Latent Dirichlet Allocation (LDA)

ANDRÉ NGUYEN

Goal

Suppose you have two million text documents in a database. Your manager informs you that he needs the contents of the entire database described and thematically labeled for a meeting that is happening in three hours. Additionally, your manager is numerically illiterate so you cannot include any numbers in your report.

How would you approach this task?

Multinomial Distribution

Suppose you have N independent trials, each of which leads to a success for exactly one of K categories. Each category, i in {1, 2, … K} has a given, fixed probability of success p_i.

The multinomial distribution models the probability of observing any particular outcome in this scenario.

For example:

Suppose you have a 6 sided dice. The dice is biased so that you never roll a 1, 2, 3, or 4. The dice will land on the 5 about 60% of the time. You roll the dice 4 times.

What are the parameters (values of N, K, p_1 … p_K) of the multinomial distribution modeling this experiment?

Multinomial Distribution

For example:

Suppose you have a 6 sided dice. The dice is biased so that you never roll a 1, 2, 3, or 4. The dice will land on the 5 about 60% of the time. You roll the dice 4 times.

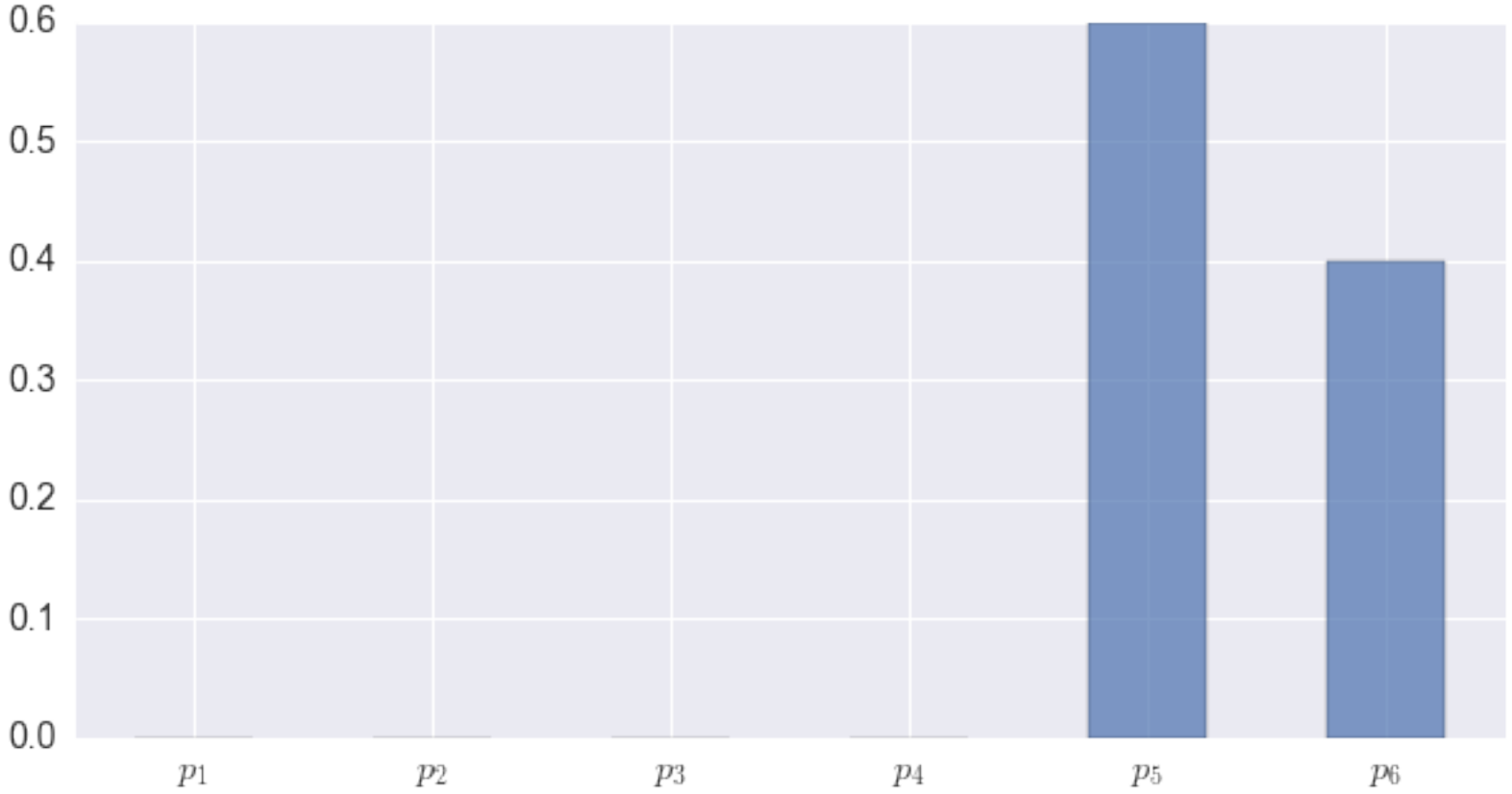
What are the parameters (values of N, K, $p_1 \ldots p_K$) of the multinomial distribution modeling this experiment?

