Table of Content

For more detail follow:

LinkedIn Page: **https://www.linkedin.com/company/pathshala-ai/**

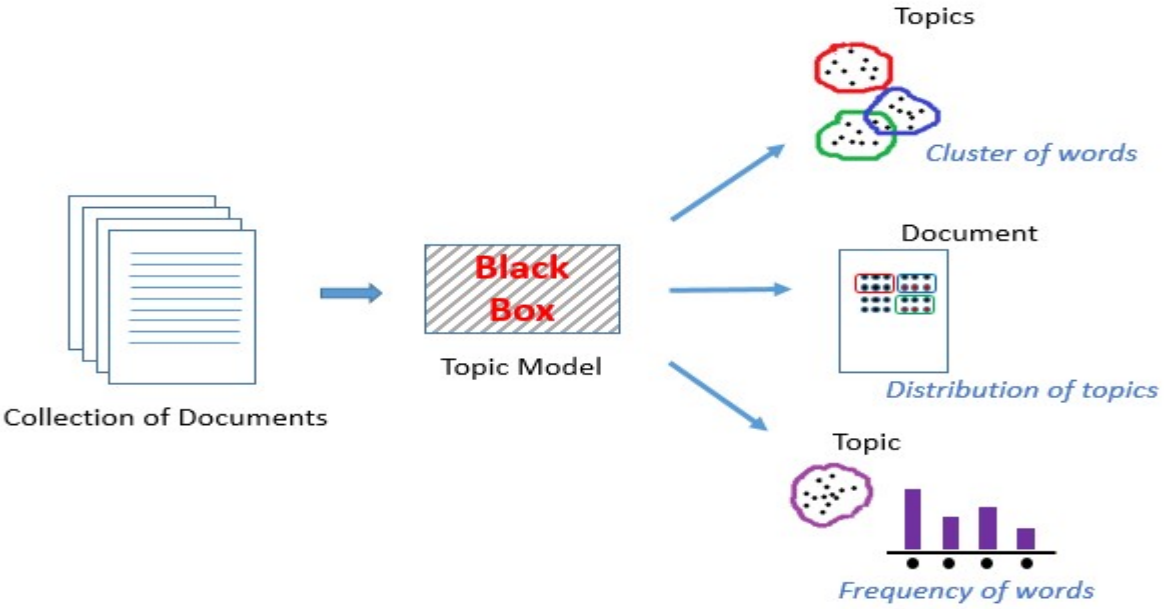
GitHub:https://github.com/shashi-ai1/DataScience/tree/master/NLP/NLP_YouTubevideo_Code

BY: SHASHI KUMAR

https://www.youtube.com/c/ShashiSAS

Topic Modeling

Huge amount of data are available in the form of news article, social media post, customers reviews or others sources. In these text data have lots of hidden information, then how to extract useful information in organize way .so, Topic modeling is the method to extract the information based on the themes.



Topic modeling provides methods for automatically organizing, understanding, searching and summarizing large electronic archives.

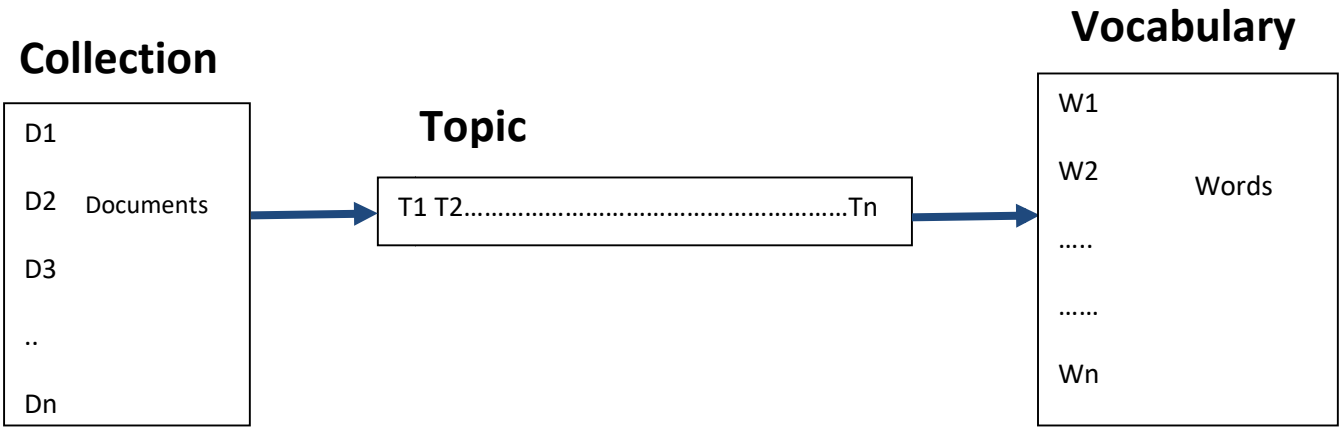
- Discover the hidden themes(topic) from collection of text (Corpus)
- Annotate the documents according to those themes
- Use annotations to organize, summarize, and search the texts

Topic Modeling provides a convenient way to analyze big unclassified text. A topic contains a cluster of words that frequently occurs together. A topic modeling can connect words with similar meanings and distinguish between uses of words with multiple meanings.

