# Latent Dirichlet Allocation Overview

Kanru Wang

March 2020

# Usage and Main Idea

- Latent Dirichlet Allocation (LDA) is the one of the most popular algorithms for document clustering.
- E.g.
  - News article clustering, which facilitates recommendation
  - Large corpus topic discovery

# Usage and Main Idea

- Latent Dirichlet Allocation (LDA) is the one of the most popular algorithms for document clustering.
- E.g.
  - News article clustering, which facilitates recommendation
  - Large corpus topic discovery
- Main idea:
  - Assume that each document, if treated as bag-of-words, can be generated by a sampling process

# Usage and Main Idea

- Latent Dirichlet Allocation (LDA) is the one of the most popular algorithms for document clustering.
- E.g.
  - News article clustering, which facilitates recommendation
  - Large corpus topic discovery
- Main idea:
  - Assume that each document, if treated as bag-of-words, can be generated by a sampling process
  - Each topic is just a specific distribution of words.

    (e.g. a Machine Learning topic may have: "model" 132, "rate" 118, …, "cat" 3, "song" 1, "football" 0)
  - Each document has a specific distribution of topics.

# Usage and Main Idea

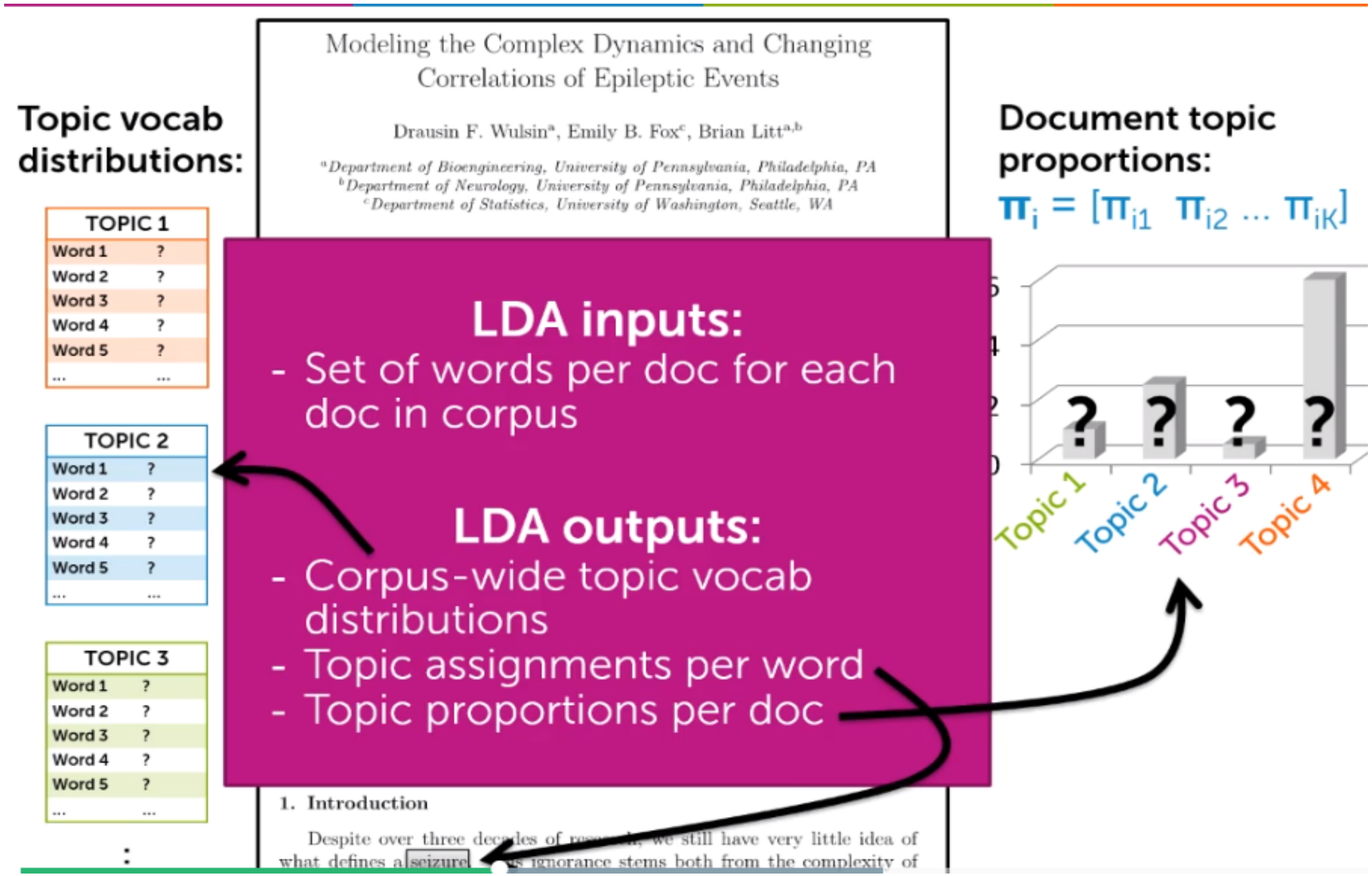
- Latent Dirichlet Allocation (LDA) is the one of the most popular algorithms for document clustering.
- E.g.
  - News article clustering, which facilitates recommendation
  - Large corpus topic discovery
- Main idea:
  - Assume that each document, if treated as bag-of-words, can be generated by a sampling process
  - Each topic is just a specific distribution of words.

