Latent Dirichlet Allocation (LDA)

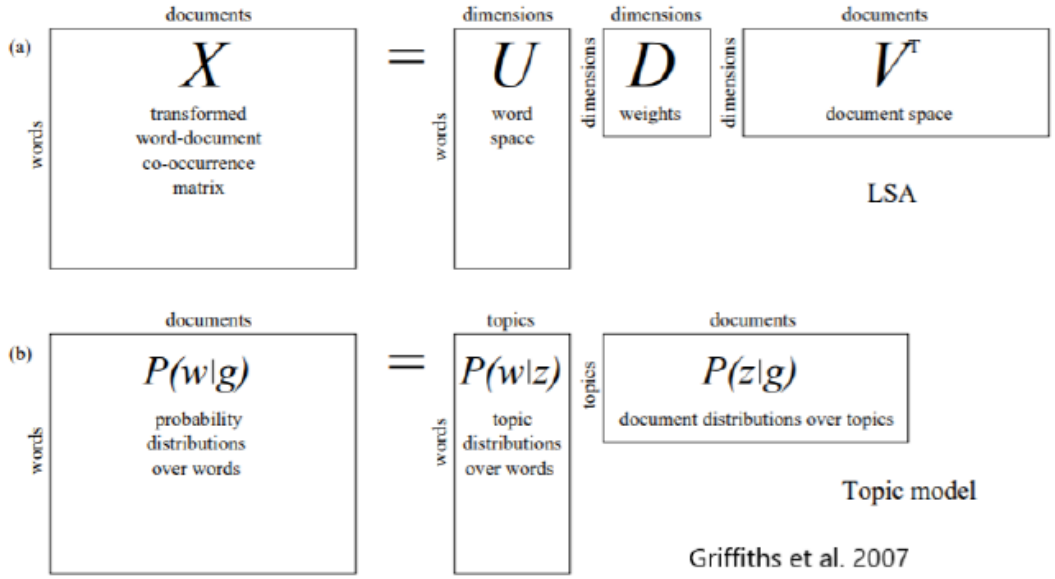
Introduction to Topic Models

Topic models, also known as probabilistic topic models, are algorithms used to automatically identify and extract topics from a collection of texts. These models categorize texts into distinct groups based on underlying topics or themes. Here are some key aspects of topic modeling:

1. **Identification of Topics**: A 'topic' in this context is a collection of words that frequently appear together. For example, in a collection of news articles, common topics might be politics, sports, economy, etc.
2. **Latent Dirichlet Allocation (LDA)**: This is one of the most popular methods for topic modeling. LDA assumes that each document is a mixture of a small number of topics and that each word in the document is attributable to one of the document's topics. LDA is an example of a 'generative model,' which means it assumes a probabilistic model for how the documents are generated, and tries to infer the parameters of this model.
3. **Document and Word Distributions**: In LDA, each topic is represented as a distribution over words, and each document is represented as a distribution over topics. This allows not just for determining what topics are present in a corpus, but also for deducing the proportion of each topic in individual documents.
4. **Applications**: Topic models are widely used for various applications, such as organizing large archives of documents, summarizing texts, and aiding in information retrieval systems. They can help in discovering the underlying themes in text data, classifying documents into topics, and even in improving the accuracy of other machine learning models.
5. **Challenges**: One of the challenges in topic modeling is determining the number of topics. Also, the interpretation of the topics requires human judgment and can sometimes be ambiguous or subjective.

Latent Semantic Analysis (covered previously) discovers hidden semantic content whereas topic models focus on the underlying subjects or themes that are present in the documents. LSA uses Singular Value Decomposition to break down the TDM into three smaller matrices for smaller dimensional representation whereas in topic modelling (while there are many different kinds, in this discussion we focus on Latent Dirichlet Allocation- LDA), DTM is broken into two major components: the distribution of topics over terms and the document distribution over topics. Unlike latent factors of LSA, topic models are generative probabilistic models where each topic is clearly identifiable and

explainable.



Real-World Example

Imagine a hotel listed on TripAdvisor with numerous reviews. As a hotel manager, you want to understand different aspects of customer feedback. For example:

- **Booking Experience**: Reviews about how easy it is to book a room.
- **Room Quality**: Feedback on the room's cleanliness and comfort.
- **Restaurant Quality**: Opinions on the food served at the hotel restaurant.

By grouping these reviews into categories, the hotel can gain specific insights into each aspect rather than just a general overview.

