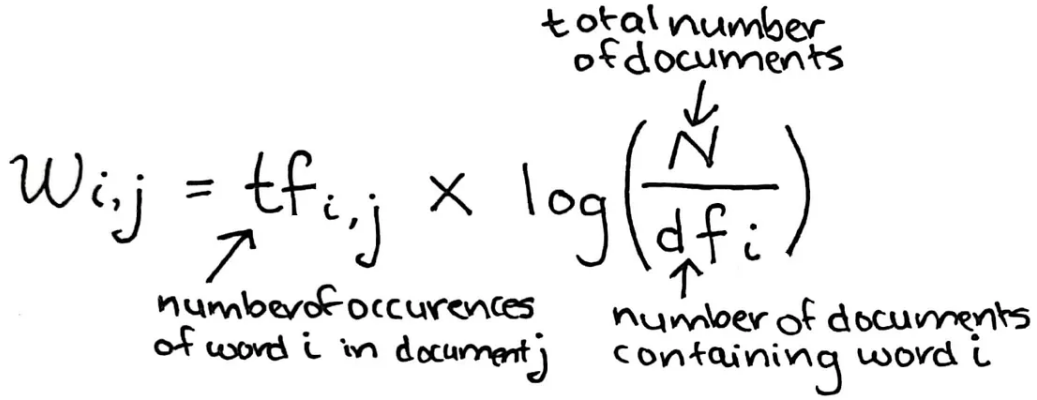

Figure 1. Equation to calculate a tf-idf vector for a term $w_{i,j}$ *. Image by Author.*

Latent Semantic Indexing

Using tf-idf as a dimensionality reduction technique identifies discriminative words for a collection of documents; however, it does not expose the inter- or intra- document statistical structure. Therefore, Latent Semantic Indexing

(Deerwester et al., 1990) was introduced as a way of overcoming the deficits of tf-idf. By using singular value decomposition (SVD) to accomplish dimensionality reduction of the term-document matrix, it is possible to identify the linear subspace within the tf-idf features that captures most of the variance in a collection of documents, allowing LSI to be a generative rather than discriminative process.

The SVD process can be visualized by the diagram below, where term-document matrix *A* is factored into a product of three matrices: *U*, *S*, and *V***T**. The rows of *U* represent document vectors in terms of topics and the rows of *V* represent term vectors in terms of topics.

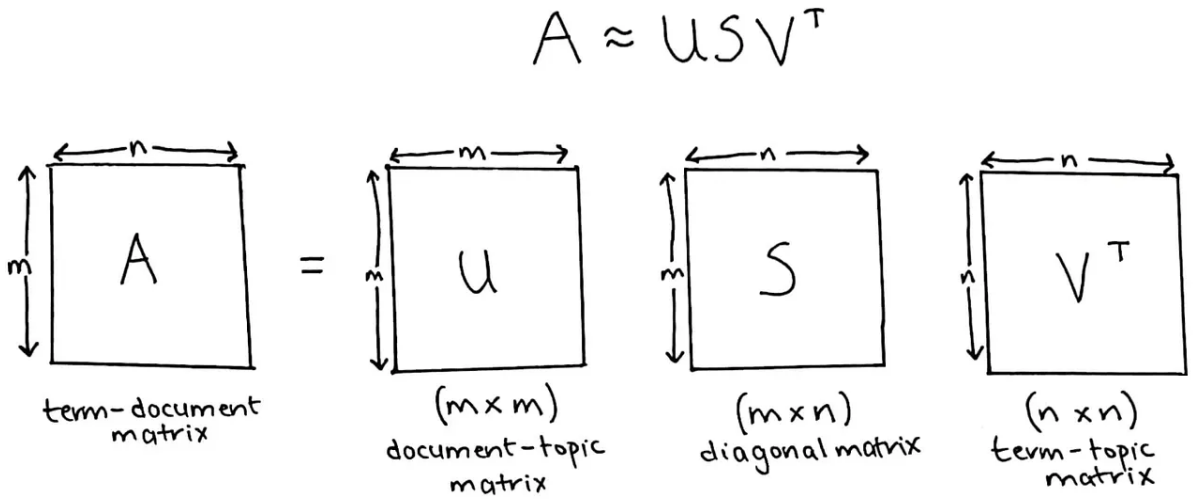

Figure 2. Singular value decomposition of term-document matrix *A*, where *U* is an orthogonal document-topic matrix, *S* is a diagonal matrix of singular values of *A*, and *V* is the transpose of the orthogonal and represents a term-topic matrix. Image by Author.