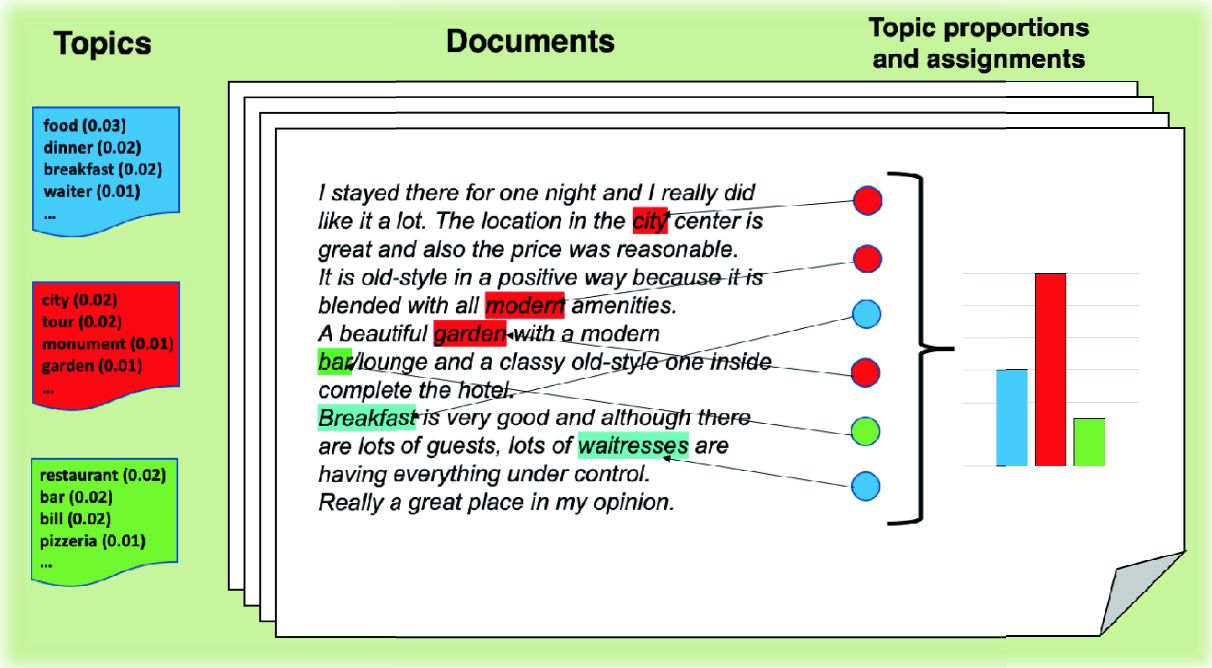
Topic modeling is to discover patterns of word-use and how to connect documents that share similar patterns. So, the idea of topic models is that term which can be working with documents and these documents are mixtures of topics, where a topic is a probability distribution over words.

Topic model gives the information about topic not about the label. Topic evolution modeling can disclose important hidden information in the document corpus, allowing identifying topics with the appearance of time, and checking their evolution with time.

Keyword: Collection, Document, Topic, Words, Probability distribution



- **Collection**: It is a mixture of documents.
- **Topics**: It is a probability distribution over the words.
- **Document**: It is probability distribution over the topics.
- **Words**: Each word is generated from one of the topics.

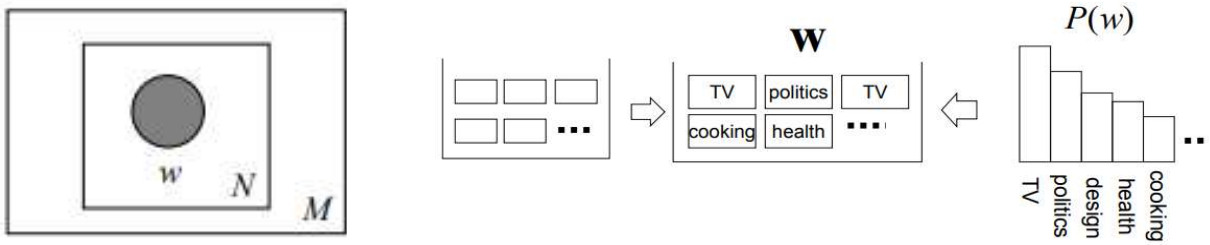


All the document in the collection share the same set of topics, but each document exhibits those topics in different proportions each word in each document is drawn from one of the topics, where the selected topic is chosen from the per-document distribution over topics.

Topic Modeling Techniques:

There are so many techniques to do topic modeling. Following are the most popular techniques.

1. **Unigram Model**



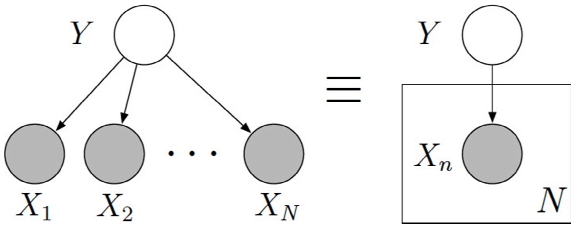
All words in all documents are generated from a distribution. Others words, the words of every document are drawn independently from a **single multinomial distribut**

$$P(\mathbf{w}) = \prod_{n=1}^{N} P(w_n)$$

Where,

- A corpus is a collection of M documents denoted by D .
- A document is a sequence of N words denoted by w = (w1,w2,...,wN), where wn is the nth word in the sequence.

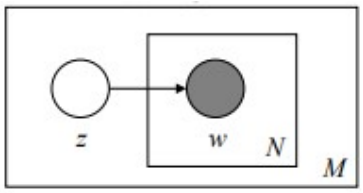
2. **Mixture of unigrams**



$$p(y, x_1, \ldots, x_N) = p(y) \prod_{n=1}^{N} p(x_n|y)$$

X represent as observed variable (shaded) and Y is hidden variable. X has a relationship with Y but not have any relationship between among the X.X_n Represents X has repeated N times or all X has grouped together and represent byX_n .

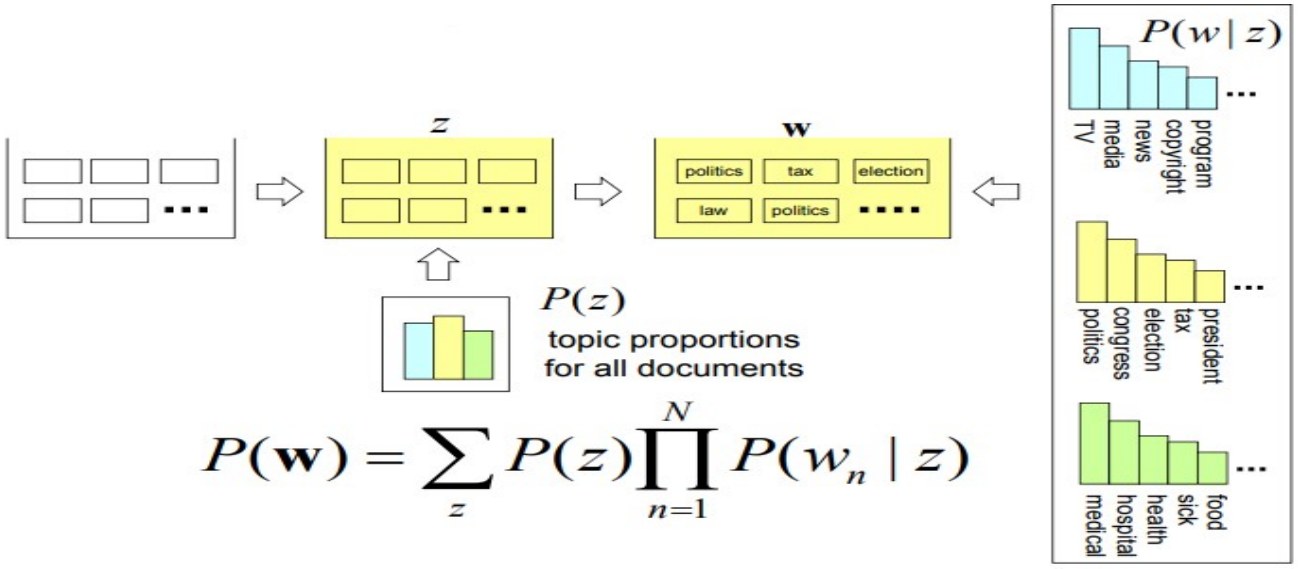
Suppose, Y=z=topic and X_n =W_n = Words then probability distribution function is same as mixture of unigram.