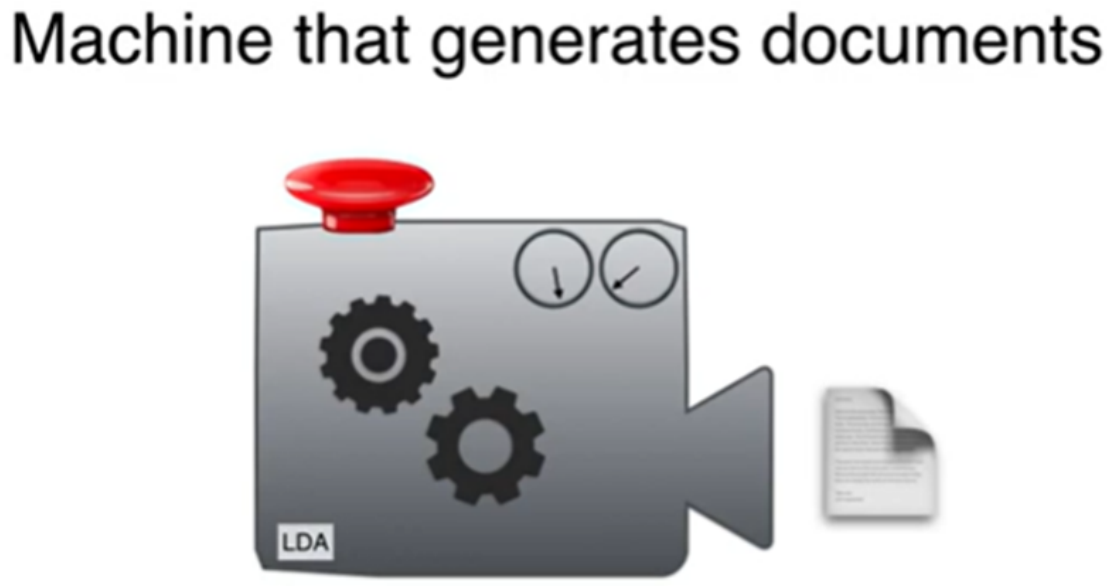
Basics of Latent Dirichlet Allocation (LDA)

LDA is a generative statistical model that assumes each document in a collection is a mixture of topics, and each topic is a mixture of words.

Key Concepts

1. **Documents**: Collections of words.
2. **Corpus**: A collection of documents.
3. **Topics**: Hidden themes or subjects that occur in a collection of documents.
4. **Words**: Elements that make up documents.

How LDA Works



(Image and idea from Luis Serrano notes)

Think of LDA as a machine that generates documents. Machine has settings that one can play with and it also has a button such that when one presses the button, some gears start turning and these gears build a document as output. Obviously, the output document would not have meaningful sentences but just have a list of words/ tokens organized into some topics.

LDA models the probability distribution of words within topics and topics within documents. Here's how it works step by step:

1. **Choosing the Number of Topics (k)**: Decide on the number of topics to classify the documents into, similar to deciding on sections in a library (e.g., Fiction, Science, History).
2. **Distribution Over Words (φk)**: For each topic, create a distribution over words.
3. **Distribution Over Topics (θm)**: For each document, create a distribution over topics.
4. **Assigning Topics to Words**: Initially, randomly assign topics to words in documents.
5. **Iteratively Refine Topic Assignments**: Refine the assignments based on the prevalence of topics in documents and words in topics.

Detailed Example with Diagrams

Step 1: Choosing the Number of Topics (k)

Suppose we have a list of words that we decide to assign to three topics in our corpus: **Marketing**, **Food**, and **Finance**.

Step 2: Distribution Over Words for Each Topic (φk)

For each topic, we create a distribution over words. Here is an example:

- **Marketing**: {brand: 0.45, consumer: 0.45, chef: 0.04, cheese: 0.04, stocks: 0.01, bonds: 0.01}
- **Food**: {brand: 0.05, consumer: 0.10, chef: 0.35, cheese: 0.45, stocks: 0.03, bonds: 0.02}

- **Finance**: {brand: 0.01, consumer: 0.01, chef: 0.01, cheese: 0.01, stocks: 0.48, bonds: 0.48}

These distributions show the probability of each word given a topic.

Step 3: Distribution Over Topics for Each Document (θm)

For each document, we create a distribution over topics. Here is an example for one document:

- **Document 1**: {Marketing: 0.55, Food: 0.35, Finance: 0.10}

This means that Document 1 is 55% about Marketing, 35% about Food, and 10% about Finance.