LSI, however, requires a special type of SVD, specifically a low-rank approximation of the matrix A is required for efficiency, therefore truncation based on the Eckart-Young Theorem (specifically the proof for the Frobenius norm) is used to build an approximate matrix. In application, a value *k* is the hyperparameter used to represent the number of topics desired, and truncated SVD is used to select only the *k* columns of *U* and *V*. The selected singular values from the diagonal *S* matrix are represented by "$\sigma_1 \ldots \sigma_k$", where $\sigma_k \geq 0$ and σ_1 has the highest importance. Truncated SVD of matrix A, can be visualized as follows:

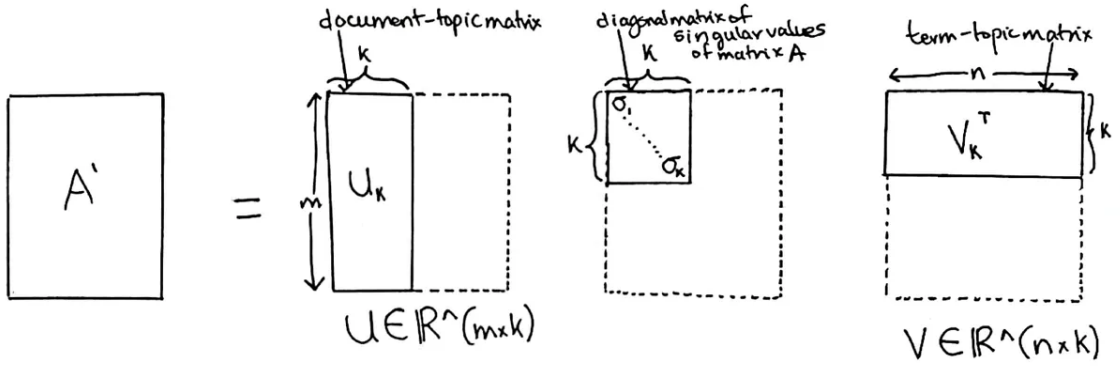

Figure 3. Latent Semantic Indexing, where term-document matrix A is transformed by truncated singular value decomposition. Image by Author.

The derived features of LSI are linear combinations of the original tf-idf features, thereby capturing some aspects of linguistic notions of synonymy and polysemy. Nonetheless, LSI unfortunately requires a very large collection of documents and terms, in addition to lacking interpretability with regards to topic content and sentiment.

Probabilistic Latent Semantic Indexing

Instead of relying solely on truncated SVD, a probabilistic model of LSI was introduced as an alternative, referred to as pLSI (Hoffman, 1999). Each word in a document is sampled from a mixture model, where the mixture components are multinomial random variables that can be viewed as representative of topics. Hence, each word is generated from a single topic, and different words within a document can be generated from different topics. Essentially, each document is reduced to a probability distribution on a fixed set of topics. This probabilistic approach can be modeled by a set of equivalent equations representing the joint probability of a document with a word, P(D,W). Furthermore, as seen in the diagram below, these equations can be derived from the truncated SVD model of LSI.

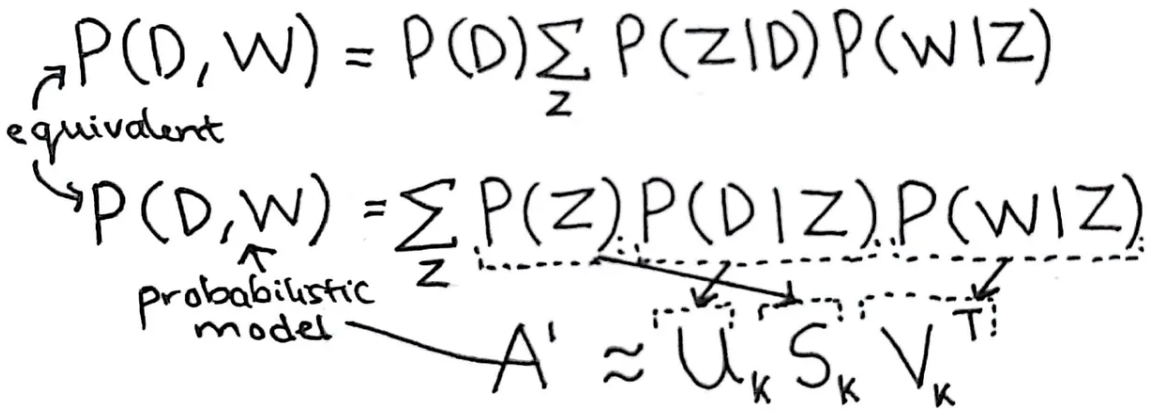

Figure 4. Pair of equivalent joint probability equations, where the generative process starting with a topic P(*z*) is representative of a probabilistic model of matrix A from LSI. Image by Author.