    (e.g. a Machine Learning topic may have: “model” 132, “rate” 118, …, “cat” 3, “song” 1, “football” 0)
  - Each document has a specific distribution of topics.
  - Each document is generated this way:

    For each word-to-fill in the document (until the whole document is generated):

    1. Based on the document’s distribution of topics, select a topic at random
    2. Based on the chosen topic’s distribution of words, select a word at random
  - Document -> (hidden) topics -> words

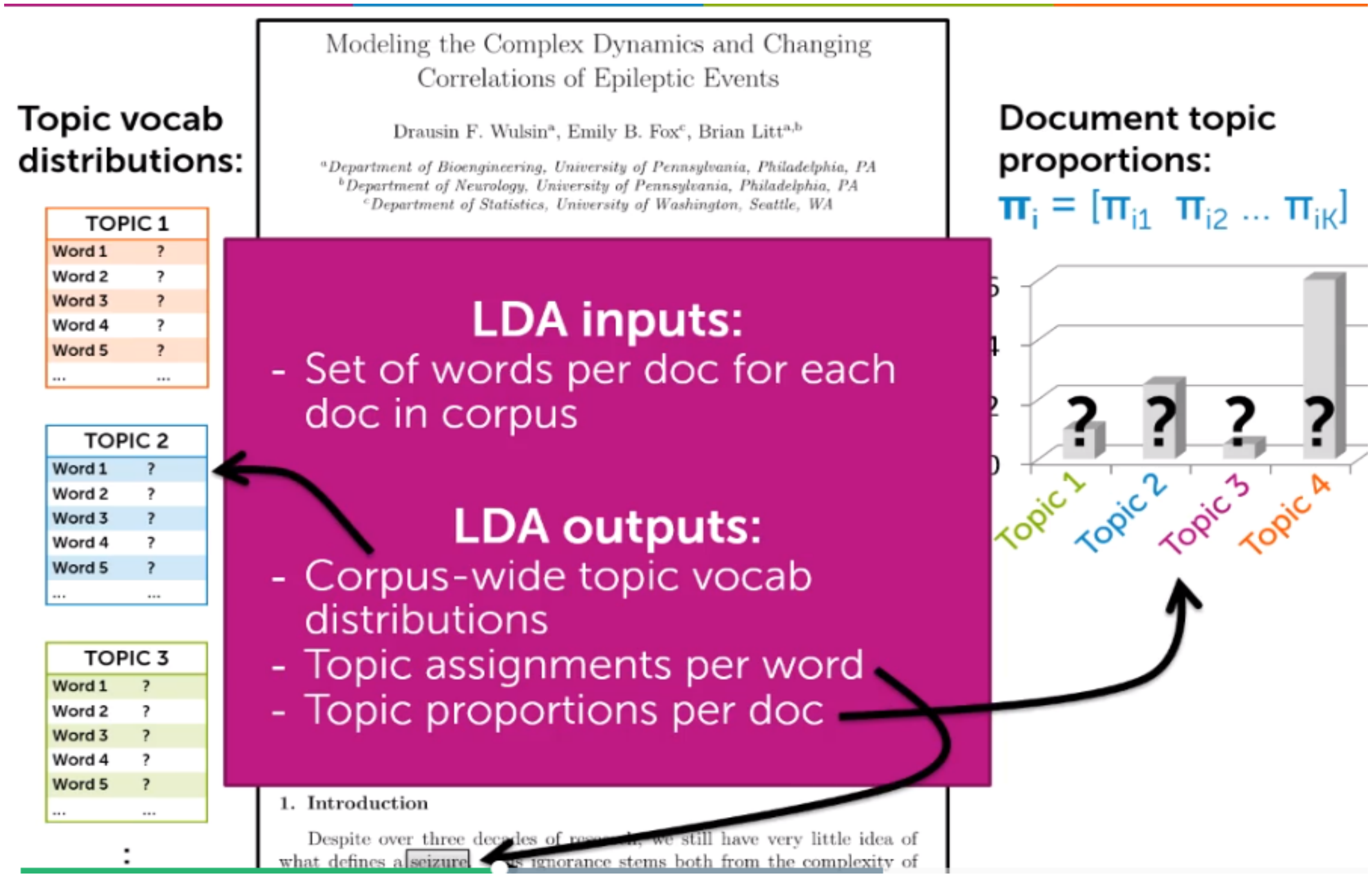
# Three useful outputs and how to get them




