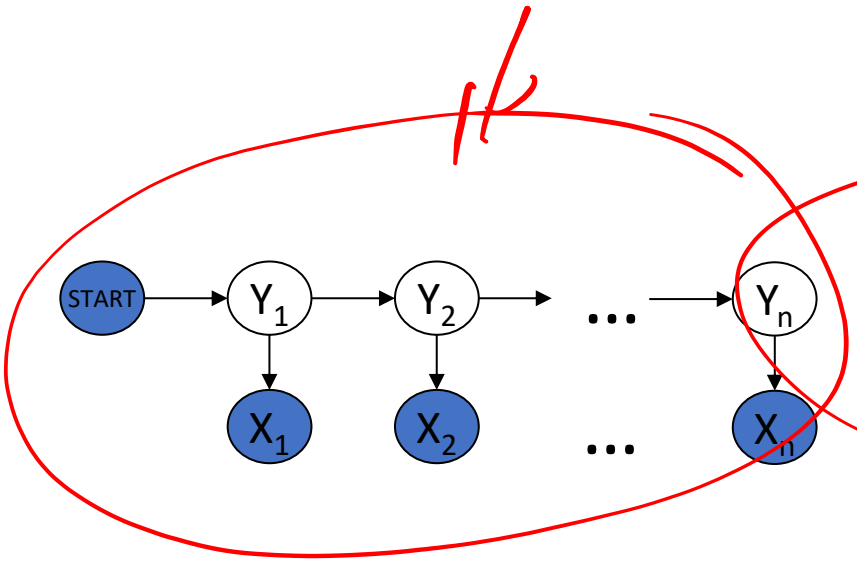
CRF (cont'd)
+
Intro to Topic Modeling

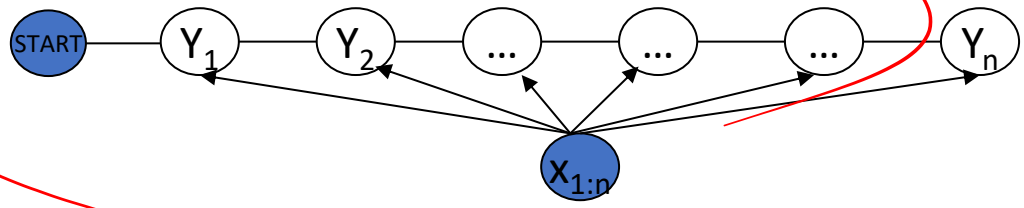
Kayhan Batmanghelich

Slides Credit:

Matt Gormley (2016)

Review: Generative vs Discriminative

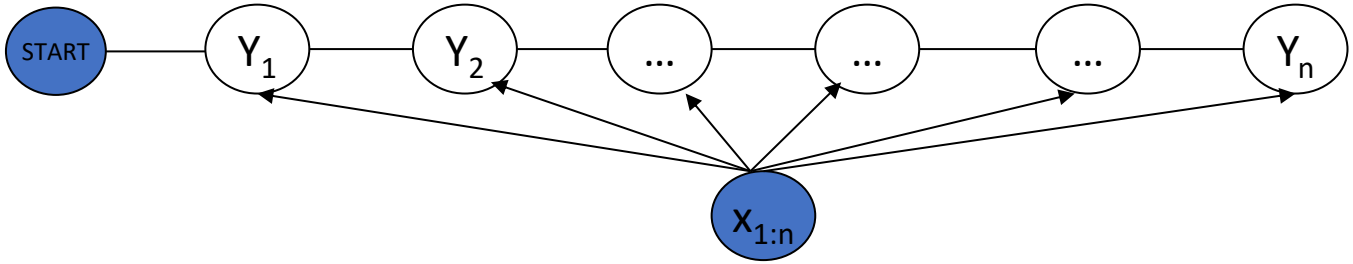




$$P(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) = \prod_{i=1}^{n} P(x_i|y_i)P(y_i|y_{i-1})$$

$$P(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}) = \frac{1}{Z(\mathbf{x}_{1:n})} \prod_{i=1}^{n} \phi(y_i, y_{i-1}, \mathbf{x}_{1:n}) = \frac{1}{Z(\mathbf{x}_{1:n}, \mathbf{w})} \prod_{i=1}^{n} \exp(\mathbf{w}^T \mathbf{f}(y_i, y_{i-1}, \mathbf{x}_{1:n}))$$

Review: Conditional Random Field



$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x},\lambda,\mu)} \exp\Big(\sum_{i=1}^{n}\Big(\sum_{k}\lambda_k f_k(y_i, y_{i-1}, \mathbf{x}) + \sum_{l}\mu_l g_l(y_i, \mathbf{x})\Big)\Big)$$

$$= \frac{1}{Z(\mathbf{x},\lambda,\mu)} \exp\Big(\sum_{i=1}^{n}\big(\lambda^T \mathbf{f}(y_i, y_{i-1}, \mathbf{x}) + \mu^T \mathbf{g}(y_i, \mathbf{x})\big)\Big)$$

$$\text{where } Z(\mathbf{x},\lambda,\mu) = \sum_{\mathbf{y}} \exp\Big(\sum_{i=1}^{n}\big(\lambda^T \mathbf{f}(y_i, y_{i-1}, \mathbf{x}) + \mu^T \mathbf{g}(y_i, \mathbf{x})\big)\Big)$$

When can I ignore $Z(\boldsymbol{x},\lambda,\mu)$:

- Computing $\arg\max_{\mathbf{y}} P(\boldsymbol{y}|\boldsymbol{x};\lambda,\mu)$?
- Computing $\max_{\lambda,\mu} \log P(\boldsymbol{y}|\boldsymbol{x};\lambda,\mu)$?

CRF learning

- Given $\{(\mathbf{x}_d, \mathbf{y}_d)\}_{d=1}^N$, find λ^*, μ^* such that

$$\lambda*, \mu* = \arg\max_{\lambda,\mu} L(\lambda,\mu) = \arg\max_{\lambda,\mu} \prod_{d=1}^{N} P(\mathbf{y}_d|\mathbf{x}_d,\lambda,\mu)$$

$$= \arg\max_{\lambda,\mu} \prod_{d=1}^{N} \frac{1}{Z(\mathbf{x}_d,\lambda,\mu)} \exp\Big(\sum_{i=1}^{n}(\lambda^T \mathbf{f}(y_{d,i}, y_{d,i-1}, \mathbf{x}_d) + \mu^T \mathbf{g}(y_{d,i}, \mathbf{x}_d))\Big)$$

$$= \arg\max_{\lambda,\mu} \sum_{d=1}^{N}\Big(\sum_{i=1}^{n}(\lambda^T \mathbf{f}(y_{d,i}, y_{d,i-1}, \mathbf{x}_d) + \mu^T \mathbf{g}(y_{d,i}, \mathbf{x}_d)) - \log Z(\mathbf{x}_d,\lambda,\mu)\Big)$$

- Computing the gradient w.r.t λ:

$$\nabla_\lambda L(\lambda,\mu) = \sum_{d=1}^{N}\Big(\sum_{i=1}^{n} \mathbf{f}(y_{d,i}, y_{d,i-1}, \mathbf{x}_d) - \sum_{\mathbf{y}}\Big(P(\mathbf{y}|\mathbf{x_d}) \sum_{i=1}^{n} \mathbf{f}(y_{d,i}, y_{d,i-1}, \mathbf{x}_d)\Big)\Big)$$

Gradient of the log-partition function in an exponential family is the expectation of the sufficient statistics.

