**LDA (Latent Dirichlet Allocation)**

Latent Dirichlet Allocation (LDA) is an example of topic modelling algorithm and is used to classify text in a document to a particular topic.

Topic modeling is the process of identifying topics in a set of documents. It can be useful for any other instance where knowing the topics of documents is important.

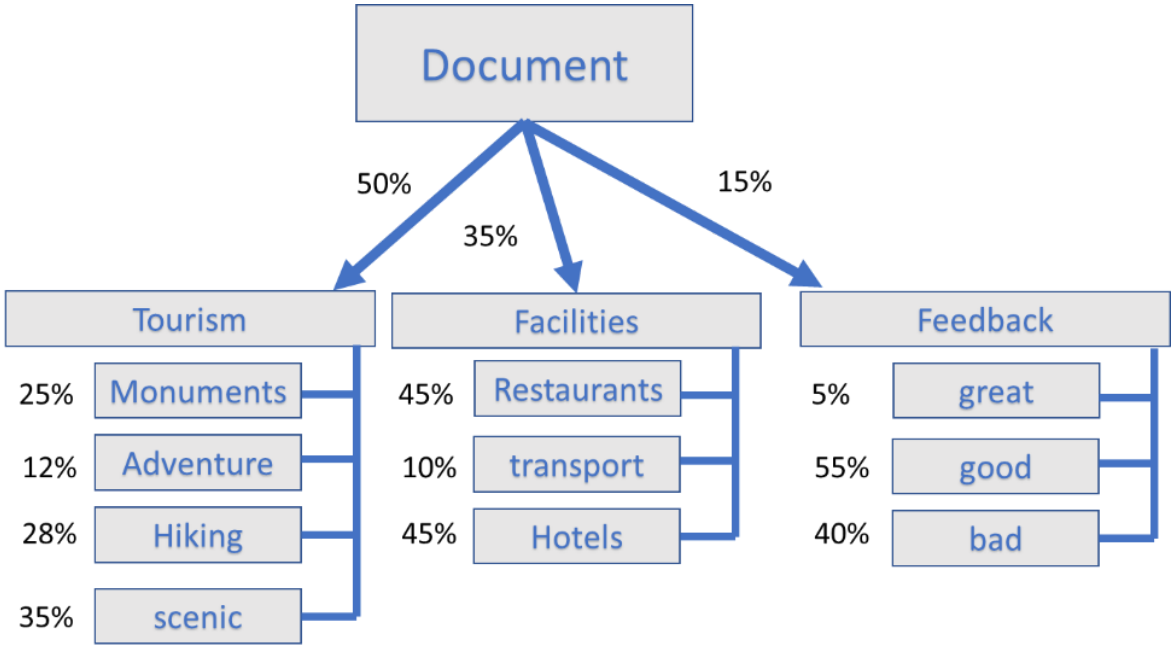
Topic modeling is also a form of dimensionality reduction technique as it breaks down and expresses a document in form of few topics.

Each document is represented as a distribution over topics.

Each topic is represented as a distribution over words.

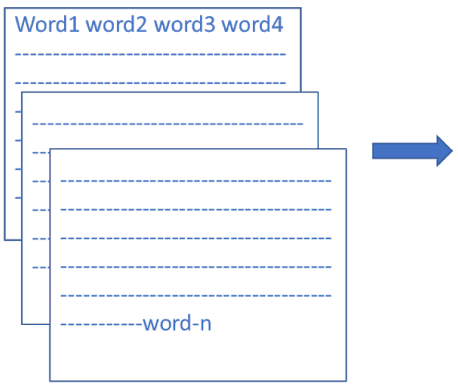
**How it works**

LDA assumes that each document is generated by a statistical generative process. That is, each document is a mix of topics, and each topic is a mix of words




LDA assumes that documents are composed of words that help determine the topics and maps documents to a list of topics by assigning each word in the document to different topics. The assignment is in terms of conditional probability.

In the following figure, the value in each cell indicates the probability of a word wj belonging to topic tk. j and k are the word and topic indices respectively. It is important to note that LDA ignores the order of occurrence of words and the syntactic information.




