10-708 Probabilistic Graphical Models

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Bayesian Nonparametrics:

Dirichlet Process

+

Dirichlet Process Mixture Model

Matt Gormley
Lecture 22
Apr. 21, 2021

Reminders

- Project Midway Milestones:
  - Midway Poster Session:
    Tue, Apr. 27 at 6:30pm – 8:30pm
  - Midway Executive Summary
    Due: Tue, Apr. 27 at 11:59pm
  - New requirement: must have baseline results
- Quiz 3
  - Mon, May 3 during lecture slot
  - Topics: Lectures 16 - 23

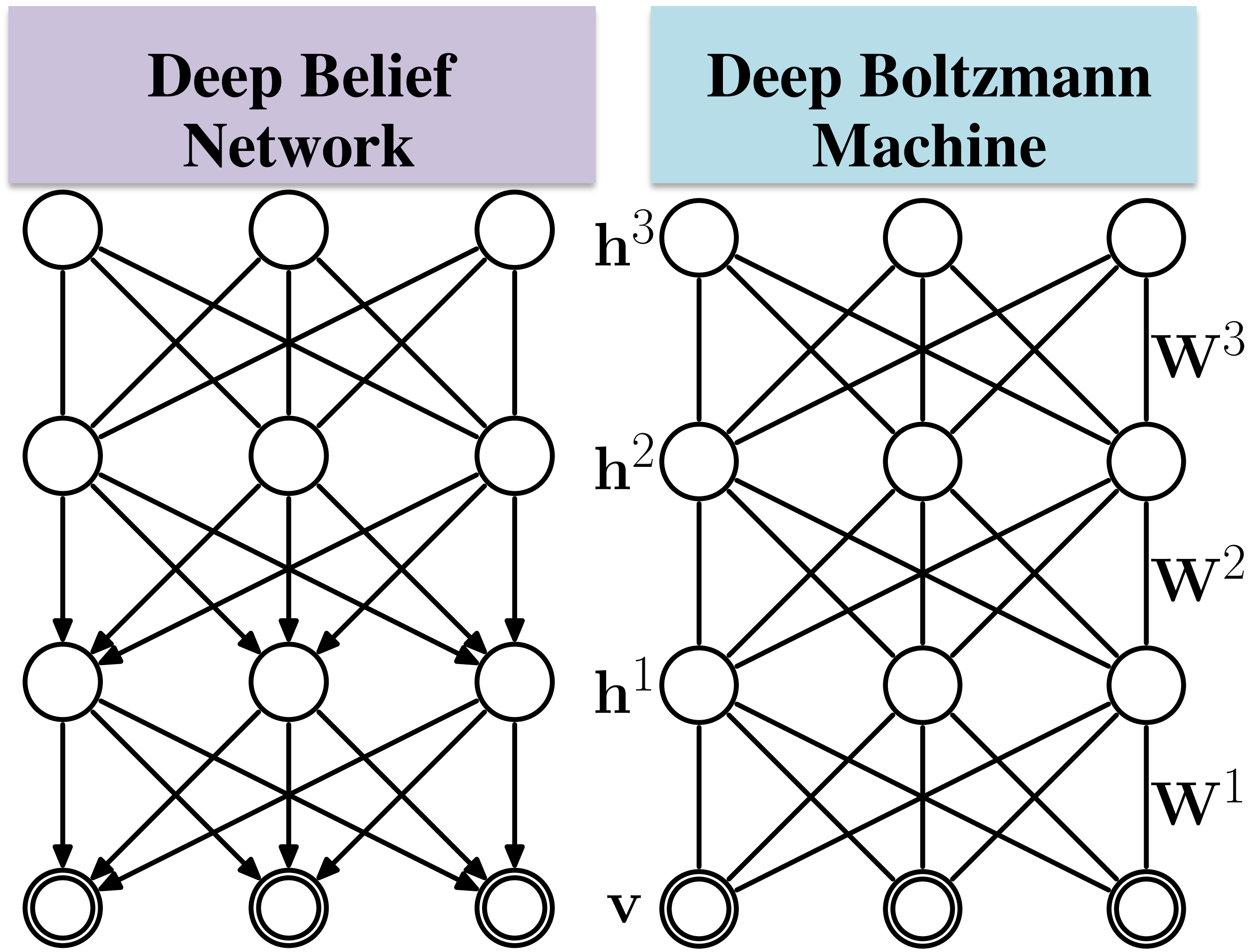
DEEP BOLTZMAN MACHINES (DBMS)

Outline

DBMs

Deep Boltzman Machines

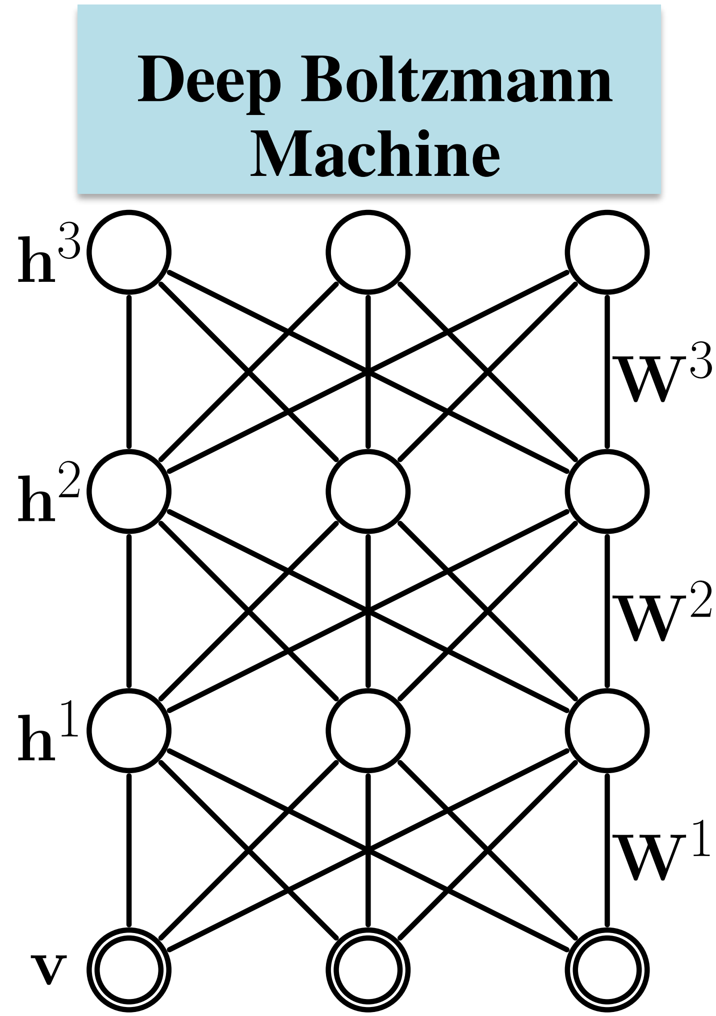
- DBNs are a hybrid directed/undirected graphical model
- DBMs are a purely undirected graphical model



DBMs

Deep Boltzman Machines

Can we use the same techniques to train a DBM?



LEARNING STANDARD BOLTZMAN MACHINES

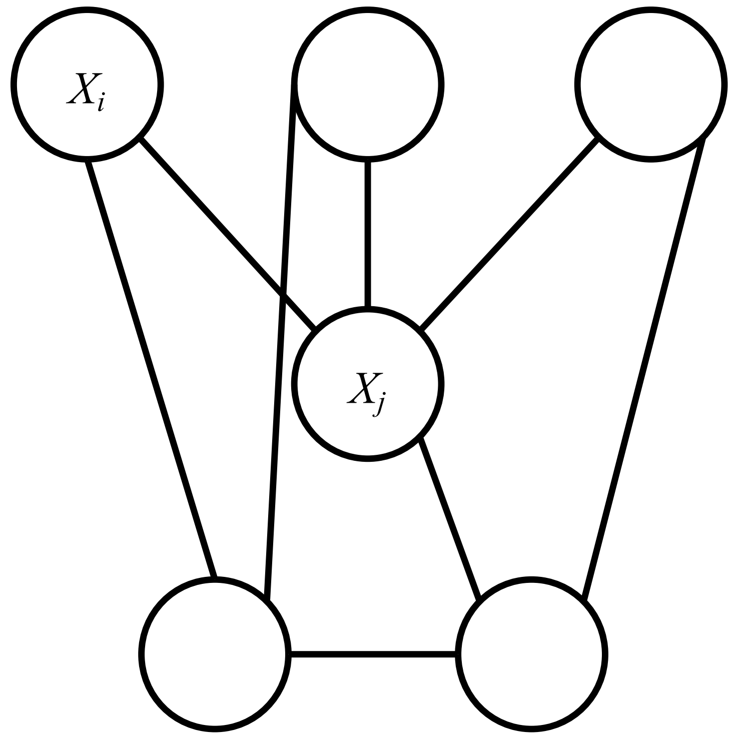
DBMs

Boltzman Machine

- Undirected graphical model of binary variables with pairwise potentials
- Parameterization of the potentials:

$$\psi_{ij}(x_i, x_j) = \exp(x_i W_{ij} x_j)$$

(In English: higher value of parameter W_{ij} leads to higher correlation between X_i and X_j on value 1)

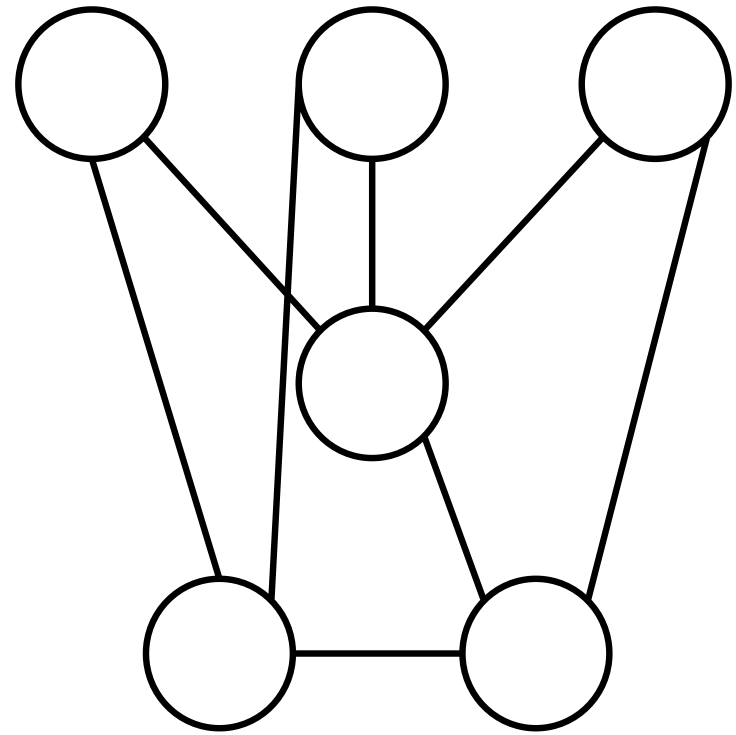


DBMs

Learning Standard Boltzman Machines

Visible units: $\mathbf{v} \in \{0,1\}^D$

Hidden units: $\mathbf{h} \in \{0,1\}^P$



Likelihood:

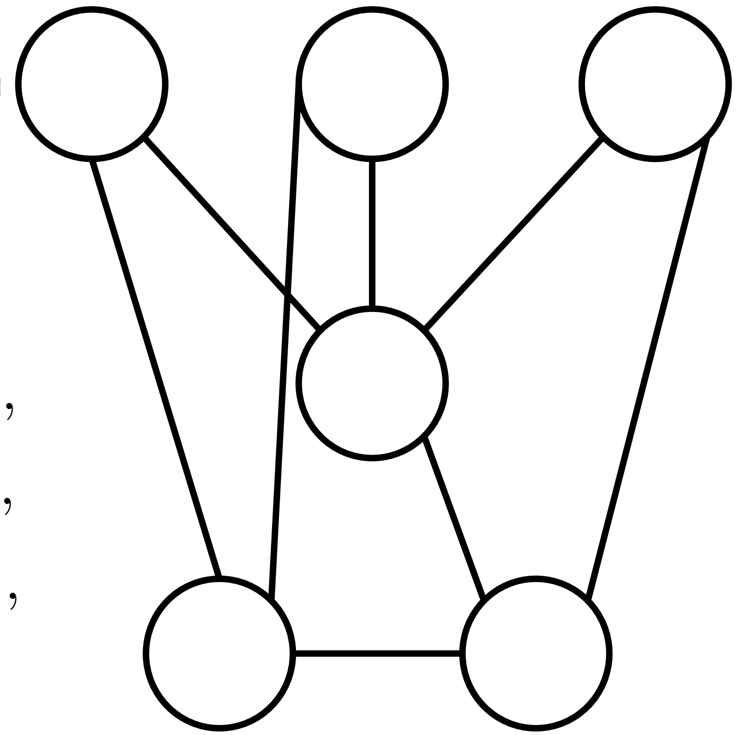
$$E(\mathbf{v},\mathbf{h};\theta) = -\frac{1}{2}\mathbf{v}^\top \mathbf{L}\mathbf{v} - \frac{1}{2}\mathbf{h}^\top \mathbf{J}\mathbf{h} - \mathbf{v}^\top \mathbf{W}\mathbf{h},$$

$$p(\mathbf{v};\theta) = \frac{p^*(\mathbf{v};\theta)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_h \exp\left(-E(\mathbf{v},\mathbf{h};\theta)\right),$$

$$Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v},\mathbf{h};\theta)\right),$$

DBMs

Learning Standard Boltzman Machines



(Old) idea from Hinton & Sejnowski (1983): For each iteration of optimization, run a separate MCMC chain for each of the data and model expectations to approximate the parameter updates.