CRF learning

$$\nabla_\lambda L(\lambda,\mu) = \sum_{d=1}^{N}\left(\sum_{i=1}^{n}\mathbf{f}(y_{d,i},y_{d,i-1},\mathbf{x}_d) - \sum_{\mathbf{y}}\left(P(\mathbf{y}|\mathbf{x_d})\sum_{i=1}^{n}\mathbf{f}(y_i,y_{i-1},\mathbf{x}_d)\right)\right)$$

- Computing the model expectations:
  - Requires exponentially large number of summations: Is it intractable?

$$\sum_{\mathbf{y}}\left(P(\mathbf{y}|\mathbf{x}_d)\sum_{i=1}^{n}\mathbf{f}(y_i,y_{i-1},\mathbf{x}_d)\right) = \sum_{i=1}^{n}\left(\sum_{\mathbf{y}}\mathbf{f}(y_i,y_{i-1},\mathbf{x}_d)P(\mathbf{y}|\mathbf{x}_d)\right)$$

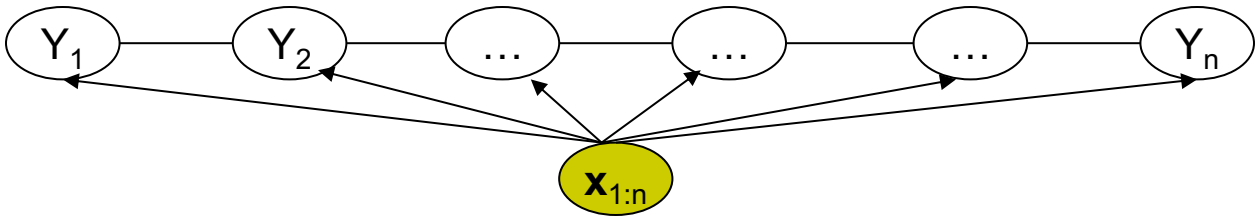
$$= \sum_{i=1}^{n}\sum_{y_i,y_{i-1}}\mathbf{f}(y_i,y_{i-1},\mathbf{x}_d)P(y_i,y_{i-1}|\mathbf{x}_d)$$

Expectation of **f** over the corresponding marginal probability of neighboring nodes!!

- Tractable!
  - Can compute marginals using the sum-product algorithm on the chain

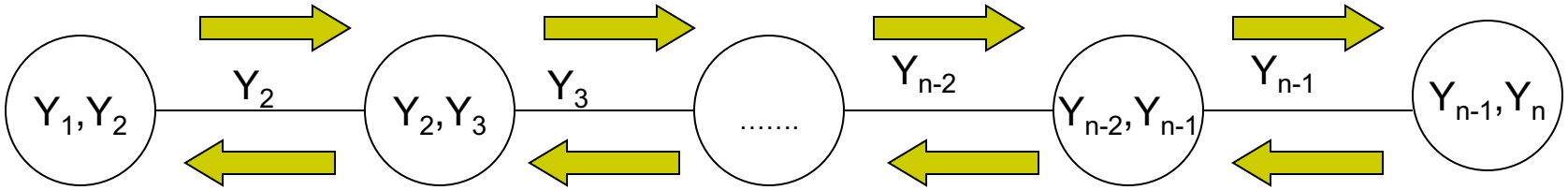
CRF learning

- Computing marginals using junction-tree calibration:



- Junction Tree Initialization:

$$\alpha^0(y_i, y_{i-1}) = \exp(\lambda^T \mathbf{f}(y_i, y_{i-1}, \mathbf{x}_d) + \mu^T \mathbf{g}(y_i, \mathbf{x}_d))$$



- After calibration:

$$P(y_i, y_{i-1}|\mathbf{x}_d) \propto \alpha(y_i, y_{i-1})$$

$$\Rightarrow P(y_i, y_{i-1}|\mathbf{x}_d) = \frac{\alpha(y_i, y_{i-1})}{\sum_{y_i, y_{i-1}} \alpha(y_i, y_{i-1})} = \alpha'(y_i, y_{i-1})$$

Also called forward-backward algorithm

CRF learning

- Computing feature expectations using calibrated potentials:

$$\sum_{y_i, y_{i-1}} \mathbf{f}(y_i, y_{i-1}, \mathbf{x}_d) P(y_i, y_{i-1} | \mathbf{x}_d) = \sum_{y_i, y_{i-1}} \mathbf{f}(y_i, y_{i-1}, \mathbf{x}_d) \alpha'(y_i, y_{i-1})$$

- Now we know how to compute $\nabla_\lambda L(\lambda,\mu)$:

$$\begin{aligned}
\nabla_\lambda L(\lambda, \mu) &= \sum_{d=1}^{N} \left(\sum_{i=1}^{n} \mathbf{f}(y_{d,i}, y_{d,i-1}, \mathbf{x}_d) - \sum_{\mathbf{y}} \left(P(\mathbf{y} | \mathbf{x_d}) \sum_{i=1}^{n} \mathbf{f}(y_i, y_{i-1}, \mathbf{x}_d) \right) \right) \\
&= \sum_{d=1}^{N} \left(\sum_{i=1}^{n} \left(\mathbf{f}(y_{d,i}, y_{d,i-1}, \mathbf{x}_d) - \sum_{y_i, y_{i-1}} \alpha'(y_i, y_{i-1}) \mathbf{f}(y_i, y_{i-1}, \mathbf{x}_d) \right) \right)
\end{aligned}$$

- Learning can now be done using gradient ascent:

$$\begin{aligned}
\lambda^{(t+1)} &= \lambda^{(t)} + \eta \nabla_\lambda L(\lambda^{(t)}, \mu^{(t)}) \\
\mu^{(t+1)} &= \mu^{(t)} + \eta \nabla_\mu L(\lambda^{(t)}, \mu^{(t)})
\end{aligned}$$

CRF learning

- In practice, we use a Gaussian Regularizer for the parameter vector to improve generalizability

$$\lambda*, \mu* = \arg\max_{\lambda,\mu} \sum_{d=1}^{N} \log P(\mathbf{y}_d|\mathbf{x}_d, \lambda, \mu) - \frac{1}{2\sigma^2}(\lambda^T\lambda + \mu^T\mu)$$