https://www.coursera.org/learn/ml-clustering-and-retrieval

# Three useful outputs and how to get them

In order to generate a document that has the same bag-of-words representation as the actual document's, we need to optimize the weights in
Topic Vocab Distributions
and
Document Topic Proportions



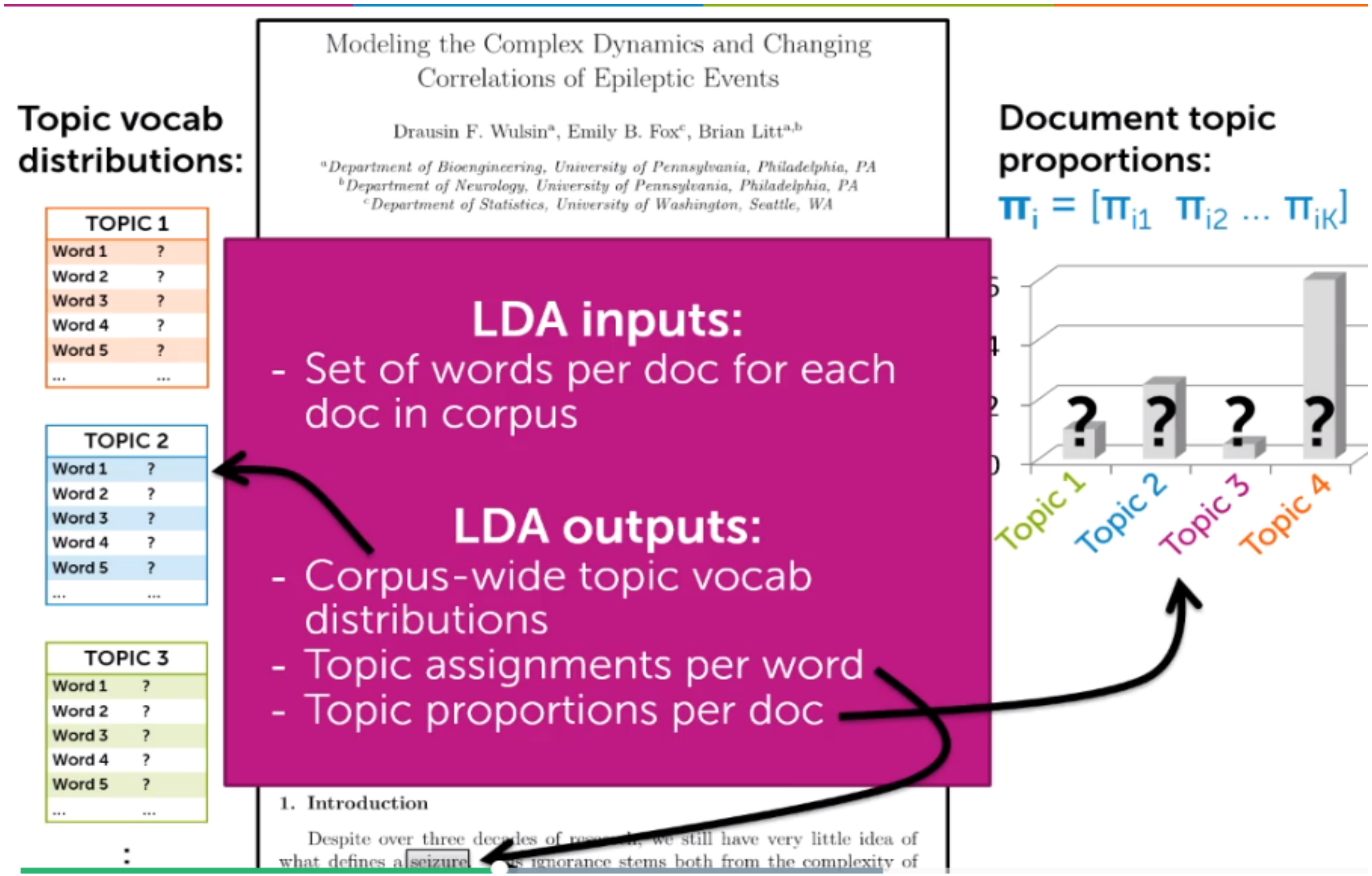

https://www.coursera.org/learn/ml-clustering-and-retrieval

# Three useful outputs and how to get them

In order to generate a document that has the same bag-of-words representation as the actual document's, we need to optimize the weights in
Topic Vocab Distributions
and
Document Topic Proportions

There are 2 ways to optimize them:

1. Collapsed Gibbs Sampling
2. Variational Bayesian Inference (won't discuss)




https://www.coursera.org/learn/ml-clustering-and-retrieval

# Collapsed Gibbs Sampling

## Maintain global statistics

| 3 | 2 | 1 | 3 | 1 |
|---|---|---|---|---|
| epilepsy | dynamic | Bayesian | EEG | model |

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| Doc i | 2 | 1 | 2 |

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| epilepsy | 1 | 0 | 35 |
| Bayesian | 50 | 0 | 1 |
| model | 42 | 1 | 0 |