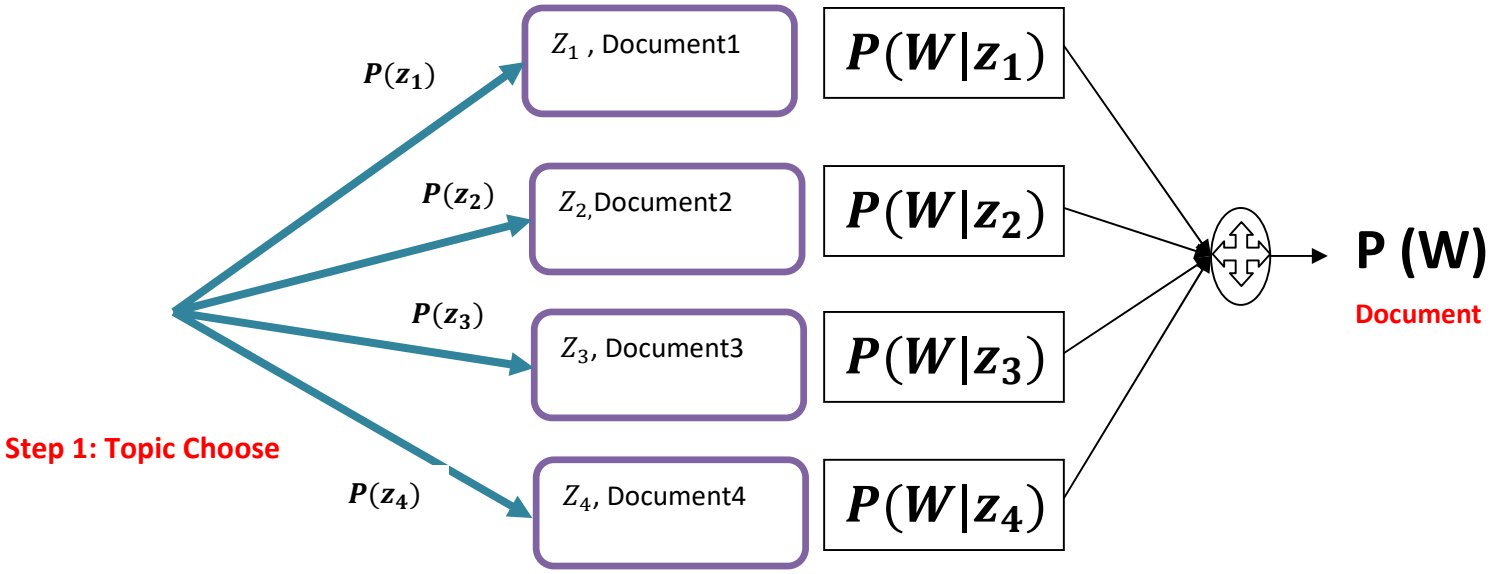
Each document is generated by first choosing a topic z and then generating N words independently from the conditional multinomial p(w|z).

Assumption: Each document having **a single topic**, all words from that topic.

Disadvantage: In real time data have one document have multiple topics.



$$P(\mathbf{w}) = \sum_{z} P(z) \prod_{n=1}^{N} P(w_n \mid z)$$

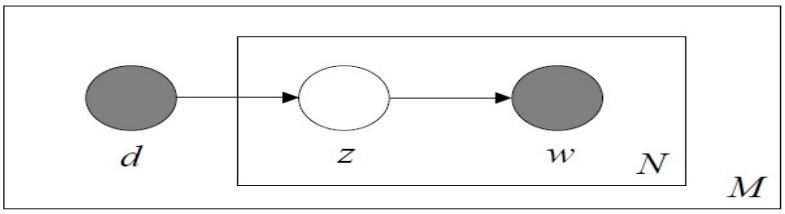


Step 2: all words in a document are generated from a distribution

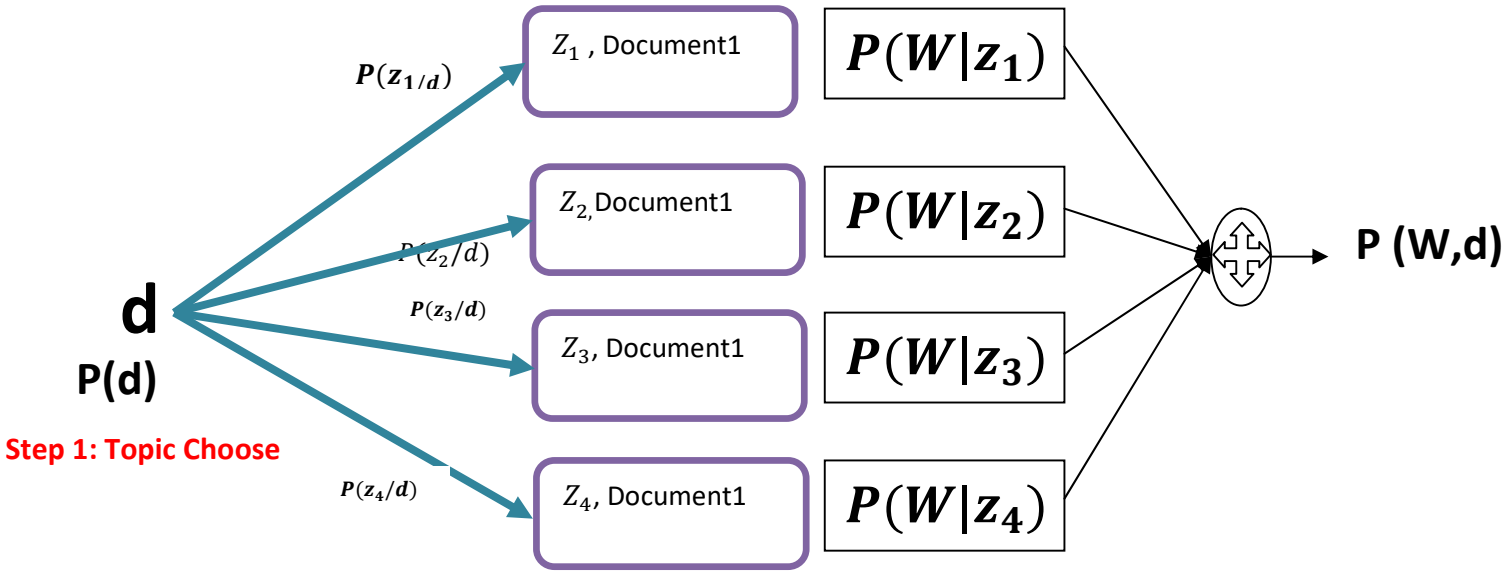
Latent Semantic Analysis:

