

Latent Dirichlet Allocation(LDA)

Suresh Kumar Choudhary

Assistant Professor,
Department of Data Science & Analytics
Central University of Rajasthan
sureshdewenda@gmail.com

Why Topic Modeling ?

- News providers can use topic modelling to understand articles quickly or cluster similar articles
- unsupervised clustering of images
- to recommend books based on your past readings
- identify important events in history by analysing text

Intuition behind LDA ?

- Doc1: Banana Apple
- Doc2: Orange Apple
- Doc3: Orange Banana
- Doc4: Tiger Cat
- Doc5: Tiger Dog
- Doc6: Cat Dog

Intuition behind LDA ?

- Doc1: Banana Apple
- Doc2: Orange Apple
- Doc3: Orange Banana
- Doc4: Tiger Cat
- Doc5: Tiger Dog
- Doc6: Cat Dog

LDA(K=2)



	Topic 1	Topic 2
Apple	33%	0%
Banana	33%	0%
Orange	33%	0%
Tiger	0%	33%
Cat	0%	33%
Dog	0%	33%

Intuition behind LDA ?

- Doc1: Banana Apple
- Doc2: Orange Apple
- Doc3: Orange Banana
- Doc4: Tiger Cat
- Doc5: Tiger Dog
- Doc6: Cat Dog

LDA(K=2)

	Topic 1	Topic 2
Apple	33%	0%
Banana	33%	0%
Orange	33%	0%
Tiger	0%	33%
Cat	0%	33%
Dog	0%	33%

	Topic 1	Topic 2
Doc1	100%	0%
Doc2	100%	0%
Doc3	100%	0%
Doc4	0%	100%
Doc5	0%	100%
Doc6	0%	100%

Input(What we have)

Output(What we want)

Intuition behind LDA ?

- Doc1: Banana Apple
- Doc2: Orange Apple
- Doc3: Orange Banana
- Doc4: Tiger Cat
- Doc5: Tiger Dog
- Doc6: Cat Dog
- Doc7: Cat Dog Apple

LDA(K=2)

	Topic 1	Topic 2
Apple	33%	0%
Banana	33%	0%
Orange	33%	0%
Tiger	0%	33%
Cat	0%	33%
Dog	0%	33%

	Topic 1	Topic 2
Doc1	100%	0%
Doc2	100%	0%
Doc3	100%	0%
Doc4	0%	100%
Doc5	0%	100%
Doc6	0%	100%
Doc7	33%	66%

Input(What we have)

Output(What we want)

What is the big idea behind LDA ?

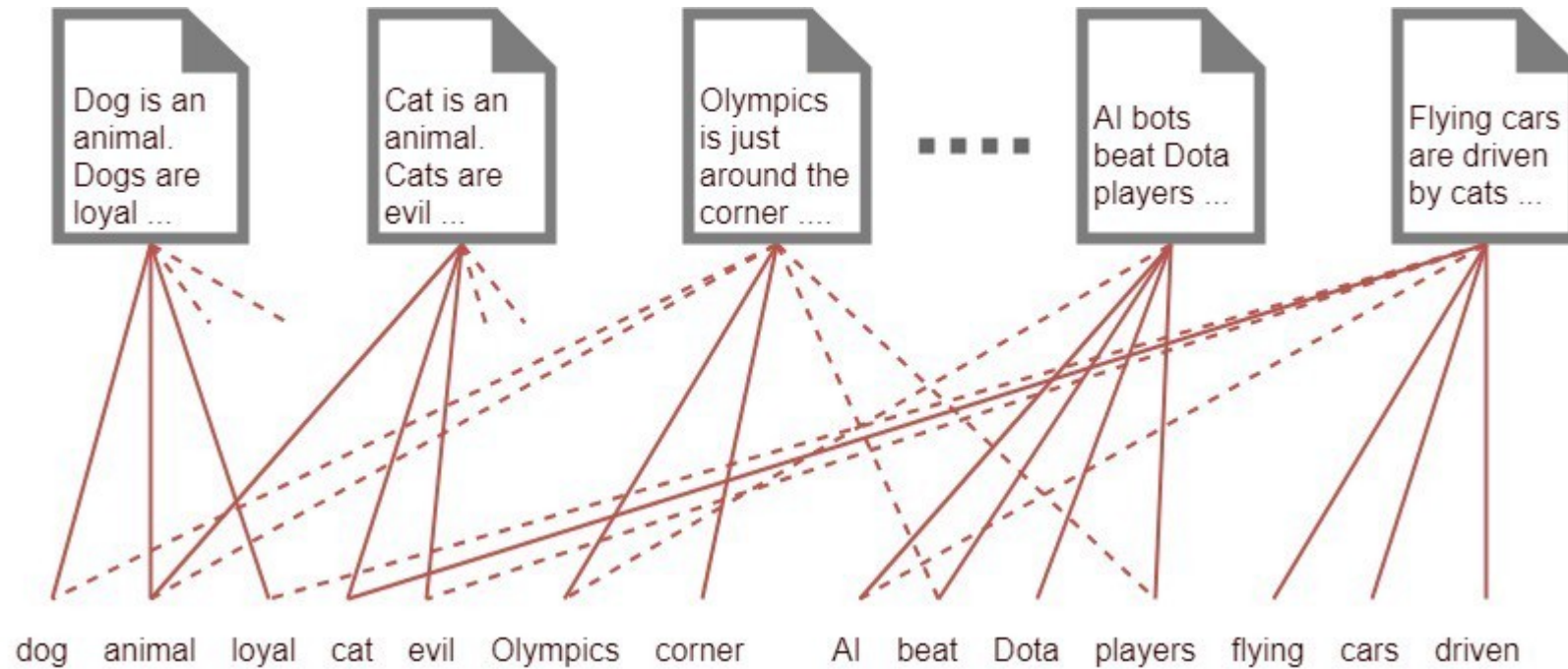
- Each document can be described by a distribution of topics and each topic can be described by a distribution of words

