Topic Modeling:

Latent Dirichlet Allocation (LDA)

Objectives

- Review topic modeling
- Review NMF
- State what LDA stands for
  - Pronounce Dirichlet, backup
- Describe what the Dirichlet distribution is
- Describe how LDA does topic modeling
  - Explain what a generative, probabilistic model is
- Perform hand calculations to understand how the *topic-document* and *word-topic* distributions are created.
- Compare/contrast LDA and NMF
- Perform LDA in sklearn and gensim

Review: Topic modeling

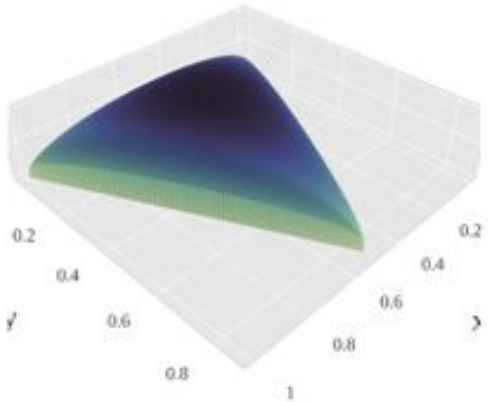
- What is a “topic”?
- What is the point of doing topic modeling?
- Is topic modeling a supervised learning technique?
  - Can it assist supervised learning?
- Is topic modeling typically the result of hard or soft clustering?
- Name some algorithms that perform topic modeling.

Review: NMF

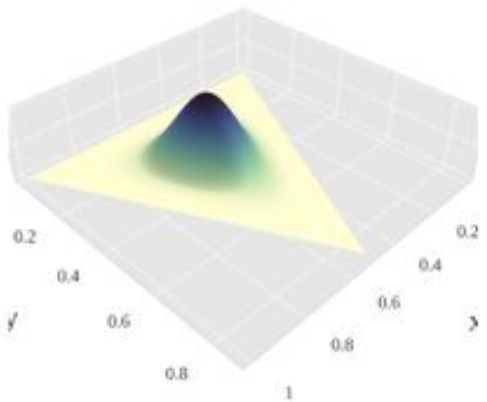
- What does it stand for?
- Conceptually, what is it?
- For a feature matrix that contains only zero and positive values, what are the advantages of using NMF relative to SVD or PCA? Disadvantages?
- What matrices result from NMF, and how do you interpret them?
- How do you know if your NMF matrices are “right”?
- Describe the algorithm used to factor a feature matrix into NMF matrices.

The Dirichlet Distribution

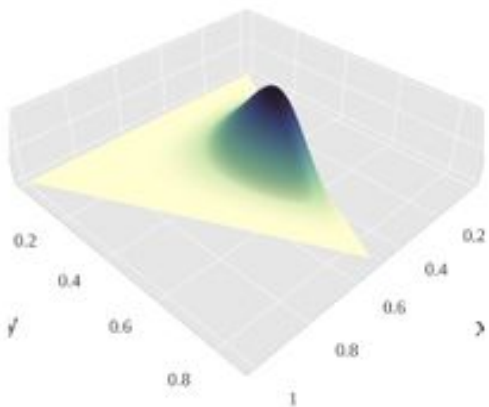
The Dirichlet distribution (after Peter Gustav Lejeune Dirichlet) is a family of continuous multivariate probability distributions parameterized by a vector α of positive values. It's a multivariate generalization of the beta distribution.



α = (1.3, 1.3, 1.3)



α = (7, 7, 7)



α = (6, 2, 6)

The support of the Dirichlet distribution is a set of *K*-dimensional vectors whose entries are real numbers in the interval (0,1), with their sum (L1 norm) equal to 1. These can be viewed as the probabilities of a *K*-way categorical event.