Step 4: Assigning Topics to Words

Initially, topics are assigned to words randomly.

Step 5: Iteratively Refine Topic Assignments

Using algorithms like Gibbs Sampling or Variational Inference, we iteratively reassign topics to words and refine the topic distributions to better fit the data.

Graphical Representation of LDA

Plate Notation Diagram

The plate notation diagram is a compact representation of the dependencies between the variables in a probabilistic graphical model. For LDA, it looks like this:

- **M**: Number of documents
- **N**: Number of words in a document
- **K**: Number of topics
- θ_m: Topic distribution for document m
- φ_k: Word distribution for topic k
- $\mathbf{z}_{mn}$: Topic assignment for word n in document m
- $\mathbf{x}_{mn}$: Word n in document m

Step-by-Step Diagram

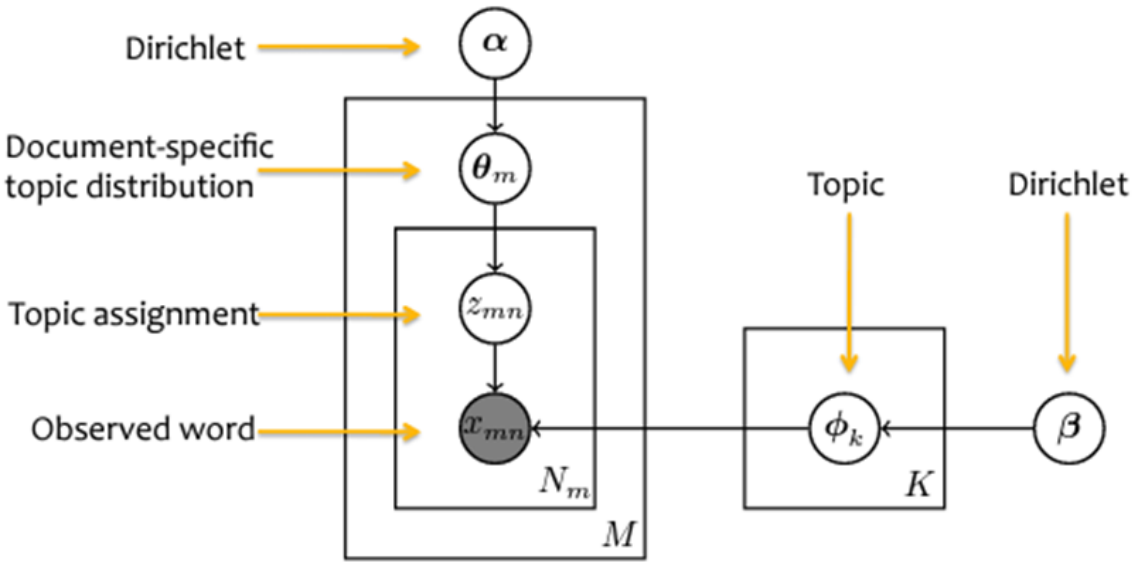


Figure 1: Plate diagram for LDA model

(From Matt Gormley Notes)

Each topic in Latent Dirichlet Allocation (LDA) is represented as a multinomial distribution over the vocabulary, parameterized by φk. Similarly, document-specific topic distribution is parametrized by multinomially distributed θ_m. The multinomial distribution is a generalization of the binomial distribution to more than two categories. It describes the probability of counts for a fixed number of independent trials, where each trial can result in one of k different outcomes.

Since LDA is an unsupervised learning method, topics are visualized through their most probable words, with a descriptive label used for identification. LDA's structure consists of two key distributions: one over words and one over topics. Before diving into the inference process, it's reasonable to ask: "Does this method plausibly explain how a corpus of documents is generated?" or "Why might this approach be effective?"

The answer to this question lies in the fact that LDA balances two objectives:

1. For each document, it tries to allocate its words to as few topics as possible.
2. For each topic, it aims to assign high probability to as few terms as possible.

Achieving these goals simultaneously is challenging because if a document is assigned to a single topic, all its words must have high probability under that topic, making the second goal difficult. Conversely, if very few words are assigned to each topic, many topics will be needed to cover all the words in a document, complicating the first goal. LDA strikes a balance between these goals to identify groups of words that frequently co-occur.