ML Reg

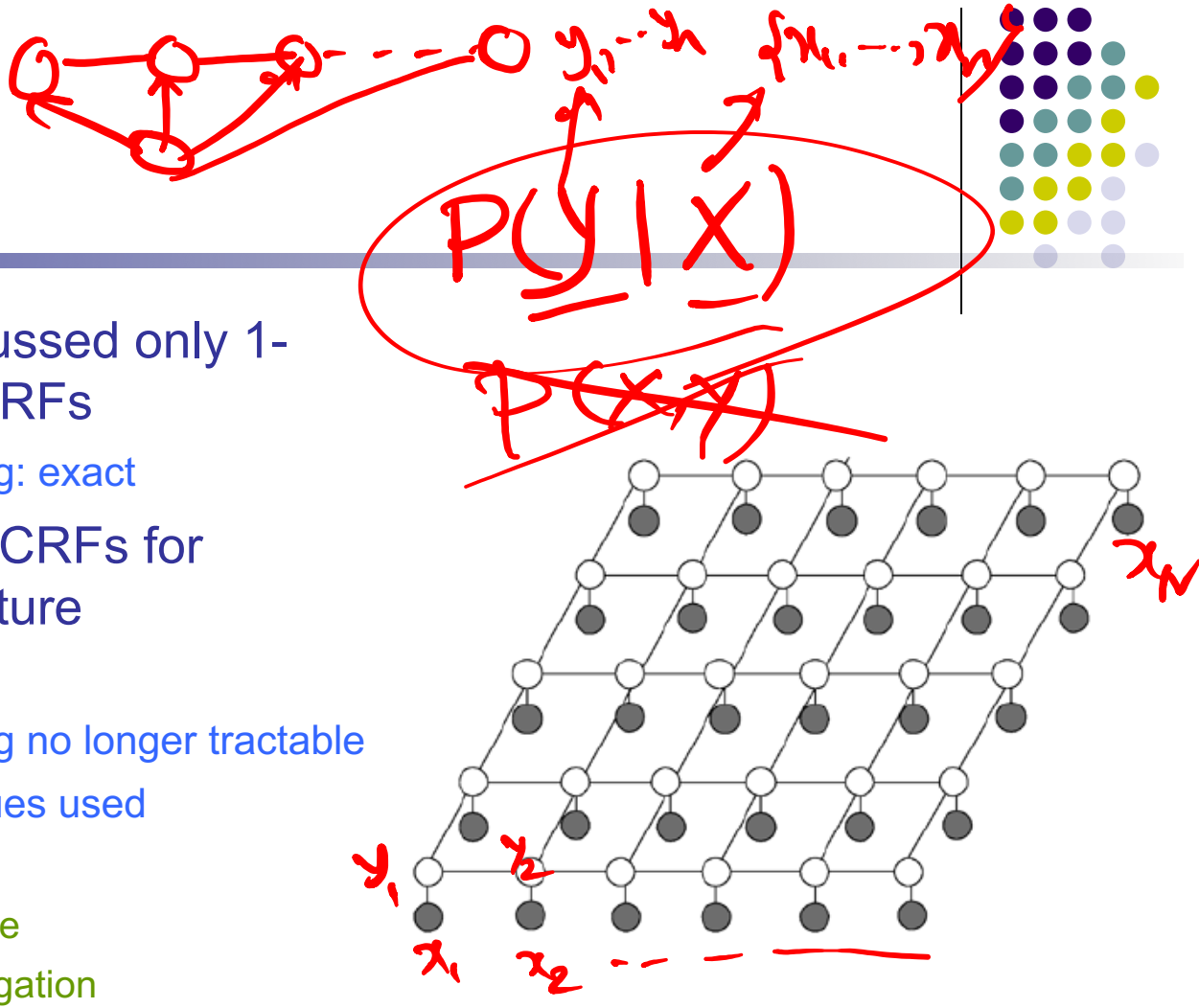
- In practice, gradient ascent has very slow convergence
  - Alternatives:
    - Conjugate Gradient method
    - Limited Memory Quasi-Newton Methods

General CRFs, Hidden-state CRFs

2. CASE STUDY: IMAGE SEGMENTATION (COMPUTER VISION)

Other CRFs

- So far we have discussed only 1-dimensional chain CRFs
  - Inference and learning: exact
- We could also have CRFs for arbitrary graph structure
  - E.g: Grid CRFs
  - Inference and learning no longer tractable
  - Approximate techniques used
    - MCMC Sampling
    - Variational Inference
    - Loopy Belief Propagation
  - We will discuss these techniques soon



Applications of CRF in Vision

Stereo Matching

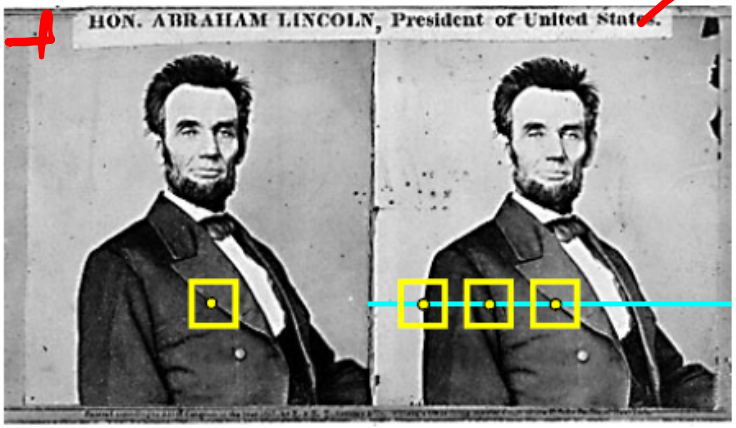


Image Segmentation

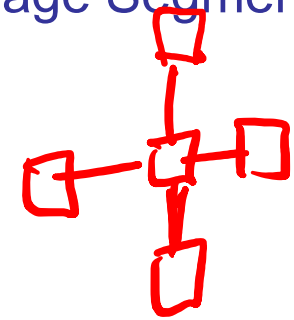
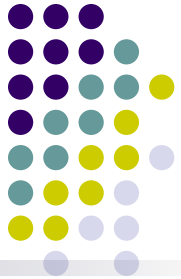


Image Restoration

Application: Image Segmentation



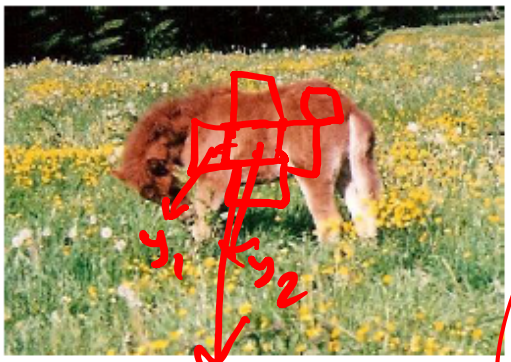
$\phi_i(y_i, x) \in \mathbb{R}^{\approx 1000}$: local image features, e.g. bag-of-words

$\rightarrow$ $\langle w_i, \phi_i(y_i, x) \rangle$: local classifier (like logistic-regression)