$N = 4$, $K = 6$,

$[\ p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5\ \ p_6\ ] = [\ 0.0, 0.0, 0.0, 0.0, 0.6, 0.4\ ]$

Multinomial Distribution

$[\ p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5\ \ p_6\ ] = [\ 0.0, 0.0, 0.0, 0.0, 0.6, 0.4\ ]$



Multinomial Distribution

$N = 4,\ K = 6,$

$[\ p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5\ \ p_6\ ] = [\ 0.0, 0.0, 0.0, 0.0, 0.6, 0.4\ ]$

Given this model, what is the probability of observing the following outcome? {5, 6, 2, 5}

Multinomial Distribution

$N = 4,\ K = 6,$

$[\ p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5\ \ p_6\ ] = [\ 0.0, 0.0, 0.0, 0.0, 0.6, 0.4\ ]$

Given this model, what is the probability of observing the following outcome? {5, 6, 2, 5}

The probability of this outcome is 0 because the probability of observing a 2 under this model is 0.

Multinomial Distribution

$N = 4,\ K = 6,$

$[\ p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5\ \ p_6\ ] = [\ 0.0, 0.0, 0.0, 0.0, 0.6, 0.4\ ]$

Given this model, what is the probability of observing the following outcome? {5, 6, 2, 5}

The probability of this outcome is 0 because the probability of observing a 2 under this model is 0.

Suppose you now set $N = 1000$. You run the experiment once. (In other words, you sample 1000 times from a multinomial distribution parametrized by [0.0, 0.0, 0.0, 0.0, 0.6, 0.4].) How many successes for each category do you expect?

Multinomial Distribution

$N = 4,\ K = 6,$

$[\ p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5\ \ p_6\ ] = [\ 0.0, 0.0, 0.0, 0.0, 0.6, 0.4\ ]$

Given this model, what is the probability of observing the following outcome? {5, 6, 2, 5}

The probability of this outcome is 0 because the probability of observing a 2 under this model is 0.

Suppose you now set $N = 1000$. You run the experiment once. (In other words, you sample 1000 times from a multinomial distribution parametrized by [0.0, 0.0, 0.0, 0.0, 0.6, 0.4].) How many successes for each category do you expect?

You expect to roll 600 fives and 400 sixes.

Tokenization of Text Data

Given a sequence of symbols, tokenization chops the sequence up into pieces called tokens. The set of all tokens is called the vocabulary.

Suppose a vocabulary, V = {“cat”, “dog”, “ate”, “food”}.

How would you tokenize the following? “The cat ate food. The dog ate the cat.”

Tokenization of Text Data

Given a sequence of symbols, tokenization chops the sequence up into pieces called tokens. The set of all tokens is called the vocabulary.

Suppose a vocabulary, *V* = {“cat”, “dog”, “ate”, “food”}.

How would you tokenize the following? “The cat ate food. The dog ate the cat.”

[“cat”, “ate”, “food”, “dog”, “ate”, “cat”]

Note that the word “the” was ignored in the tokenization because it is not part of the vocabulary. Words such as “the”, “and”, “about”, “having”… are often ignored on purpose because they do not convey much meaning. Commonly ignored words are called stop words.

Also note that tokens do not have to be single words. For example, the vocabulary can be a set of characters *V* = {“a”, “b”, “c”, “d”} or can additionally contain pairs of words (bigrams) such that *V* = {“cat”, “dog”, “ate”, “cat ate”, “dog ate”}.

