PPOL 628: Text as Data – Computational Linguistics for Social Scientists

Class 10: Topic Models

Today

- Lecture: an overview of topic models.
  - Generative process and inference.
  - Moving discussion of evaluation methods to next week.

- Lab: topic_models.R

- Website: github.com/matthewjdenny/PPOL_628_Text_As_Data

- Most slides from: https://www.cs.cmu.edu/~mgormley/courses/10701-f16/slides/lecture20-topic-models.pdf

Topic Models

- Class of models for learning about the topical content of a corpus.

- “Topic Model” can actually refer to any of of hundreds of different machine learning models.
  - We will focus on the most popular in this class of models – Latent Dirichlet Allocation.

- Having a basic understanding of the math and its relation to model performance is useful.

- Researcher degrees of freedom (preprocessing, hyper-parameters) can have a huge impact on performance.

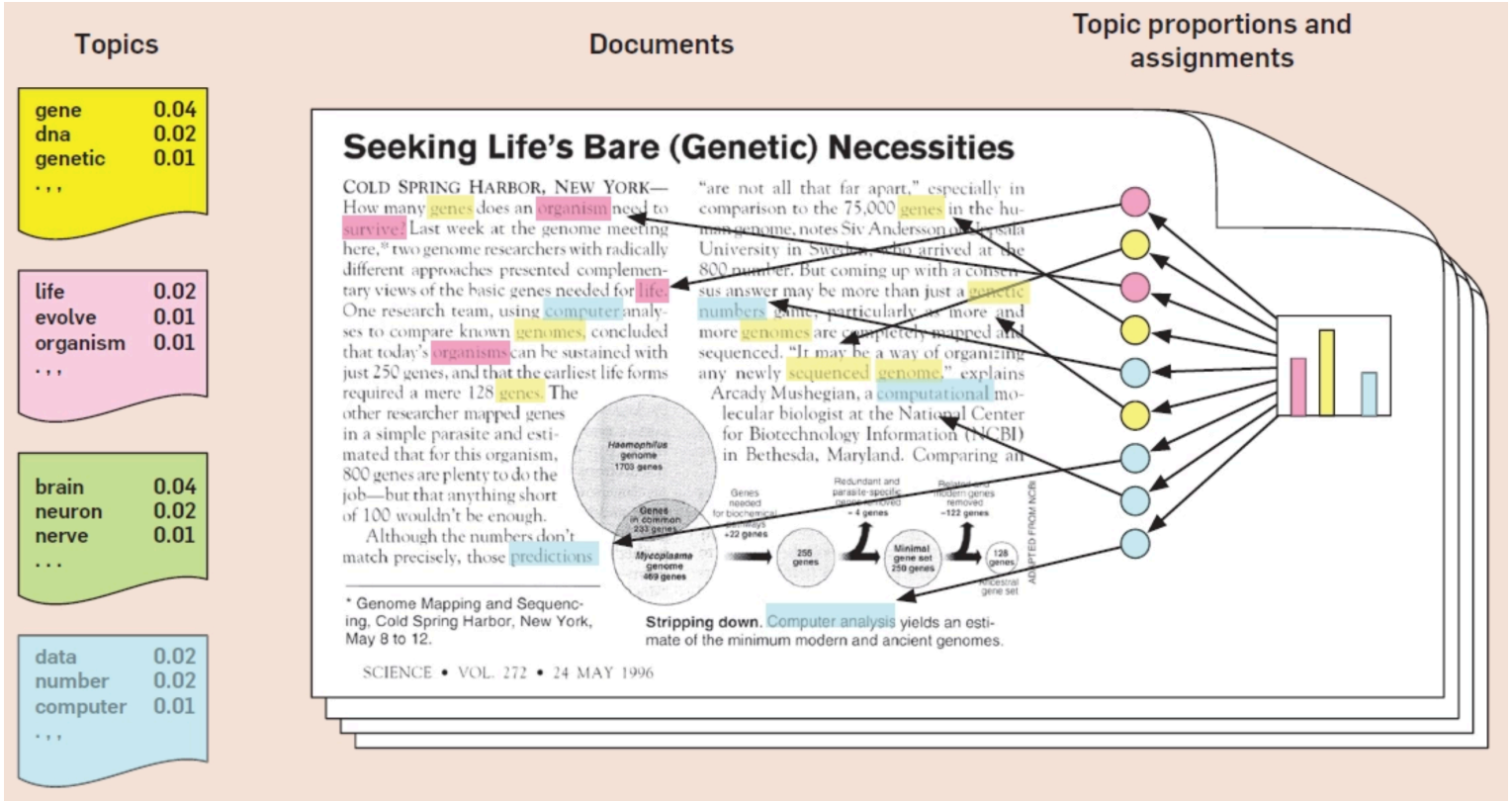
Topic Models -- Motivation

- You have a large number of documents (more than you can possibly read), and want to:
  - Sort these documents into thematic or topical categories.
  - Understand if different authors or groups of authors focus on different things.
  - Understand how themes, topics change over time, or with other covariates.

- We could use dictionaries, vanilla clustering algorithms, supervised learning, but each has problems:
  - **Dictionaries/Supervised Learning**: hard to know ahead of time what all of the topics are, and all relevant words for all topics. Words can be relevant to more than one topic. Documents can be about more than one topic.
  - **Vanilla Clustering**: documents can be about more than one topic, may want to discover topics, learn what important words in those topics are.

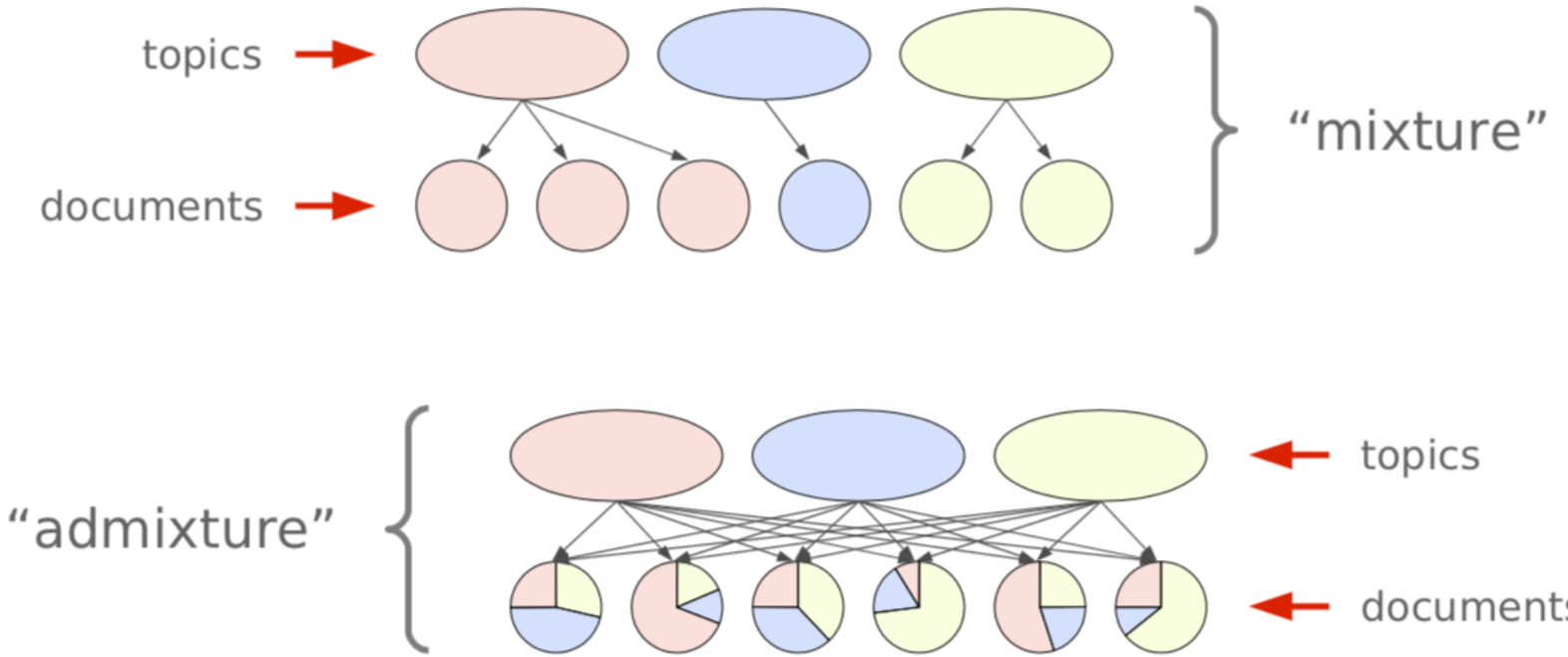
Topic Models -- Goals

- **Learn what topics each document is about:**
  - Learn the **proportion** of each topic in each document.
  - Document-topic proportions as a document covariate.
- **Learn what words best represent each topic**:
  - Learn the words with highest probability in each topic.
  - Display results visually to enable easy human interpretation.



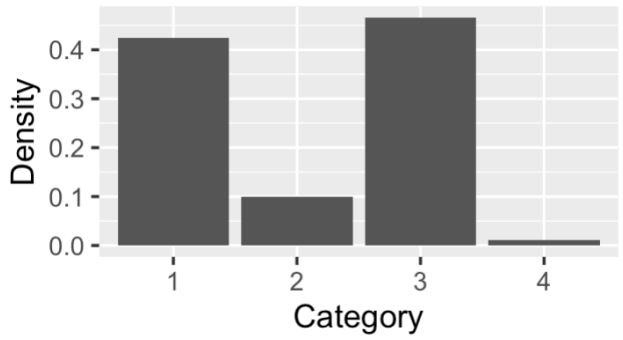
Mixture vs. Admixture Models

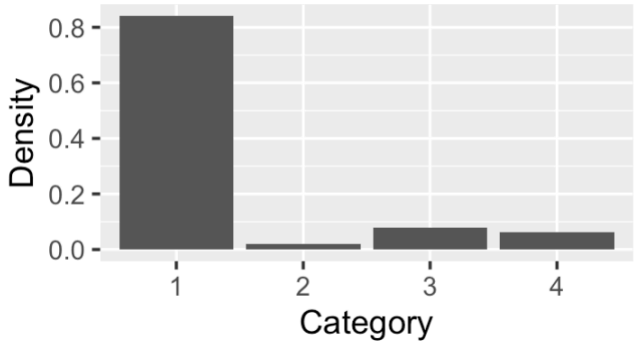
- **Mixture Model:** Each document is about one topic, common words can be important in multiple topics.
- **Admixture Model**: Each document is about multiple topics in different proportions. Words are more exclusive to topics.

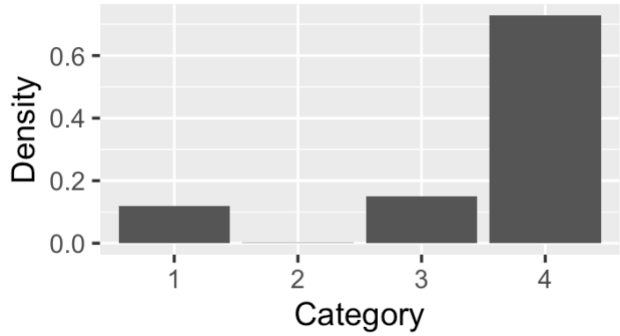


The Dirichlet and Multinomial Distributions

- **Multinomial Distribution:** distribution over categories. Can take samples from distribution which are a category label.
  - M(.5,.25,.25) → {1,2,1,1,3,2,1,2,1,3,1,1,3}
  - Useful for modeling discrete categories like words within a given topic.
- **Dirichlet Distribution**: distribution over multinomial probability vectors. Each sample is a multinomial distribution.
  - Useful for modeling distributions of categories like topics within documents, words within topics.