Delta updates to each of model parameters:

$$\begin{aligned}\Delta\mathbf{W} &= \alpha\left(\mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{h}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{v}\mathbf{h}^\top]\right),\\ \Delta\mathbf{L} &= \alpha\left(\mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{v}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{v}\mathbf{v}^\top]\right),\\ \Delta\mathbf{J} &= \alpha\left(\mathrm{E}_{P_{\mathrm{data}}}[\mathbf{h}\mathbf{h}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{h}\mathbf{h}^\top]\right),\end{aligned}$$

Full conditionals for Gibbs sampler:

$$p(h_j = 1|\mathbf{v}, \mathbf{h}_{-j}) = \sigma\Big(\sum_{i=1}^{D} W_{ij} v_i + \sum_{m=1\backslash j}^{P} J_{jm} h_j\Big),$$

$$p(v_i = 1|\mathbf{h}, \mathbf{v}_{-i}) = \sigma\Big(\sum_{j=1}^{P} W_{ij} h_j + \sum_{k=1\backslash i}^{D} L_{ik} v_j\Big),$$

DBMs

Learning Standard Boltzman Machines

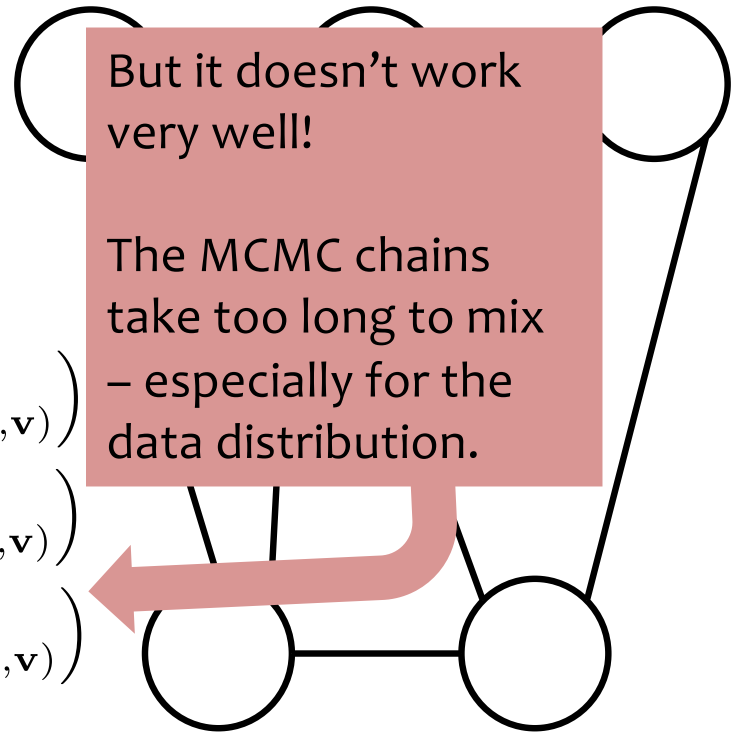
(Old) idea from Hinton & Sejnowski (1983): For each iteration of optimization, run a separate MCMC chain for each of the data and model expectations to approximate the parameter updates.

Delta updates to each of model parameters:

$$\Delta \mathbf{W} = \alpha \left(\langle \mathbf{v}\mathbf{h}^T \rangle_{\mathbf{v} \in \mathcal{D}, \mathbf{h} \sim p(\mathbf{h}|\mathbf{v})} - \langle \mathbf{v}\mathbf{h}^T \rangle_{\mathbf{v}, \mathbf{h} \sim p(\mathbf{h}, \mathbf{v})} \right)$$

$$\Delta \mathbf{L} = \alpha \left(\langle \mathbf{v}\mathbf{v}^T \rangle_{\mathbf{v} \in \mathcal{D}, \mathbf{h} \sim p(\mathbf{h}|\mathbf{v})} - \langle \mathbf{v}\mathbf{v}^T \rangle_{\mathbf{v}, \mathbf{h} \sim p(\mathbf{h}, \mathbf{v})} \right)$$

$$\Delta \mathbf{J} = \alpha \left(\langle \mathbf{h}\mathbf{h}^T \rangle_{\mathbf{v} \in \mathcal{D}, \mathbf{h} \sim p(\mathbf{h}|\mathbf{v})} - \langle \mathbf{h}\mathbf{h}^T \rangle_{\mathbf{v}, \mathbf{h} \sim p(\mathbf{h}, \mathbf{v})} \right)$$



But it doesn't work very well!

The MCMC chains take too long to mix – especially for the data distribution.

Full conditionals for Gibbs sampler:

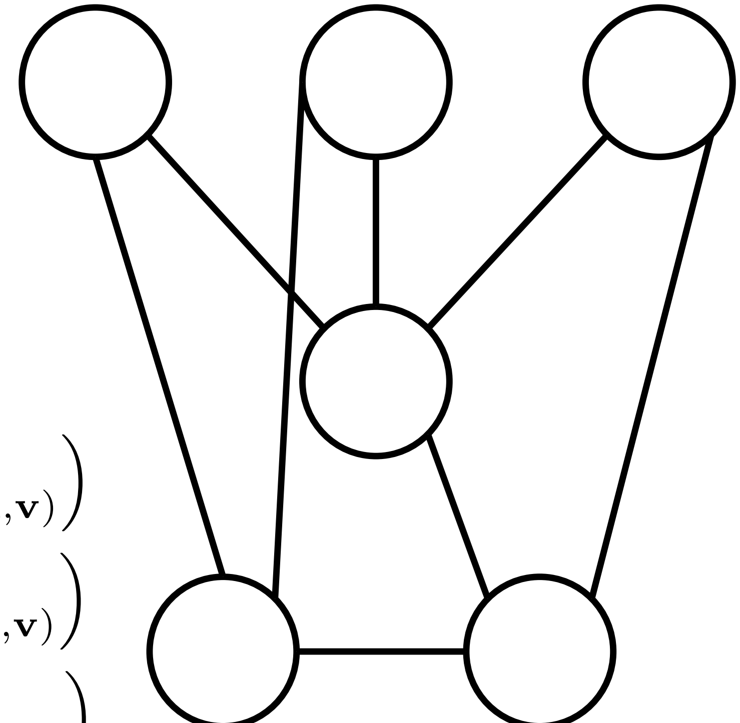
$$p(h_j = 1 | \mathbf{v}, \mathbf{h}_{-j}) = \sigma\Big(\sum_{i=1}^{D} W_{ij} v_i + \sum_{m=1 \backslash j}^{P} J_{jm} h_j \Big),$$

$$p(v_i = 1 | \mathbf{h}, \mathbf{v}_{-i}) = \sigma\Big(\sum_{j=1}^{P} W_{ij} h_j + \sum_{k=1 \backslash i}^{D} L_{ik} v_j \Big),$$

Learning Standard Boltzman Machines

(New) idea from Salakhutinov & Hinton (2009):

- Step 1) Approximate the data distribution by variational inference.
- Step 2) Approximate the model distribution with a "persistent" Markov chain (from iteration to iteration)



Delta updates to each of model parameters:

$$\Delta \mathbf{W} = \alpha \left(\langle \mathbf{v}\mathbf{h}^T \rangle_{\mathbf{v} \in \mathcal{D}, \mathbf{h} \sim p(\mathbf{h}|\mathbf{v})} - \langle \mathbf{v}\mathbf{h}^T \rangle_{\mathbf{v}, \mathbf{h} \sim p(\mathbf{h}, \mathbf{v})} \right)$$

$$\Delta \mathbf{L} = \alpha \left(\langle \mathbf{v}\mathbf{v}^T \rangle_{\mathbf{v} \in \mathcal{D}, \mathbf{h} \sim p(\mathbf{h}|\mathbf{v})} - \langle \mathbf{v}\mathbf{v}^T \rangle_{\mathbf{v}, \mathbf{h} \sim p(\mathbf{h}, \mathbf{v})} \right)$$

$$\Delta \mathbf{J} = \alpha \left(\langle \mathbf{h}\mathbf{h}^T \rangle_{\mathbf{v} \in \mathcal{D}, \mathbf{h} \sim p(\mathbf{h}|\mathbf{v})} - \langle \mathbf{h}\mathbf{h}^T \rangle_{\mathbf{v}, \mathbf{h} \sim p(\mathbf{h}, \mathbf{v})} \right)$$

DBMs

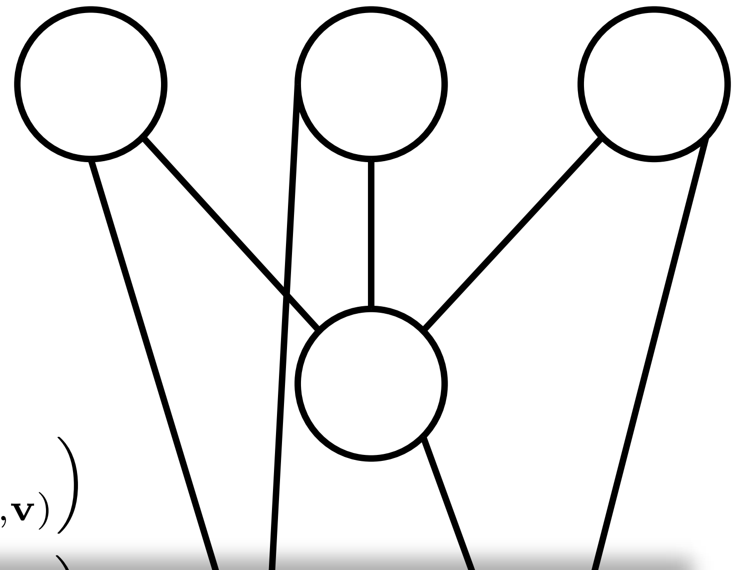
Learning Standard Boltzman Machines

(New) idea from Salakhutinov & Hinton (2009):

- Step 1) Approximate the data distribution by variational inference.
- Step 2) Approximate the model distribution with a "persistent" Markov chain (from iteration to iteration)

Delta updates to each of model parameters:

$$\Delta \mathbf{W} = \alpha \left(\langle \mathbf{v}\mathbf{h}^T \rangle_{\mathbf{v}\in\mathcal{D}, \mathbf{h}\sim p(\mathbf{h}|\mathbf{v})} - \langle \mathbf{v}\mathbf{h}^T \rangle_{\mathbf{v},\mathbf{h}\sim p(\mathbf{h},\mathbf{v})} \right)$$



Step 1) Approximate the data distribution…

Mean-field approximation:

$$q(\mathbf{h};\mu) = \prod_{j=1}^{P} q(h_i)$$

$$q(h_i = 1) = \mu_i$$

Variational lower-bound of log-likelihood:

$$\ln p(\mathbf{v};\theta) \geq \sum_{\mathbf{h}} q(\mathbf{h}|\mathbf{v};\mu) \ln p(\mathbf{v},\mathbf{h};\theta) + \mathcal{H}(q)$$

Fixed-point equations for variational params:

$$\mu_j \leftarrow \sigma\Big(\sum_i W_{ij} v_i + \sum_{m \setminus j} J_{mj} \mu_m\Big)$$

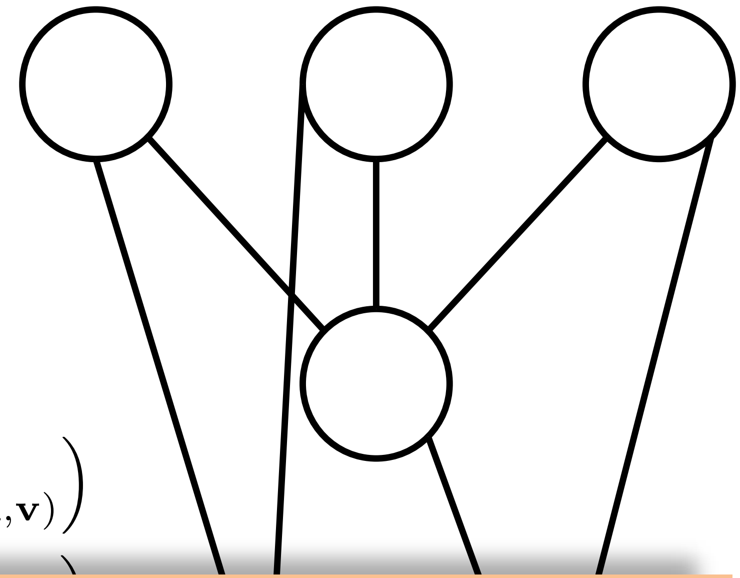
DBMs

Learning Standard Boltzman Machines

(New) idea from Salakhutinov & Hinton (2009):

- Step 1) Approximate the data distribution by variational inference.
- Step 2) Approximate the model distribution with a “persistent” Markov chain (from iteration to iteration)

Delta updates to each of model parameters:



$$\Delta \mathbf{W} = \alpha \left(\left\langle \mathbf{v}\mathbf{h}^T \right\rangle_{\mathbf{v} \in \mathcal{D}, \mathbf{h} \sim p(\mathbf{h}|\mathbf{v})} - \left\langle \mathbf{v}\mathbf{h}^T \right\rangle_{\mathbf{v}, \mathbf{h} \sim p(\mathbf{h}, \mathbf{v})} \right)$$

Step 2) Approximate the model distribution…

Why not use variational inference for the model expectation as well?

Difference of the two mean-field approximated expectations above would cause learning algorithm to **maximize** divergence between true and mean-field distributions.