'm' documents with 'n' words

| | word1 | word2 | word3 | | | | Word-n |
|---|---|---|---|---|---|---|---|
| Topic-1 | 0.024 | 0.012 | 0.014 | - | - | - | 0.086 |
| Topic-2 | 0.026 | 0.186 | 0.164 | - | - | - | 0.194 |
| Topic-3 | 0.018 | 0.112 | 0.192 | - | - | - | 0.028 |
| \| | - | - | - | - | - | - | - |
| Topic-K | 0.128 | 0.144 | 0.084 | - | - | - | 0.036 |

Once the probabilities are estimated (we will get to how these are estimated shortly), finding the collection of words that represent a given topic can be done either by picking top r probabilities of words or by setting a threshold for probability and picking only the words whose probabilities are greater than or equal to the threshold value.

LDA is a matrix factorization technique. In vector space any corpus (collection of documents) can be represented as a document-term matrix. The following matrix shows a corpus of N documents D1, D2, D3…. Dn and vocabulary size of M words W1, W2 ... Wn. The value of i, j cell gives the frequency count of word Wj in Document Di.

| | W1 | W2 | W3 | Wn |
|---|---|---|---|---|
| D1 | 0 | 2 | 1 | 3 |
| D2 | 1 | 4 | 0 | 0 |
| D3 | 0 | 2 | 3 | 1 |
| Dn | 1 | 1 | 3 | 0 |

LDA converts this Document-Term Matrix into two lower dimensional matrices in the form of M1 and M2.

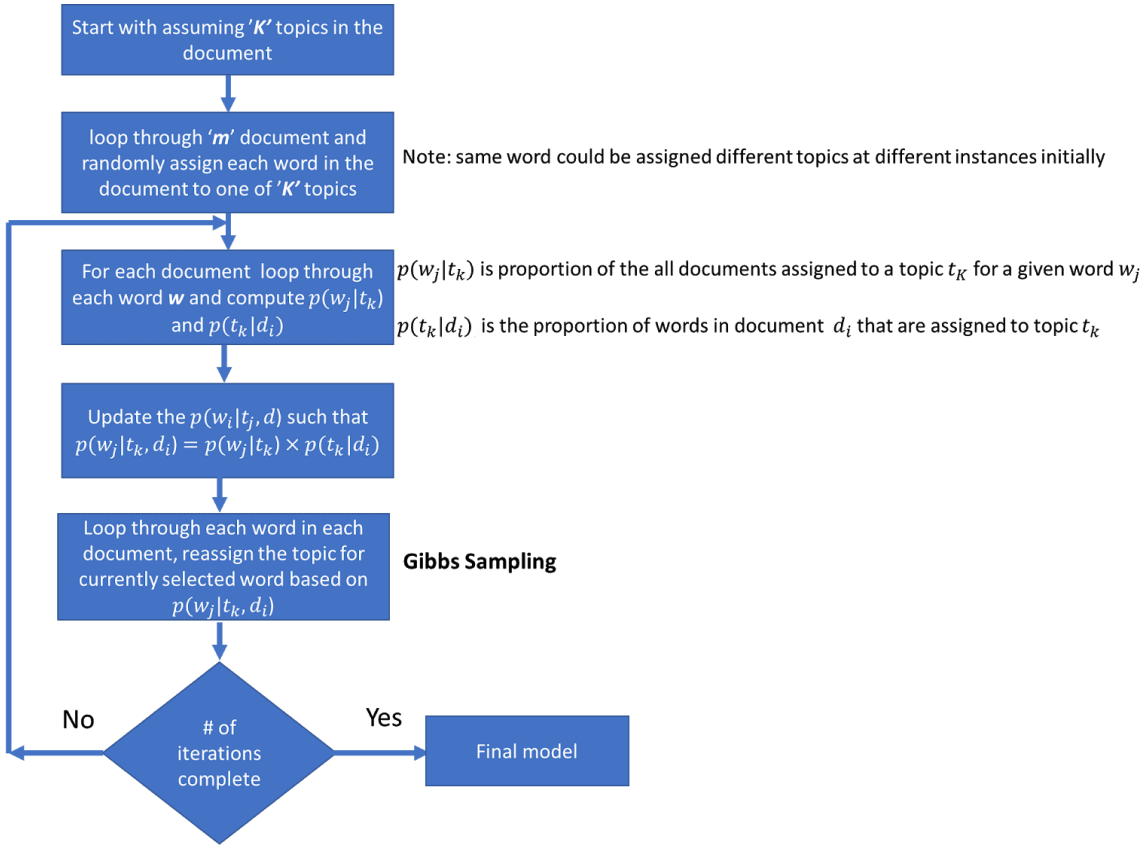
M1 is a document-topics matrix and M2 is a topic-terms matrix with dimensions (N, K) and (K, M) respectively where N is the number of documents, K is the number of topics and M is the vocabulary size.