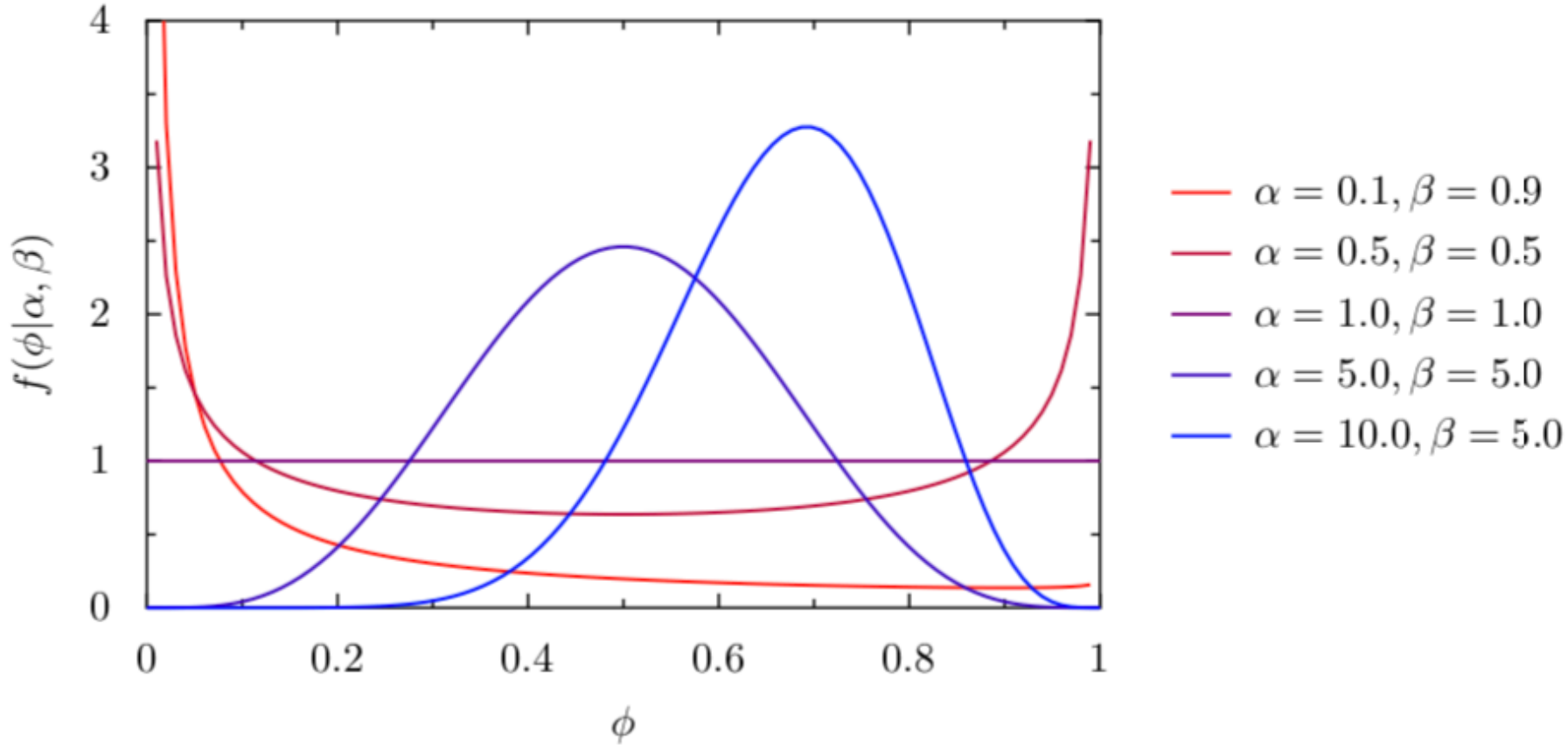






Dirichlet a as Generalization of Beta Dist.

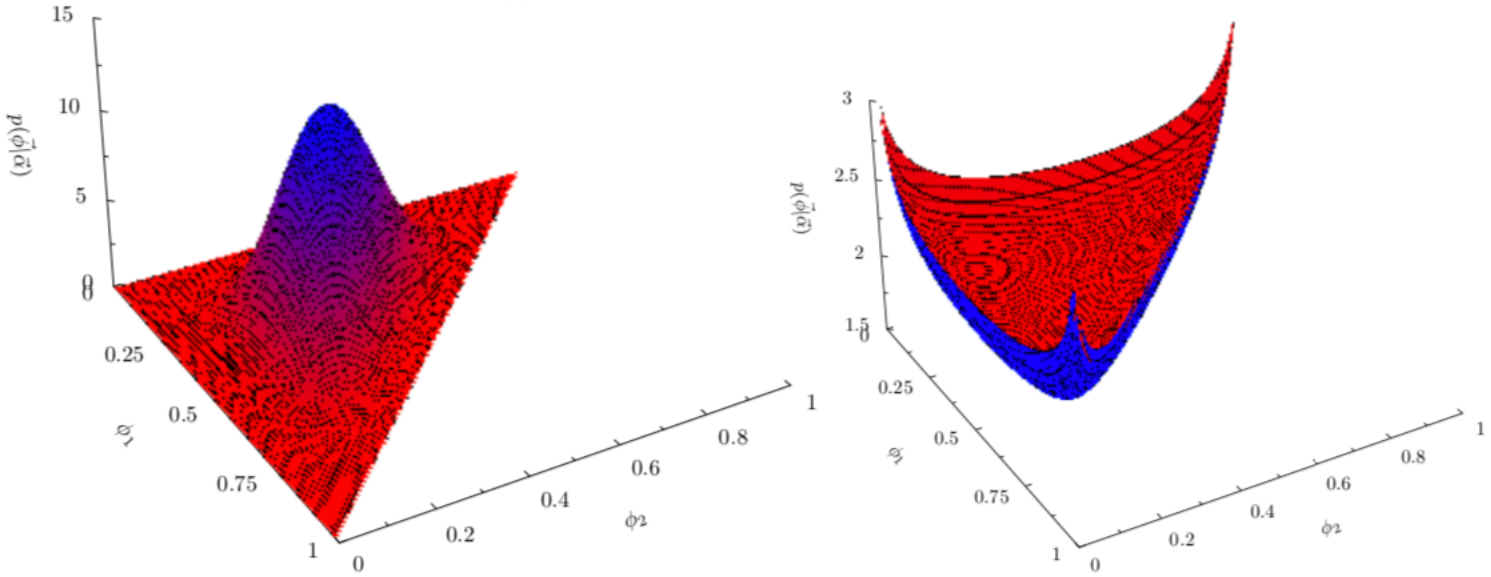
$$f(\phi|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$



Dirichelt has two hyper-parameters $\boldsymbol{\alpha}$ and $\mathbf{m}$

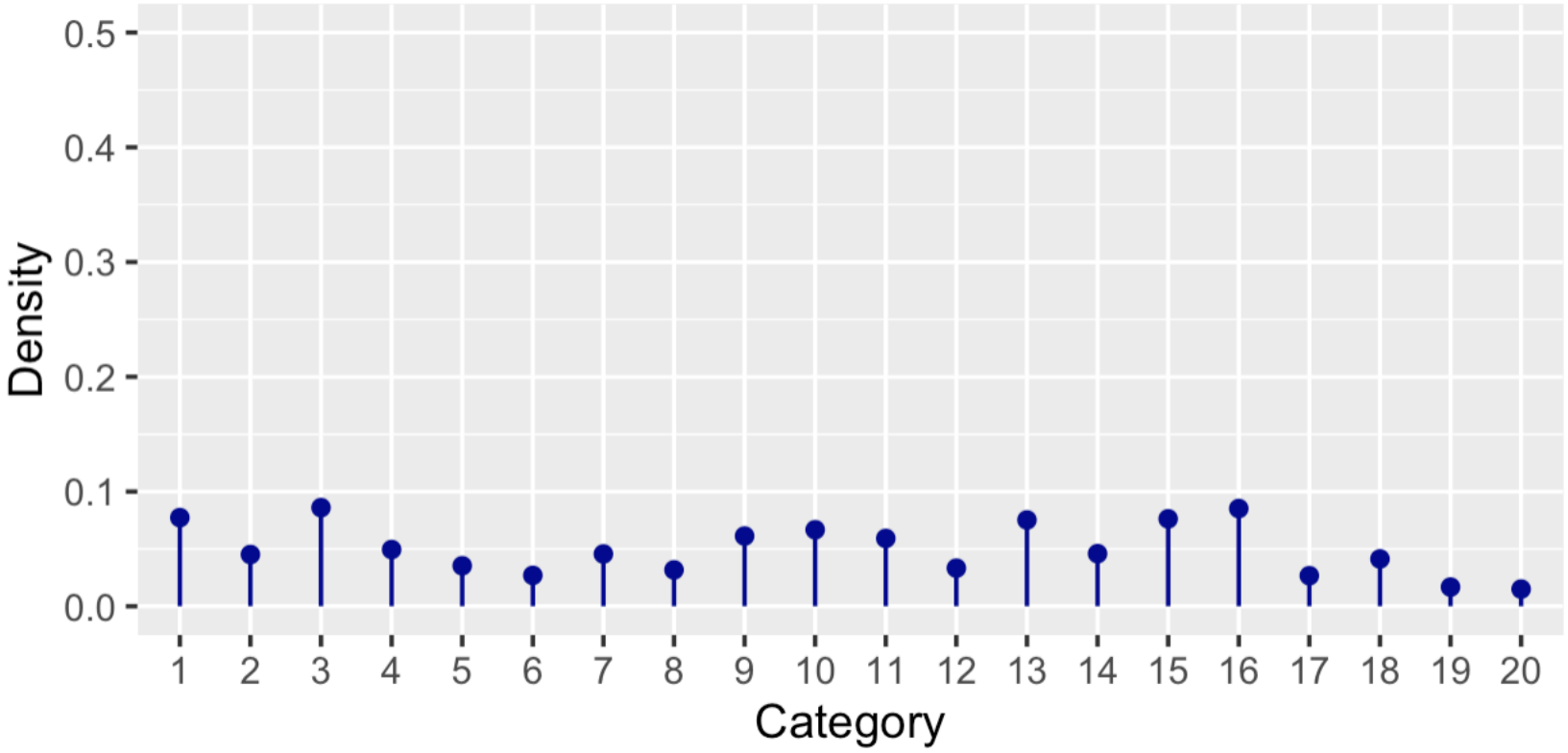
- $\boldsymbol{\alpha}$ the “smoothing parameter” controls how “smooth” distribution is.
- $\mathbf{m}$ the “base measure” (a vector over categories) controls bias towards categories.

$$p(\vec{\phi}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \phi_k^{\alpha_k - 1} \quad \text{where } B(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)}$$