Bag-of-Words

Bag-of-words is a representation model often used in natural language processing for simplifying text data. Bag-of-words ignores token order but keeps token duplicates.

Under the bag-of-words model, the following are equivalent:

[“the”, “cat”, ”ate”, “the”, “dog”]

[“the”, “dog”, ”ate”, “the”, “cat”]

[“the”, “the”, ”dog”, “cat”, “ate”]

The equivalence of “the cat ate the dog” and “the dog ate the cat” under the bag-of-words model is an example of why it might be sometimes useful to use n-grams in addition to single words in the vocabulary. Using bigrams, the two sentences are no longer equivalent:

[”cat ate”, “ate dog”]

[“dog ate”, “ate cat”]

Count Vectorization

Given a bag-of-words representation of a text document, a numerical representation of the document can be computed by counting the number of occurrences of vocabulary tokens in the document.

Given a vocabulary V = {“a”, “b”, “c”, “d”, “e”} and five documents:

d_1 = “aabaac”, d_2 = “daaae”, d_3 = “aaaaaaaa”, d_4 = “bcdebcde”, d_5 = “aab”

Bag-of-words and count vectorization would give you the following representation of the data:

	a	b	c	d	e
d_1	4	1	1	0	0
d_2	3	0	0	1	1
d_3	8	0	0	0	0
d_4	0	2	2	2	2
d_5	2	1	0	0	0

Generative Bayesian Modeling

In generative Bayesian modeling, the objective is to best model the underlying process that generates the data. In other words, we assume that each observed data point is the result of sampling from an unknown distribution. We want to infer the underlying distribution(s).

For example, suppose we have the following vocabulary: V = {“a”, “b”, “c”, “d”, “e”}

And the underlying multinomial distribution over the vocabulary is:

[p_a p_b p_c p_d p_e] = [0.0, 0.25, 0.25, 0.25, 0.25]

Generating a text document of length N would be equivalent to rolling a five sided dice with the above bias N times and recording the outcome. The dice would land on sides “b”, “c”, “d”, “e” about a quarter of the time each.

Generation of a Document

Let's generate a document of length $N = 5$ from $[\ p_a\ \ p_b\ \ p_c\ \ p_d\ \ p_e\ ] = [\ 0.0, 0.25, 0.25, 0.25, 0.25\ ]$

Roll 1: "e"

Roll 2: "b"

Roll 3: "c"

Roll 4: "c"

Roll 5: "e"

The resulting document is "ebcce".

Inference of the Underlying Distribution

Usually in data science, we perform the reverse of the generation procedure. We observe a set of documents and we try to infer the underlying distribution of the generation process.

Suppose we have a vocabulary V = {“a”, “b”, “c”}

We observe three documents: “aaab”, “abaa”, “aaabaaab”

What is a reasonable guess for the underlying distribution [p_a p_b p_c]? Should we set $p_c = 0$?

Inference of the Underlying Distribution

Usually in data science, we perform the reverse of the generation procedure. We observe a set of documents and we try to infer the underlying distribution of the generation process.

Suppose we have a vocabulary V = {“a”, “b”, “c”}

We observe three documents: “aaab”, “abaa”, “aaabaaab”

What is a reasonable guess for the underlying distribution [p_a p_b p_c]? Should we set $p_c = 0$?

We don’t want to set $p_c = 0$ because we have observed only three documents and not all possible documents (usually assumed to be an infinite number of documents). Our data suggests that p_c should be a small number, but we want to prevent it from being set to zero. In other words, we want to smooth the inferred distribution.

Something like [p_a p_b p_c] = [0.74 0.24 0.02] might be reasonable.

Bayesian statistics does this smoothing elegantly via the use of prior distributions which allow us to combine prior beliefs with observed data. As we observe new data, our beliefs slowly move towards what is suggested by the data, but our prior beliefs do not immediately disappear in the presence of data.

The Dirichlet distribution is the prior distribution of the multinomial. It can be thought of as a generative process from which biased dice can be sampled.

Hierarchical Generative Bayesian Models

Let's complicate the generative process a bit. Suppose we have a corpus of $D = 10$ documents.

We have a vocabulary V = {"cat", "dog", "Boston", "Chicago", "basketball", "soccer"} of size $K = 6$.