In the set of equations above for P(D,W), the joint probability parameters are multinomial distributions that can be trained by an expectation-maximization algorithm (EM algorithm) for inference of parameter estimates which depend on unobserved, latent variables. Importantly, pLSI only applies a probabilistic treatment to topics and words, not documents. This leads to two problems, firstly the number of parameters for pLSI grows linearly with the size of the corpus, so it is prone to overfitting. Secondly, there are no parameters to model P(D), so it is not evident how probability would be assigned to a new document. These issues led to the development of the LDA model, which allowed for better generalization and will be discussed in the following section.

Latent Dirichlet Allocation (LDA)

LDA has roots in evolutionary biology; back in 2000 researchers developed this model for the study of population genetics. A few years later, LDA was applied to the field of machine learning by Blei et al., 2003, a group that includes the renowned Andrew Ng. LDA is a generative probabilistic model, specifically it is a three-level hierarchical Bayesian model, for a collection of discrete data (such as a text corpora). LDA can be thought of as a Bayesian version of pLSI, that overcomes the weakness of the latter and thus allows for better generalization.

The theoretical underpinnings of LDA rely on exploiting the concepts of exchangeability with the de Finetti theorem (1990). Exchangeability is a major simplifying assumption of text processing that allows for computationally-

efficient methods. Both LSI and pLSI and based on the fundamental probability assumption described by the "bag-of-words" method whereby the order of words in a document can be ignored. This assumption of exchangeability extends to the treatment of documents, where one can assume that the specific order of documents in a corpus is not an important consideration.

According to de Finetti's theorem, exchangeable observations are conditionally independent relative to a latent variable, therefore an epistemic probability can be assigned to the latent variable. Furthermore, any collection of exchangeable random variables, also referred to as an exchangeable sequence of Bernoulli random variables, has a representation as a "mixture" distribution, specifically a mixture of sequences of independent and identically distributed (*i.i.d.*) Bernoulli random variables. The implication of a mixture model is the possibility of probabilistically representing the presence of sub-populations within an overall population without requiring an observed dataset to identify the sub-populations. Essentially, utilizing de Finetti's theorem, it is possible to capture significant intra-document statistical structure via the mixture distribution.

Unpacking the LDA model

The innovation of LDA is in using Dirichlet priors for document-topic and term-topic distributions, thereby allowing for Bayesian inference over a three-level hierarchical model, the third layer being the distinguishing feature in a comparison to a simple Dirichlet multinomial clustering model. With regards to Bayesian inference, plate notation is an intuitive method of graphically representing variables that repeat; a "plate" (ie. box) is used to represent replicates, and edges denote conditional dependencies. As seen in the diagram below, the outer plate represents documents, and the inner plate represents repeated choices of topics and words within a document.