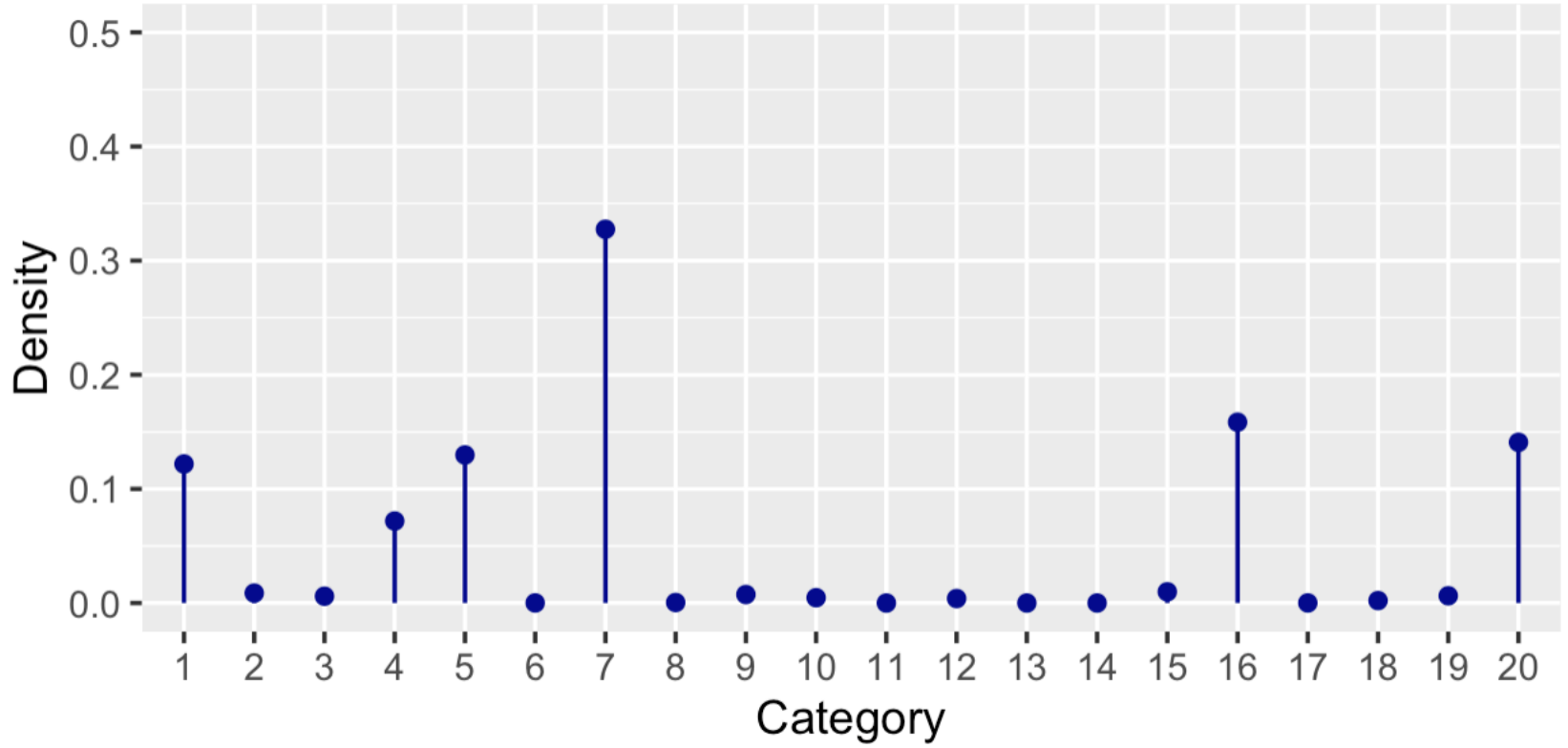
Dirichlet Distribution – Smoothing

- Very large $\boldsymbol{\alpha}$ means draws from Dirichlet are likely to be very similar, with relatively equal probability across categories.



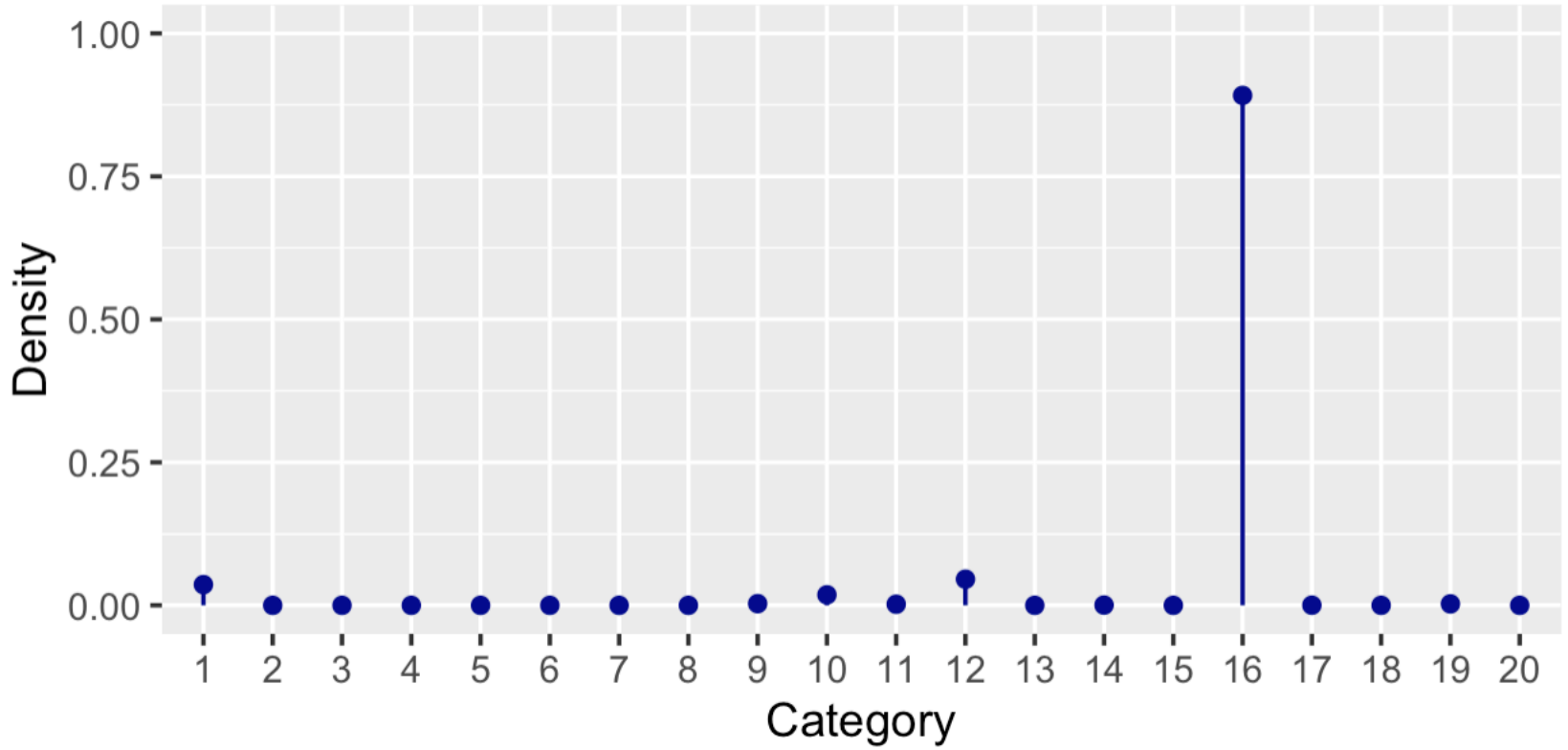
Dirichlet Distribution – Smoothing

- Medium values $\boldsymbol{\alpha}$ means draws from Dirichlet are likely to exhibit more variation. Note that "size" of alpha is dependent on number of categories (grows with number of categories).



Dirichlet Distribution – Smoothing

- Very small $\boldsymbol{\alpha}$ means draws from dirichlet are likely to be very peaked, with almost all probability mass on a single category.



Latent Dirichlet Allocation (LDA)

- The canonical topic model (Blei, Ng, & Jordan, 2003).
- A **generative model**: Posits a generative process for documents as first picking distribution over topics for that document, then picking words from topics.
- Goal of inference is then to "invert" the generative process to recover the document-topic distributions and topic-word distributions.
- Difficult inference problem with many solutions.
  - We will cover most common inference approach: collapsed Gibbs sampling.
- Dozens of extensions to this model.
  - Correlated topics, dynamic (temporal), polylingual, semi-supervised, structural (incorporating covariates -- next week), hierarchical, etc.

Fixing Notation

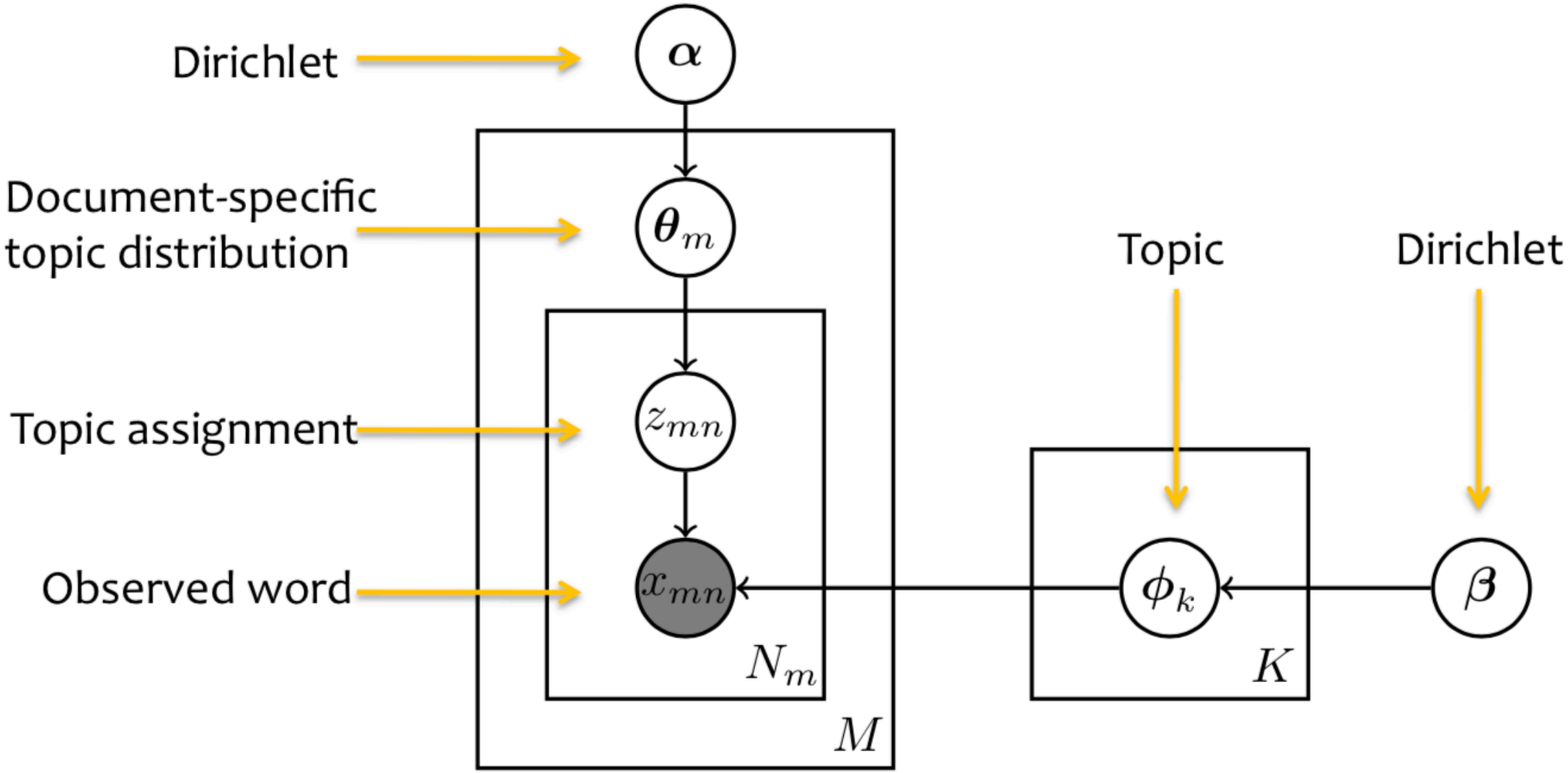
- Let $\boldsymbol{\alpha}$ be the smoothing parameter multiplied by the base measure, controlling the **distribution over topics in documents**.
- Let $\boldsymbol{\beta}$ be the smoothing parameter multiplied by the base measure, controlling the **distribution over words in topics**.
- There are $k \in \{1,\ldots,\mathbf{K}\}$ **topics**, each of which is a distribution over words.
- There are $m \in \{1,\ldots,\mathbf{M}\}$ **documents,** with $n \in \{1,\ldots,\mathbf{N_m}\}$ tokens in each document.
- For token n in document m, its topic assignment is denoted by $z_{m,n}$
- For token n in document m, its word type assignment is denoted by $x_{m,n}$
- There are $t \in \{1,\ldots,\mathbf{V}\}$ **word types** (entries in our vocabulary).