The main goal of Latent Semantic Analysis (LSA) is to create vector based representation for texts to make semantic content. By vector representation (LSA) computes the similarity between texts to pick the heist efficient related words.

3. Probabilistic latent semantic indexing (pLSI)



The pLSI model attempts to relax the simplifying assumption made in the mixture of unigram model that each document is generated from **only one topic**.



Step 2: all words in a document are generated from a distribution

In above diagram represent one document with probability p(d) have 4 topics and each topic have own distribution.

Probabilistic Latent Semantic Indexing (PLSI) is a latent variable model for co-occurrence data which associates an unobserved topic variable $z_k \in \{z_1, \cdots, z_k\}$ with the occurrence of a word $w_j \in \{w_1, \cdots, w_M\}$ in a particular document $d_i \in \{d_1, \cdots, d_N\}$. As a generative model for word/document co-occurrences, PLSI is defined by the following scheme:
1) Select a document d_i with probability$P(d_i)$,
2) Pick a topic z_k with probability P ($z_k \mid d_i$),
3) Generate a word w_j with probability P ($w_j \mid z_k$).

As a result one obtains an observation pair (d_i,w_j), while the latent topic variable z_k is discarded. Translating the data generation process into a joint probability model results in the expression

$$P(d_i, w_j) = P(d_i) * P(w_j/d_i)$$

$$p(d, w_n) = p(d) \sum_z p(w_n \mid z) p(z \mid d)$$

Disadvantage:

1. The number of parameters (KV+KM) in the model grows linearly with the size of the corpus. This leads to serious problems with over fitting and process time will increases.
2. Not clear how to assign probability to document outside of the training set or unseen documents.

Dirichlet Distribution:

The Dirichlet distribution defines a probability density for a vector valued input having the same characteristics as our multinomial parameter θ. It has support (the set of points where it has non-zero values) over.

$$x_1, \ldots, x_K \text{ where } x_i \in (0, 1) \text{ and } \sum_{i=1}^{K} x_i = 1$$

Where, *K* is the number of variables. Its probability density function has the following form:

$$\text{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{\text{Beta}(\boldsymbol{\alpha})} \prod_{i=1}^{K} \boldsymbol{\theta}_i^{\alpha_i - 1}, \text{ where } \text{Beta}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \text{ and } \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k).$$

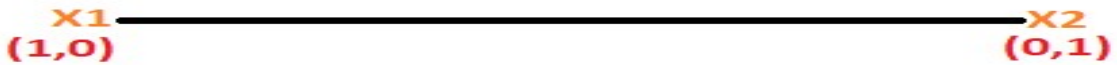
The Dirichlet distribution is parameterized by the vector *α*, which has the same number of elements *K* as our multinomial parameter θ. So we can interpret

$p(\theta|\alpha)$ as answering the question *"what is the probability density associated with multinomial distribution* θ*"*, given that our Dirichlet distribution has parameter α."

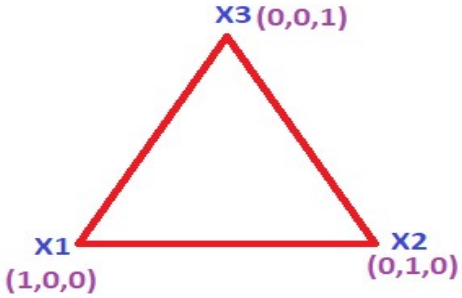
The Dirichlet distribution is an exponential family distribution over the **simplex**, i.e. positive vectors that sum to one.

We can understand Simplex:-

1-Simplex : X1+X2=1



2-Simplex: X1+X2+X3=1



X1 = {1,0,0} or {0.33,0.33,0.34} or {0.5,0.25,0.25}…etc.

The Dirichlet distribution tells what is the probability of getting distribution like {1,0,0} or {0.33,0.33,0.34} or {0.5,0.25,0.25}…etc .

Preferred distribution where one topic has high probability and other topic has low probability or the entire topic have same probability (1,1,1).so these type of constraint we can put by α.

A k-dimensional Dirichlet random variable θ can take values in the (k−1) simplex (a k-vector θ lies in the (k−1) simplex if $\theta_i \geq 0$, $\sum_{i=1}^{K} \theta_i = 1$), and has the following probability density on this Simplex:

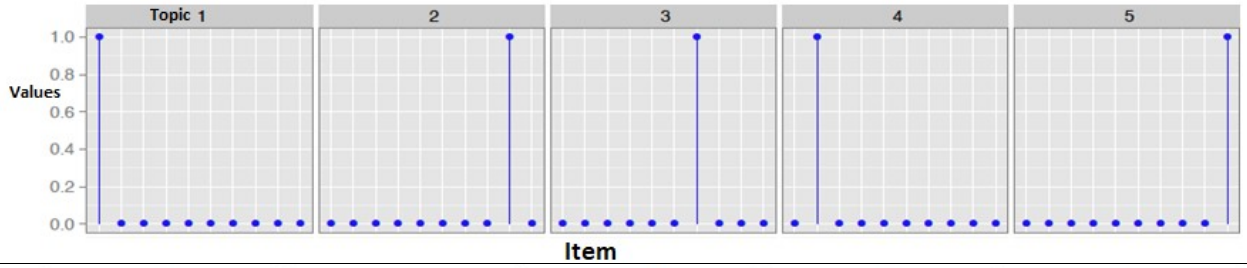
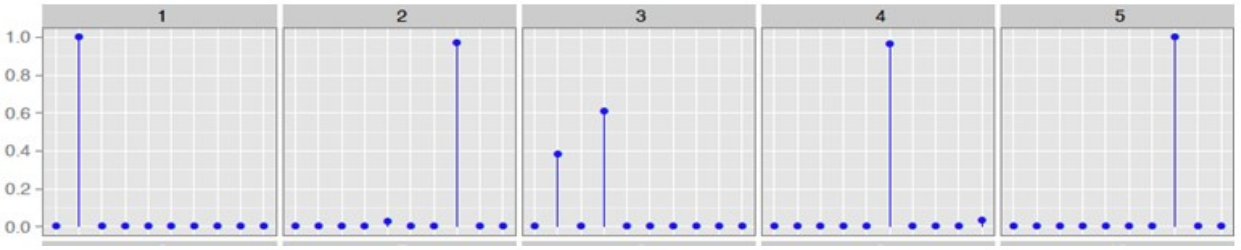
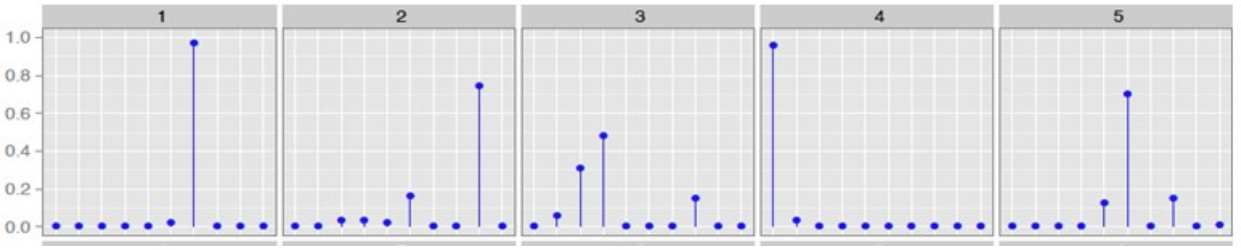
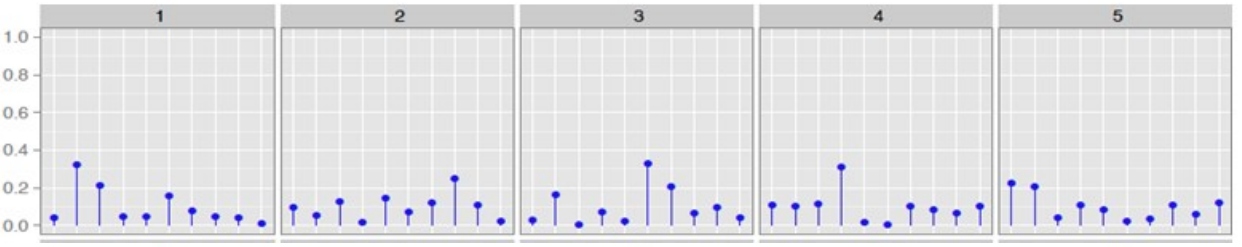
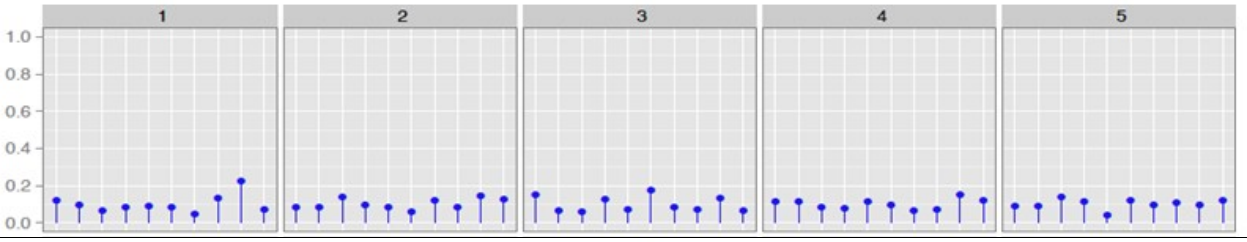
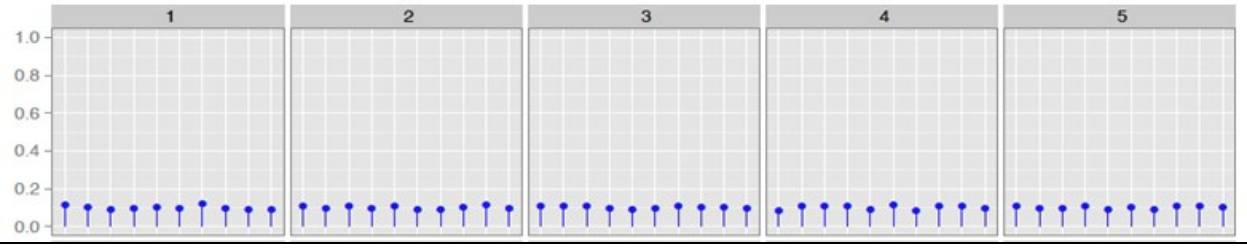
$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1}$$

This expression tells, what is the probability of θ when α has given.

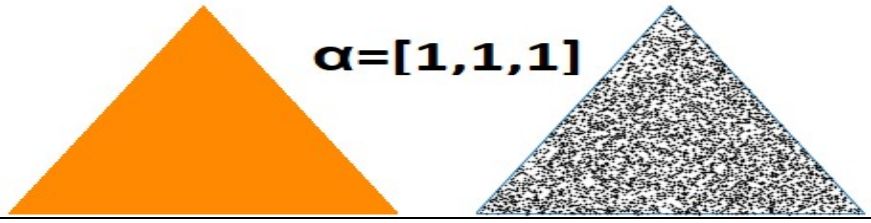
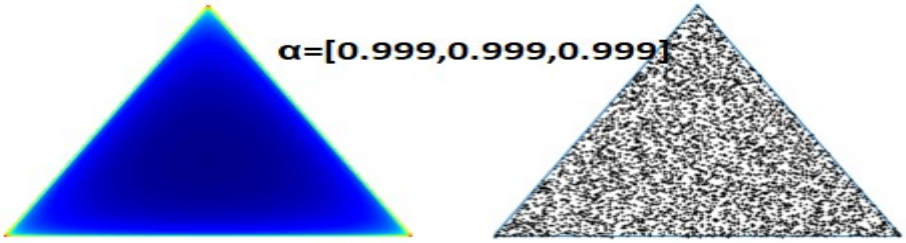
	$\prod \theta^{\alpha-1}$; α=0.1	$\prod \theta^{\alpha-1}$;α=2
θ ={0.98,0.01,0.01}	**1.01*7.94*7.94=4054**	**0.98*0.01*0.01=0.000098**
θ ={0.33,0.33,0.34}	**2.7*2.7*2.6=19.42**	**0.33*0.33*0.34=0.037**