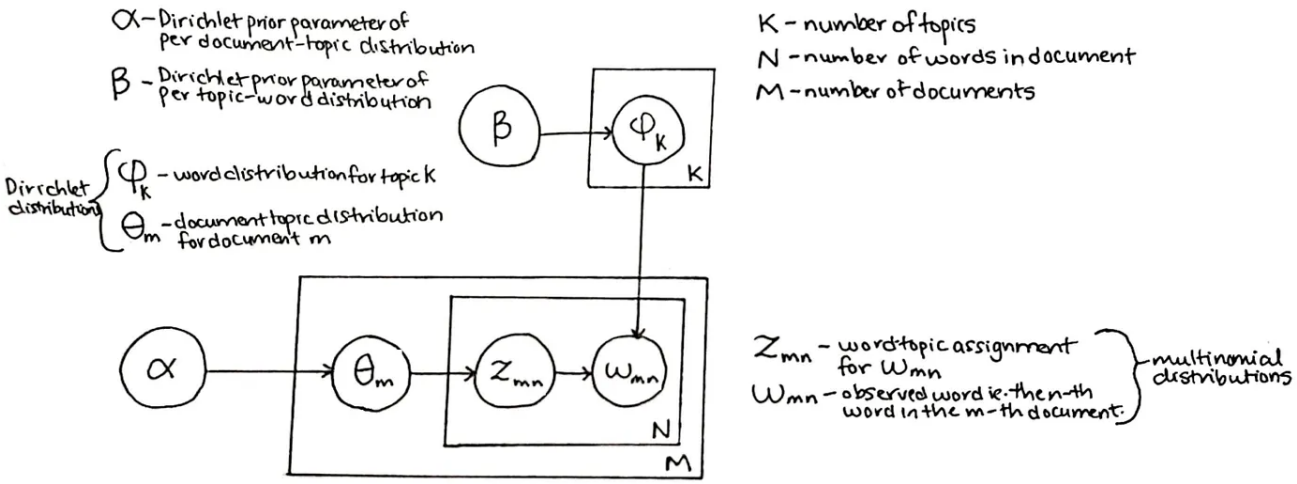

Figure 5. Probabilistic graphical representation of the LDA model with plate notation. Image by Author.

Dirichlet distributions allow for probability distribution sampling over a probability simplex in which all the numbers add up to 1, and these numbers represent probabilities over *K* distinct categories. A *K*-dimensional Dirichlet distribution has *k*-parameters and represents uncertainty as a probability distribution. The Dirichlet prior parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are corpus-level parameters that are sampled once in the process of generating a corpus, and parameter $\boldsymbol{\Theta}_m$ is a document-level variable that is sampled once per document. Variables z_{mn} and w_{mn} are word-level variables that are sampled once for each word in each document. Lastly, $\boldsymbol{\varphi}_k$ represents the word probability distribution for a topic *k*.

Procedure and corpus probability

With the LDA model, documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. This generative process for each document within a corpus can be written simply as follows:

Draw each topic parameter $\boldsymbol{\beta}_k \sim$ Dirichlet ($\boldsymbol{\varphi}$), for $k \in \{1 \ldots K\}$

For each document:

1. Choose the topic distribution $\boldsymbol{\Theta}_m \sim$ Dirichlet ($\boldsymbol{\alpha}$)
2. For each of the *N* words w_n :

a) Choose a topic $z_{mn} \sim \text{Multinomial}(\boldsymbol{\Theta}_m)$

b) Choose a word $w_n \sim \text{Multinomial}(\boldsymbol{\beta}_k)$

The probability of a corpus (D) is the result of taking the product of the marginal probabilities of single documents, and the marginal distribution for a single document is obtained by integrating over $\boldsymbol{\Theta}$ and summing over z topics.

$$P(D|\alpha,\beta) = \prod_{m=1}^{M} \int P(\theta_m|\alpha)\left(\prod_{n=1}^{N_m} \sum_{z_{mn}} P(z_{mn}|\theta_m)P(w_{mn}|z_{mn},\beta)\right) m\theta_m$$

Figure 6. From the LDA model, the probability of a corpus D is modeled by the probability equation $P(D|\boldsymbol{\alpha},\boldsymbol{\beta})$. Image by Author.

Understanding the importance of Dirichlet distributions in LDA requires an understanding of Bayes' Theorem which states that, if the prior is Dirichlet distributed ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$) and the likelihood is multinomial distributed (z_{mn}, w_{mn}), the posterior will be Dirichlet distributed and can therefore be computed. Nevertheless, the posterior distribution is intractable for exact inference, and there are several approximating inference algorithms such as Laplace approximation, Markov chain Monte Carlo (eg. Gibbs sampling) and my choice of a variational Bayes algorithm which will be discussed later in the implementation section covering inference.

Implementing the LDA model

A reasonably large dataset is required to build a LDA model. Minimum necessary size is dependent on the characteristics and average length of the documents. Generally, the larger the dataset the better the results, due to an increase in observations. For example, when working with news articles, the minimum number of articles required for modeling is 600; however, results improve when the dataset includes more than 1,000 articles. For tweets, the minimum size depends on the heterogeneity of the conversation, and since tweets have a short sequence length, the minimum dataset size is significantly larger than what is required for news articles. A dataset of 5,000 to 10,000 tweets is sufficient for modelling, and results improve at a slower rate

compared to news articles when the dataset grows. For policy documents, which have a longer average length compared to news articles, fewer samples are required to build a model; minimum size depends on the scope of the collection.