LDA Generative Process – Algorithm

- We draw topic-word distributions, then document-specific topic distributions, and use both of these to generate the actual words in the documents.

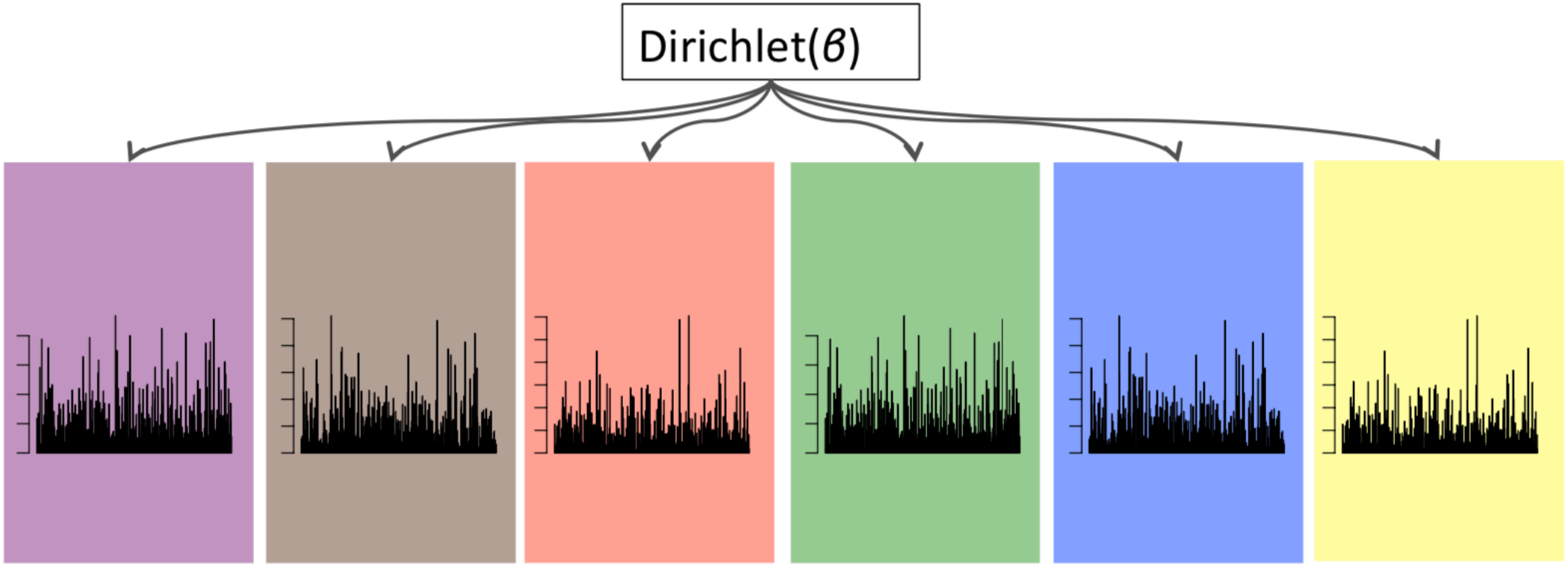
For each topic $k \in \{1, \ldots, K\}$:
  $\boldsymbol{\phi}_k \sim \text{Dir}(\boldsymbol{\beta})$ [*draw distribution over words*]
For each document $m \in \{1, \ldots, M\}$
  $\boldsymbol{\theta}_m \sim \text{Dir}(\boldsymbol{\alpha})$ [*draw distribution over topics*]
  For each word $n \in \{1, \ldots, N_m\}$
    $z_{mn} \sim \text{Mult}(1, \boldsymbol{\theta}_m)$ [*draw topic assignment*]
    $x_{mn} \sim \boldsymbol{\phi}_{z_{mi}}$ [*draw word*]

LDA Plate Model



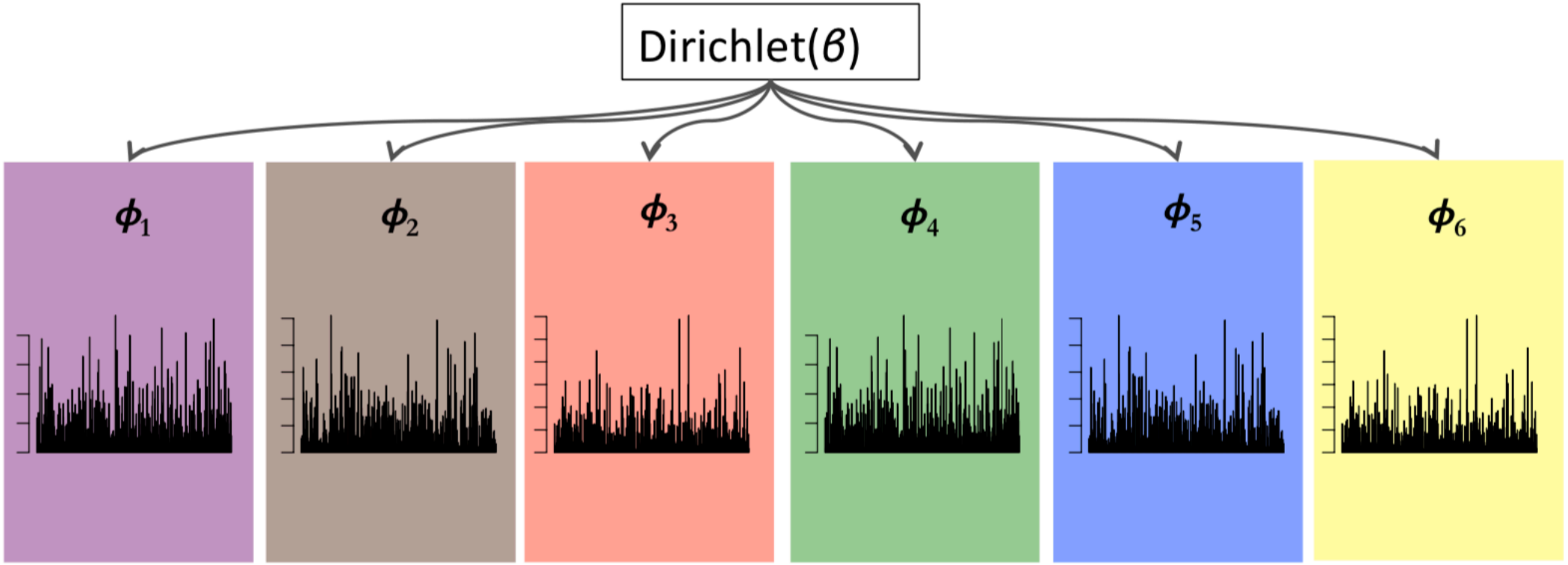
LDA Generative Process

- Each topic is a distribution over words.



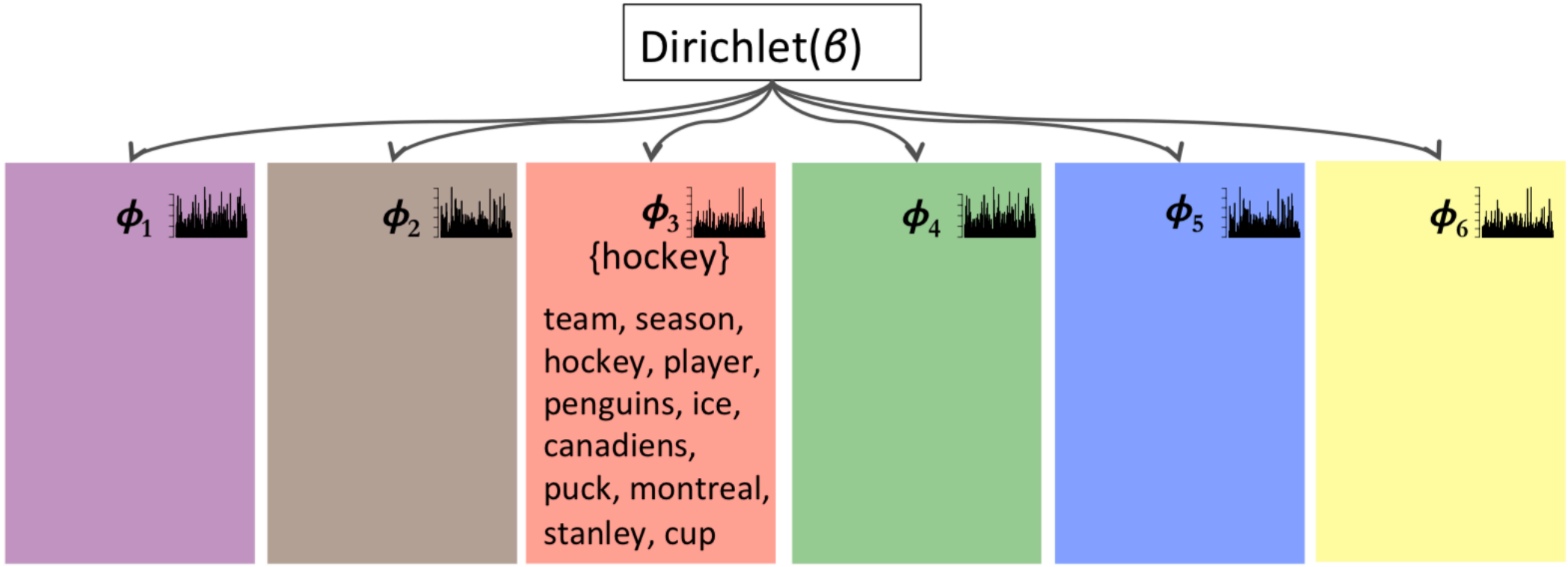
LDA Generative Process

- We represent these topic-word distributions by thetas.