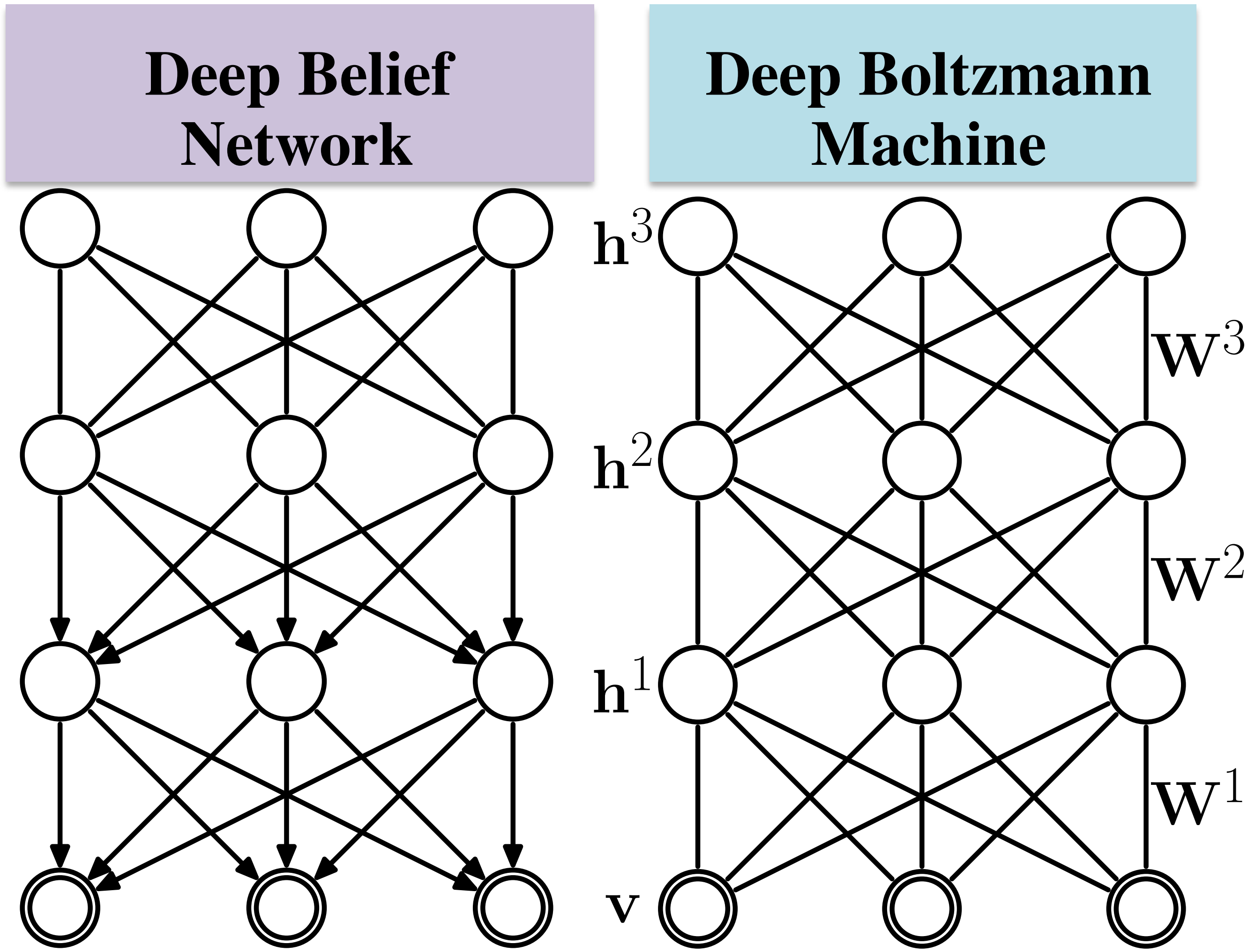
Persistent CD adds correlations between successive iterations, but not an issue.

LEARNING DEEP BOLTZMAN MACHINES

DBMs

Deep Boltzman Machines

- DBNs are a hybrid directed/undirected graphical model
- DBMs are a purely undirected graphical model

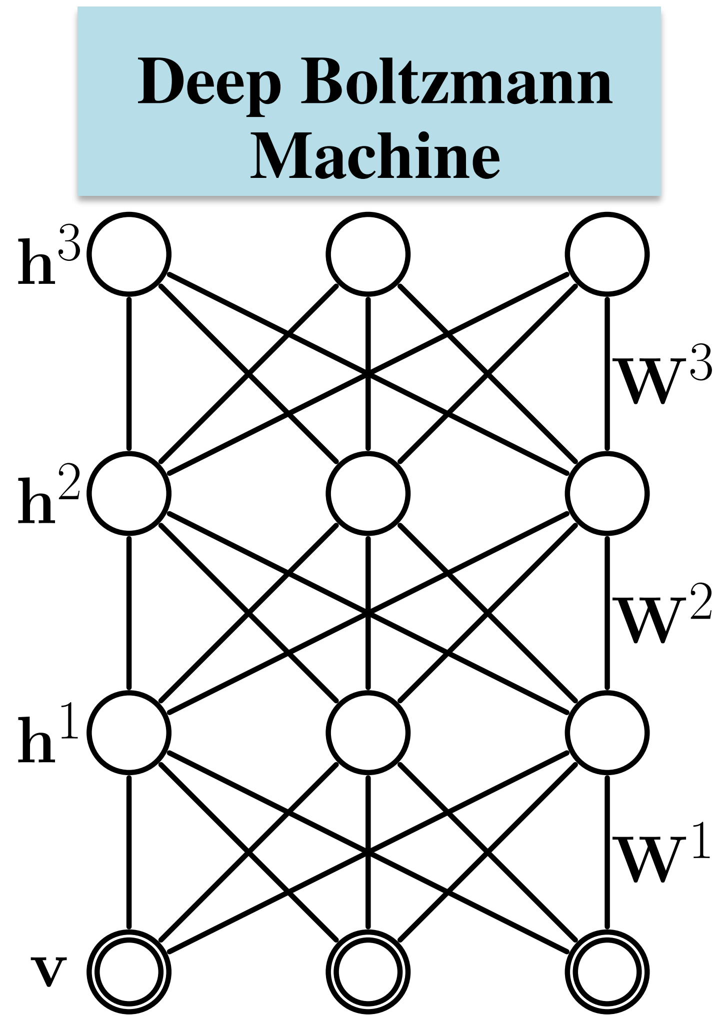


Learning Deep Boltzman Machines

Can we use the same techniques to train a DBM?

I. Pre-train a stack of RBMs in greedy layerwise fashion (requires some caution to avoid double counting)

II. Use those parameters to initialize two step mean-field approach to learning full Boltzman machine (i.e. the full DBM)



Document Clustering and Retrieval

Clustering Results

- Goal: cluster related documents
- Figures show projection to 2 dimensions
- Color shows true categories

PCA

DBM

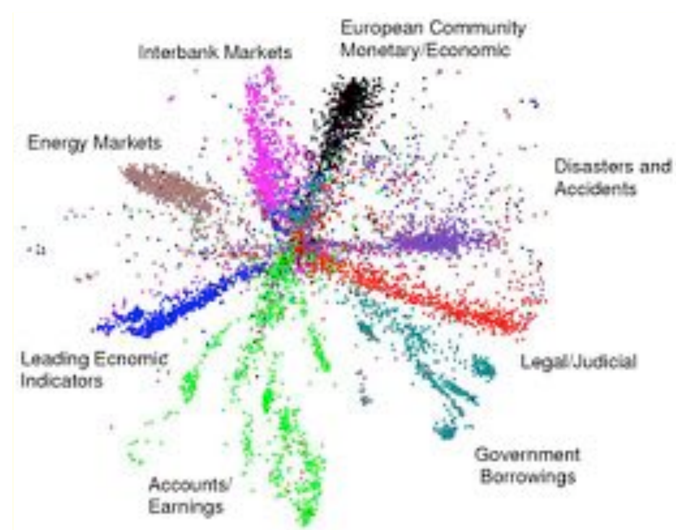


Figure from (Salakhutdinov and Hinton, 2009)

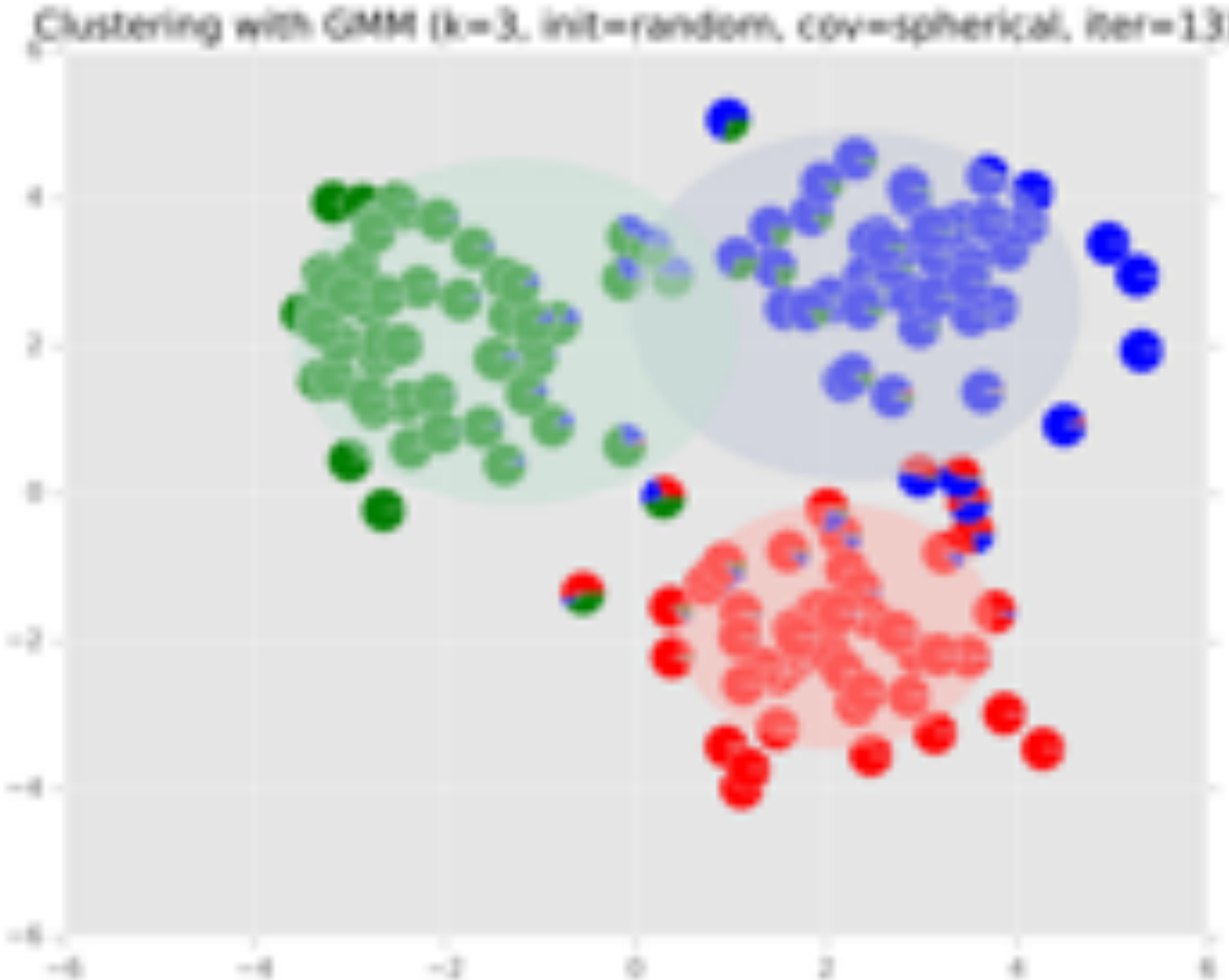
EXAMPLE: K-MEANS & GMM

K-Means Algorithm

- **Given** unlabeled feature vectors
  $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$
- **Initialize** cluster centers $c = \{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(K)}\}$
  and cluster assignments $z = \{z^{(1)}, z^{(2)}, \ldots, z^{(N)}\}$
- **Repeat** until convergence:
  - for j in $\{1, \ldots, K\}$
    $\mathbf{c}^{(j)}$ = **mean** of **all** points assigned to cluster j
  - for i in $\{1, \ldots, N\}$
    $z^{(i)}$ = **index** j of cluster center **nearest** to $\mathbf{x}^{(i)}$

K-Means Example: Real-World Dataset

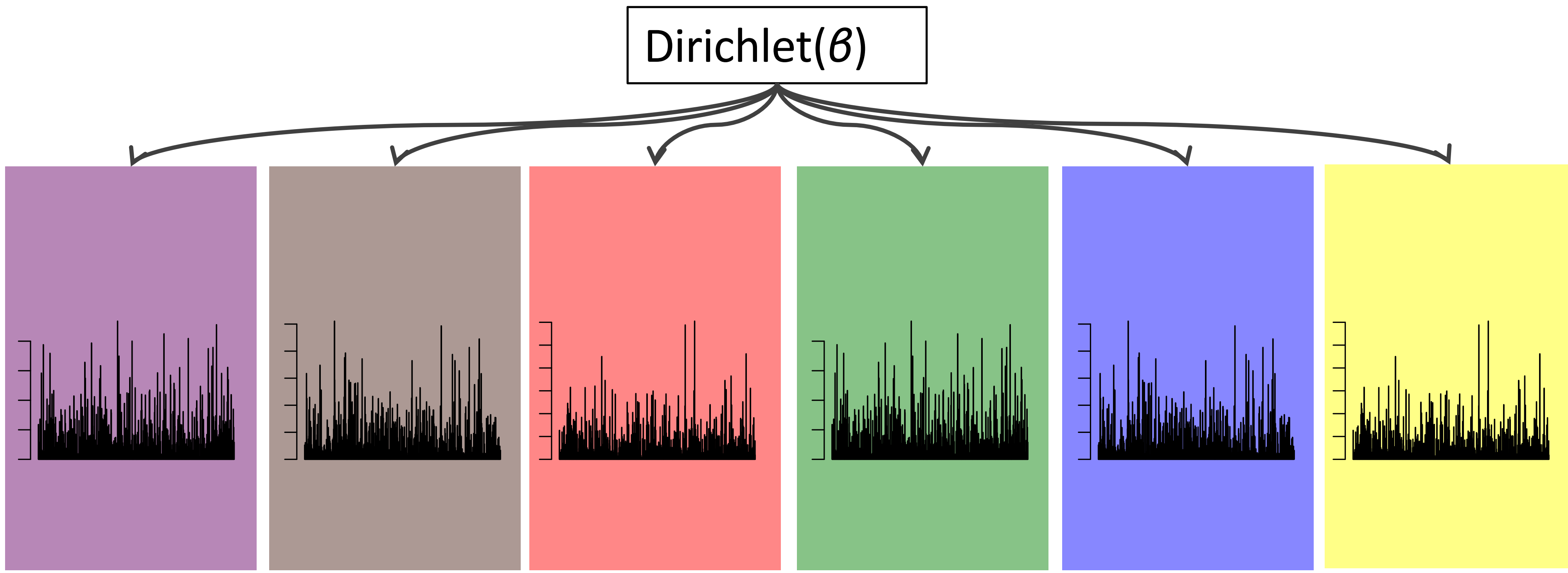
Example: GMM



LATENT DIRICHLET ALLOCATION (LDA)