**But why do we use this LDA
idea?**

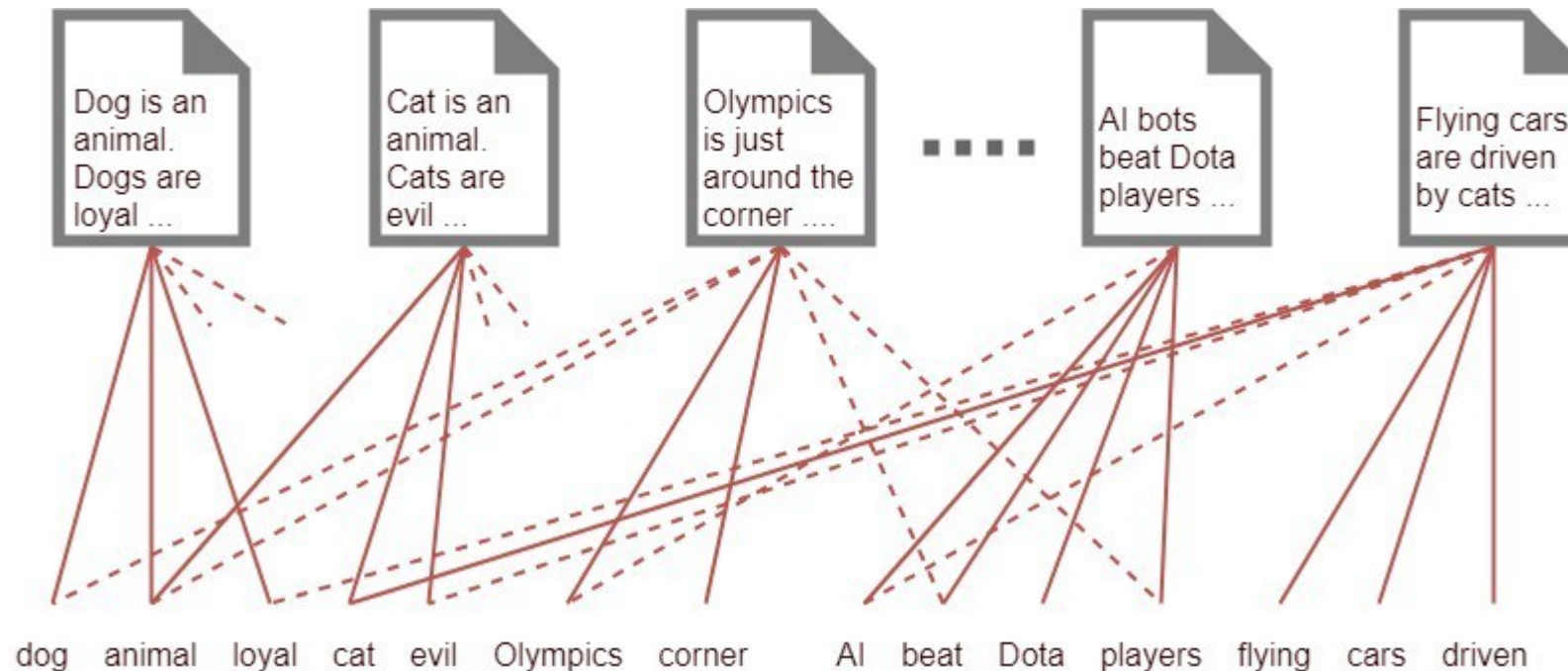
But why do we use this LDA idea?