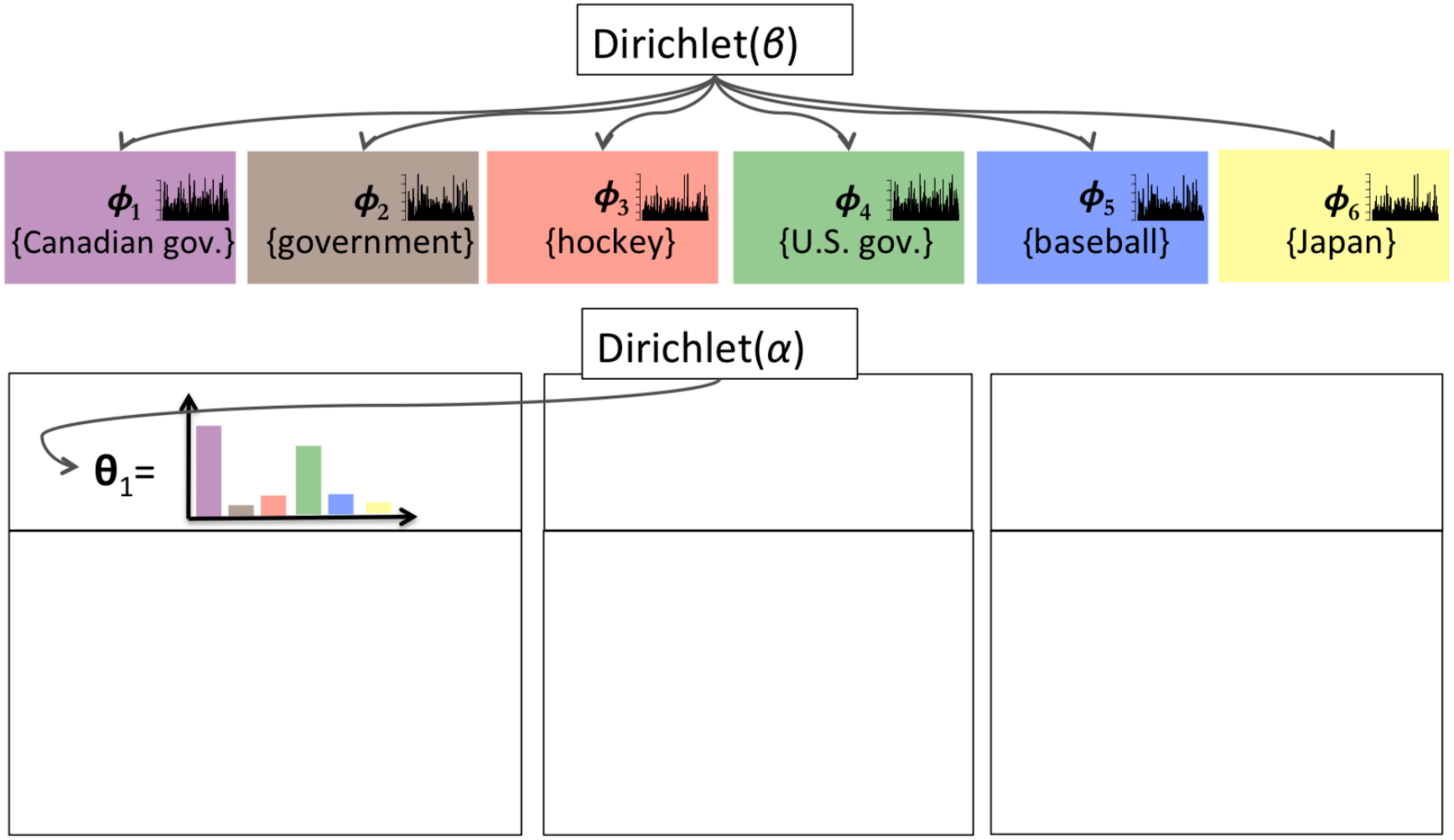


LDA Generative Process

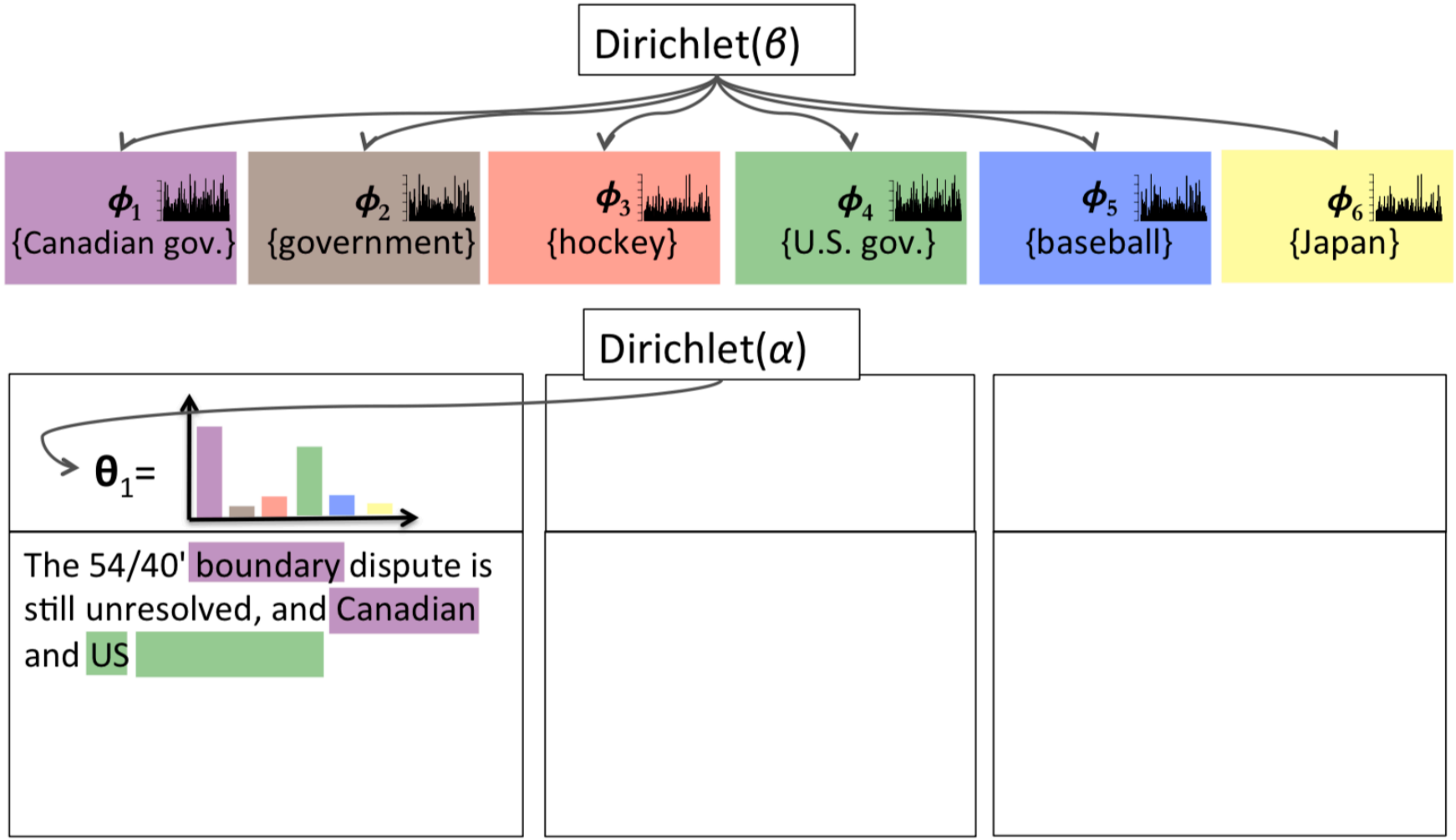
- Each topic is usually represented by the highest probability words in that topic, and researchers typically assign an overarching label to the topic.