(Blei, Ng, & Jordan, 2003)

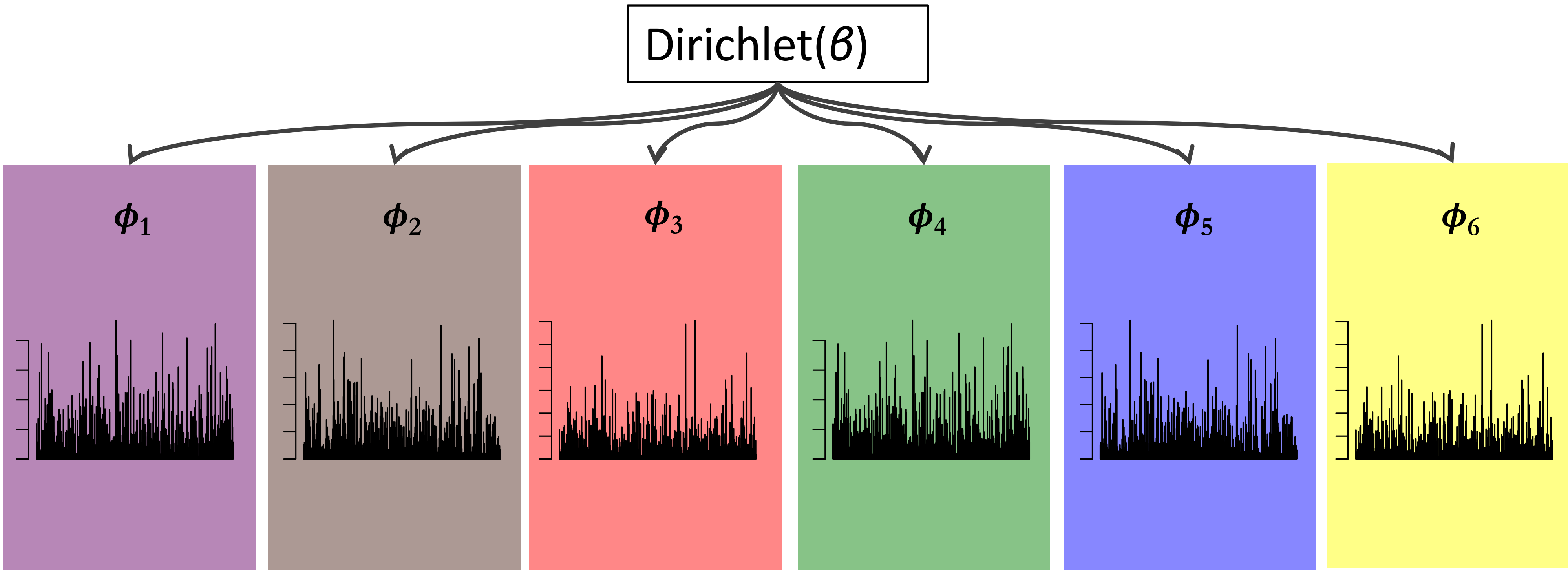
LDA for Topic Modeling



- The **generative story** begins with only a **Dirichlet prior** over the topics.
- Each **topic** is defined as a **Multinomial distribution** over the vocabulary, parameterized by $\boldsymbol{\phi}_k$

(Blei, Ng, & Jordan, 2003)

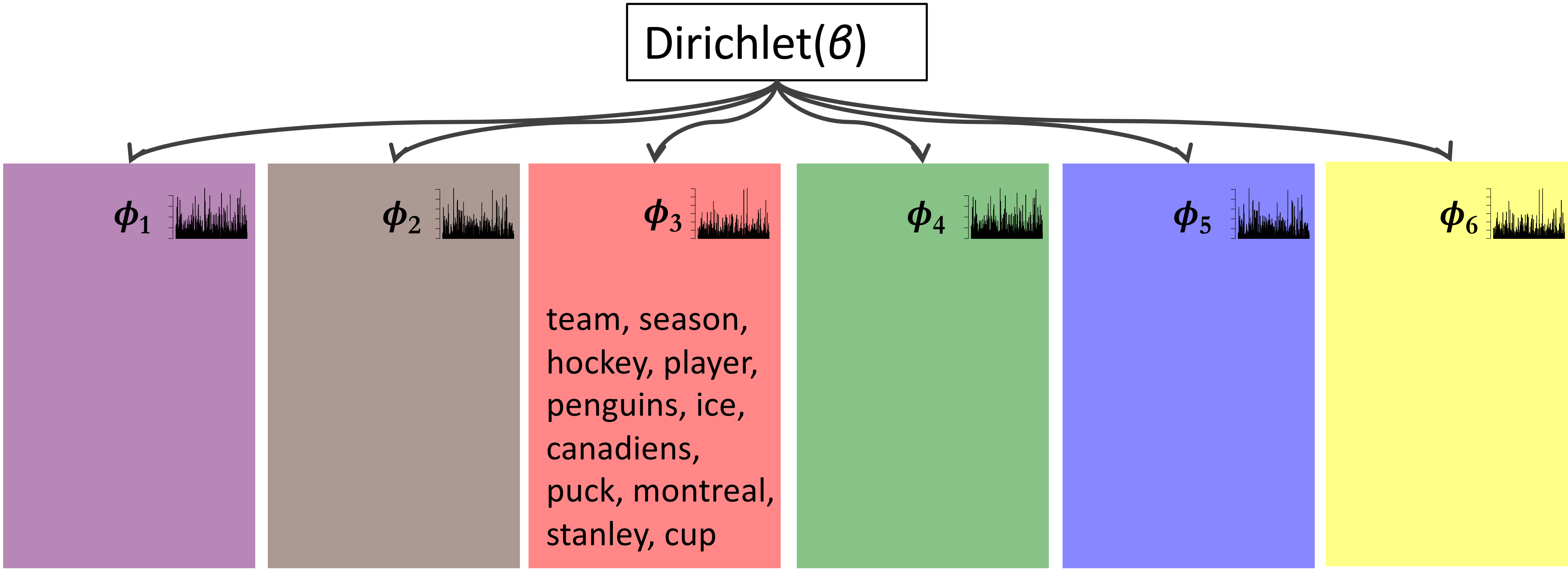
LDA for Topic Modeling



- The **generative story** begins with only a **Dirichlet prior** over the topics.
- Each **topic** is defined as a **Multinomial distribution** over the vocabulary, parameterized by ϕ_k

(Blei, Ng, & Jordan, 2003)

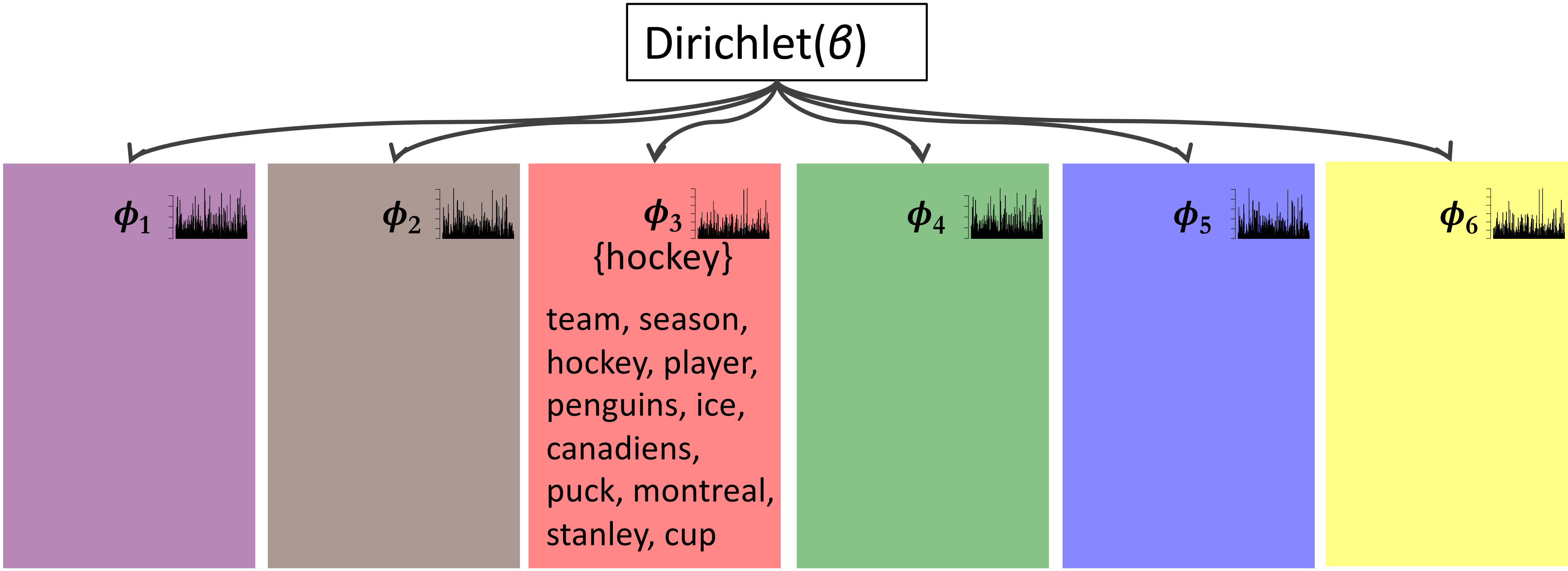
LDA for Topic Modeling



- A topic is visualized as its **high probability words.**

(Blei, Ng, & Jordan, 2003)

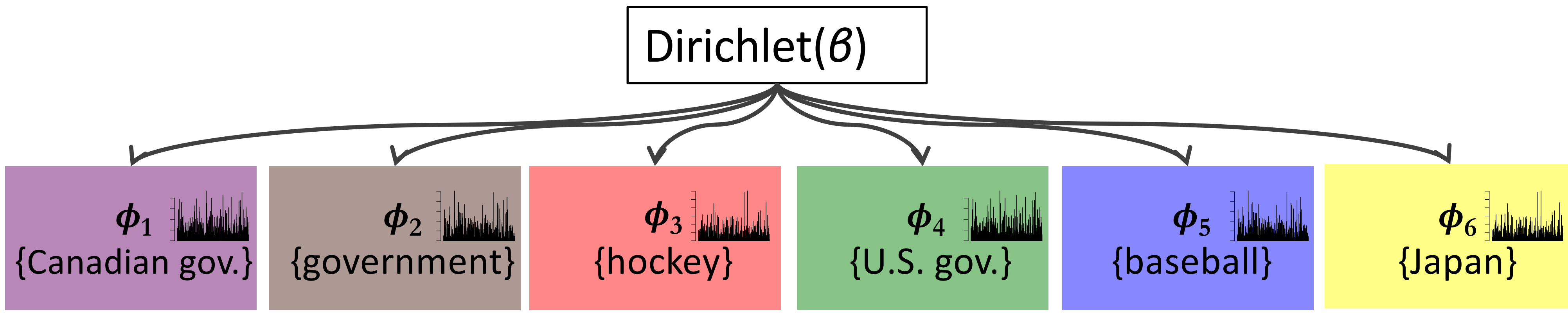
LDA for Topic Modeling



- A topic is visualized as its **high probability words.**
- A pedagogical **label** is used to identify the topic.