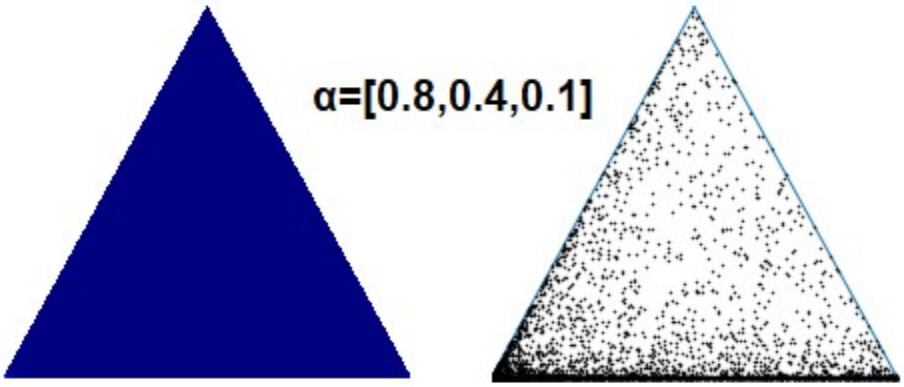
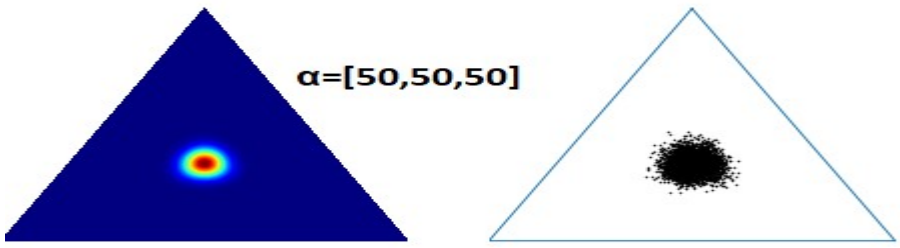
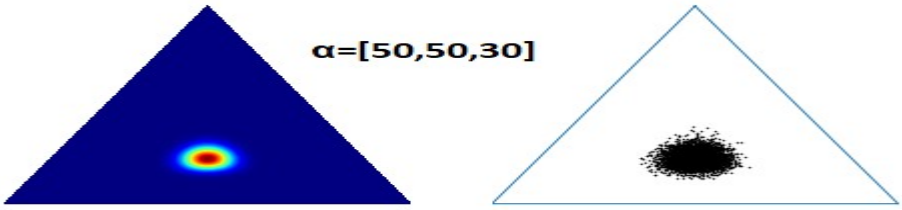
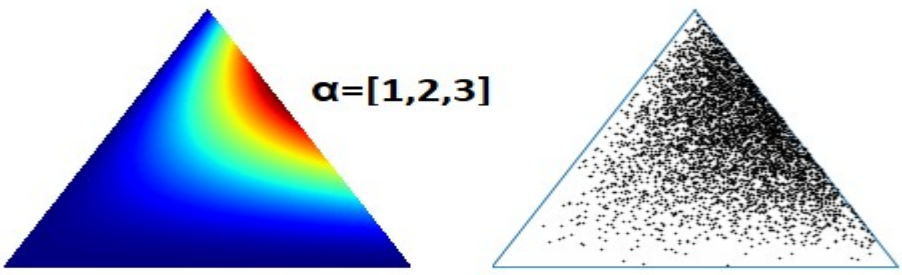
- If increase α,it will prefer to have distribution where have equal probability (θ = {0.33,0.33,0.34}).
- If α is small, it will prefer to have distribution where one topic having high probability and other having low probability.

Effect of ‘α’ on topic distribution

α	Simulations for 10 Topic in 5 documents	
0.001		If α values decreases, it will start favouring distribution where one topic has high probability and another topic has low probability.
0.01		
0.1		
1		All 5 documents have different distribution
10		If α values increases ,it will start favouring distribution which have equal probability
100		

Github : https://github.com/shashi-ai1/DataScience/blob/master/NLP/NLP_YouTubevideo_Code/Visualizing%20Dirichlet%20Distributions.ipynb

<table>
<tr>
<td> 
</td>
<td>The case of α= (1,1,1)yields a uniform distribution, where all points on the simplex are equally probable.</td>
</tr>
<tr>
<td> 
</td>
<td rowspan="2">For values of α<1, the distribution concentrates in the corners and along the boundaries of the simplex. The color scale runs from dark blue (lowest values) to red (highest values). Notice in the plot only the corners of the simplex are red and intermediate values lie along the boundary of the simplex.

For values of α<1 and α1≠ α α2≠ α3, distribution is tends toward one topic has higher probability and another topic has lower probability.</td>
</tr>
<tr>
<td> 
</td>
</tr>
<tr>
<td> 
</td>
<td rowspan="2">For values of α>1, the distribution tends toward the center of the simplex. As α increase, the distribution becomes more tightly concentrated around the center of the simplex.</td>
</tr>
<tr>
<td> 
</td>
</tr>
<tr>
<td> 
</td>
<td rowspan="2">Asymmetric (no central) Dirichlet distribution with a higher value for α.</td>
</tr>
<tr>
<td> 
</td>
</tr>
</table>

4. Latent Dirichlet Allocation

- '***Latent***' has the same sense in LDA as in Latent semantic indexing, i.e. capturing topics as **latent variables**
- The distribution that is used to draw the per-document topic distributions is called a **Dirichlet** distribution. This result is used to **allocate** the words of the documents to different topics.

The reason of appearance of Latent Dirichlet Allocation (LDA) model is to improve the way of mixture models that capture the exchangeability of both words and documents from the old way by PLSA and LSA.

Disadvantage in pLSI:

1. The number of parameters (KV+KM) in the model grows linearly with the size of the corpus. This leads to serious problems with over fitting and process time will increases.
2. Not clear how to assign probability to document outside of the training set or unseen documents.