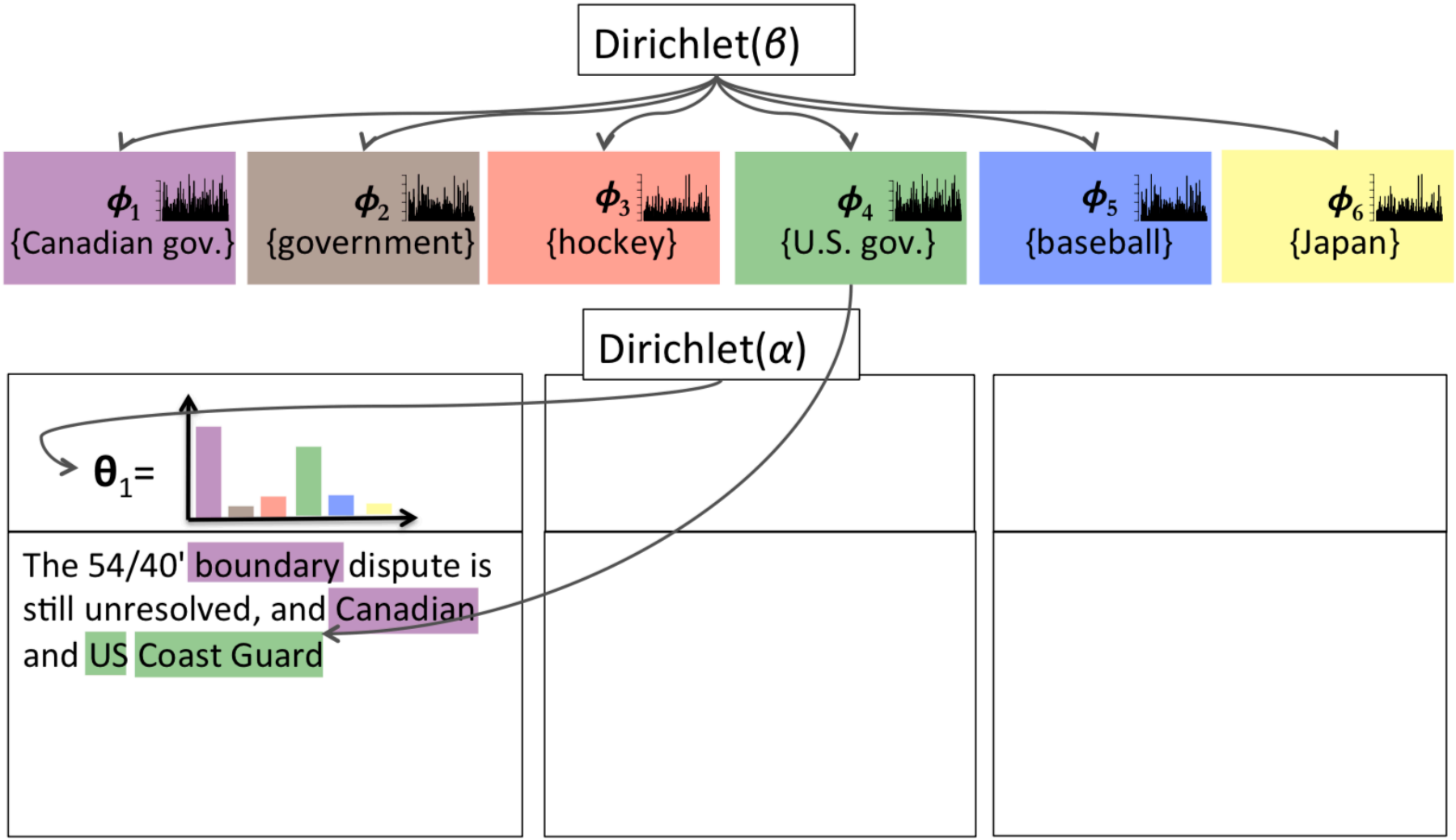
LDA Generative Process



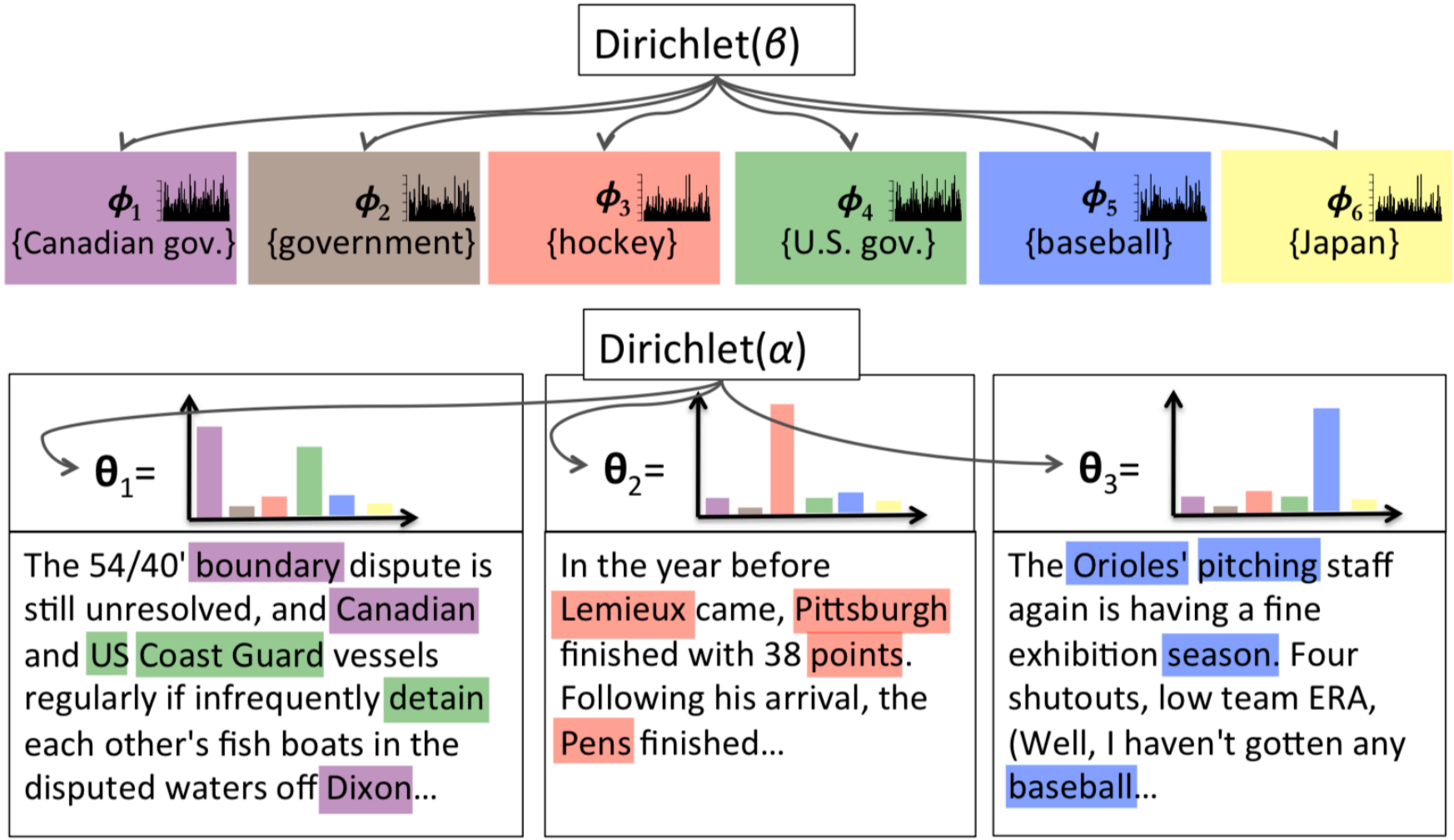
LDA Generative Process



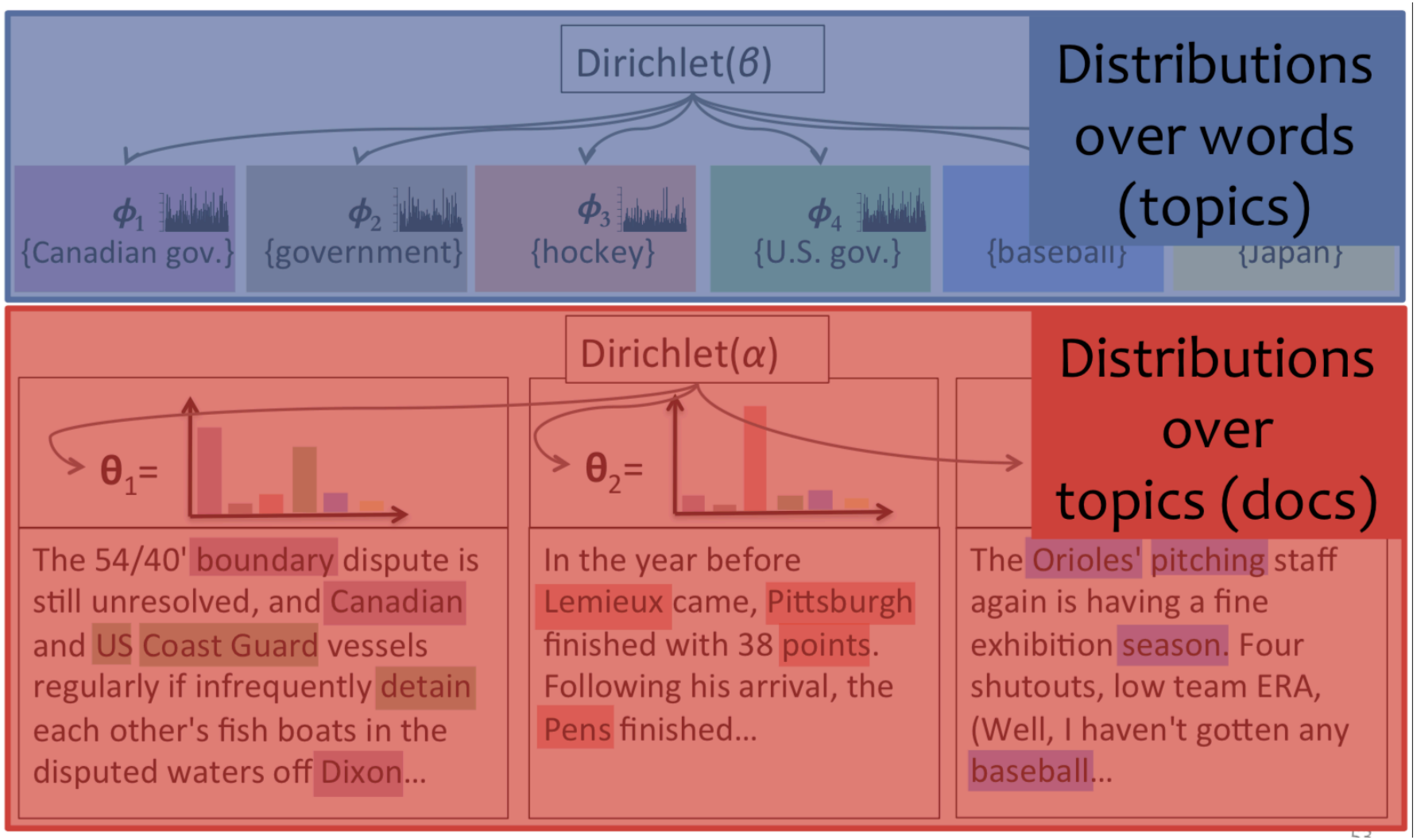
LDA Generative Process



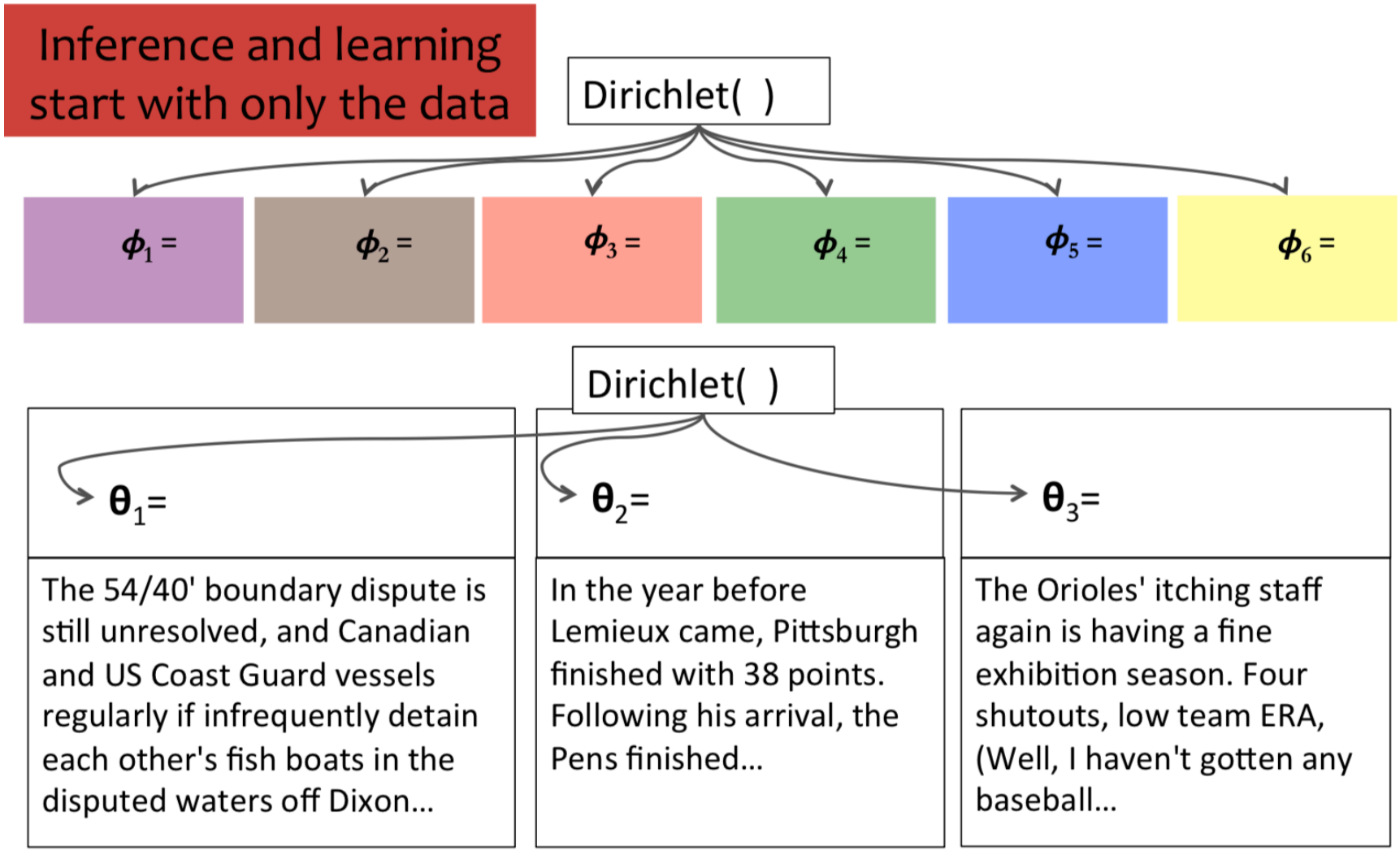
LDA Generative Process



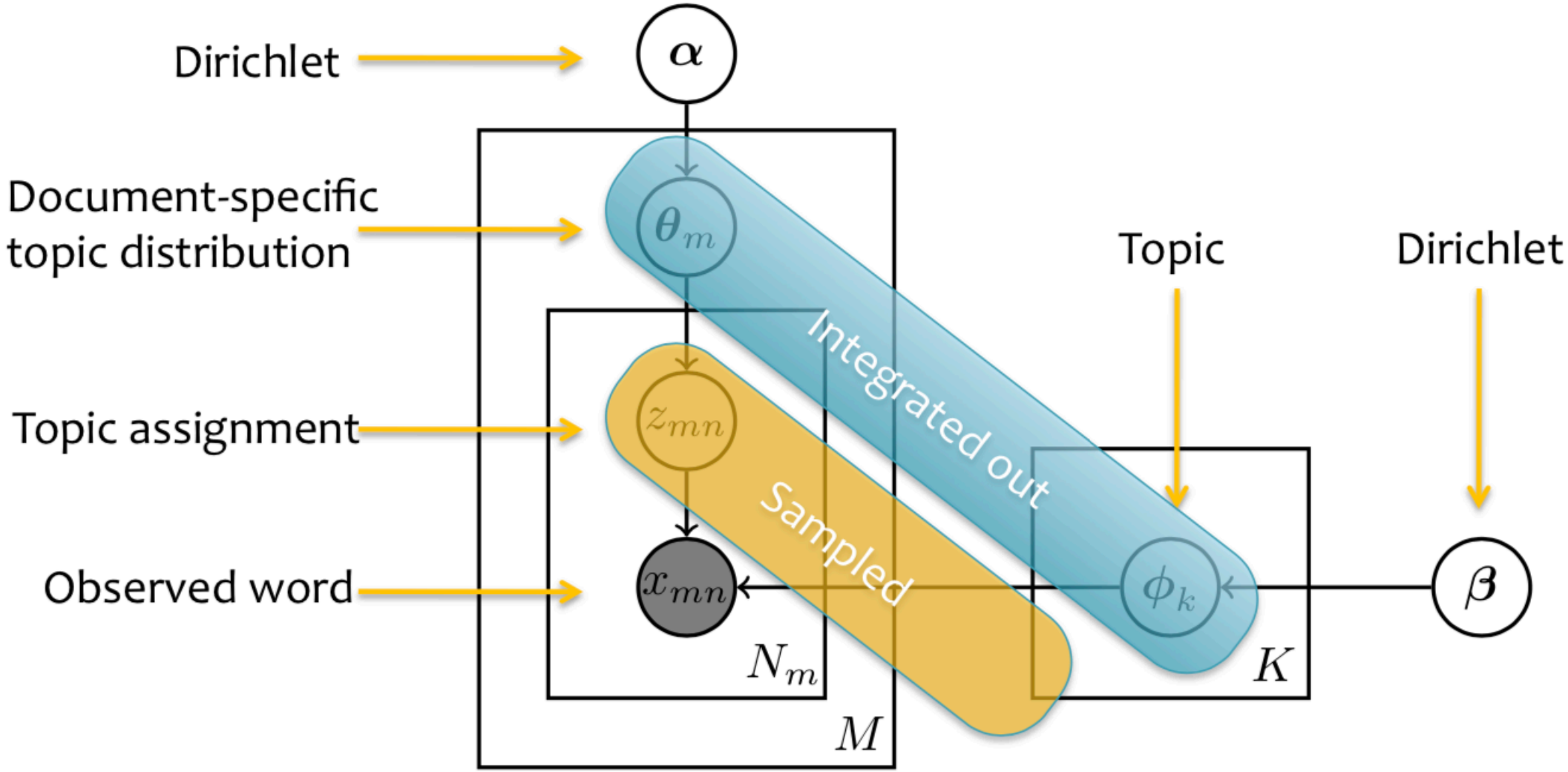
LDA Generative Process



LDA Problem of Inference



Collapsed Gibbs Sampling



Collapsed Gibbs Sampling Algorithm

- **Collapsed**: because we integrate out document-topic and topic-word distributions, for inference all we have to do is resample individual word-topic assignments.
- **Gibbs Sampling**: initialize to random token-topic assignments and keep resampling token-topic assignments until empirical distribution over topic assignments in documents converges.

Algorithm:

- While not done…
  - For each document, m:
    - For each word, n:
      - » Resample a single topic assignment using the full conditionals for z_{mn}

The Math to Determine Conditionals

$$
\begin{aligned}
p(z_i = k | Z^{-i}, X, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{p(X, Z | \boldsymbol{\alpha}, \boldsymbol{\beta})}{p(X, Z^{-i} | \boldsymbol{\alpha}, \boldsymbol{\beta})} \\
&\propto p(X, Z | \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= p(X | Z, \boldsymbol{\beta}) p(Z | \boldsymbol{\alpha}) \\
&= \int_{\Phi} p(X | Z, \Phi) p(\Phi | \boldsymbol{\beta}) \, d\Phi \int_{\Theta} p(Z | \Theta) p(\Theta | \boldsymbol{\alpha}) \, d\Theta \\
&= \left(\prod_{k=1}^{K} \frac{B(\vec{n}_k + \boldsymbol{\beta})}{B(\boldsymbol{\beta})} \right) \left(\prod_{m=1}^{M} \frac{B(\vec{n}_m + \boldsymbol{\alpha})}{B(\boldsymbol{\alpha})} \right) \\
&= \frac{n_{kt}^{-i} + \beta_t}{\sum_{v=1}^{T} n_{kv}^{-i} + \beta_v} \cdot \frac{n_{mk}^{-i} + \alpha_k}{\sum_{j=1}^{K} n_{mj}^{-i} + \alpha_j}
\end{aligned}
$$

where t, m are given by i

Resampling Token Topic Assignments

- Below we have the probability of the i'th token in document *m* belonging in topic *k*, conditional on the current topic assignments of all of the other token-topic assignments in the document/corpus excluding the assignments of the current token.