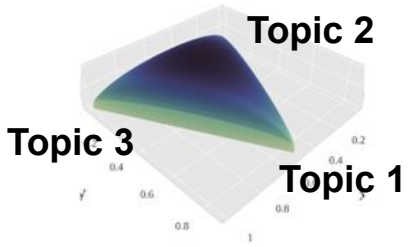
Wikipedia

The Dirichlet Distribution in Topic Modeling

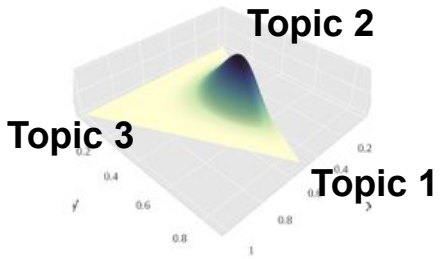
The Dirichlet distribution describes, for a given document, its distribution of topics.

documents are distributions of topics

Document 1



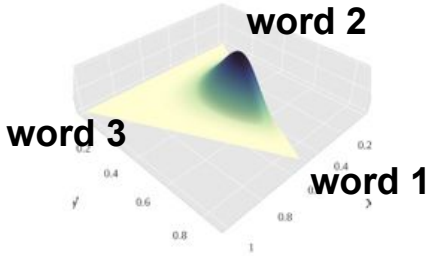
Document 2



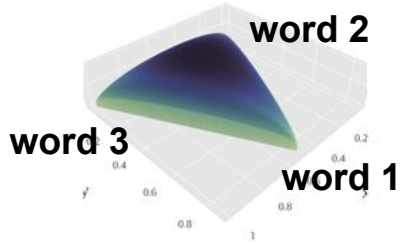
The Dirichlet distribution also describes, for a given topic, its distribution of words.

topics are distributions of words

Topic 1



Topic 2



Latent Dirichlet Allocation

In NLP, **Latent Dirichlet allocation** (**LDA**) is a *generative statistical model* that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. Wikipedia

Similar to NMF, LDA yields two matrices, denoted θ and ϕ: where θ represents topic-document distribution (topics in document) and ϕ represents the word-topic distribution (words in topic).

LDA example

Suppose you have the following set of sentences:

1. I like to eat broccoli and bananas.
2. I ate a banana and spinach smoothie for breakfast.
3. Chinchillas and kittens are cute.
4. My sister adopted a kitten yesterday.
5. Look at this cute hamster munching on a piece of broccoli.

Given these sentences and asked for 2 topics, LDA might produce something like

- **Sentences 1 and 2**: 100% Topic A
- **Sentences 3 and 4**: 100% Topic B
- **Sentence 5**: 60% Topic A, 40% Topic B
- **Topic A**: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, … (at which point, you could interpret topic A to be about food)
- **Topic B**: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, … (at which point, you could interpret topic B to be about cute animals)

E. Chen's blog

LDA example

Suppose you have the following set of sentences:

1. I like to eat broccoli and bananas.
2. I ate a banana and spinach smoothie for breakfast.
3. Chinchillas and kittens are cute.
4. My sister adopted a kitten yesterday.
5. Look at this cute hamster munching on a piece of broccoli.

ϕ: word-topic*

	broccoli	bananas	chinchillas	kittens
Topic A	0.3	0.15	0	0
Topic B	0	0	0.2	0.2

*does not contain all words - rows should sum to 1

θ: topic-document

	Topic A	Topic B
Sentence 1	1	0
Sentence 2	1	0
Sentence 3	0	1
Sentence 4	0	1
Sentence 5	0.6	0.4

LDA is a generative model

LDA represents documents as **mixtures of topics** that “spit out” words according to probability distributions.

LDA assumes that documents are produced in the following fashion.