The following four sections will cover the details necessary to successfully implement LDA for topic modeling of text data. The first section provides a description of select hyperparameters of the LDA model, including the Dirichlet priors. Secondly, it is important to consider the problem of inference; specifically, I focus on the solution of a variational Bayes algorithm. Following that, I provide an explanation of how to heuristically fine-tune the Dirichlet prior parameters with empirical experiments (ie. without relying on an EM algorithm for estimations, the method used in the introductory paper by Blei et al., 2003). Lastly, a practical introduction to a method for model evaluation that relies on a t-distributed stochastic neighbor embedding (t-SNE) visualization and perplexity scores.

Hyperparameters of the LDA model

There are several Python libraries with LDA modules. Currently, I prefer using Sci-Kit Learn (sklearn), though Gensim is a very popular choice with a multicore option for parallelization of LDA. Regardless of the choice of package, the hyperparameters remain the same, so the following discussion will be general in nature.

When implementing LDA it is necessary to fine-tune the hyperparameters, specifically the number of topics (k), the number of features (V, ie. fixed vocabulary size), and the α and β Dirichlet prior parameters. It is possible to fine-tune other parameters such as the number of iterations, the learning method (batch or online), the learning offset, perplexity tolerance, and others. For the sake of practicality, however, I will focus on the four parameters first mentioned. In my experience, working with a variety of datasets ranging from tweets (short and informal) to policy documents (long and formal), I have found that the best approach is to choose hyperparameters heuristically and then refine them with empirical experiments.

Number of topics (k)

Often, the most important hyperparameter is the number of topics, the choice of which depends on the characteristics and size of the dataset. For example, the larger the dataset the greater the number of topics, only if the dataset is representative of a diverse collection. However, a collection of a few thousand scientific articles on a particular subject, might not contain more topics if several thousand similar articles are added to the initial dataset.

The heuristic approach is to leverage knowledge of the content of the dataset to estimate a probable target, and make adjustments based on model evaluation and dimensionality reduction based visualizations. The optimal number of topics can be determined by calculating topic coherence scores over a range of topic numbers and plotting the resultant topic coherence trend. However, this is a computationally expensive task (ie. very time-intensive), therefore it is far more efficient to heuristically estimate a starting value for *k* and use model evaluation techniques to empirically guide adjustments.

Number of features (*V*)

The same approach can be used for choosing the number of features, which is equivalent to setting a fixed size for the vocabulary. The greater the number of features, the longer the LDA model will take to train; however, a sufficiently-sized vocabulary is necessary to capture the most important words for clustering of topics. Generally, setting the number of features to 10,000 is a good starting point for most models. This value needs to be fine-tuned depending on the size of the dataset and the amount of diversity of the words in the collection.

Importantly, increasing the number of features has diminishing returns on clustering accuracy. For example, if the total vocabulary for a dataset is 20,000 words, setting the number of features to 10,000 will definitely capture the vast majority of important words. On the other hand, assuming a total vocabulary of 100,000 words, increasing the number of features to 15 or 20,000 is enough to capture the majority of important words, since ordering the 100,000 words by tf-idf vectorization values will result in only a small percent being feature relevant.

The α and β hyperparameters

Most LDA models assume symmetric distribution, and the α and β parameters act as prior to the posterior calculation. This assumption of symmetry would mean that each topic is evenly distributed throughout a document, whereas for an asymmetric distribution (as measured by skewness) certain topics would be favoured over others. As Dirichlet prior concentration parameters, α and β, are representative of document-topic density and topic-word density respectively.

The α parameter will specify prior beliefs about topic sparsity and uniformity; visualized as a matrix each row is a document and each column is a topic. With a high α value, documents are assumed to contain more topics. Essentially, this means that each document is likely to contain a mix of many topics and not a single topic specifically. Conversely, a low α value assumes that a document will contain a mixture of just a few or a single topic. This happens because as the value of α decreases, sparsity increases such that, when the distribution is sampled, most values will be zero or close to zero.

Additionally, if the distributions are asymmetrical, a high α value results in a more specific topic distribution per document. Initially, the α parameter can be set to a real number value divided by the number of topics, and the results should reveal a sense of the sparsity and symmetry of the distribution. Therefore, the heuristic approach for choosing an α value, is to estimate the topical sparsity of each document on average. Subsequent adjustments should be determined by model evaluation and then tested empirically.