We have $T = 3$ dice (which we will call topics) with the following multinomial distributions over the vocabulary:

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.4, 0.4, 0.05, 0.05, 0.05, 0.05]$

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.4, 0.4, 0.05, 0.05]$

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.05, 0.05, 0.4, 0.4]$

Hierarchical Generative Bayesian Models

Let's complicate the generative process a bit. Suppose we have a corpus of $D = 10$ documents.

We have a vocabulary V = {"cat", "dog", "Boston", "Chicago", "basketball", "soccer"} of size $K = 6$.

We have $T = 3$ dice (which we will call topics) with the following multinomial distributions over the vocabulary:

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.4, 0.4, 0.05, 0.05, 0.05, 0.05]$ (animal topic)

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.4, 0.4, 0.05, 0.05]$ (city topic)

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.05, 0.05, 0.4, 0.4]$ (sports topic)

Each document in the corpus will be encoded as a 3 sided document dice encoding a distribution over topics. A document about animals in cities would look something like:

$[\ p_{animal}\ p_{city}\ p_{sports}\ ] = [0.5, 0.4, 0.1]$

Note that all topics share the same words in different proportions, and that all documents share the same topics in different proportions.

Generation of a Document

Let's generate a document of length $N = 5$ from $[\ p_{animal}\ p_{city}\ p_{sports}\ ] = [0.5, 0.4, 0.1]$

With the following topics:

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.4, 0.4, 0.05, 0.05, 0.05, 0.05]$ (animal topic)

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.4, 0.4, 0.05, 0.05]$ (city topic)

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.05, 0.05, 0.4, 0.4]$ (sports topic)

Word 1: Roll document dice: "animal", then roll "animal" topic dice: "dog"

Word 2: Roll document dice: "animal", then roll "animal" topic dice: "soccer"

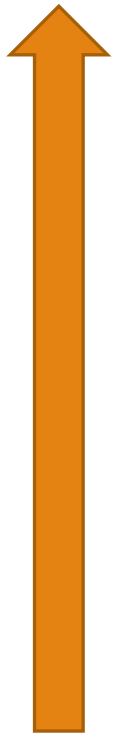
Word 3: Roll document dice: "city", then roll "city" topic dice: "Chicago"

Word 4: Roll document dice: "animal", then roll "animal" topic dice: "dog"

Word 5: Roll document dice: "sports", then roll "sports" topic dice: "cat"

The resulting document is: "dog soccer Chicago dog cat"

Inference



Let's generate a document of length $N = 5$ from $[\ p_{animal}\ p_{city}\ p_{sports}\ ] = [0.5, 0.4, 0.1]$

With the following topics:

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.4, 0.4, 0.05, 0.05, 0.05, 0.05]$ (animal topic)

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.4, 0.4, 0.05, 0.05]$ (city topic)

$[\ p_{cat}\ p_{dog}\ p_{Boston}\ p_{Chicago}\ p_{basketball}\ p_{soccer}\ ] = [0.05, 0.05, 0.05, 0.05, 0.4, 0.4]$ (sports topic)

Word 1: Roll document dice: "animal", then roll "animal" topic dice: "dog"

Word 2: Roll document dice: "animal", then roll "animal" topic dice: "soccer"

Word 3: Roll document dice: "city", then roll "city" topic dice: "Chicago"

