Topic Modeling

Topic Modeling refers to the task of identifying "topics" that best describe a set of documents

↳ This is an unsupervised learning method (we do NOT require preexisting training data)

↳ Similar to clustering of numerical data

[Topic modeling is not to be confused with topic classification which is a supervised technique based on tagged (training) documents]

Topic modeling provides a method for automatically organizing & summarizing large collections of documents ex: Take untagged reports to make a small number of "topics" which can be used to more quickly find the relevant information [for a given problem, most text is likely uninformative]

This can be done with each document in a single "topic"

OR as "soft clustering" in which documents can be tagged with multiple "topics"

As in clustering, there are multiple algorithms that can provide differing results:

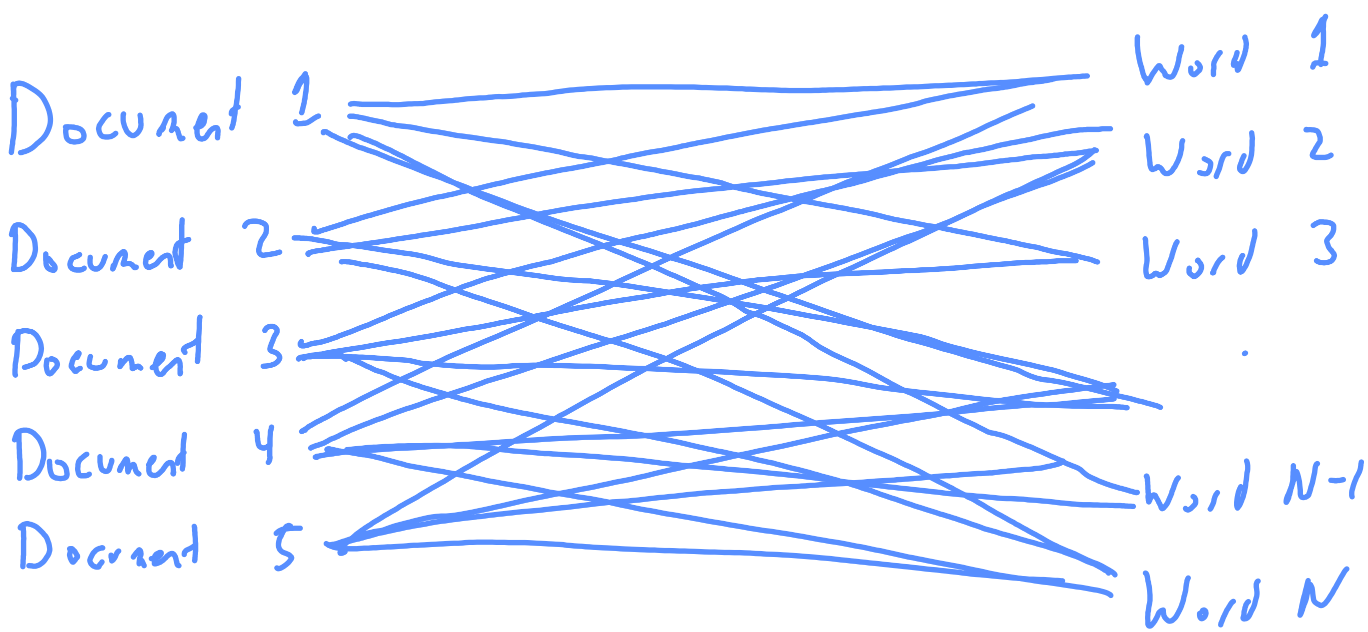
1) Latent Dirichlet Allocation
2) Latent Semantic Allocation
3) Pachinko Allocation Model (similar to Latent Dirichlet Allocation)
4) Non-Negative Matrix Factorization
5) ...

Latent Dirichlet Allocation (LDA)

The aim is to find topics that a document belongs to based on the words/tokens that are in it

↪ Solves the problem by introducing a hidden / latent layer of topics

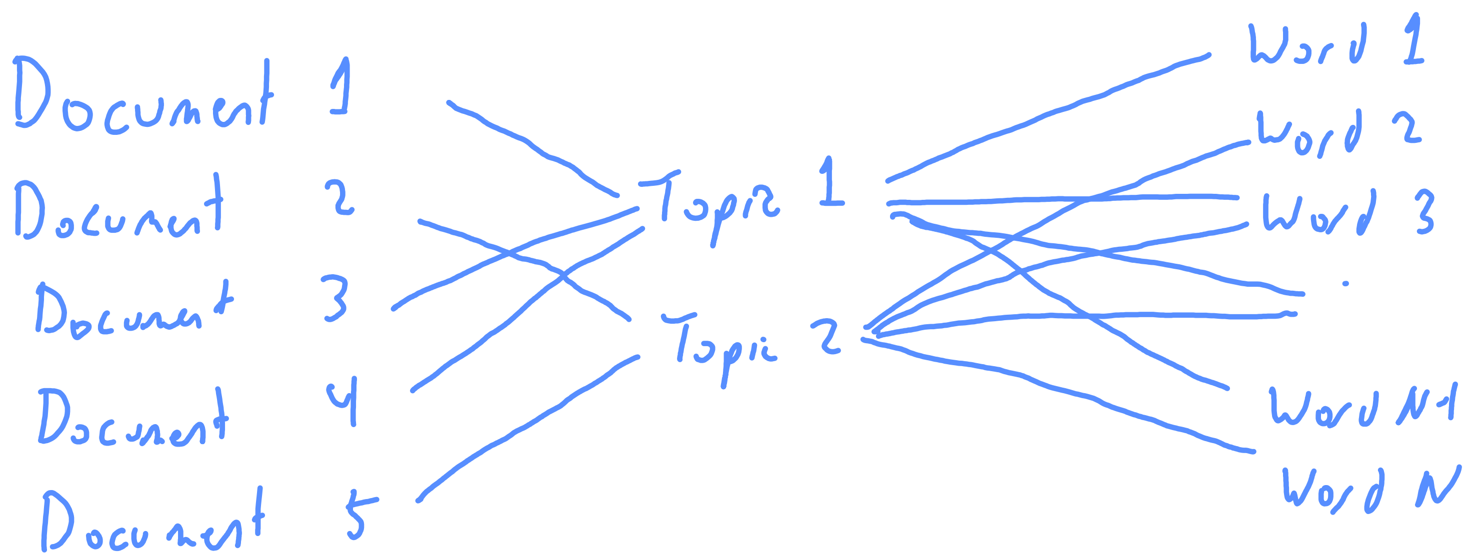
Idea: Think of each document as a bag of words (from some vocabulary of terms)
We can draw a bipartite graph connecting documents & words