| EEG | 0 | 0 | 20 |
| dynamic | 10 | 8 | 1 |
| ... | | | |

Total counts from **all** docs

- For example we have a document that has just five words, "Epilepsy dynamic Bayesian EEG model".
- The five numbers in the top table are topic indicators.
- At the start of the entire algorithm, the topic assignment of each word is random.

https://www.coursera.org/learn/ml-clustering-and-retrieval

# Collapsed Gibbs Sampling

## Randomly reassign topics

| | | | | |
|---|---|---|---|---|
| 3 | ~~2~~ | 1 | 3 | 1 |
| epilepsy | dynamic | Bayesian | EEG | model |

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| epilepsy | 1 | 0 | 35 |
| Bayesian | 50 | 0 | 1 |
| model | 42 | 1 | 0 |
| EEG | 0 | 0 | 20 |
| dynamic | 10 | 7 ~~8~~ | 1 |
| ... | | | |

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| Doc i | 2 | 0 ~~1~~ | 2 |

decrementing counts after removing current assignment $z_{iw}=2$

- Need to reassign every word, for each document, for many iterations
- Let's start from reassigning the word "dynamic".

# Collapsed Gibbs Sampling

## Probability of new assignment

| 3 | ? | 1 | 3 | 1 |
|---|---|---|---|---|
| epilepsy | dynamic | Bayesian | EEG | model |