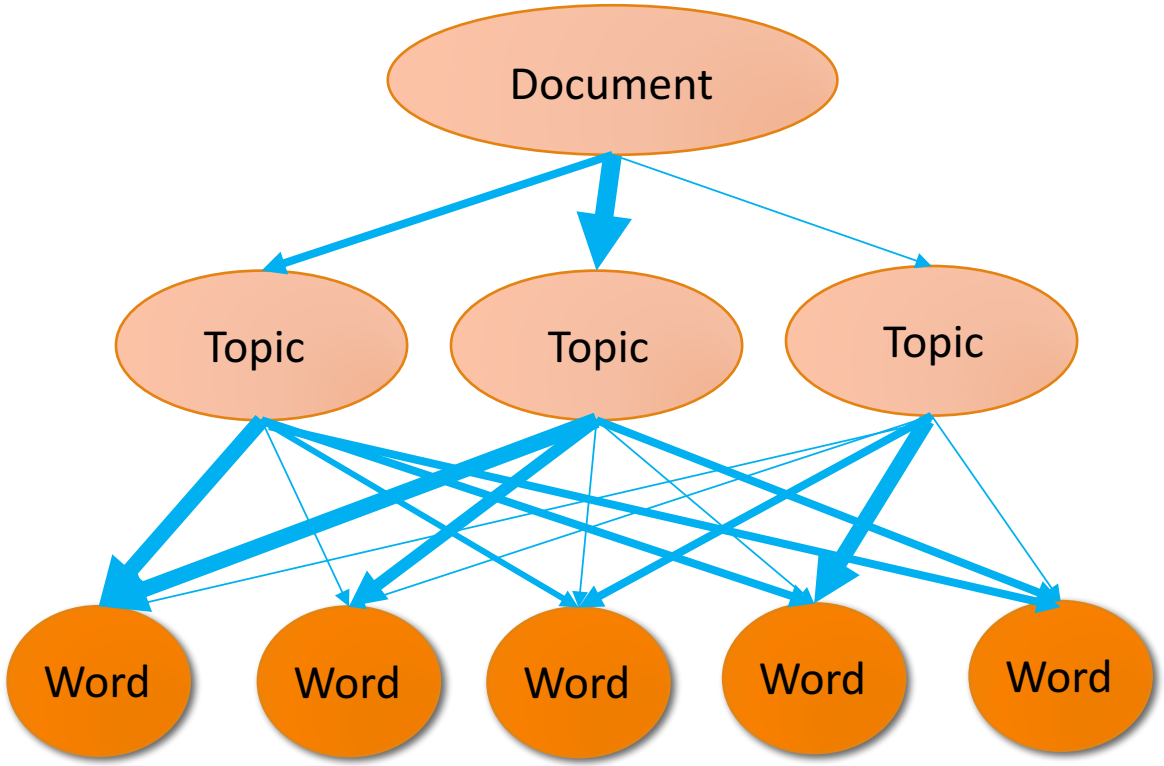
Word 4: Roll document dice: "animal", then roll "animal" topic dice: "dog"

Word 5: Roll document dice: "sports", then roll "sports" topic dice: "cat"

The resulting document is: "dog soccer Chicago dog cat"

This is the generative model assumed by Latent Dirichlet Allocation (LDA). The goal of the algorithm is to reverse engineer and learn the dice biases given the data.