LDA balances these two objectives through its probabilistic framework and iterative algorithm:

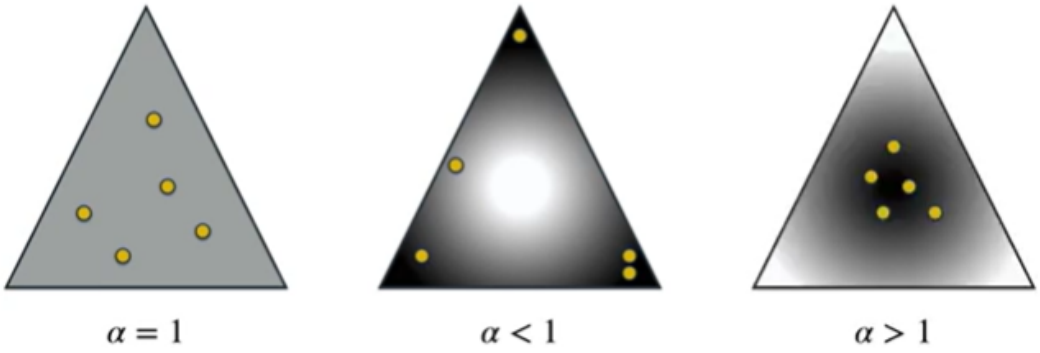
1. **For each document, allocate its words to as few topics as possible:**
   LDA assumes that each document is a mixture of a small number of topics. During the inference process, LDA iteratively assigns words to topics in a way that tends to minimize the number of topics per document. This is achieved by adjusting the topic proportions for each document, which are influenced by the prior distribution (**Dirichlet prior**) that encourages sparsity, meaning that fewer topics are more likely to be dominant in a document.

2. **For each topic, assign high probability to as few terms as possible:**
   LDA also assumes that each topic is characterized by a small number of terms with high probability. During the inference, LDA updates the word distributions for each topic in a way that maximizes the likelihood of the observed words under the current topic assignments. The **Dirichlet prior** on the topic-word distributions encourages sparsity, meaning that each topic will ideally have a few words with high probability and many words with low probability.

Through the iterative process of Gibbs sampling or variational inference, LDA repeatedly refines the topic assignments for words in documents. The interaction between the document-topic distributions and the topic-word distributions ensures that the model finds a balance where documents are represented by a few topics, and each topic is defined by a few high-probability words. This trade-off allows LDA to capture the underlying thematic structure of the corpus effectively.

Understanding Dirichlet Distribution:

Think of the Dirichlet distribution as a way to generate probabilities for different outcomes, like picking different flavors of ice cream.



Example:

Imagine you own an ice cream shop with three flavors: Chocolate, Vanilla, and Strawberry. You want to understand how customers choose these flavors on different days.

Using the Dirichlet Distribution:

1. **Setting Up Preferences (α values):**
   - We use α =1 values to set up our initial guess about customer preferences. Let's say we set preferences as [0.3, 0.4, 0.3] for the three flavors.
   - This means we think customers will prefer a mix of flavors, but we aren't sure which one they'll prefer more.
2. **α < 1 (e.g., α = 0.1):**

- The distribution might look like [0.9, 0.05, 0.05] meaning the preference is predominantly about one flavor and very little about the others.

Applying to Documents and Topics in Context of LDA:

- **α < 1 (e.g., α = 0.1):** Documents will have a sparse distribution over topics, focusing on a few topics with high probability.

- **α = 1:** Documents will have a balanced distribution over topics, with no strong preference for any particular topic.
- **α > 1:** Documents will have a more even distribution over topics, with each topic having a roughly equal chance of being chosen.

To summarize, **α** is simply the first Dirichlet distribution of the word for documents and topics, **β** is the one for topics and words. From **α** we get θ which is a multinomial distribution for picking topics, and from **β** we get φ which is a bunch of multinomial distributions for picking words. From θ we get Z which is a list of topics. We combine this with φ to obtain a list of words per topic. Finally, we concatenate these words to obtain a document. We do this as many times as number of documents in the corpus to create a corpus and then we compare that to the original one to find the arrangements of points inside the Dirichlet Distribution that maximize this probability.

Thus, the purpose of LDA is to run it “backwards”: we do not run LDA to generate document but to estimate the probability distributions that were most likely to generate such a document. This is similar to the context where given a dataset generated by normal distribution, one has to infer population parameters such as mean and variance. In this case, we ask the question, “what are the most likely Dirichlet priors and probability distributions that generated this data?”. So we take the DTM generated by the bag-of-words model, analyze the frequencies and infer the probability distributions that could have generated such a dataset.