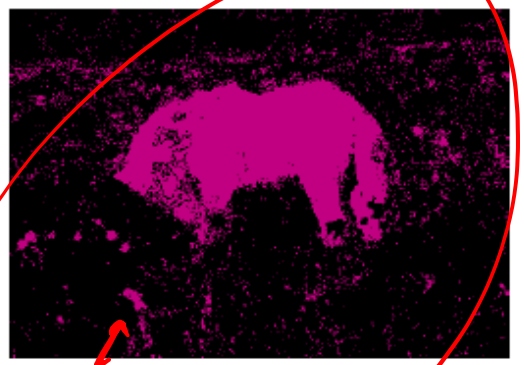
$\phi_{i,j}(y_i, y_j) = [\![y_i = y_j]\!] \in \mathbb{R}^1$: test for same label

$\rightarrow$ $\langle w_{ij}, \phi_{ij}(y_i, y_j) \rangle$: penalizer for label changes (if $w_{ij} > 0$)

combined: $\text{argmax}_y \, p(y|x)$ is smoothed version of local cues



original



local classification

local + smoothness

Case Study: Image Segmentation

- Image segmentation (FG/BG) by modeling of interactions btw RVs
  - Images are noisy.
  - Objects occupy continuous regions in an image.

[Nowozin,Lampert 2012]

Input image

Pixel-wise separate optimal labeling

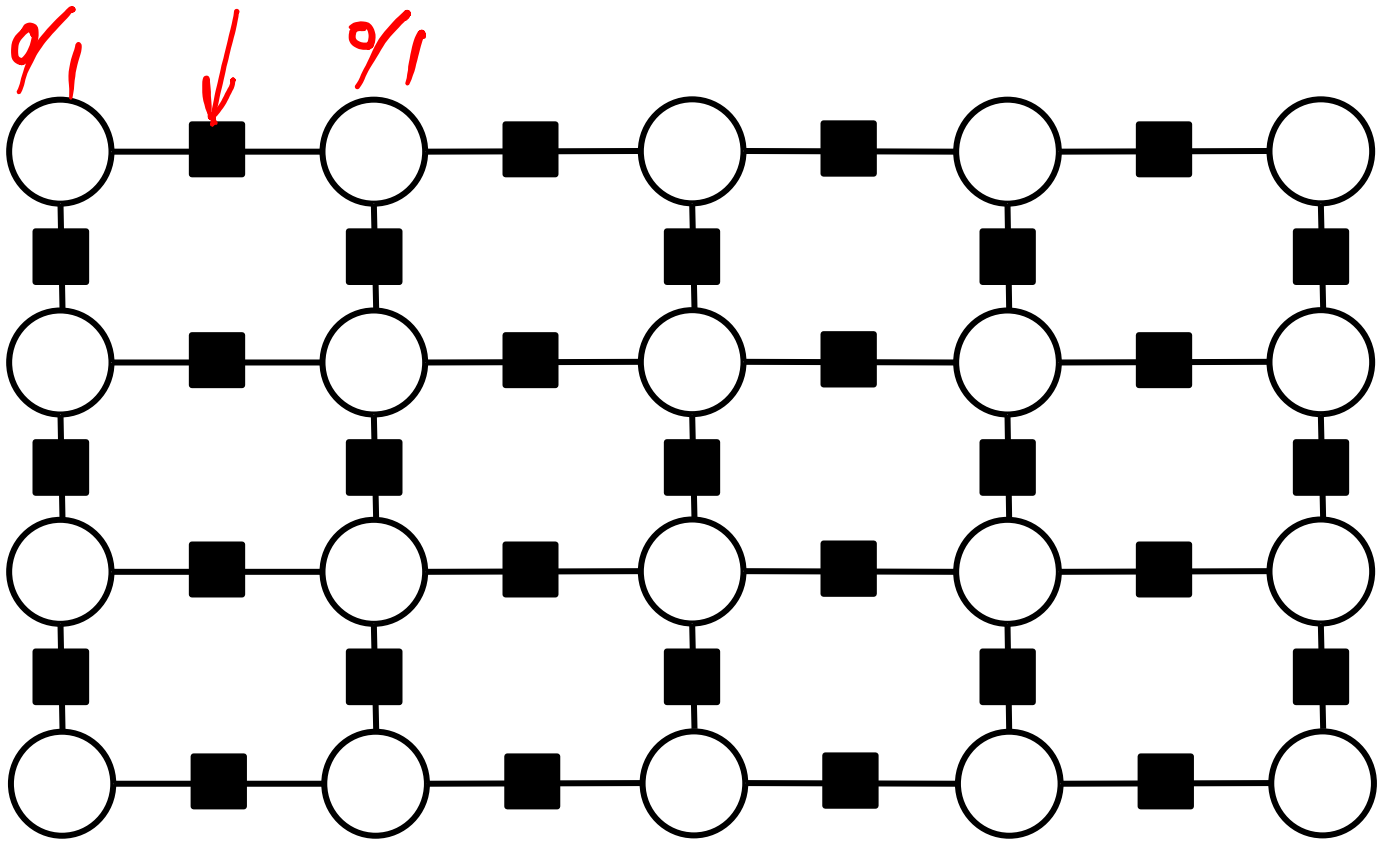
Locally-consistent joint optimal labeling

$$Y^* = \arg\max_{y \in \{0,1\}^n} \left[\underbrace{\sum_{i \in S} V_i(y_i, X)}_{\text{Unary Term}} + \underbrace{\sum_{i \in S} \sum_{j \in N_i} V_{i,j}(y_i, y_j)}_{\text{Pairwise Term}} \right].$$