How much doc "likes" each topic based on other assignments in doc

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| Doc i | 2 | 0 | 2 |

$$\frac{n_{ik} + \alpha}{N_i - 1 + K\alpha}$$

\# current assignments to topic k in doc i

\# words in doc i

smoothing param from Bayes prior

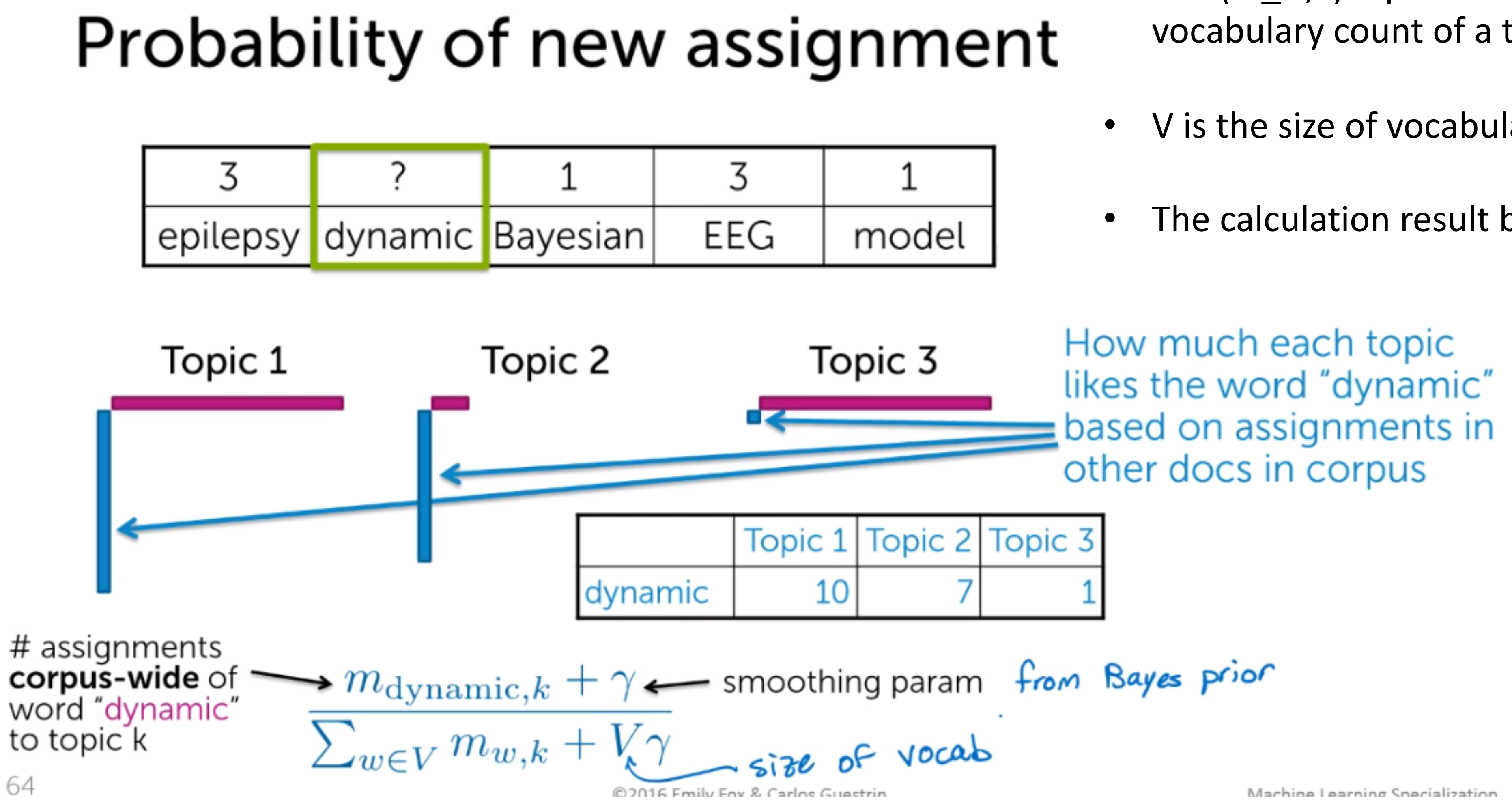
ignore current word



- K is number of topics.
- The calculation result becomes the bar length.

https://www.coursera.org/learn/ml-clustering-and-retrieval

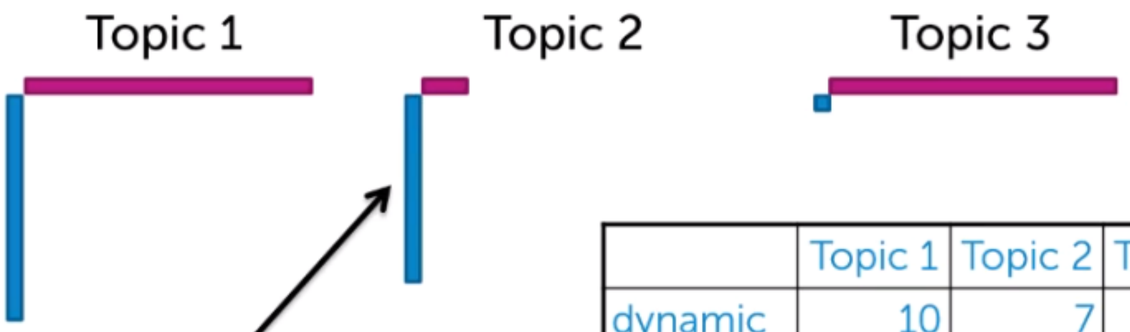
# Collapsed Gibbs Sampling




- sum(m_w,k) equals to the sum of all vocabulary count of a topic.
- V is the size of vocabulary.
- The calculation result becomes the bar length.