(Blei, Ng, & Jordan, 2003)

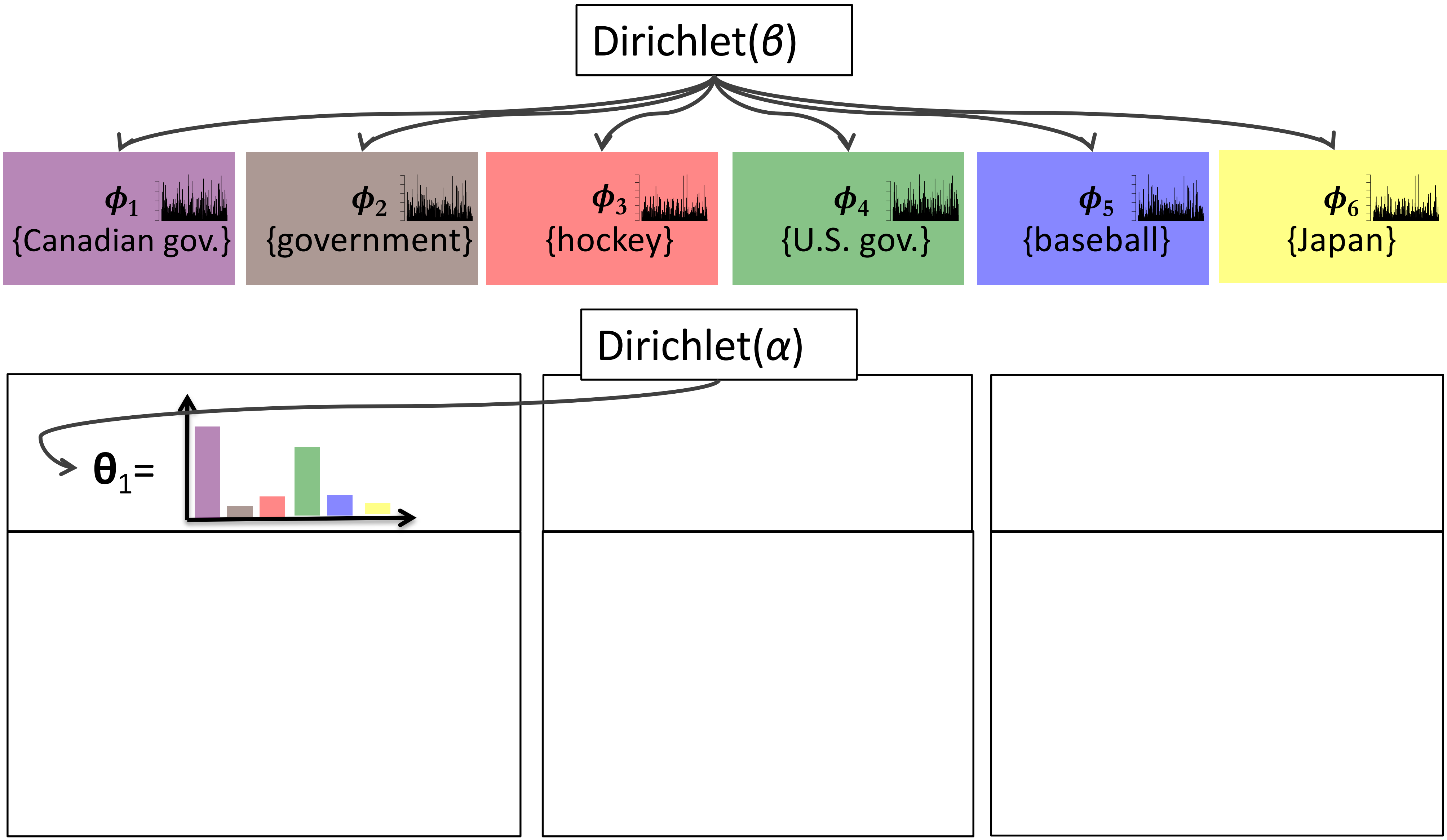
LDA for Topic Modeling



- A topic is visualized as its high probability words.
- A pedagogical **label** is used to identify the topic.

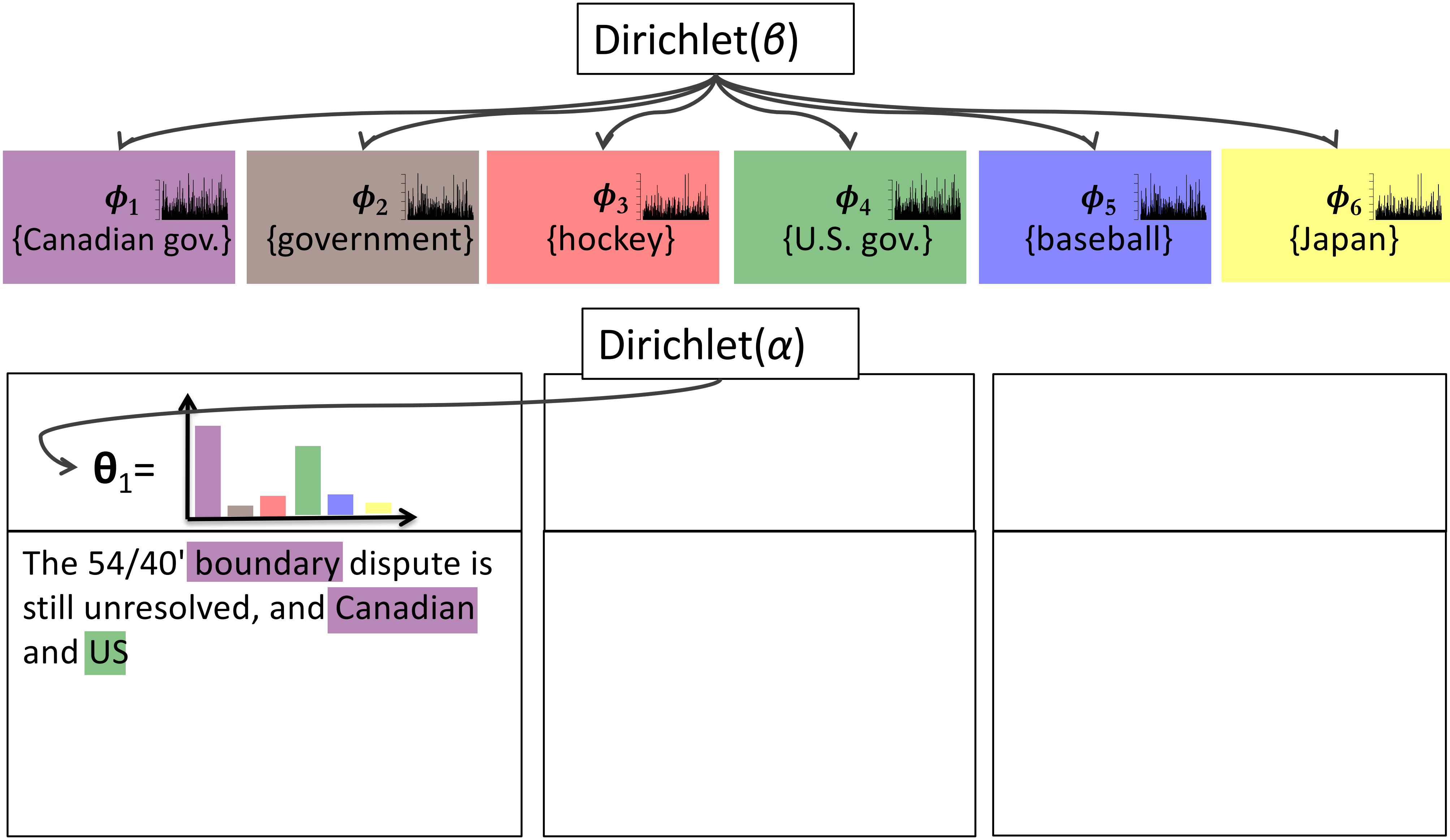
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



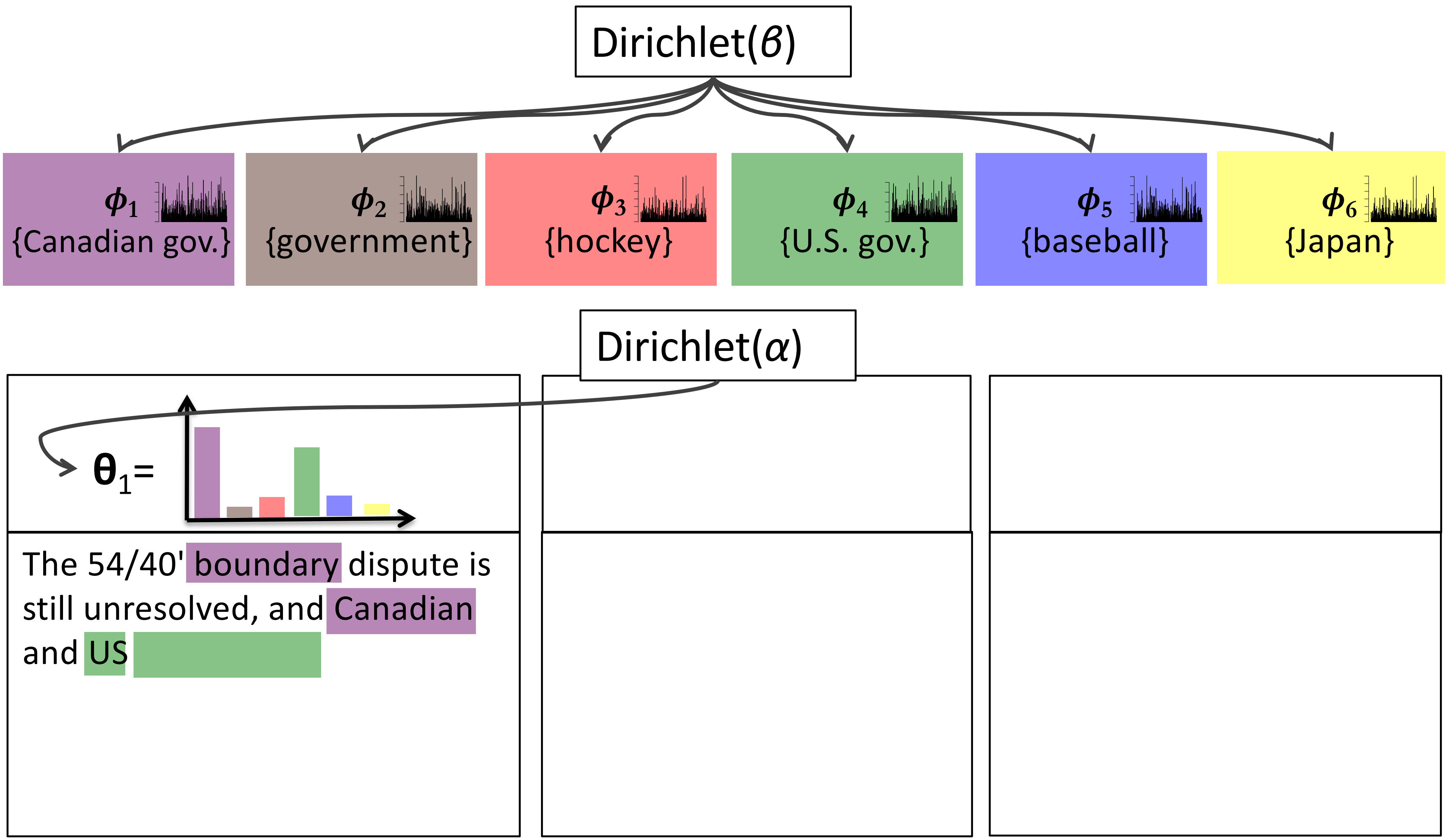
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



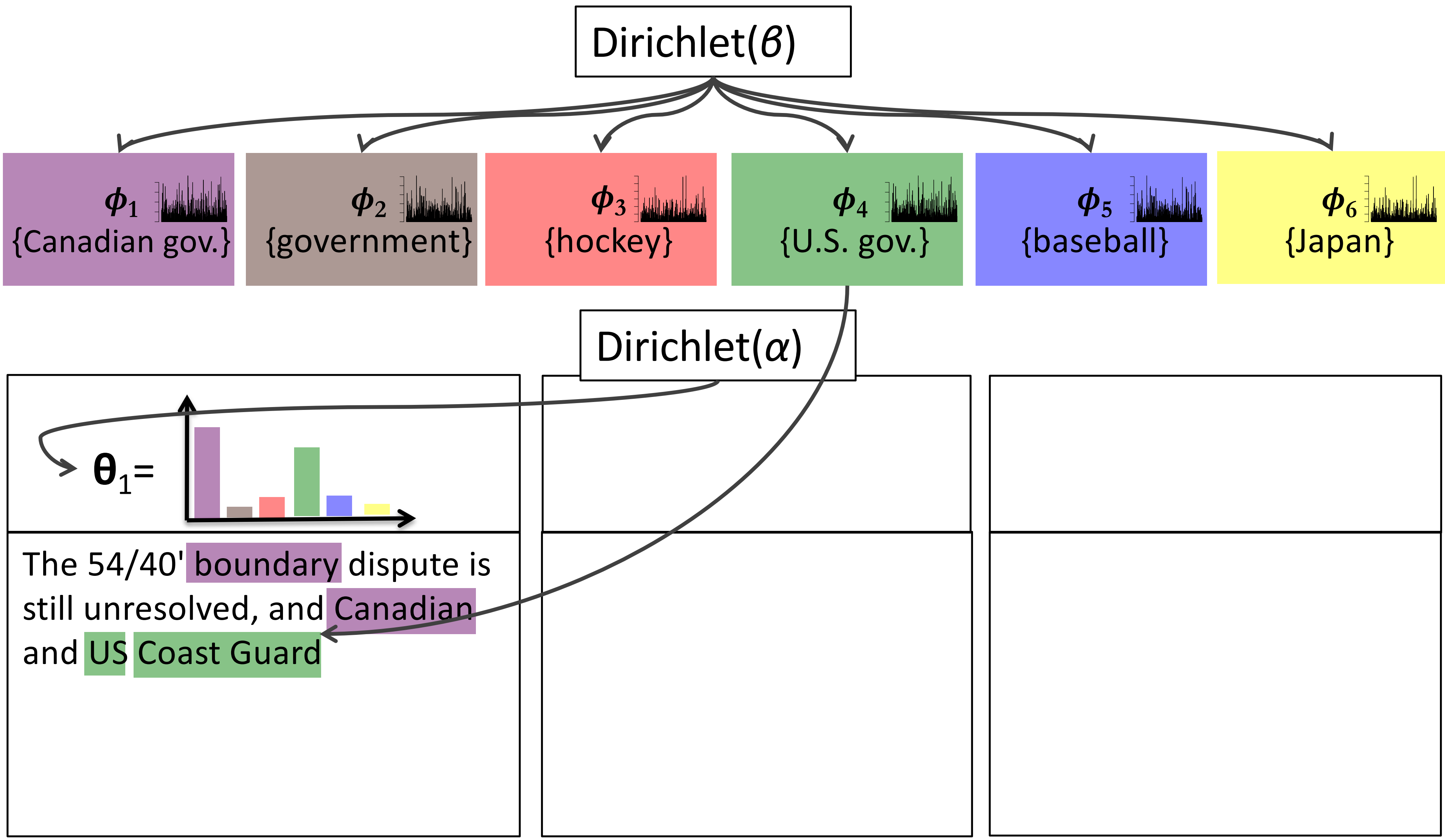
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