LDA overcomes both of these problems:

1. The topic mixture weights as a k-parameter hidden random variable rather than a large set of individual parameters which are explicitly linked to the training set.
2. The k+kV parameters in a k-topic LDA model do not grow with the size of the training corpus that LDA does not suffer from the same over fitting issues as pLSI.

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.

Document-Term Matrix

	W1	W2	W3	W4	Wn
D1	0	3	1	0	2
D2	2	0	6	8	3
D3	5	5	3	0	5
D4	1	0	1	1	2
Dn	7	8	0	4	0

=

Document- Topics Matrix

	z1	z2	z3	z4	zk
D1	0	1	1	1	0
D2	1	0	1	6	2
D3	0	1	0	0	4
D4	5	1	4	3	0
Dn	1	0	7	1	1

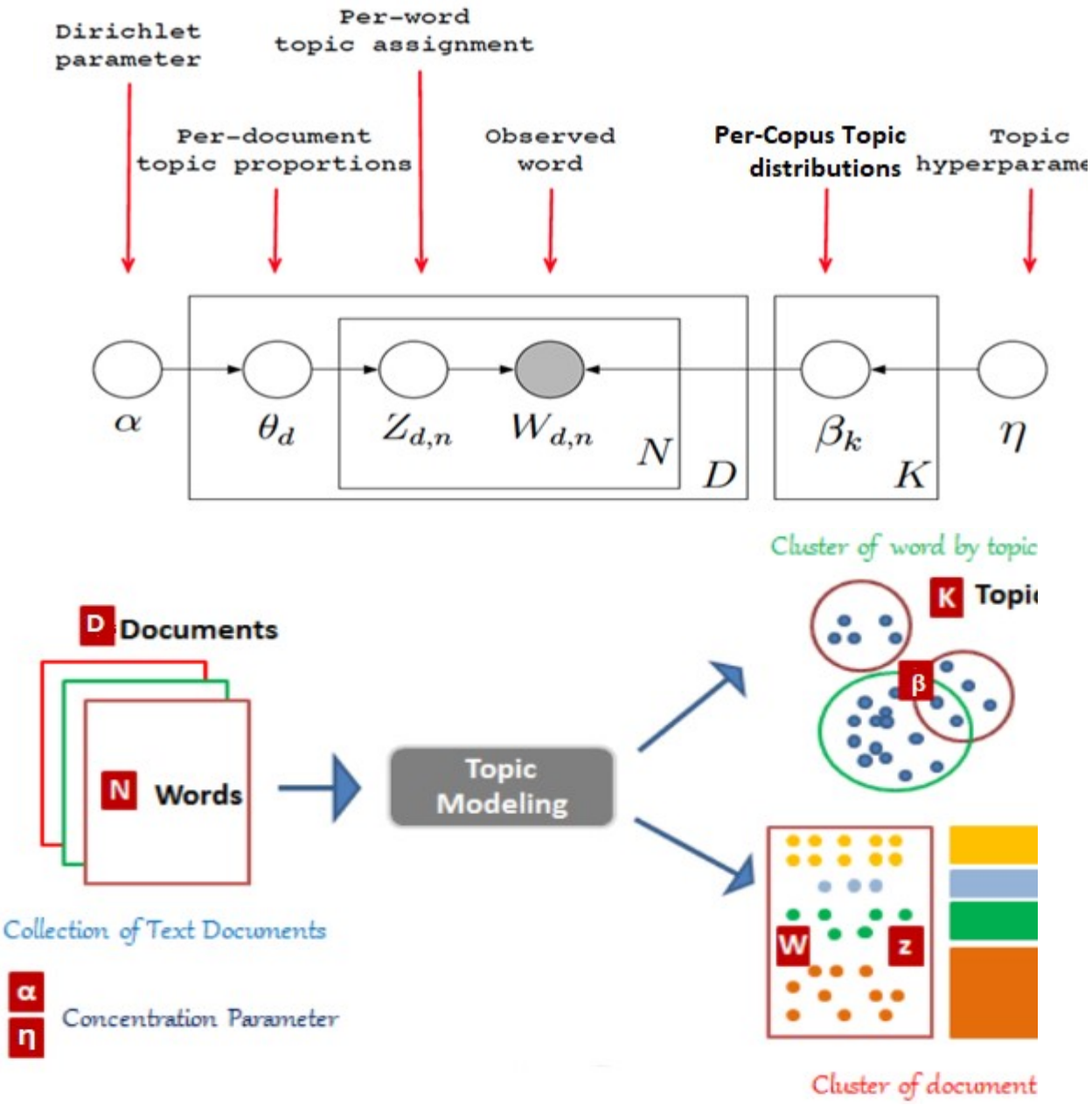
+

Topic -Term Matrix

	W1	W2	W3	W4	Wn
z1	7	6	5	0	1
z2	0	8	0	4	2
z3	4	3	8	7	4
z4	8	0	7	8	9
zk	9	8	4	3	0

LDA converts this Document-Term Matrix into two lower dimensional matrices 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.

Graphical representation of LDA:



The basic idea of the process is, each document is modeled as a mixture of topics, and each topic is a discrete probability distribution that defines how likely each word is to appear in a given topic. These topic probabilities provide a concise representation of a document. Here, a "document" is a "bag of words" with no structure beyond the topic and word statistics. LDA models each of D documents as a mixture over K latent topics, each of which describes a multinomial distribution over a W word vocabulary.

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

Assumptions in Basic Model:

a) The dimensionality k of the Dirichlet distribution (dimensionality of the topic variable z) is assumed known and fixed.
b) The word probabilities are parameterized by a k x V matrix β where βi j = p(wj = 1 | zi = 1)
c) N is independent of all the other data generating variables (θ and z).

In LDA, we assume that words are generated by topics and that those topics are infinitely exchangeable within a document. By **de Finetti's** theorem, the probability of a sequence of words and topics must therefore have the form:

– PLSA + Dirichlet prior on topic proportions
– Bayesian [Blei et. al. JMLR2003]

$$P(\mathbf{w}) = \int P(\theta) \prod_{n=1}^{N} \sum_{z} P(z \mid \theta) P(w_n \mid z) d\theta$$

Dirichlet prior | topic proportions | word distribution

How internally LDA decides topics for a document?