- Say you have a set of 1000 words (i.e. most common 1000 words found in all the documents)
- you have 1000 documents.
- Assume that each document on average has 500 of these words appearing in each.
- How can you understand what category each document belongs to?
- One way is to connect each document to each word by a thread based on their appearance in the document.

Something like this?



Something like this?



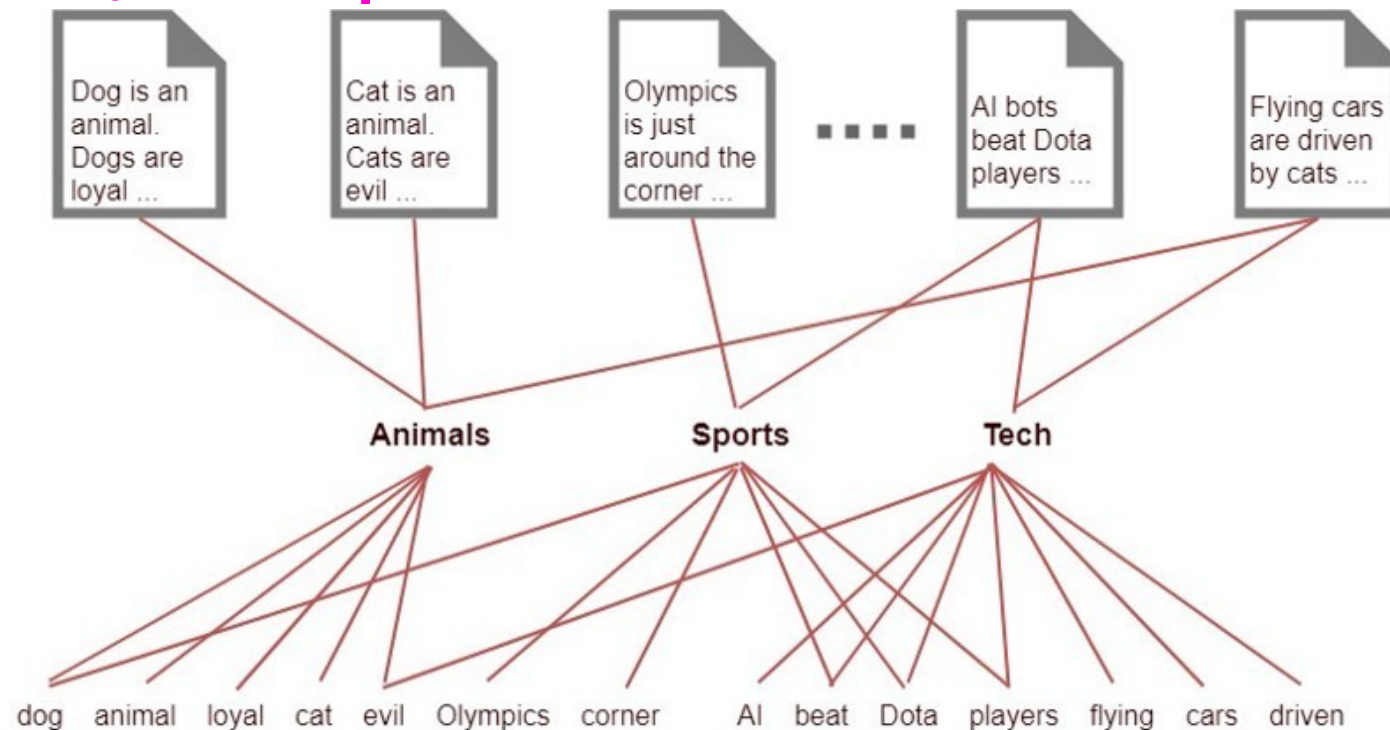
You need around $500 \times 1000 = 500,000$ threads for it

How you can solve this problem?

- by introducing a latent (i.e. hidden) layer
- Say we know 10 topics/themes that occur throughout the documents
- But these topics are not observed, we only observe words and documents, thus topics are latent.

How you can solve this problem?

- by introducing a latent (i.e. hidden) layer
- Say we know 10 topics/themes that occur throughout the documents
- But these topics are not observed, we only observe words and documents, thus topics are latent.