Y: labels
X: data (features)
S: pixels
N_i: neighbors of pixel i

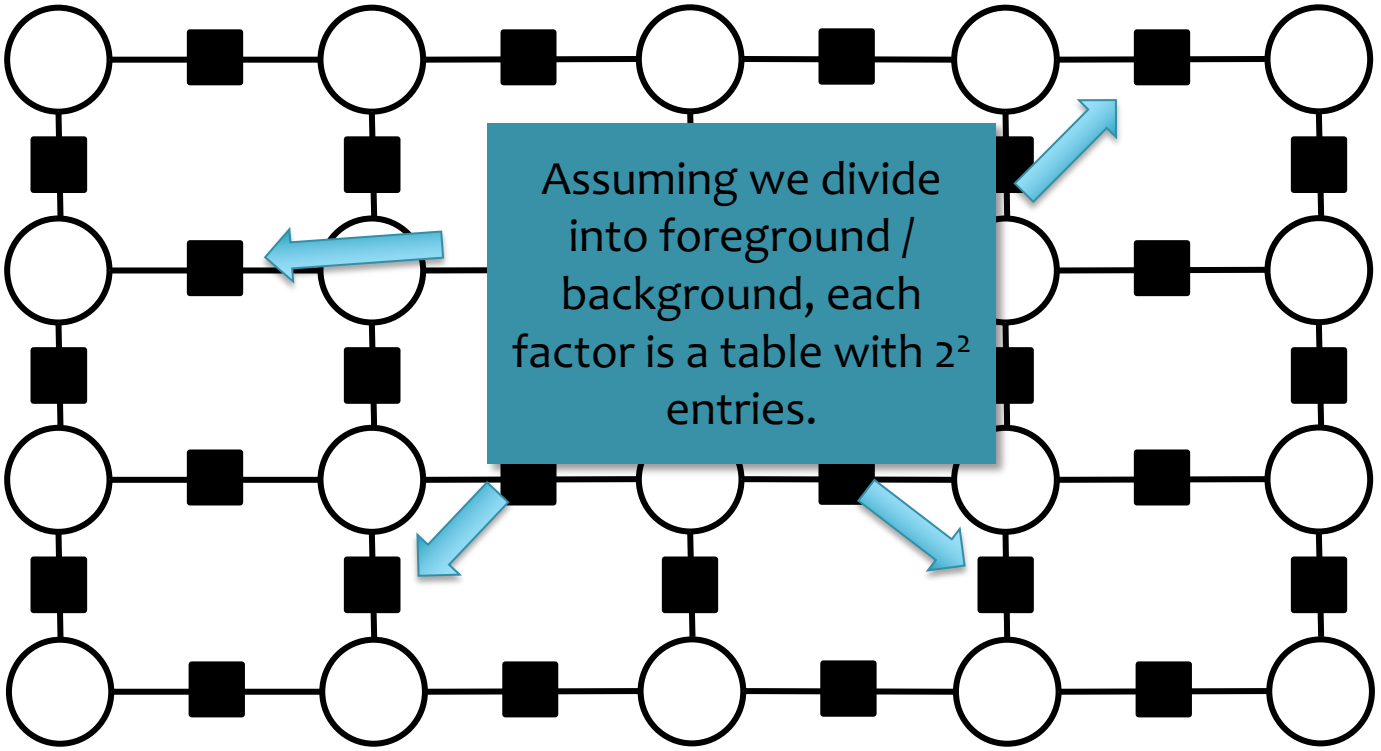
Grid CRF

- Suppose we want to image segmentation using a grid model



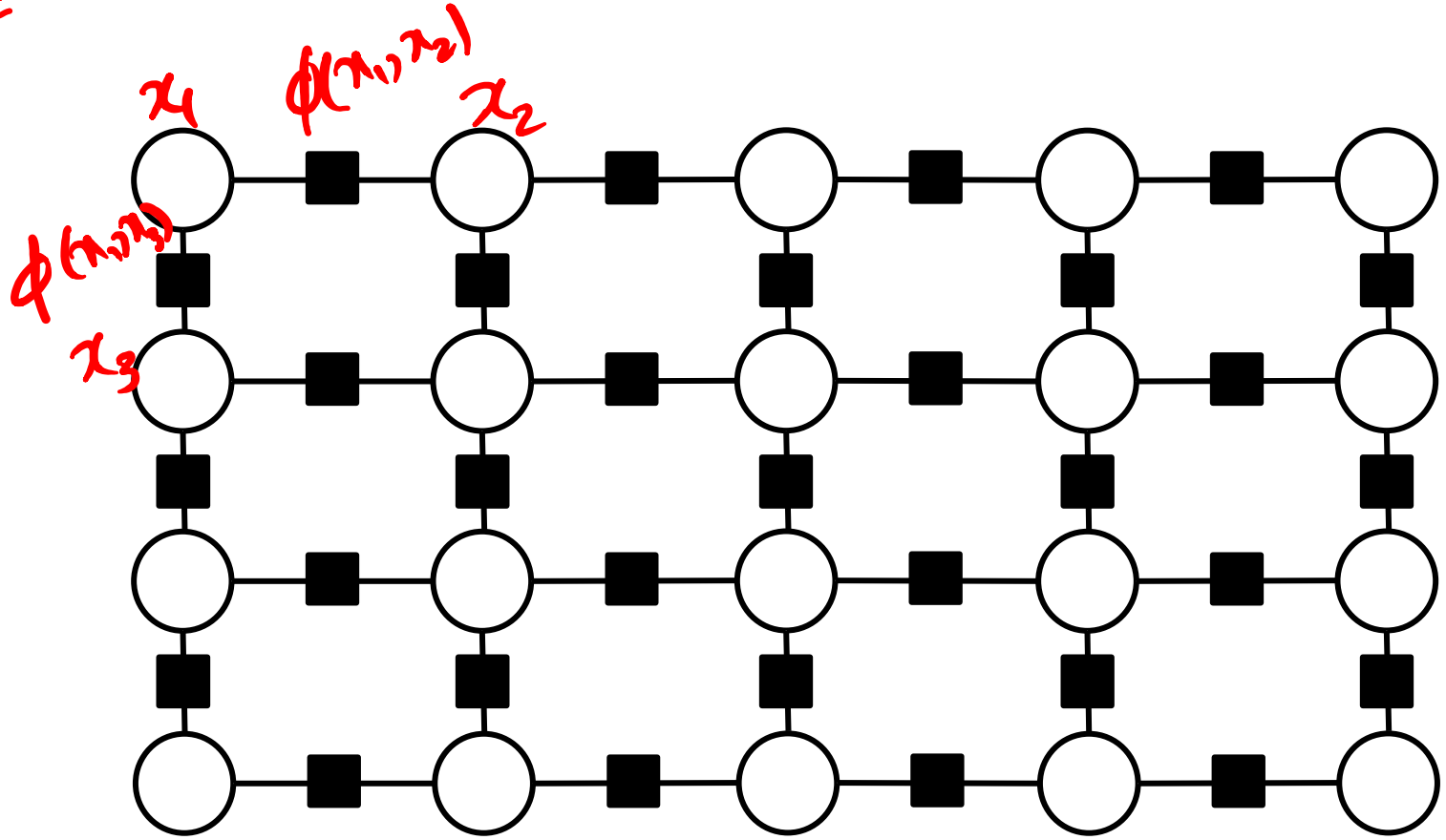
Grid CRF