As mentioned, the β parameter represents topic-word density, and it is a matrix where each row represents a topic and each column represents a word. The β parameter will specify prior beliefs about word sparsity and uniformity within topics, adjusting for bias that certain topics will favour certain words. With a high β value, topics are assumed to be made up of most words in the fixed-sized vocabulary and this results in a more specific word mixture for each topic.

Conversely, with a low β value, a topic may contain a mixture of just a few of the words in the fixed-sized vocabulary. Furthermore, if the distribution is

asymmetrical, a high β value will result in a more specific word distribution. The topics, however, will be more similar in terms of words contained. Generally, it is sufficient to set the β value to 0.01 which is the value commonly used when the word distribution is sparse (usually true); however, it may be necessary to adjust this parameter.

Variational Bayesian Inference

By definition, Bayesian inference derives the posterior probability as a consequence of two antecedents: a prior probability and a likelihood function derived from a statistical model for the observed data. With regards to LDA, the problem is how to compute the posterior distribution of the hidden variables, given a document. Normalizing the posterior distribution by marginalizing over the hidden variables results in a function which is intractable, due to the coupling of Θ and β in summation over latent topics. This joint probability can be modeled with plate notation as seen in Figure 5, and expressed with the following equation:

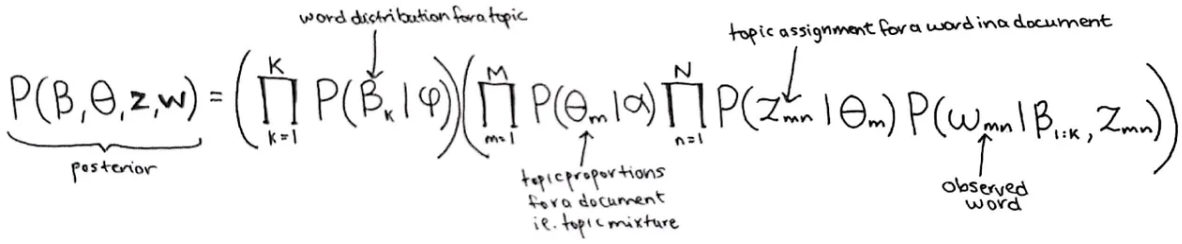

Figure 7. Joint probability equation for the posterior distribution of a LDA model. Image by Author.

In this form, it is clear that integrating over Θ, β is intractable, and as mentioned earlier, this makes exact inference of the posterior distribution very difficult. One solution is to use variational inference, specifically variational Bayesian methods that allow for approximating intractable integrals arising from Bayesian inference. Variational inference uses Jensen's inequality to obtain an adjustable lower bound on the log likelihood; therefore, variational parameters are chosen by an optimization procedure aimed at finding the tightest possible lower bound. The optimizing values of the variational parameters are found by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior distribution. I

recommend this blog post about KL divergence for an understanding of the role this dissimilarity measure plays in Bayesian inference.

A variational Bayes algorithm provides a locally-optimal exact-analytical solution to an approximation of the posterior distribution, in other words, it is an extension of the EM algorithm. The advantage of this approach over other options is the speed; therefore, I choose to use SciKit-Learn's LDA module which by default relies on a variational Bayes algorithm for inference. A slower, but nonetheless popular alternative is to use the Java-based library MALLET, which offers an optimized version of collapsed Gibbs sampling, that can also be accessed with a Python wrapper through Gensim.

Fine tuning the α and β hyperparameters

Based on my experience, generalized for working with any dataset, the guidelines for a heuristic approach are as follows:

1. Given knowledge of the topics, is it expected that the distribution of topics in each document will be sparse, such that each document contains only a few topics? If yes, then choose an $\alpha < 1$
2. Given knowledge of the total vocabulary, is it expected that the distribution of words in each topic will be sparse, such that certain topics favour certain words? If yes, then choose a $\beta < 1$

Alternately, if knowledge of the dataset is limited or if the distribution is asymmetrical, it is possible to empirically rely on model evaluation to inform fine tuning of the α and β parameters. This is accomplished by calculating perplexity scores and adjusting for sparsity. The general procedure can be described by the following steps:

1. Choose an α_m value from [0.05, 0.1, 0.5, 1, 5, 10, 50]
2. Choose a β_m value from [0.01, 0.05, 0.1, 0.5, 1, 5, 10]
3. Train the LDA model with α_m and β_m, while keeping all other parameters constant
4. Calculate model perplexity score on holdout data
5. Chose (α_m, β_m) pair with the minimum perplexity score

Personally, I favour combining the heuristic approach with empirical experiments. Calculating model perplexity scores is a method of model evaluation, which will be discussed further in the following section.