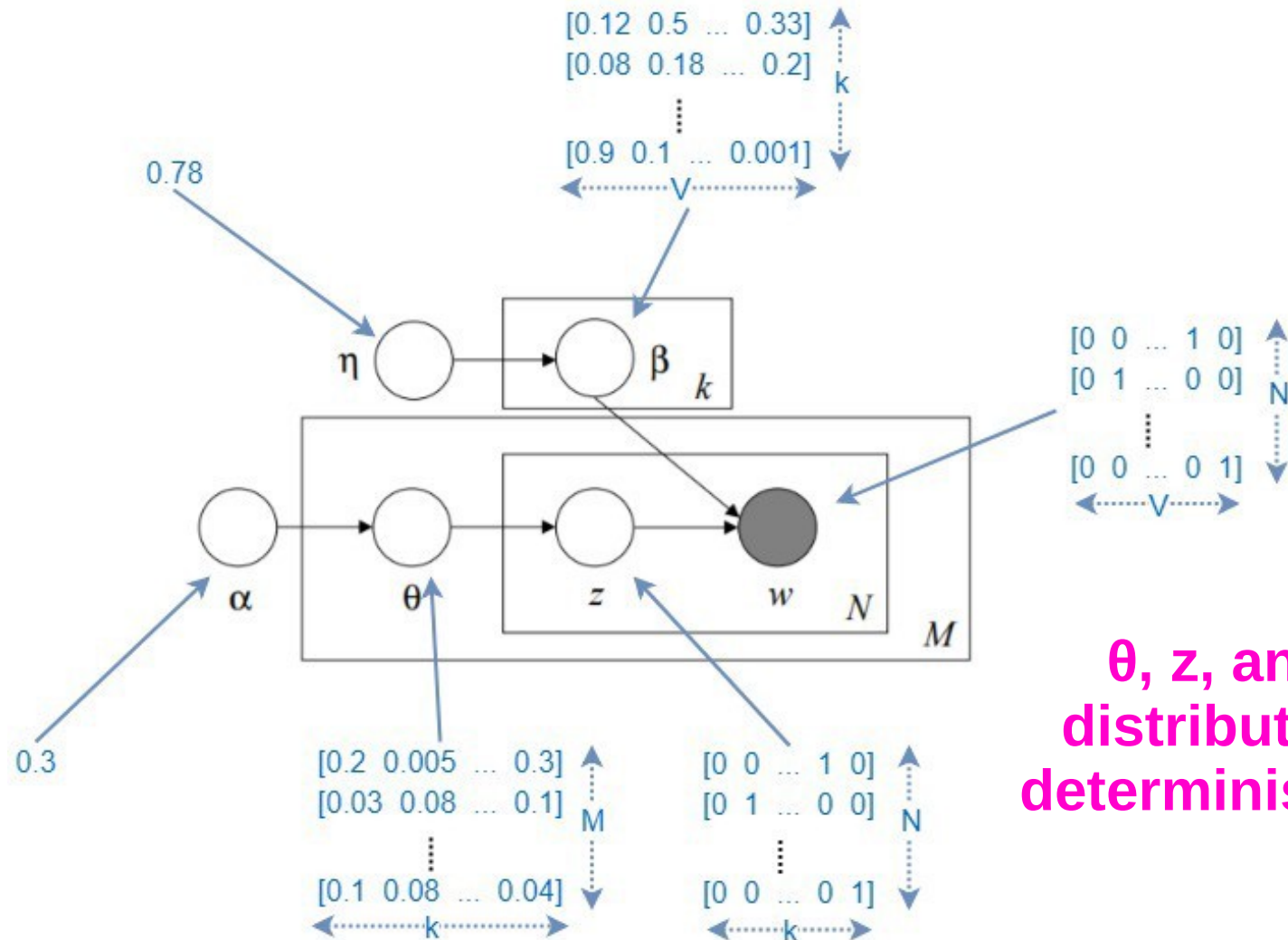
Latent Dirichlet Allocation(LDA)

- LDA does not care the order of the words in the document.
- LDA use the bag-of-words feature representation to represent a document

LDA Notations

- k —Number of topics a document belongs to (a fixed number)
- V —Size of the vocabulary
- M —Number of documents
- N —Number of words in each document
- w —A word in a document. This is represented as a one hot encoded vector of size V (i.e. V —vocabulary size)
- \mathbf{w} (bold w): represents a document (i.e. vector of “ w ”s) of N words
- D —Corpus, a collection of M documents
- z —A topic from a set of k topics. A topic is a distribution words. For example it might be, Animal = (0.3 Cats, 0.4 Dogs, 0 AI, 0.2 Loyal, 0.1 Evil)

Graphical model of the LDA



θ , z , and β are distributions, not deterministic values

θ and β ?

- θ is a random matrix
- where $\theta(i,j)$ represents the probability of the i th document to containing words belonging to the j th topic
- $\beta(i,j)$ represents the probability of the i th topic containing the j th word
- Both follows Dirichlet Distribution.