- Suppose we want to image segmentation using a grid model



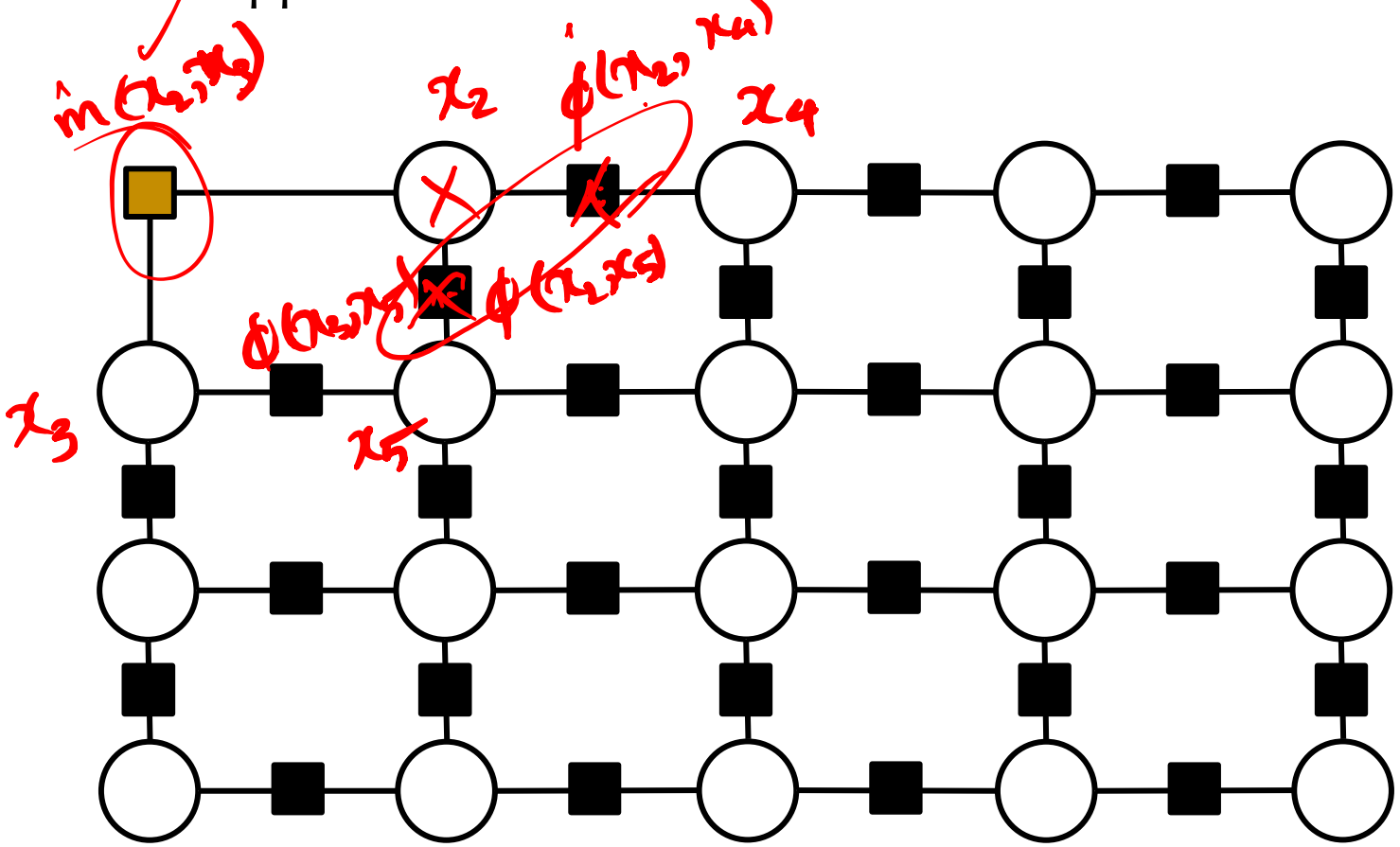
Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



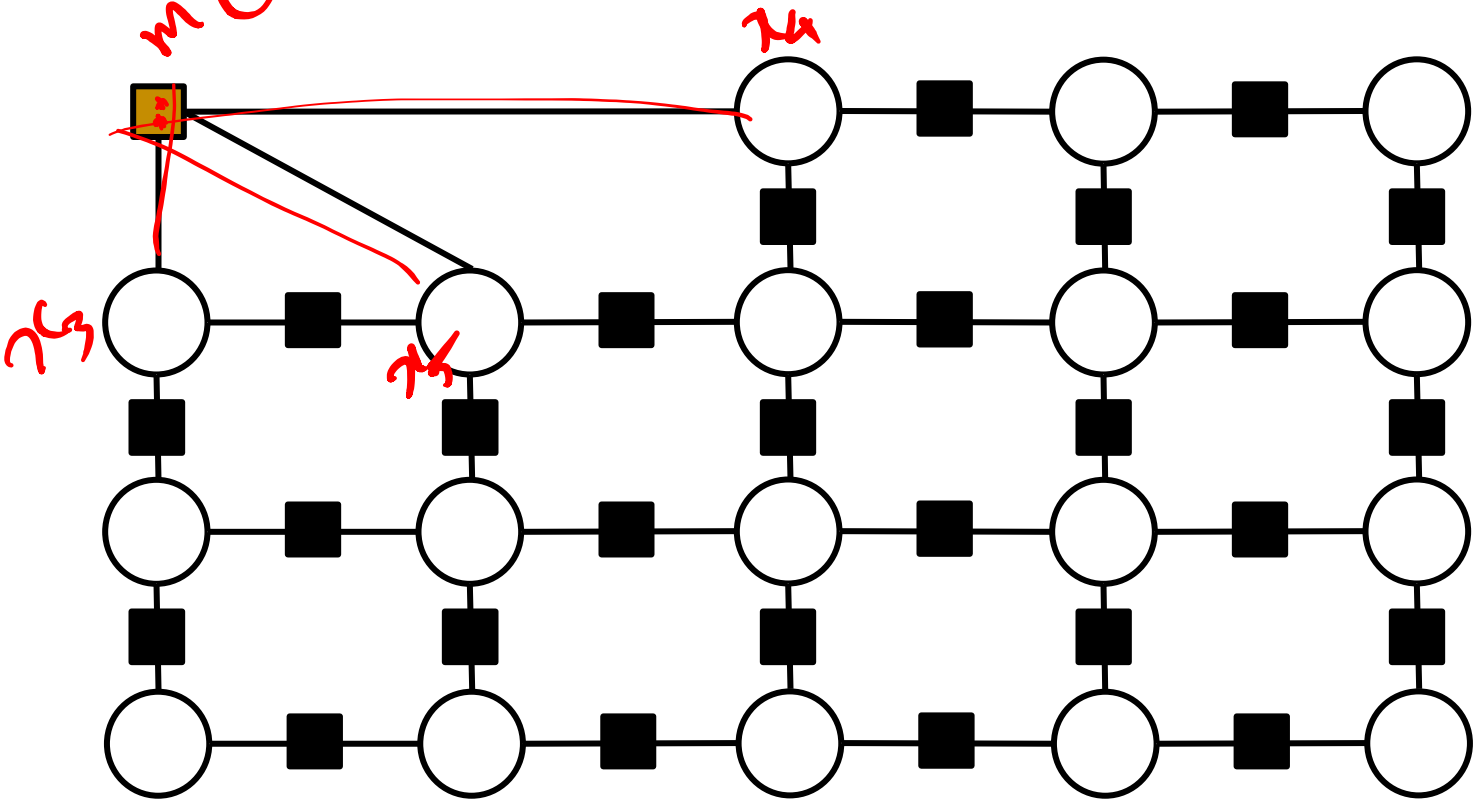
Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