| | K1 | K2 | K3 | K |
|---|---|---|---|---|
| D1 | 1 | 0 | 0 | 1 |
| D2 | 1 | 1 | 0 | 0 |
| D3 | 1 | 0 | 0 | 1 |
| Dn | 1 | 0 | 1 | 0 |

| | W1 | W2 | W3 | Wm |
|---|---|---|---|---|
| K1 | 0 | 1 | 1 | 1 |
| K2 | 1 | 1 | 1 | 0 |
| K3 | 1 | 0 | 0 | 1 |
| K | 1 | 1 | 0 | 0 |

Notice that these two matrices already provides topic word and document topic distributions. However, these distribution needs to be improved which is the main aim of LDA. LDA makes use of Gibbs sampling technique in order to improve these matrices.

Steps




Let's understand the process by taking a simple example. Let's consider a corpus of 'm' documents with five words vocabulary

Document-1

| | | | | |
|---|---|---|---|---|
| monuments | restaurants | great | good | transport |

m documents

**Step-1**
Randomly assign 'K' topics to all the words in 'm' documents

| T3 | T2 | T1 | T3 | T1 |
|---|---|---|---|---|
| monuments | restaurants | great | good | transport |

**Step-2**
Create document wise topic count
*local statistic to each document*

| | T1 | T2 | T3 |
|---|---|---|---|
| Document-1 | 2 | 1 | 2 |
| \| | | | |
| Document-m | | | |

**Step-3**
Create topic wise assignment of word count from all documents
*Global statistic* for the whole vocabulary
(numbers shown are populated for illustration only and are not actual values)

| | T1 | T2 | T3 |
|---|---|---|---|
| monuments | 1 | 0 | 35 |
| restaurants | 50 | 0 | 1 |
| great | 42 | 1 | 0 |
| good | 0 | 0 | 20 |
| transport | 10 | 8 | 1 |

**Step-4**
Resample a word.
Remove topic assignment

Document-1

| T3 ? | T2 | T1 | T3 | T1 |
|---|---|---|---|---|
| monuments | restaurants | great | good | transport |

**Step-5**
Decrement the count for the respective topic allocated from the document-topic matrix

| | T1 | T2 | T3 |
|---|---|---|---|
| Document-1 | 2 | 1 | 2-1=1 |
| \| | | | |
| Document-m | | | |

**Step-6**
Decrement the count for the respective topic allocated from the document-topic matrix

| | T1 | T2 | T3 |
|---|---|---|---|
| monuments | 1 | 0 | 35-1 =34 |
| restaurants | 50 | 0 | 1 |
| great | 42 | 1 | 0 |
| good | 0 | 0 | 20 |
| transport | 10 | 8 | 1 |

**Step-7**
calculate
$p(t_k|d_i)$
Indicates how much document $d_i$ likes topic $t_k$

$$p(t_k|d_i) = \frac{n_{ik} + \alpha}{N_i - 1 + K\alpha}$$

| | T1 | T2 | T3 |
|---|---|---|---|
| Document-1 | 2 | 1 | 2-1=1 |
| \| | | | |
| Document-m | | | |