Dirichlet distribution

Review of Multinomial and Dirichlet distributions:

1. Multinomial:

$$P(X_1 = x_1, \dots, X_K = x_K) = \frac{n!}{x_1! \dots x_K!} p_1^{x_1} \dots p_K^{x_K} \quad X_i \in \{0, \dots, n\} \quad \sum_{i=1}^K X_i = n$$

2. Dirichlet: Good for modeling a distribution over distributions

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \dots \theta_K^{\alpha_K-1} \quad \alpha = k - \text{dimensional vector} \quad \alpha_i > 0$$

variable θ can take values in the $(k-1)$ simplex: $\theta_i > 0$ and $\sum_{i=1}^K \theta_i = 1$

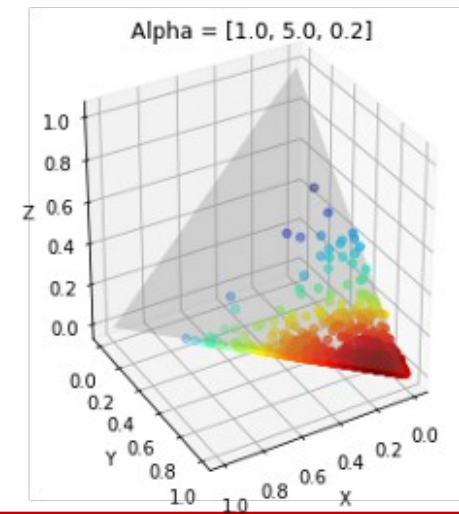
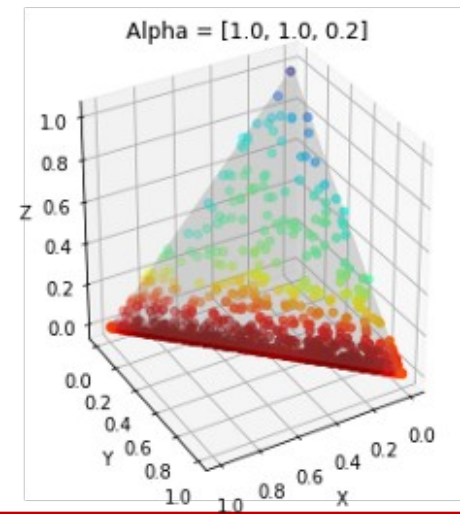
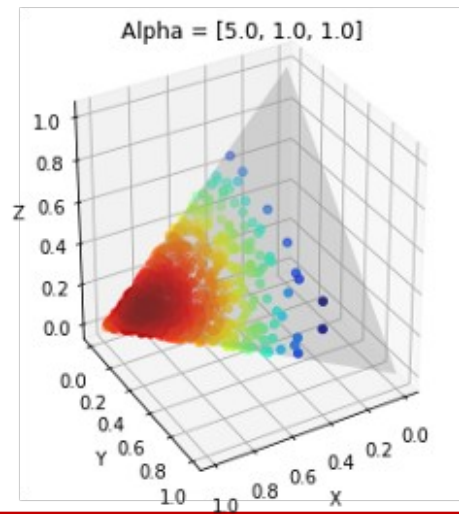
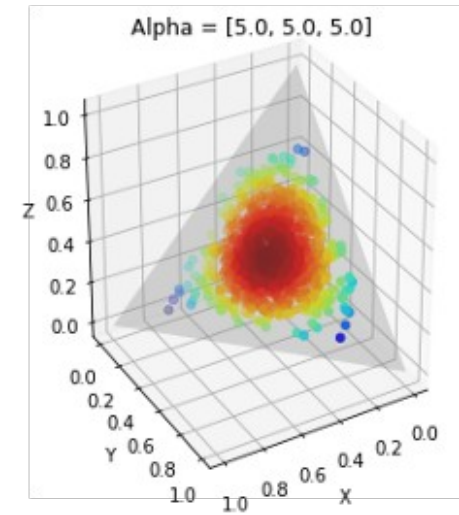
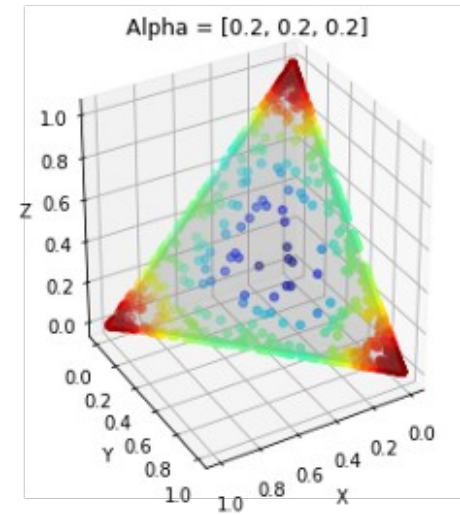
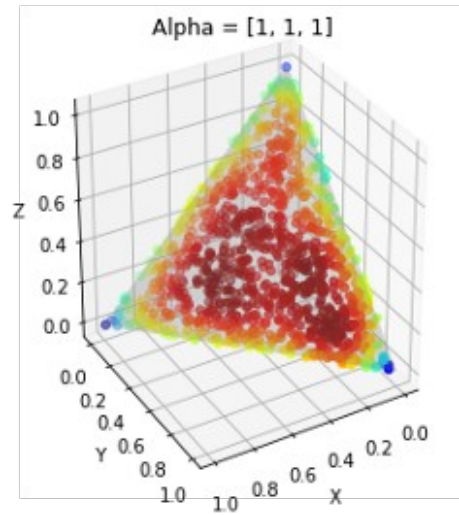
Dirichlet distribution