This graph has so many connections that it is generally useless

(Though this can give a general picture of your corpus of documents)

Instead, we want to introduce a latent layer between the 2 sides of the graph



Goal is to construct/find topics that explain "most" of the document-word connections

⌈As in clustering, we label the clusters after the fact

The algorithm cannot provide human readable labels for you

↪ look at the words/documents each topic references ⌋

LDA is often described as:

- Each document is described by a distribution of topics
- Each topic is described by a distribution of words

↑ In comparison to the networks we drew above, the links are "probabilistic" rather than fixed

LDA assumes that documents are represented as random mixtures over the latent topics

↪ As with K-Means Clustering, we choose a fixed number K of topics before beginning

Recall: The K topics have no ex ante meaning
only ex post explanations

Methodology

Consider a vocabulary of V words of interest

Consider a corpus of M documents

document i has N_i words

Let w_{ij} be the j^{th} word of document i

↳ Represented by one-hot-encoding of size V

$W_i = (w_{i1}, w_{i2}, \dots, w_{iN_i})$ is a $V \times N_i$ matrix of words in document i

[The collection $(W_1, W_2, \dots, W_M)$ of document matrices are the only observable variables]

We will denote the k^{th} topic as z_k [K topics total]

Let $Dir(\alpha)$ [Dirichlet distribution] provide a prior of the per-document topic distribution

Let $Dir(\beta)$ provide a per-topic word distribution

Dirichlet Distribution

Often used for modeling distributions of probabilities / distributions

If α is N-dimensional and $x \sim Dir(\alpha)$

then $x_i \geq 0$ for all $i = 1, \ldots, N$ and $\sum_{i=1}^{N} x_i = 1$

(x defines an N-dimensional discrete distribution)

LDA assumes the corpus of M documents is generated by:

1) Sample $\theta_i \sim Dir(\alpha)$ for each $i = 1, \dots, M$

   ↳ distribution of topics for each document

2) Sample $\varphi_k \sim Dir(\beta)$ for each $k = 1, \dots, K$

   ↳ distribution of words for each topic

3) For each word position (i, j) for document $i = 1, \dots, M$ + position $j = 1, \dots, N_i$

   a) Sample a topic $z_{ij} \sim Multinomial(\theta_i)$

   b) Sample a word $w_{ij} \sim Multinomial(\varphi_{z_{ij}})$

      ↳ single trials only, also called the categorical distribution

Goals: Find
- α to define the distribution of topics for documents
- β to define the distribution of words for topics
- θ_{ik} is the probability that document i is in topic k
- φ_{kj} is the probability that the k^{th} topic contains word $j = 1, \dots, V$

Approach: Maximum Likelihood

↳ maximize the probability:

"evidence" $= (w_1, \ldots, w_M)$

$$\mathbb{P}(\underbrace{\theta, z, \varphi}_{\text{Hidden Variables}} \mid W)$$

probability each document is in a topic (θ)

assignment of words to topics (z)

word frequency for each topic (φ)

Model the joint probability:

$$\mathbb{P}(\varphi, \theta, z, W) = \underbrace{\left(\prod_{k=1}^{K} \mathbb{P}(\varphi_k \mid \beta)\right)\left(\prod_{i=1}^{M} \mathbb{P}(\theta_i \mid \alpha) \prod_{n=1}^{N_i} \mathbb{P}(z_{in} \mid \theta_i)\, \mathbb{P}(w_{in} \mid \varphi, z_{in}\right)}_{\text{We already know these conditional probabilities}})$$

$$\hookrightarrow \mathbb{P}(\varphi, \theta, z \mid W) = \mathbb{P}(\varphi, \theta, z, W) \Big/ \underbrace{\int_{\varphi}\int_{\theta}\sum_{z} \mathbb{P}(\varphi, \theta, z, W)}_{\text{hard to compute}}$$

Posterior is approximated with variance inference

Financial Applications

1) Clustering of Unstructured Documents (ex earnings calls)

Information retrieval from structured documents (10Ks ...)
is "easy" (because of the structure)

For unstructured documents, this can be searching for a needle in a haystack

Even finding the relevant document can be challenging

Clustering into topics can help us narrow our search parameters

Can also assist with summarization based on ex post explainability

Can also be combined with sentiment analysis to discover polarities of
the different topics

⌈This can also be done with text classification if a sufficiently large
training set of tagged documents can be constructed⌋

2) Customer Service

Banking is traditionally about maintaining (customer) relationships

Topic modeling can assist in automatically sorting comments/messages
to the appropriate team

Possibly apply text classification first, then use topic modeling on those
documents with low certainty of classification

3) Sorting Social Media

Social media can have important information for, e.g., sentiment analysis

But it also has a lot of noise

Cluster by topics, then determine which clusters are "informative"

↳ again provides ex post explainability