- Use this to build a probability vector to sample new token topic assignment out of.

$$p(z_i = k | Z^{-i}, X, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{n_{kt}^{-i} + \beta_t}{\sum_{v=1}^{T} n_{kv}^{-i} + \beta_v} \cdot \frac{n_{mk}^{-i} + \alpha_k}{\sum_{j=1}^{K} n_{mj}^{-i} + \alpha_j}$$

where t, m are given by i

n_{kt} = # times topic k appears with type t

n_{mk} = # times topic k appears in document m

LDA (collapsed Gibbs Sampling)– Initialization

```
// initialisation
zero all count variables, n_m^(k), n_m, n_k^(t), n_k
for all documents m ∈ [1, M] do
    for all words n ∈ [1, N_m] in document m do
        sample topic index z_{m,n}=k ~ Mult(1/K)
        increment document–topic count: n_m^(k) += 1
        increment document–topic sum: n_m += 1
        increment topic–term count: n_k^(t) += 1
        increment topic–term sum: n_k += 1
```

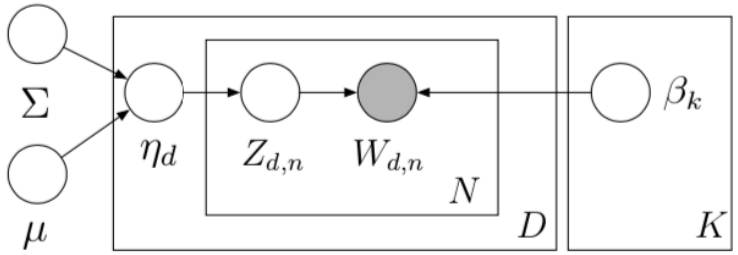
LDA (collapsed Gibbs Sampling)– Inference

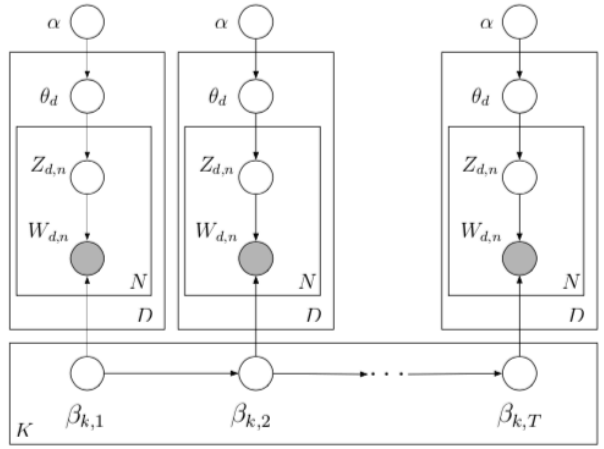
```
// Gibbs sampling over burn-in period and sampling period
while not finished do
    for all documents m ∈ [1, M] do
        for all words n ∈ [1, N_m] in document m do
            // for the current assignment of k to a term t for word w_{m,n}:
            decrement counts and sums: n_m^(k) −= 1; n_m −= 1; n_k^(t) −= 1; n_k −= 1
            // multinomial sampling acc. to Eq. 78 (decrements from previous step):
            sample topic index k̃ ~ p(z_i | z⃗_{¬i}, w⃗)
            // for the new assignment of z_{m,n} to the term t for word w_{m,n}:
            increment counts and sums: n_m^(k̃) += 1; n_m += 1; n_k̃^(t) += 1; n_k̃ += 1
```

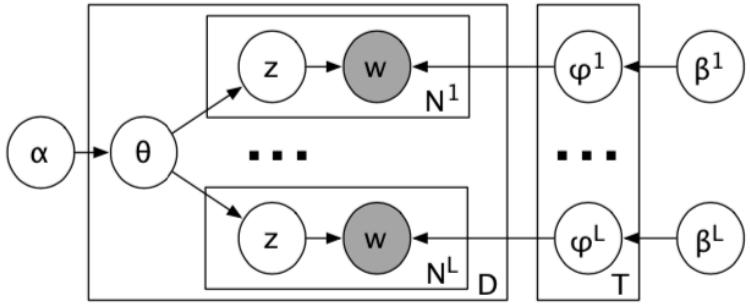
LDA Extensions

- Correlated topic models
  - Logistic normal prior over topic assignments
- Dynamic topic models
  - Learns topic changes over time
- Polylingual topic models
  - Learns topics aligned across multiple languages