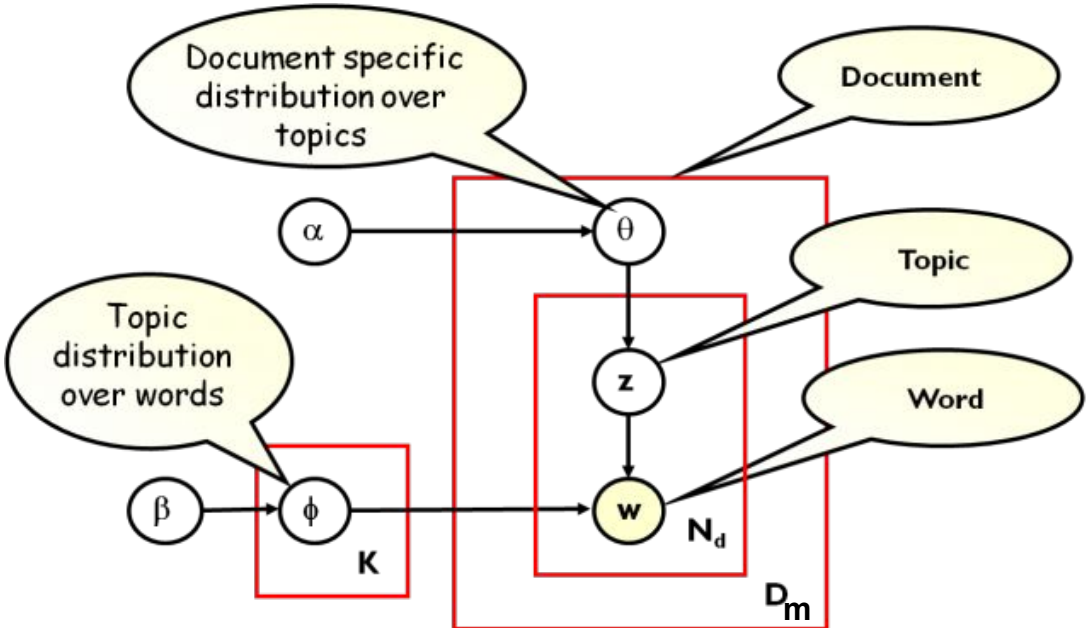
When writing each document, you:

- Decide on the number of words N the document will have (say, according to a Poisson distribution).
- Choose a topic mixture for the document (according to a Dirichlet distribution over a fixed set of K topics). For example, assuming that we have the two topics (food and animal), you might choose the document to consist of 1/3 food and ⅔ animal.
- Choose words from those topics to make the document.

adapted from E. Chen’s blog

LDA depicted graphically

- α is a hyper-prior for the Dirichlet Process over per-document topic distributions
- β is the hyper-prior for the Dirichlet Process over per-topic word distributions
- θ is the topic distribution for document *m*
- ϕ is the word distribution for topic *k*
- $z_{m,n}$ is the topic for the *n*th word in document *m*
- $w_{m,n}$ is the specific word



LDA generative process - more mathematically

1. Choose $\theta_m \sim \text{Dir}(\alpha)$, where $m \in \{1,\dots M\}$ and $\text{Dir}(\alpha)$ is the Dirichlet distribution for α
2. Choose $\phi_k \sim \text{Dir}(\beta)$, where $k \in \{1,\dots K\}$
3. For each of the word positions (m,n), where $n \in \{1,\dots N\}$, and $m \in \{1,\dots M\}$
   - Choose a topic $z_{m,n} \sim \text{Multinomial}(\theta_m)$
   - Choose a word $w_{m,n} \sim \text{Multinomial}(\phi_{m,n})$

M is the number of documents
N is the number of words in a document (different for each document)
K is the total number of topics

LDA: learning the topic-doc., word-topic distributions

Initially:
Go through each document, and randomly assign each word in the document to one of the K topics - this gives you your initial θ and ϕ matrices (not good ones!).

Loop (to improve θ and ϕ):
for each document **d**:
    for each word **w** in **d**:
        for each topic **t**, calculate:
            1) P(**t** | **d**) = the proportion of words in **d** that are currently assigned to **t**
            2) P(**w** | **t**) = the proportion of assignments to **t** over all documents that come from word **w**.
        update the topic assignment for **w** by choosing the topic with the highest probability: P(**t** | **d**) * P(**w** | **t**)

Process called Gibbs sampling

LDA: Breakout

Suppose you have the following set of sentences and are looking for two topics, **A** and **B**:

- I like to eat broccoli and bananas.
- Chinchillas and kittens are cute.
- Look at this cute hamster munching on a piece of broccoli.

Random assignment:

	eat	broccoli	banana	chinchilla	kitten	cute	hamster	munch
Sentence 1	A	B	A					
Sentence 2				B	A	B		
Sentence 3		A				A	B	B

Update the topic assignments for the words in **Sentence 1** (by calculating the conditional probabilities described in the prior slide). **Look for help on the next slide.**

LDA: Breakout

1. Calculate probabilities:

Num w in A =
Num w in B =
P(topic **A** | sentence 1) =
P(topic **B** | sentence 1) =
P(eat | topic **A**) =
P(eat| topic **B**) =
P(broccoli | topic **A**) =
P(broccoli | topic **B**) =
P(banana | topic **A**) =
P(banana| topic **B**) =