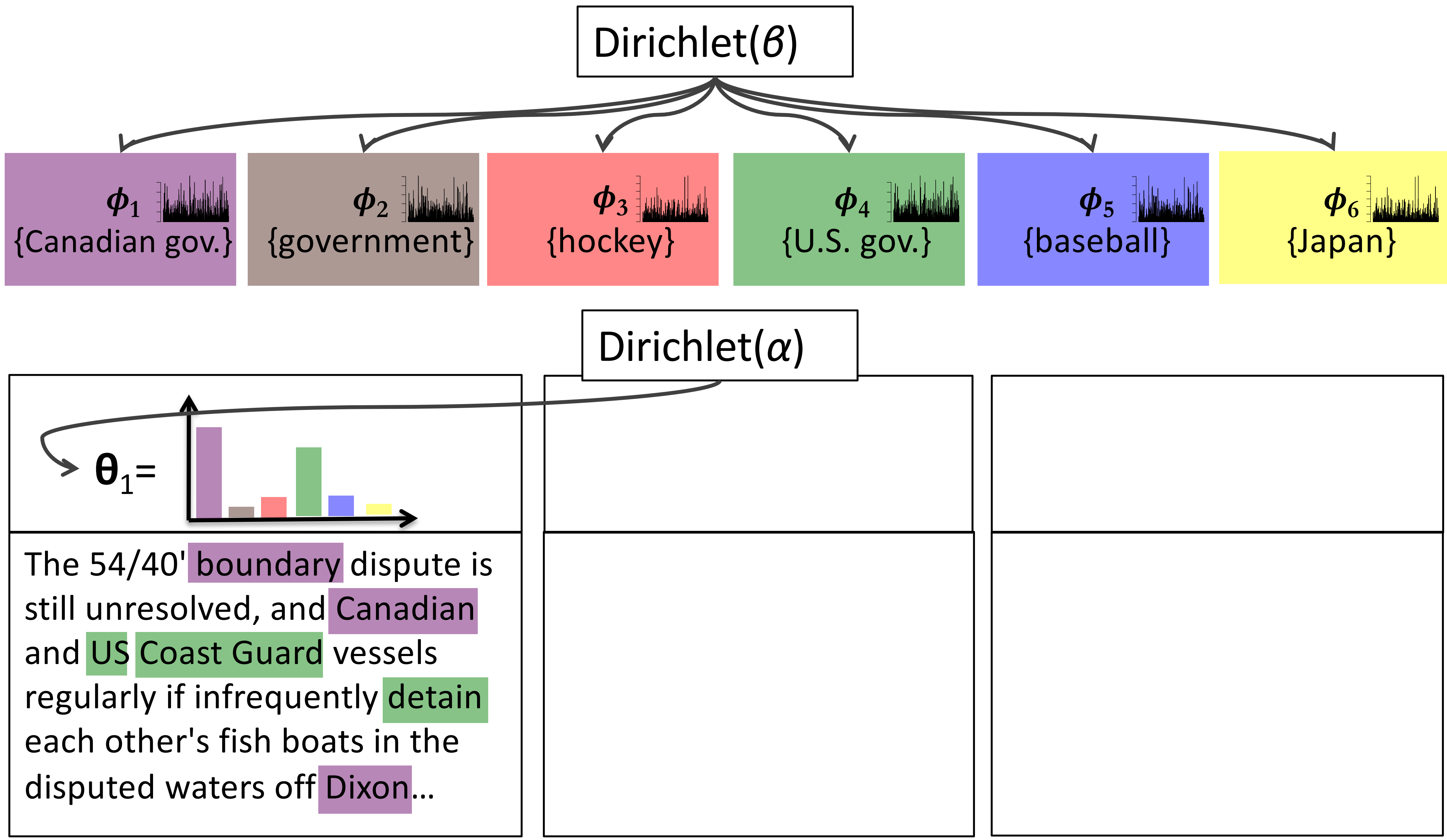
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



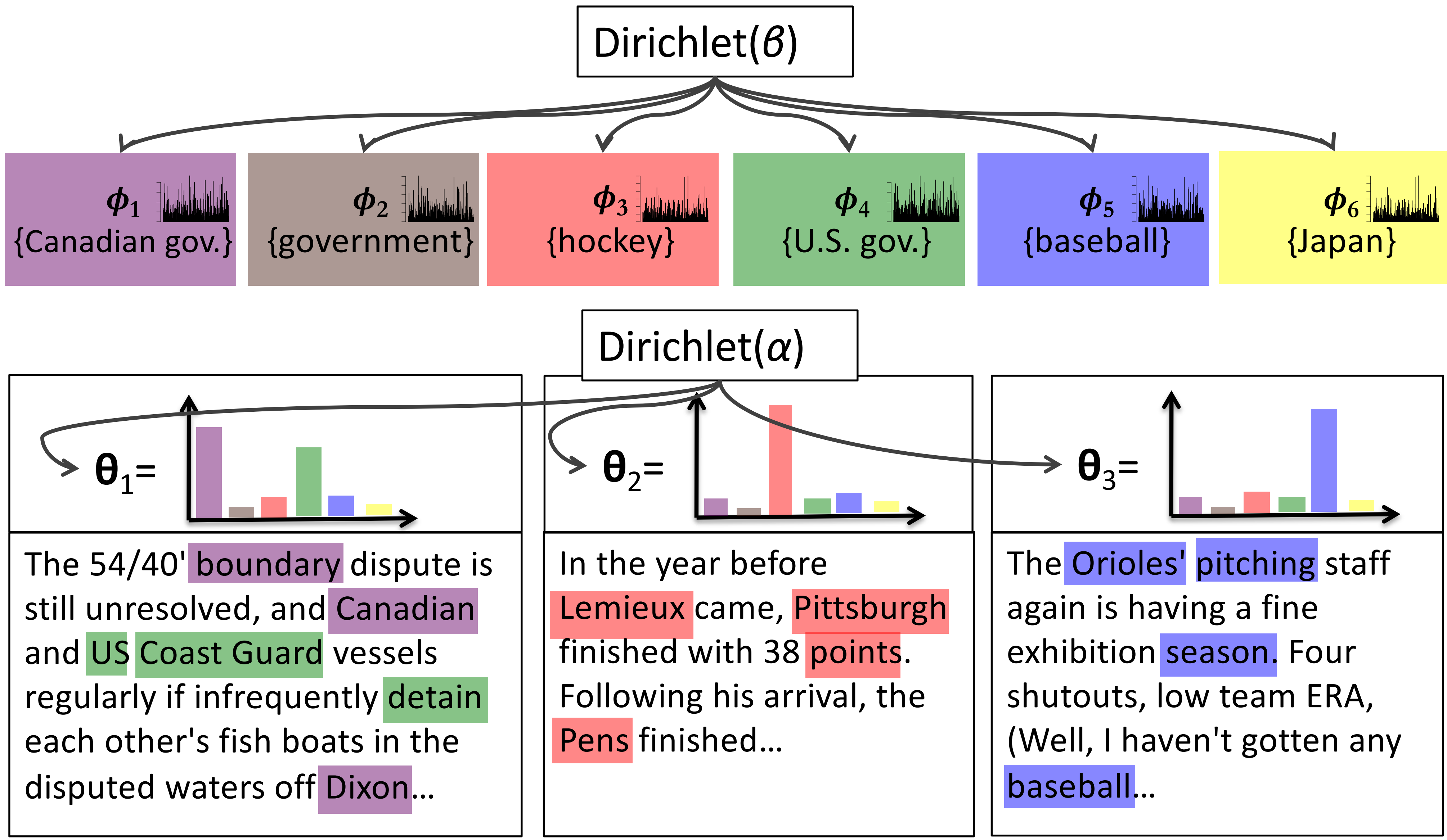
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



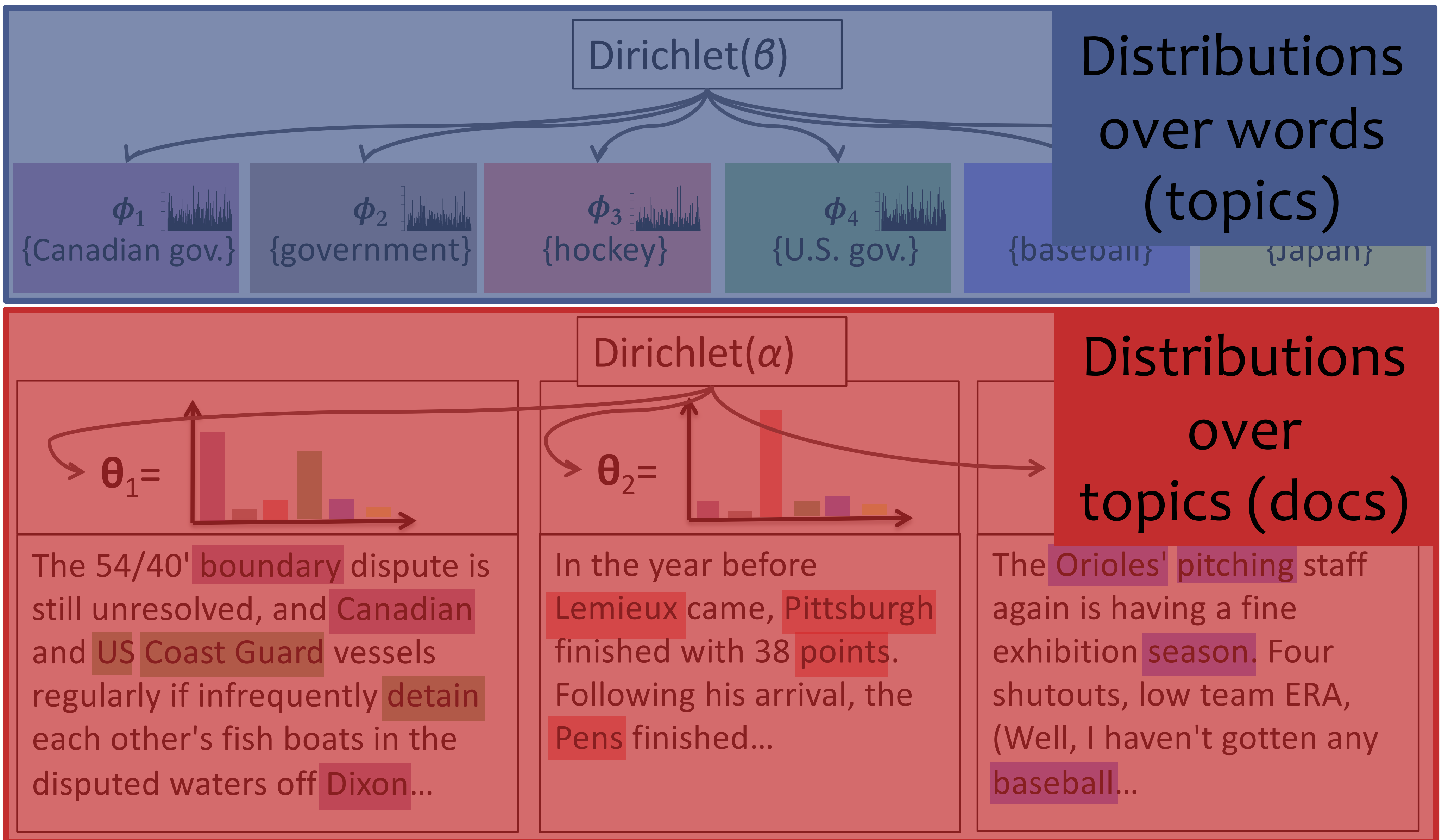
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



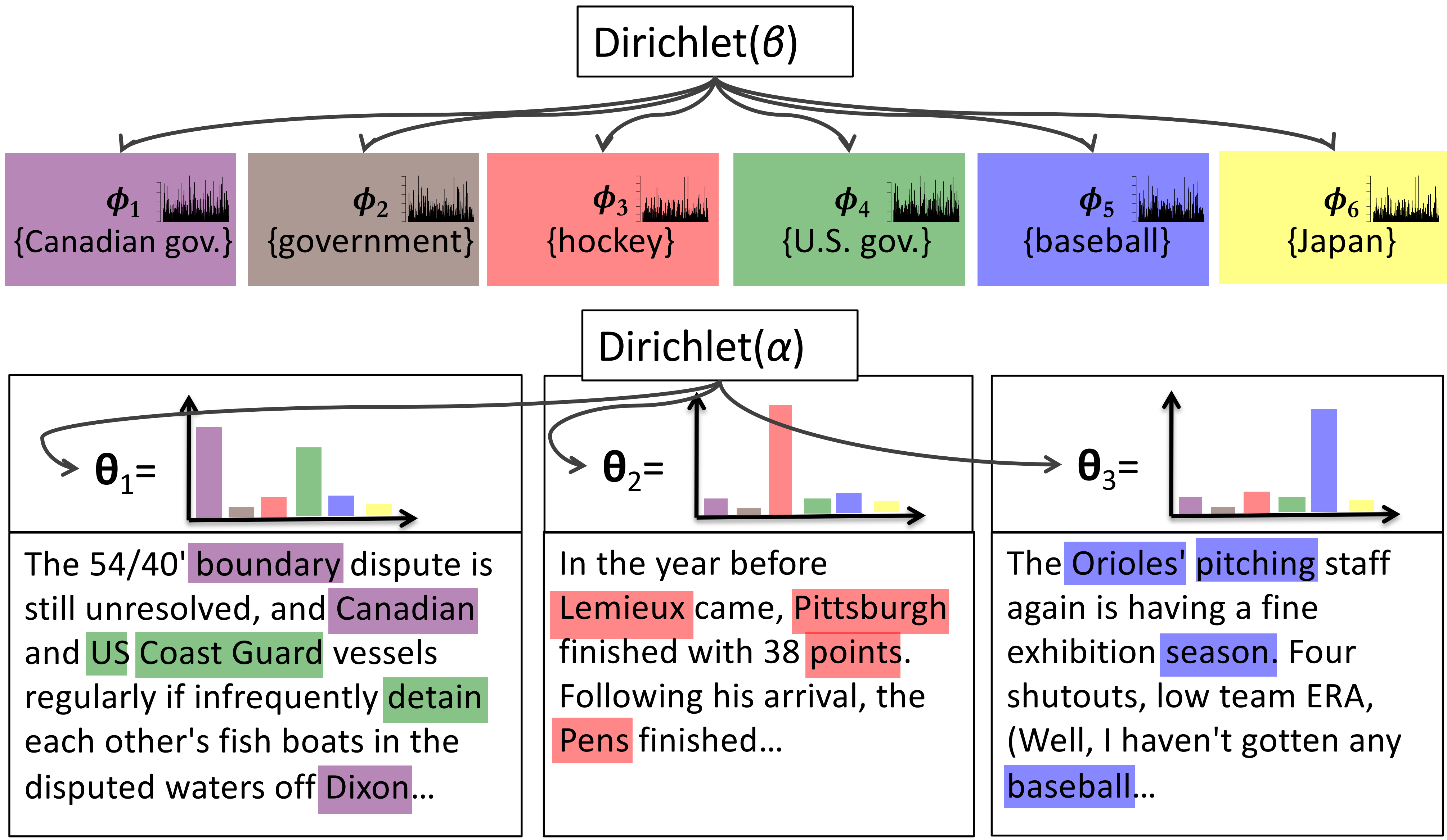
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling

Inference and learning start with only the data

Dirichlet( )

$\phi_1 =$ | $\phi_2 =$ | $\phi_3 =$ | $\phi_4 =$ | $\phi_5 =$ | $\phi_6 =$

Dirichlet( )

$\theta_1=$

The 54/40' boundary dispute is still unresolved, and Canadian and US Coast Guard vessels regularly if infrequently detain each other's fish boats in the disputed waters off Dixon…

$\theta_2=$

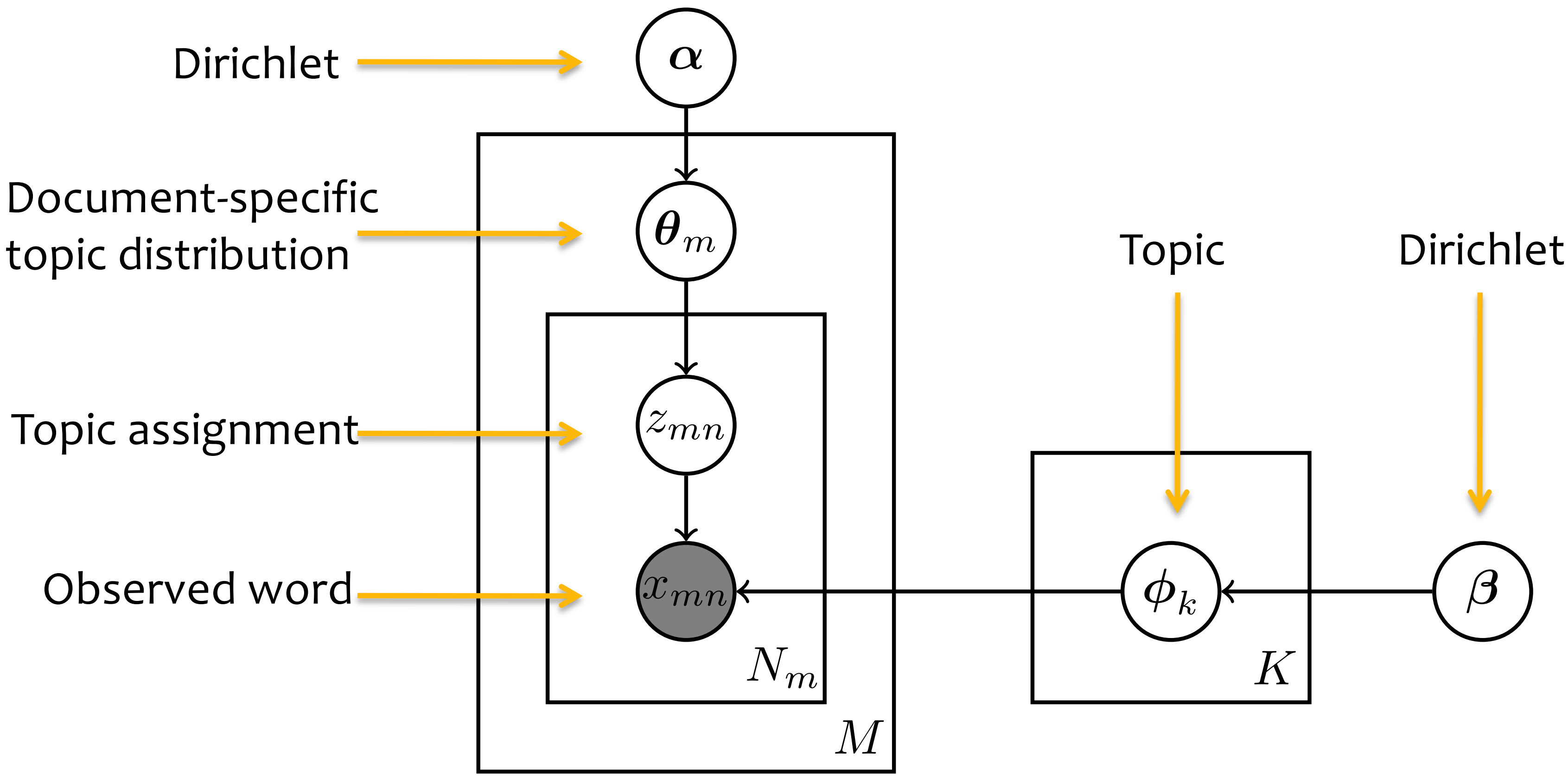
In the year before Lemieux came, Pittsburgh finished with 38 points. Following his arrival, the Pens finished…

$\theta_3=$

The Orioles' itching staff again is having a fine exhibition season. Four shutouts, low team ERA, (Well, I haven't gotten any baseball…

Latent Dirichlet Allocation

- Plate Diagram



Familiar models for unsupervised learning:

1. **K-Means**
2. **Gaussian Mixture Model (GMM)**
3. **Latent Dirichlet Allocation (LDA)**

But without labeled data, how do we know the right number of clusters / topics?

Outline

- **Motivation / Applications**
- **Background**
  - de Finetti Theorem
  - Exchangeability
  - Aglommerative and decimative properties of Dirichlet distribution
- **CRP and CRP Mixture Model**
  - Chinese Restaurant Process (CRP) definition
  - Gibbs sampling for CRP-MM
  - Expected number of clusters
- **DP and DP Mixture Model**
  - Ferguson definition of Dirichlet process (DP)
  - Stick breaking construction of DP
  - Uncollapsed blocked Gibbs sampler for DP-MM
  - Truncated variational inference for DP-MM
- **DP Properties**
- **Related Models**
  - Hierarchical Dirichlet process Mixture Models (HDP-MM)
  - Infinite HMM
  - Infinite PCFG

BAYESIAN NONPARAMETRICS

Parametric vs. Nonparametric

- **Parametric models:**
  - **Finite** and **fixed** number of parameters
  - Number of parameters is **independent of the dataset**
- **Nonparametric models:**
  - **Have** parameters (“**infinite dimensional**” would be a better name)
  - Can be understood as having an **infinite** number of parameters
  - Can be understood as having a **random** number of parameters
  - Number of parameters can **grow with the dataset**
- **Semiparametric models:**
  - Have a **parametric** component and a **nonparametric** component

Table adapted from Jordan ICML 2005

Parametric vs. Nonparametric

	Frequentist	Bayesian
Parametric	Logistic regression, ANOVA, Fisher discriminant analysis, ARMA, etc.	Conjugate analysis, hierarchical models, conditional random fields
Semiparametric	Independent component analysis, Cox model, nonmetric MDS, etc.	[Hybrids of the above and below cells]
Nonparametric	Nearest neighbor, kernel methods, boostrap, decision trees, etc.	Gaussian processes, Dirichlet processes, Pitman-Yor processes, etc.

Table adapted from Ghahramani 2015

Parametric vs. Nonparametric

Application	Parametric	Nonparametric
function approximation	polynomial regression	Gaussian processes
classification	logistic regression	Gaussian process classifiers
clustering	mixture model, k-means	Dirichlet process mixture model
time series	hidden Markov model	infinite HMM
feature discovery	factor analysis, pPCA, PMF	infinite latent factor models

Parametric vs. Nonparametric

- **Def**: a *model* is a collection of distributions

$$\{p_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$$

- *parametric model*: the parameter vector is finite dimensional

$$\Theta \subset \mathcal{R}^k$$

- *nonparametric model*: the parameters are from a possibly infinite dimensional space, $\mathcal{F}$

$$\Theta \subset \mathcal{F}$$

Motivation #1

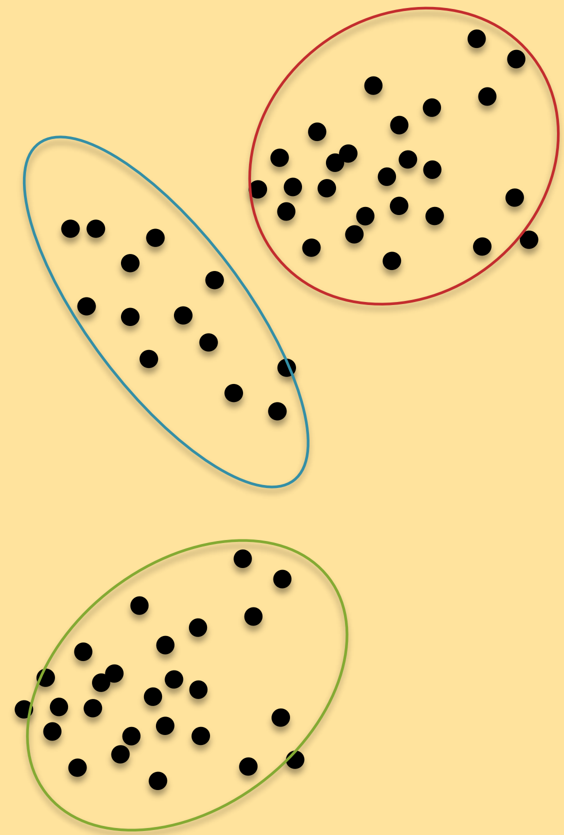
Model Selection

- For clustering:
  How many clusters in a **mixture model**?
- For topic modeling:
  How many topics in **LDA**?
- For grammar induction:
  How many non-terminals in a **PCFG**?
- For visual scene analysis:
  How many objects, parts, features?

Motivation #1

Model Selection

- For clustering:
  How many clusters in a **mixture model**?
- For topic modeling:
  How many topics in **LDA**?
- For grammar induction:
  How many non-terminals in a **PCFG**?
- For visual scene analysis:
  How many objects, parts, features?



Motivation #1

Model Selection

- For clustering:
  How many clusters in a **mixture model**?
- For topic modeling:
  How many topics in **LDA**?
- For grammar induction:
  How many non-terminals in a **PCFG**?
- For visual scene analysis:
  How many objects, parts, features?

1. **Parametric approaches:** cross-validation, bootstrap, AIC, BIC, DIC, MDL, Laplace, bridge sampling, etc.
2. **Nonparametric approach:** average of an infinite set of models

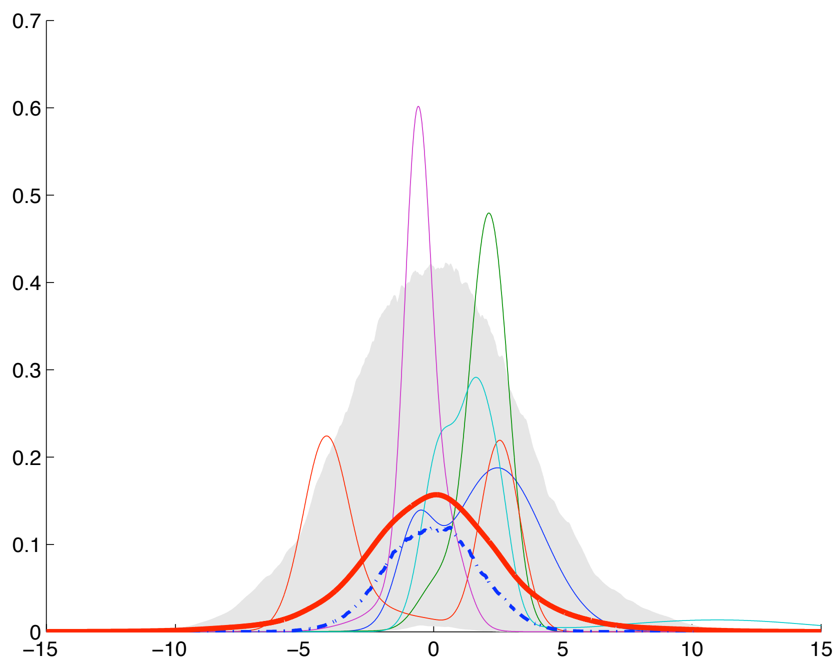
Figure from Teh MLSS 2007

Motivation #2

Density Estimation

- Given data, estimate a probability density function that best explains it
- A nonparametric prior can be placed over an infinite set of distributions

Prior:



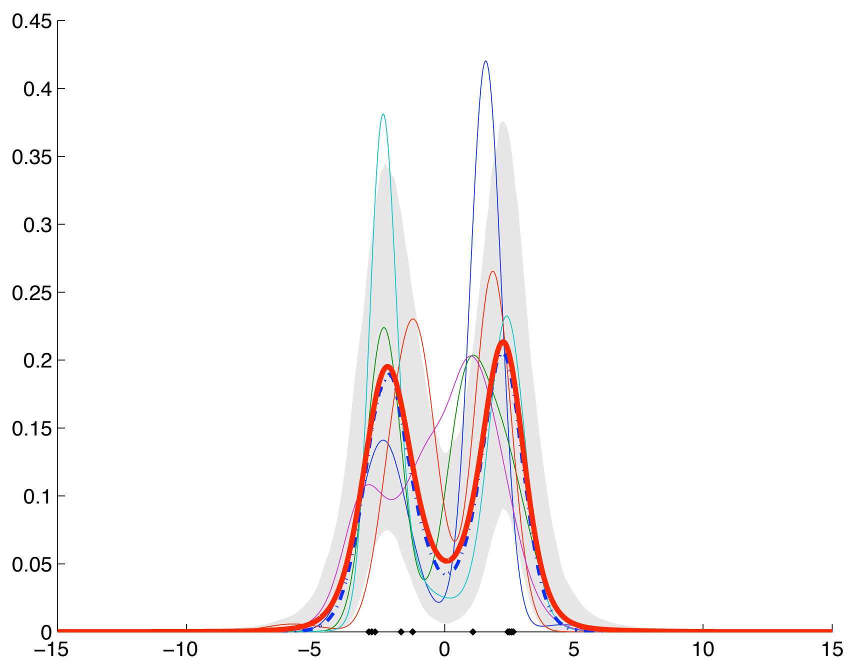
Red: mean density. Blue: median density. Grey: 5-95 quantile.
Others: draws.