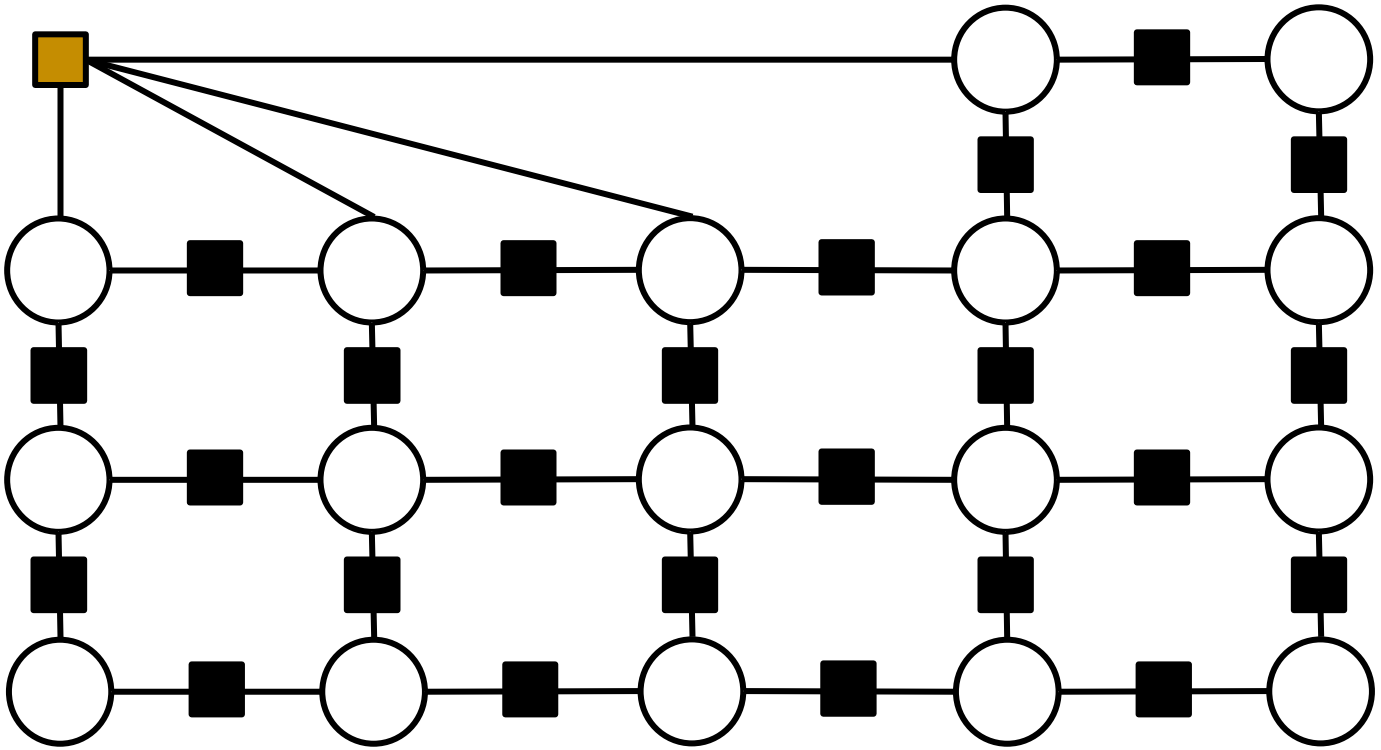
Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



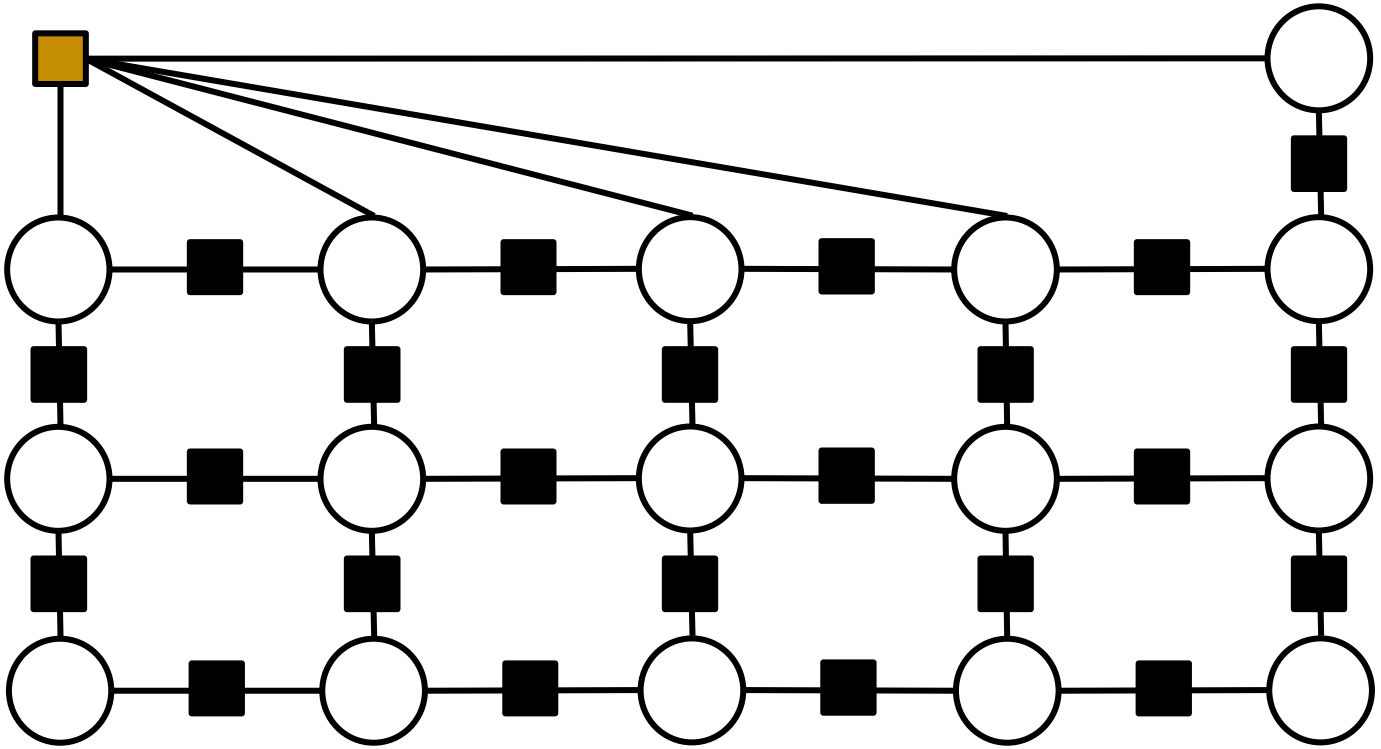
Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