2. Find highest probability for each word:

eat:
P(topic **A** | sentence 1) P(eat | topic **A**) =
P(topic **B** | sentence 1) P(eat| topic **B**) =
broccoli:
P(topic **A** | sentence 1) P(broccoli | topic **A**) =
P(topic **B** | sentence 1) P(broccoli| topic **B**) =
banana:
P(topic **A** | sentence 1) P(banana | topic **A**) =
P(topic **B** | sentence 1) P(banana| topic **B**) =

	eat	broccoli	banana	chinchilla	kitten	cute	hamster	munch
Sentence 1	A	B	A					
Sentence 2				B	A	B		
Sentence 3		A				A	B	B

LDA: Breakout

1. Calculate probabilities:

Num w in A = 5
Num w in B = 5
P(topic **A** | sentence 1) = 2/3
P(topic **B** | sentence 1) = 1/3
P(eat | topic **A**) = 1/5
P(eat| topic **B**) = 0/5
P(broccoli | topic **A**) = 1/5
P(broccoli | topic **B**) = 1/5
P(banana | topic **A**) = 1/5
P(banana| topic **B**) = 0/5

2. Find highest probability for each word:

eat:
P(topic **A** | sentence 1) P(eat | topic **A**) = ⅔*⅕= **2/15**
P(topic **B** | sentence 1) P(eat| topic **B**) = ⅓*0/5=0
broccoli:
P(topic **A** | sentence 1) P(broccoli | topic **A**) = ⅔*⅕ = **2/15**
P(topic **B** | sentence 1) P(broccoli| topic **B**) = ⅓*⅕ = 1/15
banana:
P(topic **A** | sentence 1) P(banana | topic **A**) = ⅔*⅕ = **2/15**
P(topic **B** | sentence 1) P(banana| topic **B**) = ⅓*0/5 = 0

	eat	broccoli	banana	chinchilla	kitten	cute	hamster	munch
Sentence 1	A	B	A					
Sentence 2				B	A	B		
Sentence 3		A				A	B	B

LDA: Breakout

1. Calculate probabilities:

Num w in A = 5
Num w in B = 5
P(topic **A** | sentence 1) = 2/3
P(topic **B** | sentence 1) = 1/3
P(eat | topic **A**) = 1/5
P(eat| topic **B**) = 0/5
P(broccoli | topic **A**) = 1/5
P(broccoli | topic **B**) = 1/5
P(banana | topic **A**) = 1/5
P(banana| topic **B**) = 0/5

2. Find highest probability for each word:

eat:
P(topic **A** | sentence 1) P(eat | topic **A**) = ⅔*⅕= **2/15**
P(topic **B** | sentence 1) P(eat| topic **B**) = ⅓*0/5=0
broccoli:
P(topic **A** | sentence 1) P(broccoli | topic **A**) = ⅔*⅕ = **2/15**
P(topic **B** | sentence 1) P(broccoli| topic **B**) = ⅓*⅕ = 1/15
banana:
P(topic **A** | sentence 1) P(banana | topic **A**) = ⅔*⅕ = **2/15**
P(topic **B** | sentence 1) P(banana| topic **B**) = ⅓*0/5 = 0

	eat	broccoli	banana	chinchilla	kitten	cute	hamster	munch
Sentence 1	A	A	A					
Sentence 2				B	A	B		
Sentence 3		A				A	B	B

Reassign

LDA compared to NMF

NMF	LDA
Relies on linear algebra - matrix factorization	Based on probabilistic graphical modeling - uses a Dirichlet prior on top of the data generating process
Takes TF-IDF matrix as input	Takes bag of words (term frequency) matrix as input
Produces W and H to reproduce bag-of-words matrix with lowest error	Produces θ and ϕ matrices to reproduce bag-of-words matrix with lowest error

LDA in sklearn and gensim

`lda.ipynb`

Objectives

- Review topic modeling
- Review NMF
- State what LDA stands for
  - Pronounce Dirichlet, backup
- Describe what the Dirichlet distribution is
- Describe how LDA does topic modeling
  - Explain what a generative, probabilistic model is
- Perform hand calculations to understand how the *topic-document* and *word-topic* distributions are created.
- Compare/contrast LDA and NMF
- Perform LDA in sklearn and gensim