https://www.coursera.org/learn/ml-clustering-and-retrieval

# Collapsed Gibbs Sampling




| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| dynamic | 10 | 7 | 1 |

Topic 2 also really likes "dynamic",
but in a different context...
e.g., a topic on fluid dynamics


Topic fits word **and** document

Topic fits word, but not doc

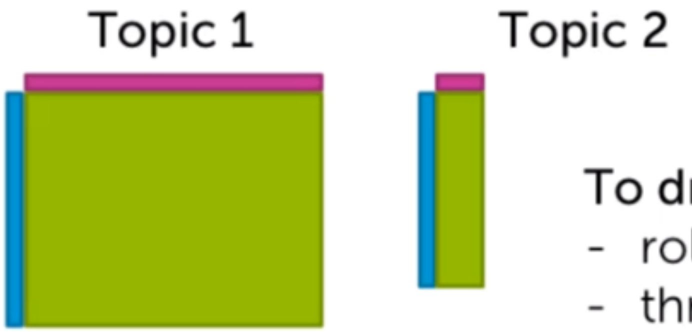
Topic fits doc, but not word

https://www.coursera.org/learn/ml-clustering-and-retrieval

# Collapsed Gibbs Sampling

## Randomly draw a new topic indicator

| | | | | |
|---|---|---|---|---|
| 3 | ? | 1 | 3 | 1 |
| epilepsy | dynamic | Bayesian | EEG | model |





**To draw new topic assignment** (equivalently):

- roll K-sided die with these probabilities
- throw dart at these regions

Normalize this product of terms over K possible topics!

How much doc likes topic

$$\boxed{\frac{n_{ik}+\alpha}{N_i-1+K\alpha}\;\frac{m_{\text{dynamic},k}+\gamma}{\sum_{w\in V} m_{w,k}+V\gamma}}$$

How much topic likes word

- Two terms are multiplied together to get the area (probability).

https://www.coursera.org/learn/ml-clustering-and-retrieval

# Collapsed Gibbs Sampling

## Update counts

| | | | | |
|---|---|---|---|---|
| 3 | 1 | 1 | 3 | 1 |
| epilepsy | dynamic | Bayesian | EEG | model |

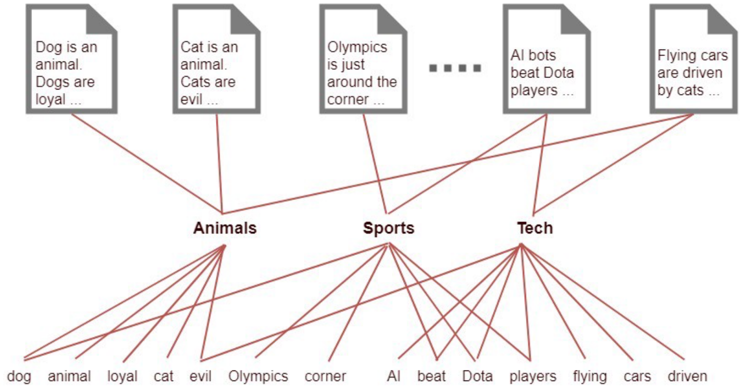
| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| epilepsy | 1 | 0 | 35 |
| Bayesian | 50 | 0 | 1 |
| model | 42 | 1 | 0 |
| EEG | 0 | 0 | 20 |
| dynamic | 11 ~~10~~ | 7 | 1 |
| ... | | | |

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| Doc i | 3 ~~2~~ | 0 | 2 |

increment counts
based on new
assignment of
$z_{iw}=1$

- After updating the topic assignment of the word “dynamic”, go to the next word “Bayesian” in this five-word document “Epilepsy dynamic Bayesian EEG model”.
- After we finish this doc, go to the next doc.
- After we finish the corpus, go through the corpus again.

# So why called Latent Dirichlet Allocation?




- Latent

The topics are hidden.

- Dirichlet

E.g. A machine can produce different dice with different biased weights, and each dice itself is a distribution as we get multiple values when we roll a dice. This is what it means to be a distribution of distributions and this is what Dirichlet is.
Here, in the context of topic modelling, the Dirichlet is the
[first step] distribution of topics in documents, and
[second step] distribution of words in the topic.
(Notice that this is the document generating process, which is different from the Gibbs Sampling.)

- Allocation

We allocate topics to the documents, and words (of the document) to topics.

https://medium.com/analytics-vidhya/topic-modeling-using-lda-and-gibbs-sampling-explained-49d49b3d1045