Motivation #2

Density Estimation

- Given data, estimate a probability density function that best explains it
- A nonparametric prior can be placed over an infinite set of distributions

Posterior:



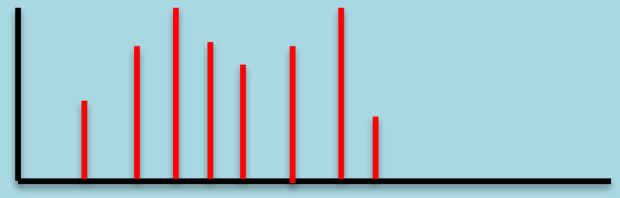
Red: mean density. Blue: median density. Grey: 5-95 quantile.
Black: data. Others: draws.

EXCHANGEABILITY AND DE FINETTI'S THEOREM

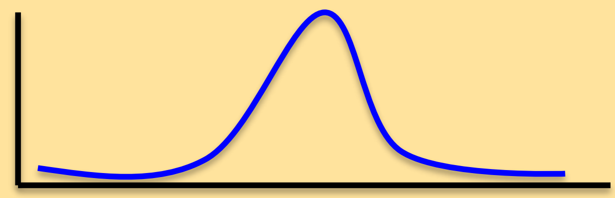
Background

Suppose we have a random variable X drawn from some distribution $P_\theta(X)$ and X ranges over a set $\mathcal{S}$.

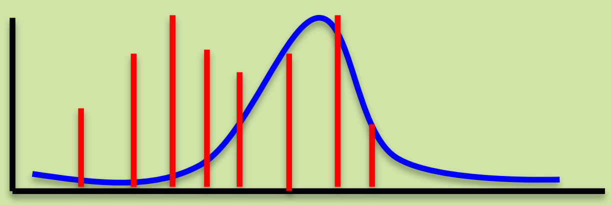
- Discrete distribution:
  $\mathcal{S}$ is a countable set.



- Continuous distribution:
  $P_\theta(X = x) = 0$ for all $x \in \mathcal{S}$



- Mixed distribution:
  $\mathcal{S}$ can be partitioned into two disjoint sets $\mathcal{D}$ and $\mathcal{C}$ s.t.
  1. $\mathcal{D}$ is countable and $0 < P_\theta(X \in D) < 1$
  2. $P_\theta(X = x) = 0$ for all $x \in \mathcal{C}$



Background

Whiteboard

- Mixed distribution

Exchangability and de Finetti's Theorem

Exchangeability:

- **Def #1**: a joint probability distribution is **exchangeable** if it is invariant to permutation
- **Def #2:** The possibly infinite sequence of random variables $(X_1, X_2, X_3, \ldots)$ is **exchangeable** if for any finite permutation s of the indices $(1, 2, \ldots n)$:

$$P(X_1, X_2, \ldots, X_n) = P(X_{s(1)}, X_{s(2)}, \ldots, X_{s(n)})$$

Notes:

- *i.i.d.* and *exchangeable* are not the same!
- the latter says that if our data are reordered it doesn't matter

Exchangability and de Finetti's Theorem

Slide from Jordan ICML 2005

Theorem (De Finetti, 1935). *If $(x_1, x_2, \ldots)$ are infinitely exchangeable, then the joint probability $p(x_1, x_2, \ldots, x_N)$ has a representation as a mixture:*

$$p(x_1, x_2, \ldots, x_N) = \int \left(\prod_{i=1}^{N} p(x_i \mid \theta) \right) dP(\theta)$$

for some random variable θ.

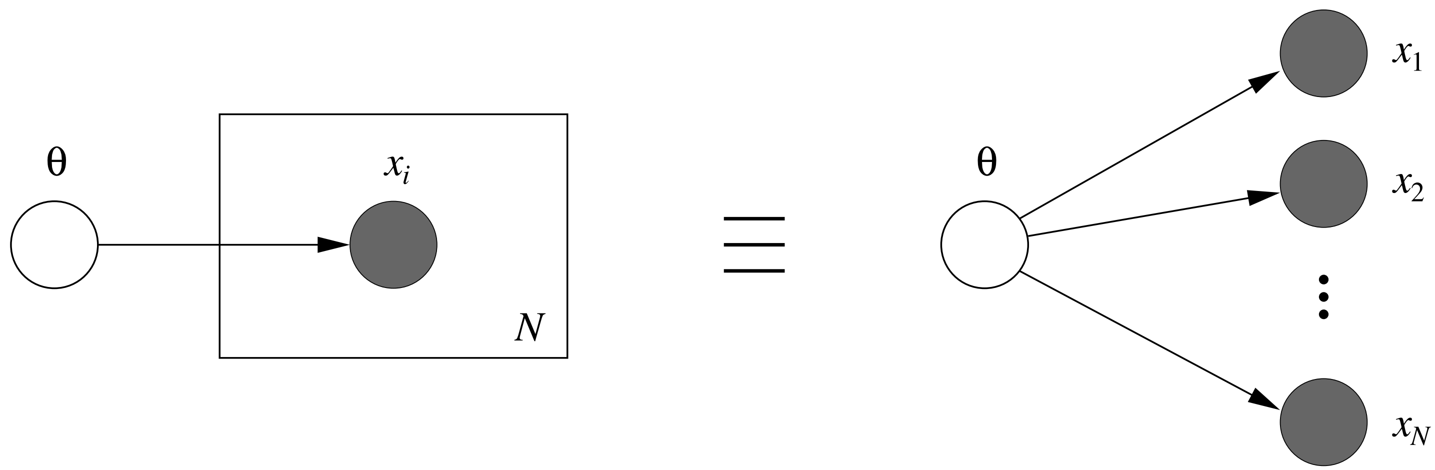
- The theorem wouldn't be true if we limited ourselves to parameters θ ranging over Euclidean vector spaces
- In particular, we need to allow θ to range over measures, in which case $P(\theta)$ is a measure on measures
  - the Dirichlet process is an example of a measure on measures...

Actually, this is the Hewitt-Savage generalization of the de Finetti theorem. The original version was given for the Bernoulli distribution

Exchangability and de Finetti's Theorem

Slide from Jordan ICML 2005

- A *plate* is a "macro" that allows subgraphs to be replicated:



- Note that this is a graphical representation of the De Finetti theorem

$$p(x_1, x_2, \ldots, x_N) = \int p(\theta) \left(\prod_{i=1}^{N} p(x_i \,|\, \theta) \right) d\theta$$

Parametric vs. Nonparametric

Type of Model	Parametric Example	Nonparametric Example	
		Construction #1	Construction #2
distribution over counts	Dirichlet-Multinomial Model	Dirichlet Process (DP)	
		Chinese Restaurant Process (CRP)	Stick-breaking construction
mixture	Gaussian Mixture Model (GMM)	Dirichlet Process Mixture Model (DPMM)	
		CRP Mixture Model	Stick-breaking construction
admixture	Latent Dirichlet Allocation (LDA)	Hierarchical Dirichlet Process Mixture Model (HDPMM)	
		Chinese Restaurant Franchise	Stick-breaking construction

Chinese Restaurant Process & Stick-breaking Constructions

DIRICHLET PROCESS

Dirichlet Process

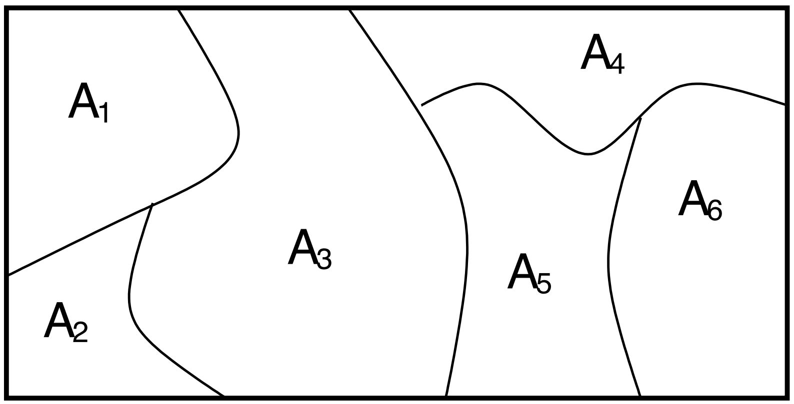
Ferguson Definition

- Parameters of a DP:
  1. Base distribution, H, is a probability distribution over Θ
  2. Strength parameter, $\alpha \in \mathcal{R}$
- We say $G \sim \mathrm{DP}(\alpha, H)$
  if for any partition $A_1 \cup A_2 \cup \ldots \cup A_K = \Theta$
  we have:

$$(G(A_1), \ldots, G(A_K)) \sim \mathrm{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_K))$$

In English: the DP is a distribution over probability measures s.t. marginals on finite partitions are Dirichlet distributed

A partition of the space Θ

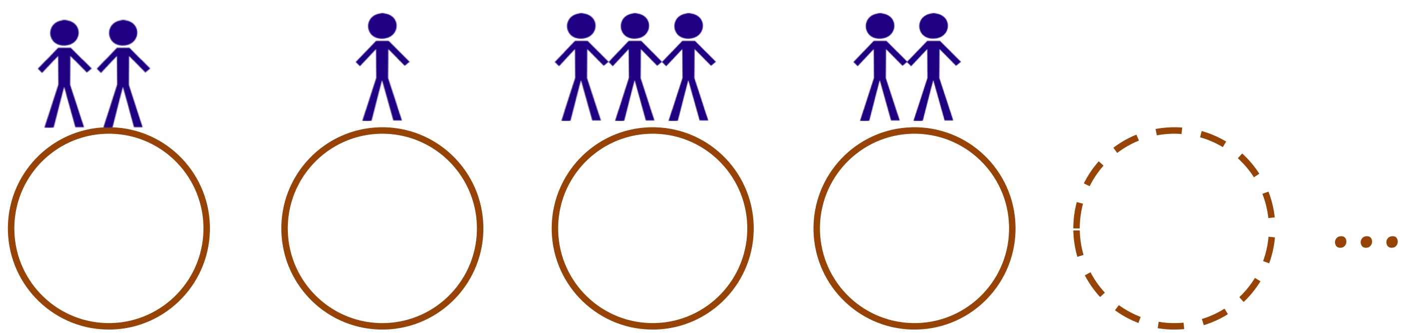


Chinese Restaurant Process

- Imagine a Chinese restaurant with an infinite number of tables
- Each customer enters and sits down at a table
  - The first customer sits at the first unoccupied table
  - Each subsequent customer chooses a table according to the following probability distribution:

    $p(\textit{kth occupied table}) \propto n_k$
    $p(\textit{next unoccupied table}) \propto \alpha$

    where n_k is the number of people sitting at the table k



$$\frac{2}{8+\alpha} \quad \frac{1}{8+\alpha} \quad \frac{3}{8+\alpha} \quad \frac{2}{8+\alpha} \quad \frac{\alpha}{8+\alpha}$$

Chinese Restaurant Process

Properties:

1. CRP defines a **distribution over clusterings** (i.e. partitions) of the indices $1,\ldots,n$
   - customer = index
   - table = cluster
2. We write $z_1, z_2, \ldots, z_n \sim CRP(\alpha)$ to denote a **sequence of cluster indices** drawn from a Chinese Restaurant Process
3. The CRP is an **exchangeable process**
4. **Expected number of clusters** given n customers (i.e. observations) is $O(\alpha \log(n))$
   - *rich-get-richer effect* on clusters: popular tables tend to get more crowded
5. Behavior of CRP with α:
   - As α goes to 0, the number of clusters goes to 1
   - As α goes to $+\infty$, the number of clusters goes to n

Dirichlet Process

Whiteboard

– Stick-breaking construction of the DP

CRP vs. DP

Dirichlet Process: For both the CRP and stick-breaking constructions, if we marginalize out G, we have the following predictive distribution:

$$\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{1}{\alpha + n}\left(\alpha H + \sum_{i=1}^{n} \delta_{\theta_i}\right)$$

(Blackwell-MacQueen Urn Scheme)

The Chinese Restaurant Process is just a different construction of the Dirichlet Process where we have marginalized out G

Dirichlet Process

Whiteboard

- Dirichlet Process
  (Polya urn scheme version)

Properties of the DP

1. Base distribution is the "mean" of the DP:

$$\mathbb{E}[G(A)] = H(A) \text{ for any } A \subset \Theta$$

2. Strength parameter is like "inverse variance"

$$V[G(A)] = H(A)(1 - H(A))/(\alpha + 1)$$

3. Samples from a DP are discrete distributions
   (stick-breaking construction of $G \sim \mathrm{DP}(\alpha, H)$
   makes this clear)

4. Posterior distribution of $G \sim \mathrm{DP}(\alpha, H)$
   given samples $\theta_1, \ldots, \theta_n$ from G is a DP

$$G|\theta_1, \ldots, \theta_n \sim \mathrm{DP}\left(\alpha + n, \frac{\alpha}{\alpha+n} H + \frac{n}{\alpha+n} \frac{\sum_{i=1}^{n} \delta_{\theta_i}}{n}\right)$$