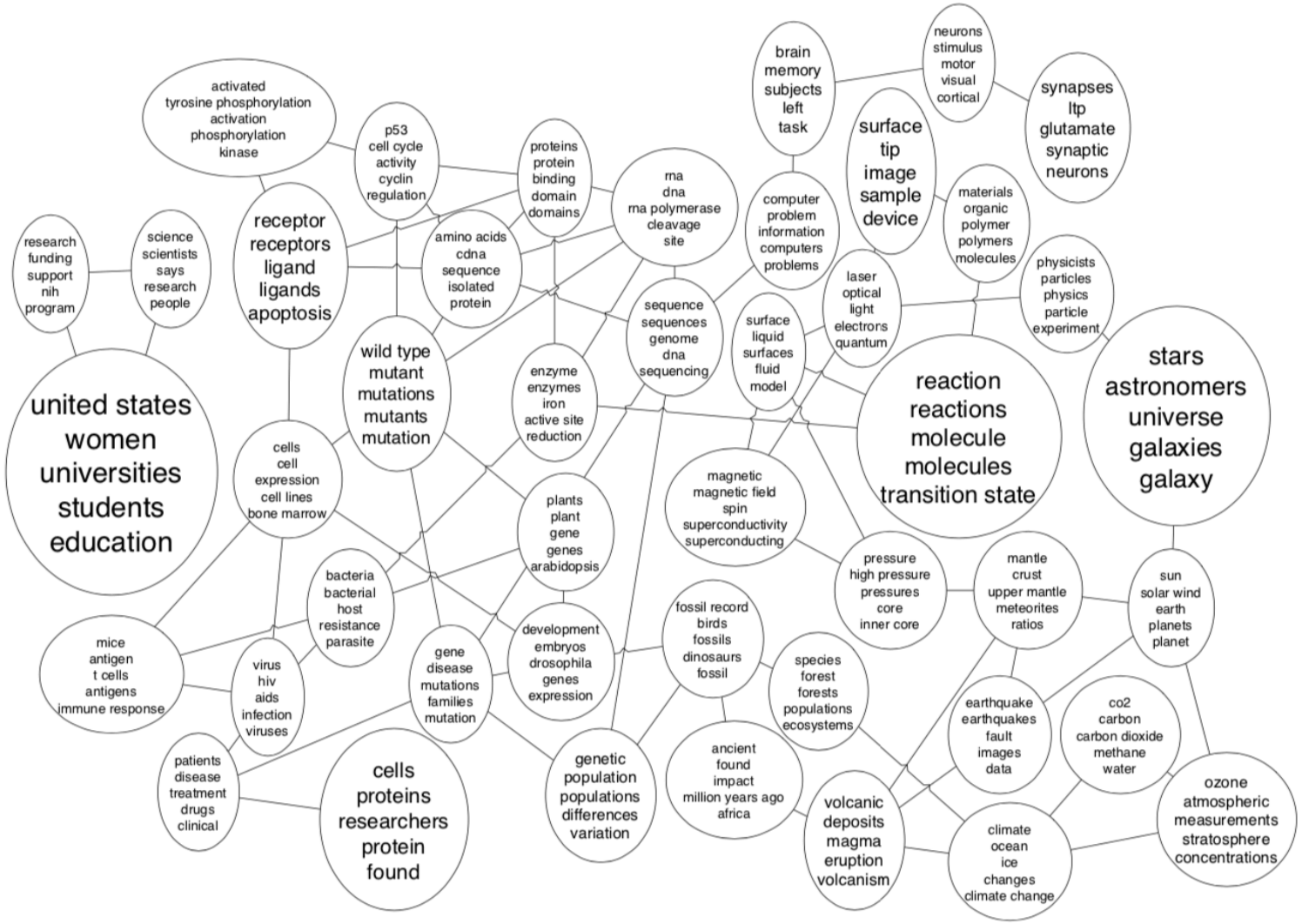
…



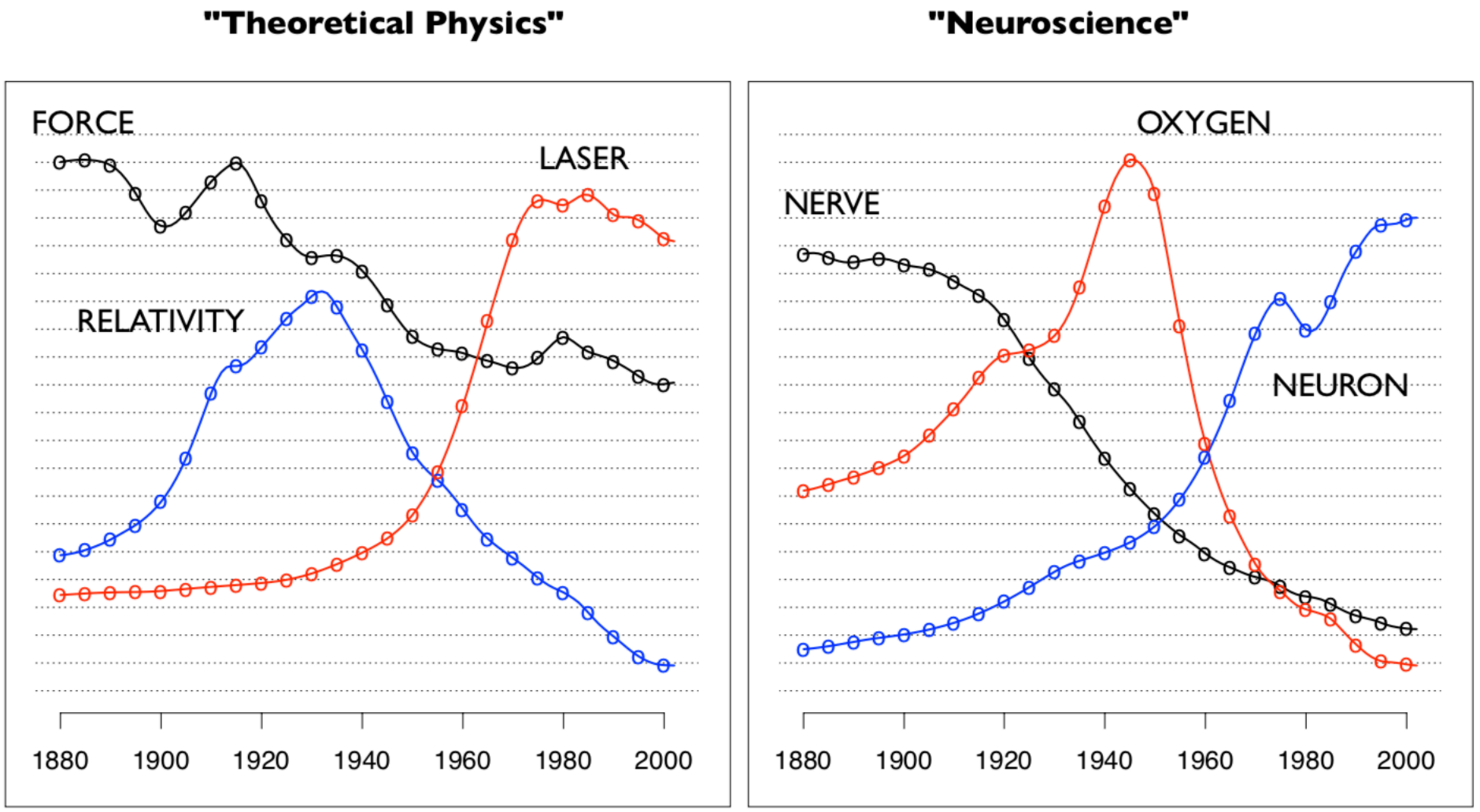




Correlated Topic Model



Dynamic Topic Model



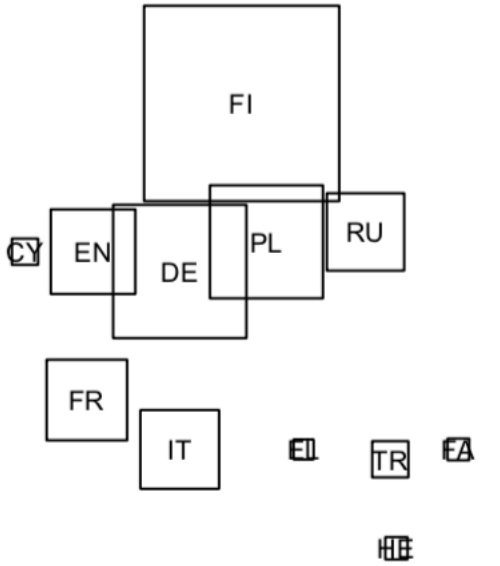
Polylingual Topic Models

Topic 1 (twelve languages)

CY	sadwrn blaned gallair at lloeren mytholeg
DE	space nasa sojus flug mission
EL	διαστημικό sts nasa αγγλ small
EN	**space mission launch satellite nasa spacecraft**
FA	فضایی ماموریت ناسا مدار فضانورد ماهواره
FI	sojuz nasa apollo ensimmäinen space lento
FR	spatiale mission orbite mars satellite spatial
HE	החלל הארץ חלל כדור א תוכנית
IT	spaziale missione programma space sojuz stazione
PL	misja kosmicznej stacji misji space nasa
RU	космический союз космического спутник станции
TR	uzay soyuz ay uzaya salyut sovyetler

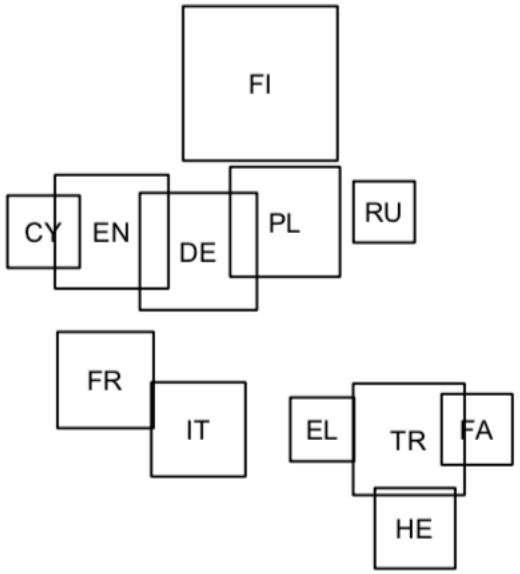
Polylingual Topic Models

world
ski
km
won



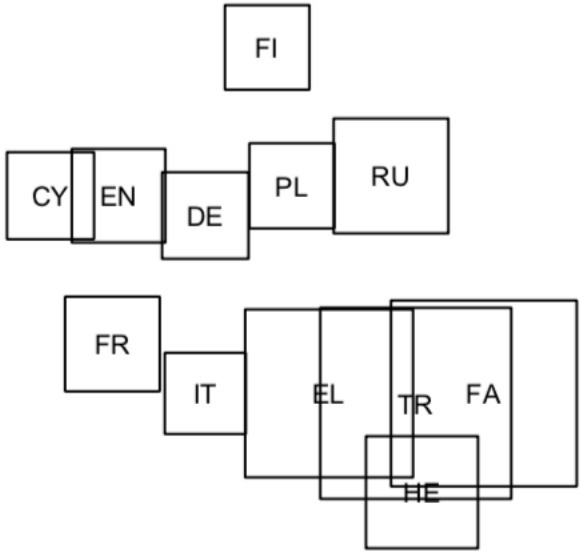
Analysis: mostly Nordic countries

actor
role
television
actress



Analysis: uniform across countries

ottoman
empire
khan
byzantine



Analysis: mostly countries near Istanbul

Topic Models – My Take

- Topic models can be a valuable tool for discovering themes/topics in a corpus.
- There are lots of different extensions (which have varying usefulness, some of which have questionable properties).
- As much as topic models are unsupervised, and thus (theoretically free from human bias), researcher decisions have a huge impact on results.
  - PreText (Denny and Spirling, 2018)
  - Choice of hyper-parameters (number of topics, concentration parameters, iterations, etc.).
  - Rerunning model with a different seed will change results.

Topic Models – My Take

- To serve as a valid basis for measurement, need to run your topic model only once.
- Validation (quantitative and qualitative is key).
- Experience (understanding how to preprocess data generally for interpretable results, how to select model hyper parameters, etc.) is key to not cherry-picking results from multiple model runs.
- Accessing many of these models will require you to step outside of R/python.
- Implementing collapsed Gibbs sampling for LDA from scratch is a good exercise if you are interested in programming.