- Dirichlet distribution is the multivariate generalisation of the Beta distribution
- For an N-dimensional Dirichlet distribution you have a N length vector as α
- Lets take an example of a 3-dimensional problem, where we have 3 parameters in α that affects the shape of θ (i.e. distribution)
- Large α values push the distribution to the middle of the triangle, where smaller α values push the distribution to the corners.

Dirichlet distribution

- In the LDA model, we want the topic mixture proportions for each document to be drawn from some distribution.
 - distribution = “probability distribution”, so it sums to one
- So, we want to put a prior distribution on multinomials. That is, k -tuples of non-negative numbers that sum to one.
 - We want probabilities of probabilities
 - These multinomials lie in a $(k-1)$ -simplex
 - Simplex = generalization of a triangle to $(k-1)$ dimensions.
- Our prior:
 - Needs to be defined for a $(k-1)$ -simplex.
 - Conjugate to the multinomial

Dirichlet distribution



Effect of Alpha

- When $\alpha < 1.0$, the majority of the probability mass is in the "corners" of the simplex, generating mostly documents that have a small number of topics.
- When $\alpha > 1.0$, the most documents contain most of the topics.

How do we learn the LDA?

let us list down the latent (hidden) variable we need to find:

- α —Distribution related parameter that governs what the distribution of topics is for all the documents in the corpus looks like
- θ —Random matrix where $\theta(i,j)$ represents the probability of the i th document to containing the j th topic
- η —Distribution related parameter that governs what the distribution of words in each topic looks like
- β —A random matrix where $\beta(i,j)$ represents the probability of i th topic containing the j th word.

Mathematical Formulation, what do we need to learn?

$$P(\theta_{1:M}, \mathbf{z}_{1:M}, \beta_{1:k} | \mathcal{D}; \alpha_{1:M}, \eta_{1:k})$$

I have a set of M documents, each document having N words, where each word is generated by a single topic from a set of K topics. I'm looking for the joint posterior probability of:

- θ —A distribution of topics, one for each document,
- \mathbf{z} — N Topics for each document,
- β —A distribution of words, one for each topic,

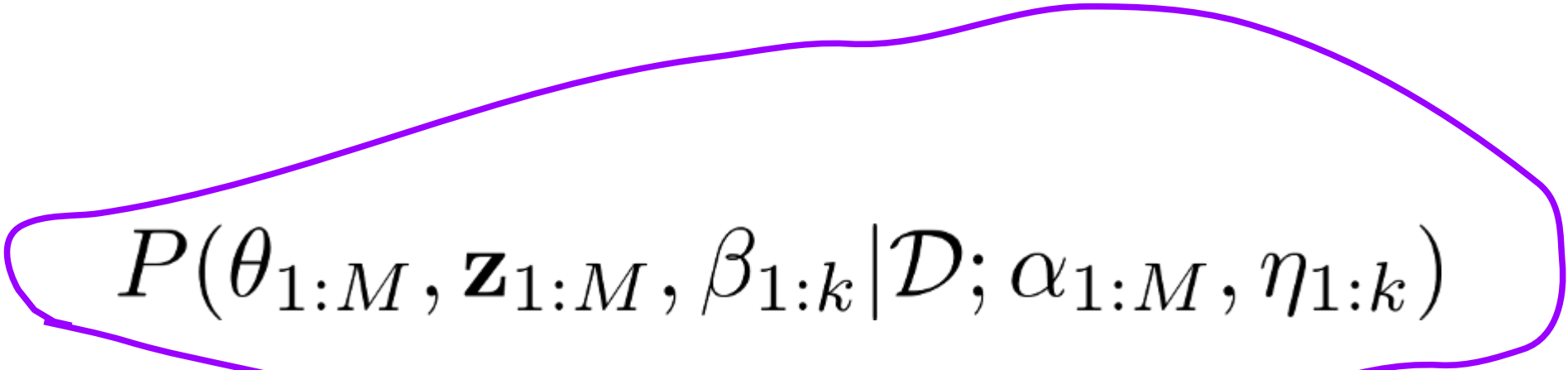
Given:

- \mathcal{D} —All the data we have (i.e. the corpus),

and using parameters:

- α —A parameter vector for each document (document—Topic distribution)
- η —A parameter vector for each topic (topic—word distribution)

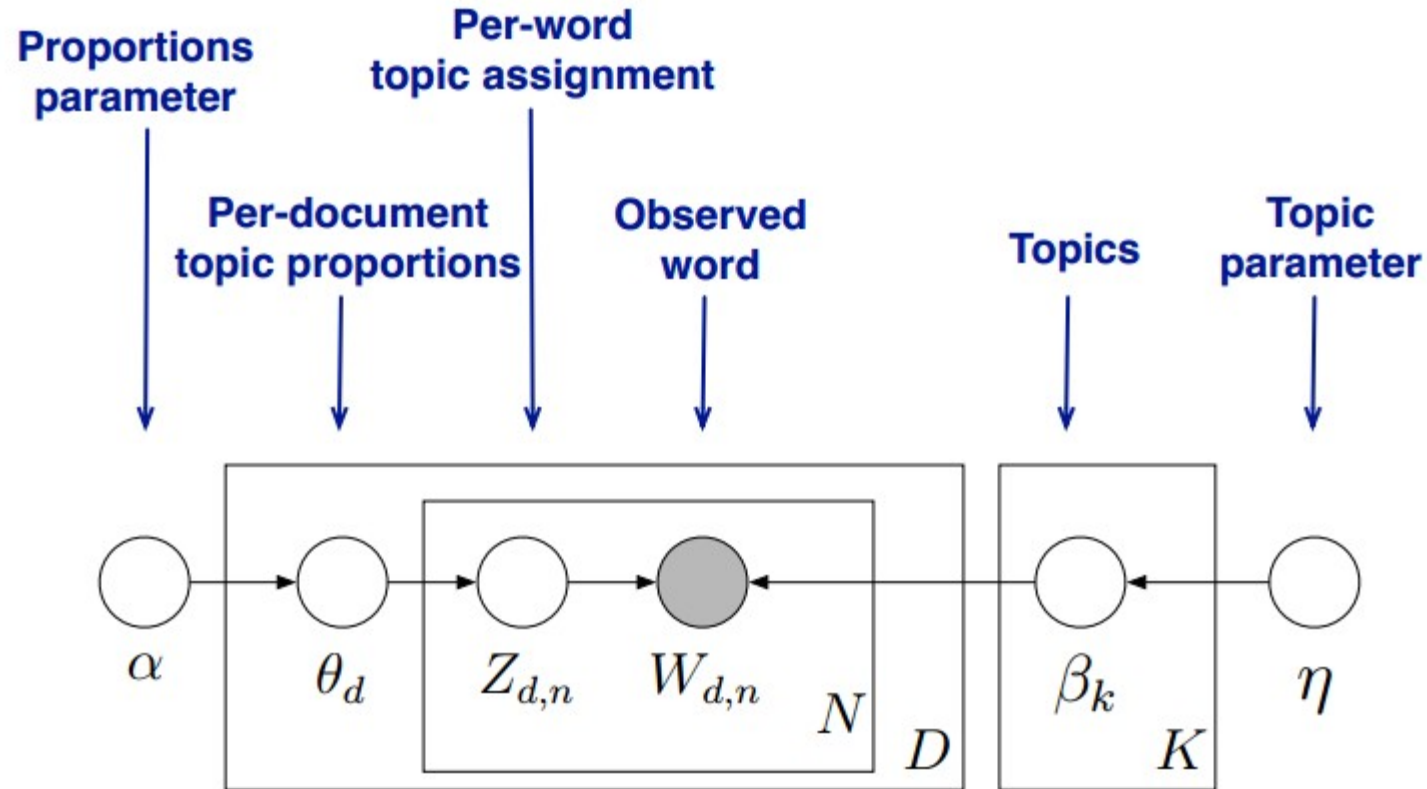
Mathematical Formulation, what do we need to learn?


$$P(\theta_{1:M}, \mathbf{z}_{1:M}, \beta_{1:k} | \mathcal{D}; \alpha_{1:M}, \eta_{1:k})$$



How can I solve this?