LDA assumes the following generative process for each document w in a corpus D:
Step 1. Draw each topic $\beta_i \sim Dir(\eta)$, for $i \in \{1, \ldots, K\}$.
For each document:
Step 2. Choose topic proportional θ ~ Dir (α).
Step 3. for each of the N words W_n:
(a) Choose a topic Z_n =Multinomial (θ).
(b) Choose a word W_n from p(W_n / Z_n,β), a multinomial probability conditioned on the topic Z_n.

Step 3(a): It assigns a random topic to each word.

Step 3(b): It iterates to each word 'w' for each document and tries to adjust current topic-word assignment with a new assignment. A new topic 'k' is being assigned to the word 'w' with probability 'P' which is product of 2 probabilities; P1 and P2. So for every topic assigned to a word, there are 2 probabilities calculated.

Step 4:
p1=proportion of (topic 'Z'/document'd') i.e proportion of words in document 'd' that are currently assigned to topic 'Z'.
p2= proportion of (word 'w'/topic 'Z') i.e proportion of assignments to topic 'Z' that come from word 'w' across all docs.

LDA computes p1*p2, based upon which it finds optimal topic 'k' for each word 'w'.

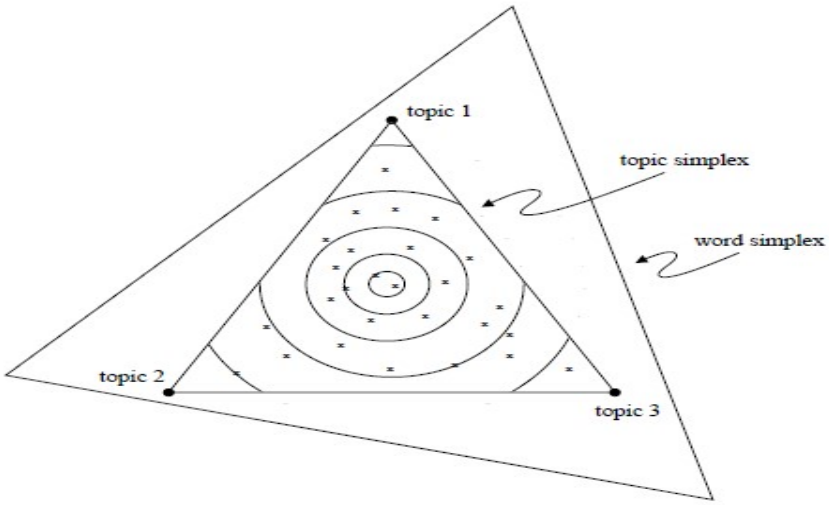
Step 5: This is being repeated many times until a steady stage is achieved where doc topic and topic term distributions are fairly good. This is where LDA converges.

Hyperparametrs:

- **High α** means every document is likely to contain a mixture of most of the topics and not just any single topic specifically.
- **Low α** means a document is more likely to be represented by just few of the topic.
- **High β** means each topic is likely to contain a mixture of most of the words not just any word specifically.
- **Low β** means topic may contain a mixture of just a few of words.

A geometric interpretation

A good way of illustrating the differences between LDA and the other latent topic models is by considering the geometry of the latent space, and seeing how a document is represented in that geometry under each model.



The topic simplex for three topics embedded in the word simplex for three words. The corners of the word simplex correspond to the three distributions where each word (respectively) has probability one. The three points of the topic simplex correspond to three different distributions over words. The **mixture of unigrams** places each document at one of the corners of the topic simplex. The **pLSI** model induces an empirical distribution on the topic simplex denoted by x. **LDA** places a smooth distribution on the topic simplex denoted by the contour lines.
All four of the models described above—unigram, mixture of unigrams, pLSI, and LDA operate in the space of distributions over words. Each such distribution can be viewed as a point on the (V −1)-simplex, which we call the word simplex.

1. The **unigram model** finds a single point on the word simplex and posits that all words in the corpus come from the corresponding distribution. The latent variable models consider k points on the word simplex and form a sub-simplex based on those points, which we call the topic simplex. Note that any point on the topic simplex is also a point on the word simplex. The different latent variable models use the topic simplex in different ways to generate a document.

2. The **mixture of unigrams model** posits that for each document, one of the k points on the word simplex (that is, one of the corners of the topic simplex) is chosen randomly and all the words of the document are drawn from the distribution corresponding to that point.

3. The **pLSI model** posits that each word of a training document comes from a randomly chosen topic. The topics are themselves drawn from a document-specific distribution over topics, i.e., a point on the topic simplex. There is one such distribution for each document; the set of training documents thus defines an empirical distribution on the topic simplex.

4. **LDA** posits that each word of both the observed and unseen documents is generated by a randomly chosen topic which is drawn from a distribution with a randomly chosen parameter. This parameter is sampled once per document from a smooth distribution on the topic simplex.

References:

1. Latent Dirichlet Allocation *DM Blei*, AY Ng, *MI Jordan* - the Journal of machine Learning research, 2003 (https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf)
2. https://towardsdatascience.com/dirichlet-distribution-a82ab942a879
3. https://gist.github.com/tboggs/8778945
4. https://www.cse.iitk.ac.in/users/piyush/courses/tpmi_winter19/tpmi_w19_lec19_slides.pdf
5. http://cbl.eng.cam.ac.uk/pub/Intranet/MLG/ReadingGroup/RCC29Mar2012.pdf
6. https://www.slideshare.net/SoojungHong2/latent-dirichletallocation-presentation
7. https://www.slideshare.net/clauwa/topic-models-lda-and-correlated-topic-models?next_slideshow=1
8. https://cse.iitkgp.ac.in/~pawang/courses/topic_models.pdf
9. https://thesai.org/Downloads/Volume6No1/Paper_21-A_Survey_of_Topic_Modeling_in_Text_Mining.pdf
10. https://www.coursera.org/lecture/text-mining/3-1-probabilistic-topic-models-mixture-of-unigram-language-models-EbbsO
11. https://slideplayer.com/slide/783946/
12. https://bookdown.org/Maxine/tidy-text-mining/latent-dirichlet-allocation.html
13. https://towardsdatascience.com/dirichlet-distribution-a82ab942a879
14. http://blog.bogatron.net/blog/2014/02/02/visualizing-dirichlet-distributions/
15. https://medium.com/analytics-vidhya/nlp-a-complete-guide-for-topic-modeling-latent-dirichlet-allocation-lda-using-gensim-8c836bd3519f
16. https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/
17. https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/
18. https://thinkinfi.com/latent-dirichlet-allocation-for-beginners-a-high-level-overview/