Where $n_{ik}$ is the total number of words in '$i^{th}$' document in '$k^{th}$' topic, $N_i$ is the number of words in the '$i^{th}$' document, K is the number of topics considered. $\alpha$ is a hyper parameter

For instance, for document-1 and topic-3, $p(t_k|d_i) = \frac{1+0.1}{5-1+3\times0.1} = 0.155$ assuming $\alpha = 0.1$

**Step-8**
calculate
$p(w_j|t_k)$
Indicates how much topic $t_k$ likes word $w_j$

$$p(w_j|t_k) = \frac{m_{j,k} + \beta}{\sum_{j \in V} m_{j,k} + V\beta}$$

| | T1 | T2 | T3 |
|---|---|---|---|
| monuments | 1 | 0 | 35-1 =34 |
| restaurants | 50 | 0 | 1 |
| great | 42 | 1 | 0 |
| good | 0 | 0 | 20 |
| transport | 10 | 8 | 1 |

Where $m_{j,k}$ is the corpus wide assignment of word $wj$ to $k^{th}$ topic. $V$ is the vocabulary of the corpus. $\beta$ is a hyper parameter

For instance, for the word monument in topic-3, $p(w_j|t_k) = \frac{34+0.5}{56+5\times0.5} = 0.56$ assuming $\beta = 0.5$

**Step-9**
calculate for word $w_j$ $p(w_j|t_k,d_i)$

$$p(w_j|t_k,d_i) = p(t_k|d_i) \times p(w_j|t_k)$$

**Step-10**
Resampling/ Reassign

For a given word $w_j$ in a document $d_i$, find the topic 'k' for which $p(w_j|t_k,d_i)$ is maximum and reassign the word to the '$k^{th}$' topic

For instance, for the word 'monument' in 'document-1' if $P(monument|T2, Document-1) = 0.4$, the highest value, then reassign the word 'monument' to 'Topic-2'.

| T2 | T2 | T1 | T3 | T1 |
|---|---|---|---|---|
| monuments | restaurants | great | good | transport |

**Step-11**
Repeat for all

Repeat steps 4 to 10 for all the words in all the document

**Step-12**
Repeat process

Repeat steps 2 to 11 for a predefined number of iterations

What are the important LDA hyper parameters?

There are three important hyper parameters

- document-topic density factor ‘α’

The ‘α’ hyperparameter controls the number of topics expected in the document. Low value of ‘α’ is used to imply that fewer number of topics is expected and a higher value implies that one would expect the documents to have higher number topics

- topic-word density factor ‘β’

The ‘β’ hyper parameter controls the distribution of words per topic. At lower values of ‘β’, the topics will likely have fewer words and at higher values topics will likely have more words.

- the number of topics ‘K’ to be considered.

The K hyperparameter specifies the number of topics expected in the corpus of documents. Choosing a value for K is generally based on domain knowledge. An alternate way is to train different LDA models with different numbers of K values and compute the coherence score. Choose the value of K for which the coherence score is highest.

Ideally we would like to see a few topics in each document and few words in each of the topics. So α and β are typically set below one.

How to measure performance of LDA?

Coherence

A set of statements or facts is said to be coherent if they support each other.

Topic Coherence

Topic coherence score is always used to measure how well the topics are extracted. Here we quantify the coherence of a topic by measuring the degree of semantic similarity between its high scoring words.

C_V score and Umass score

There are two major types of coherent scores

- c_v score

Typically, $0 < x < 1$

Best coherence for c_v is typically the maximum.

- umass  score

Typically, $-14 < x < 14$.

Best coherence for umass is typically the minimum.

**Example**

- I eat fish and vegetables.
- Cats are pets.
- My kitten eats fish.

Given the above sentences, LDA might classify the red words under the Topic F which we might label as food. Similarly, blue words might be classified under a separate Topic P which we might label as pets

**Sentence 1**: 100% Topic F
**Sentence 2**: 100% Topic P
**Sentence 3**: 33% Topic P and 67% Topic F