Model Evaluation

The challenge with many machine learning models, including the LDA model is how to interact with the high-dimensional data in a meaningful way that is interpretable for humans. Therefore, my primary method of evaluation is to use t-distributed Stochastic Neighbor Embedding (t-SNE) as a tool to visualize the high-dimensional data. This dimensionality reduction technique was introduced by Laurens van der Maaten and Geoffrey Hinton in 2008. The best resource I have found for implementing t-SNE is the personal blog of Laurens van der Maaten; the FAQ section on the t-SNE page in particular, offers valuable tips for understanding the visualization.

Essentially, the t-SNE technique works to convert similarities between data points to joint probabilities, and then tries to minimize the KL divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. The dimensionality of the LDA model is determined by the number of features set during training (usually a minimum of 10,000); therefore, t-SNE can be used to reduce the dimensions to a 2-D embedding that offers a visualization of the clustering determined by LDA. To interpret the t-SNE, a simple visual evaluation of the clusters offers insight, as does the KL divergence score, for which the value closest to 0 is optimal. In the image below, created with the Python plotting library Bokeh and a dataset of 67,000 tweets, the differently coloured clusters represent the abstract topics, and positioning is determined by the dimensionality reduction algorithm.

Figure 8. Example t-SNE plot. Image by Author.

As mentioned earlier, model perplexity scores can be calculated to evaluate the effect of various hyperparameters for empirical testing. I use sklearn to calculate perplexity, and this blog post provides an overview of how to assess perplexity in language models. When building a LDA model I prefer to set the perplexity tolerance to 0.1 and I keep this value constant so as to better utilize t-SNE visualizations. It is important to note that t-SNE has a non-convex objective function, where the objective function is minimized using a gradient descent optimization that is initiated randomly. Therefore, different initializations will result in different solutions, so it is possible (and often advisable) to run t-SNE multiple times with the same data and perplexity, and then choose the visualization with the lowest KL divergence.

Limitations of LDA

Practically, an assessment of the LDA model reveals a few weaknesses, namely the necessity of a fixed *k* value, the inability of Dirichlet distributions in capturing correlations, the static nature does not show the evolution of topics over time, and lastly the simplifying "bag-of-words" exchangeability assumption. Of these limitations, none are sufficient to abandon this topic modeling method, but an awareness is necessary to understand the boundaries of results.

In my experience, LDA can be used consistently and successfully to model text collections of news articles and policy documents, yet the results can be mixed for unconventional datasets, such as collections of tweets which are short and informal. Generally, a dataset is unsuitable for topic modeling if the length of the documents is too short, the data set is too small, or if there are too many topics within a collection (eg. book). In the past, ad hoc heuristics have been successfully employed to preprocess documents, such as aggregating tweets into longer "documents" (Hong and Davison, 2010). These measures, however, can be blind since many of the common assumptions of limitations are not theoretically justified; for example, the deficiency in handling shorter documents has not been explained by theory.

In a paper from 2014 by Tang et al. an attempt was made to understand the limiting factors of LDA with posterior contraction analysis. It was found that Liebig's law of the minimum is applicable to LDA, whereby the scarcest resource acts as the limiting factor. Of the four guidelines proposed, the most important one is regarding the number of documents (*M*); a sufficiently sized dataset is absolutely necessary. Once a viable *M* value is achieved, further increasing *M* may not significantly improve performance, unless the document length is also suitably increased. Lastly, when a large number of topics (*K*) is used to fit an LDA model, the statistical inference may become inescapably inefficient. This is because the convergence rate deteriorates quickly to a non-parametric rate, depending on the number of topics used to fit the LDA model. Therefore, in practice it is very important to avoid selecting an overly large *k* value.

In conclusion, I believe that an awareness of the LDA model's deficiencies along with practical guidelines for choosing datasets and hyperparameters will allow for successful implementation of this method for topic modeling. I

welcome all feedback, whether questions or comments, so please feel free to connect with me on Linkedin.