Mathematical Formulation, what do we need to learn?



$$p(\beta, \theta, \mathbf{z}, \mathbf{w}) = \left(\prod_{i=1}^K p(\beta_i | \eta) \right) \left(\prod_{d=1}^D p(\theta_d | \alpha) \prod_{n=1}^N p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

Variational Inference

- There are many ways to solve it.
- we're going to approximate that with some known probability distribution that closely matches the true posterior
- minimise the KL divergence between the approximation and true posterior as an optimisation problem

$$\gamma^*, \phi^*, \lambda^* = \operatorname{argmin}_{(\gamma, \phi, \lambda)} D(q(\theta, \mathbf{z}, \beta | \gamma, \phi, \lambda) || p(\theta, \mathbf{z}, \beta | \mathcal{D}; \alpha, \eta))$$

- γ , ϕ and λ represent the free variational parameters we approximate θ, \mathbf{z} and β with, respectively
- $D(q||p)$ represents the KL divergence between q and p
- by changing γ, ϕ and λ , we get different q distributions having different distances from the true posterior p .

Gibbs Sampling

- Go through each document, and randomly assign each word in the document to one of the K topics.
 - Notice that this random assignment already gives you both topic representations of all the documents and word distributions of all the topics (albeit not very good ones).
 - So to improve on them, for each document d
 -Go through each word w in d
 -And for each topic t , compute two things:
 - 1) $p(\text{topic } t \mid \text{document } d)$ = the proportion of words in document d that are currently assigned to topic t , and
 - 2) $p(\text{word } w \mid \text{topic } t)$ = the proportion of assignments to topic t over all documents that come from this word w . Reassign w a new topic, where you choose topic t with probability $p(\text{topic } t \mid \text{document } d) * p(\text{word } w \mid \text{topic } t)$ (according to our generative model, this is essentially the probability that topic t generated word w , so it makes sense that we resample the current word's topic with this probability).
-

Gibbs Sampling

- In other words, in this step, we're assuming that all topic assignments except for the current word in question are correct, and then updating the assignment of the current word using our model of how documents are generated.
- After repeating the previous step a large number of times, you'll eventually reach a roughly steady state where your assignments are pretty good.
- So use these assignments to estimate the topic mixtures of each document (by counting the proportion of words assigned to each topic within that document) and the words associated to each topic (by counting the proportion of words assigned to each topic overall).

Gibbs Sampling

$$p(z_{d,n} = k | \vec{z}_{-d,n}, \vec{w}, \alpha, \lambda) = \frac{n_{d,k} + \alpha_k}{\sum_i^K n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \lambda_{w_{d,n}}}{\sum_i v_{k,i} + \lambda_i}$$

where:

- $n(d,k)$: Number of times document d use topic k
-
- $v(k,w)$: Number of times topic k uses the given word
-
- α_k : Dirichlet parameter for document to topic distribution
-
- λ_w : Dirichlet parameter for topic to word distribution

Gibbs Sampling with Example

- Word Topic Assignment Randomly

India	enters	world	cup	final
1	3	1	2	4

- count matrix $v(k,w)$

	1	2	3	4
India	70	5	0	8
enters	2	3	15	6
world	28	4	12	1
cup	6	43	6	0
final	7	0	9	31

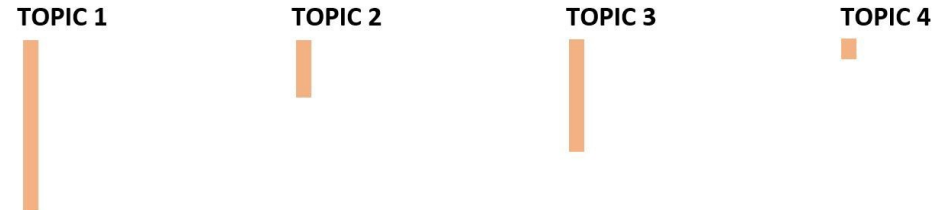
Now let's change the assignment of word "world" in the document

Gibbs Sampling with Example

- First, we will reduce the count of world in topic 1 from 28 to 27 as we don't know to what topic world belongs.
- Second let's represent the matrix $n(d,k)$ in the following way to show how much a document use each topic



- Third, let's represent $v(k,w)$ in the following way to show how many times each topic is assigned to this word



- Fourth, we will multiply these two matrices to get our conditional probabilities
 - Finally, we will randomly pick any of the topic and will assign that topic to world and we will repeat these steps for all other words as well. Intuitively, topic with highest conditional probability should be selected but as we can see other topics also have some chance to get selected
-

Any Question?