Latent Dirichlet Allocation

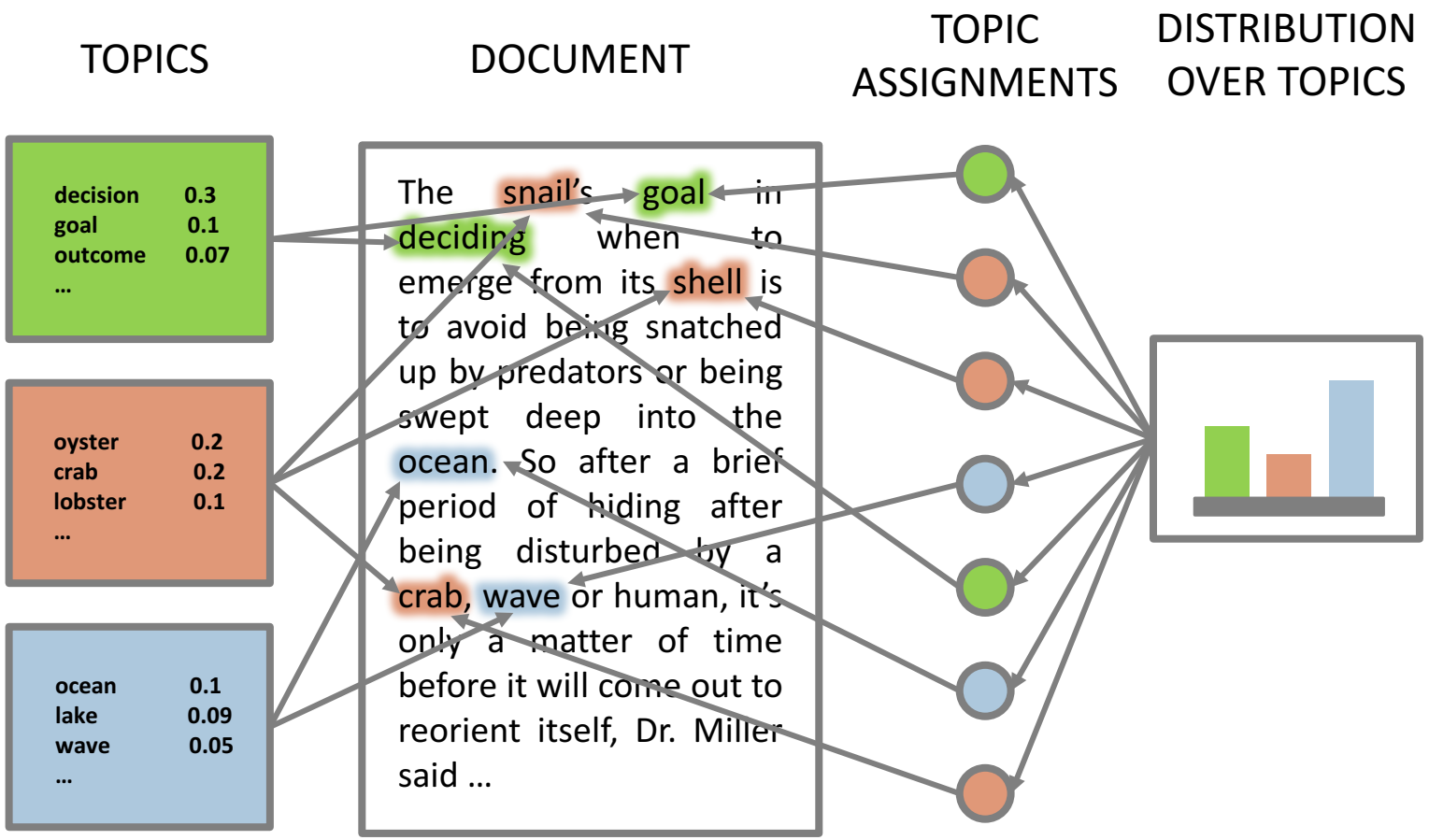


Latent Dirichlet Allocation (LDA) is a three level generative Bayesian hierarchical model of a corpus of documents.

- A corpus is modeled as a set of documents.
- A document is modeled as a multinomial distribution (dice) over topics.
- A topic is modeled as a multinomial distribution (dice) over words.

- LDA can be viewed as a form of data compression where raw text documents are summarized and labeled by their top (most probable) topics. Topics can themselves be summarized by their top (most probable) words.
- The name of the algorithm comes from the fact that only the words of a document are observed, so the document dice and topics dice are unobserved or latent. The document and topic dice are assumed to be drawn from a Dirichlet prior distribution.

LDA Generative Process Example



For each word in the document, a topic assignment is sampled from the distribution over topics.

Then, based on the topic assignment, a word is sampled from the appropriate topic's distribution over words.

A corpus is created by sampling multiple documents in this fashion, each document having its own distribution over topics.

LDA is used to infer the Topics and the Distribution over Topics for each document. LDA is fully unsupervised.

Goal and Solution

Suppose you have two million text documents in a database. Your manager informs you that he needs the contents of the entire database described and thematically labeled for a meeting that is happening in three hours. Additionally, your manager is numerically illiterate so you cannot include any numbers in your report.

Minimal effort solution:

Use LDA to extract thematic topics and use these topics to label all of the documents in the database.

1) Transform all of the documents using count vectorization (tokenization and bag-of-words).

2) Input the transformed documents to LDA. Run LDA.

3) LDA returns topics as weighted mixtures of words and returns documents represented as weighted mixtures of topics.

4) Use top words in each topic to describe the topic. Use top topics in each document to label the documents.

Papers and Extensions

Easy paper about LDA:
- https://www.cs.princeton.edu/~blei/papers/Blei2012.pdf

More technical paper about LDA, including all of the math for the inference procedure:
- https://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf

Extension of LDA where topics evolve over time:
- https://mimno.infosci.cornell.edu/info6150/readings/dynamic_topic_models.pdf

Extension of LDA where the number of topics is inferred by the model:
- http://people.eecs.berkeley.edu/~jordan/papers/hdp.pdf

LDA applied in computer vision:
- http://groups.csail.mit.edu/vision/app/pubs/wangEnips07.pdf

Top topic model researcher David Blei's website:
- http://www.cs.columbia.edu/~blei/