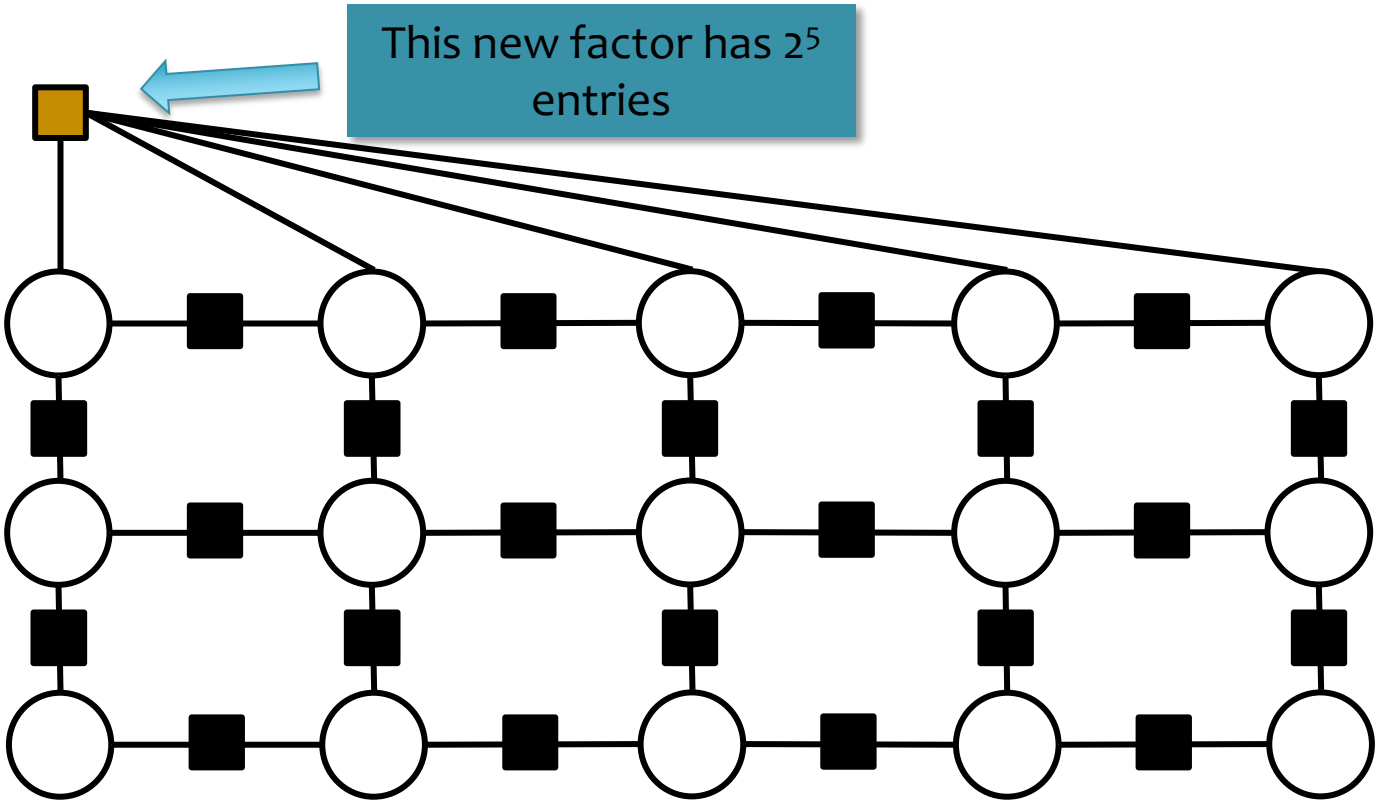
Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



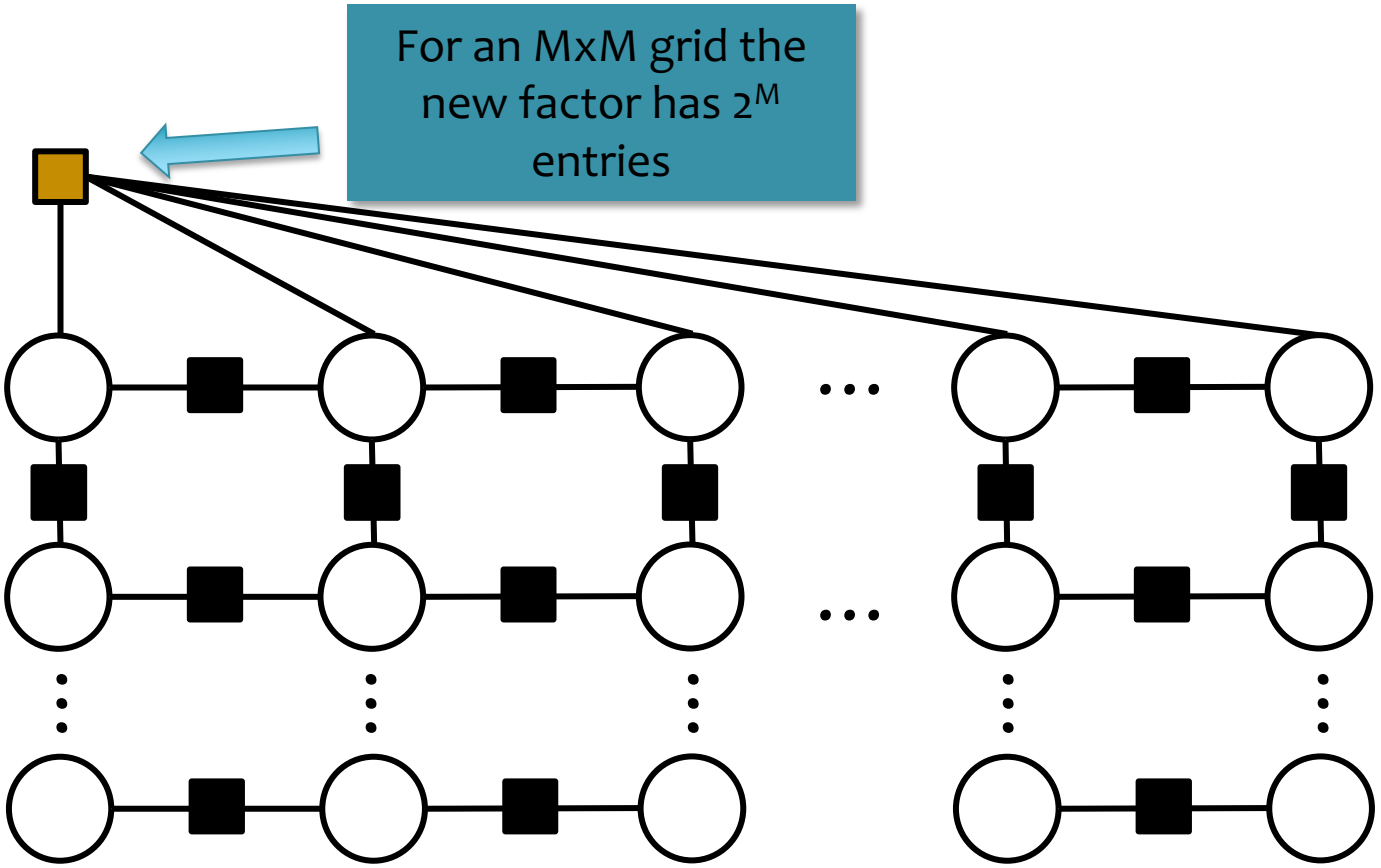
Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



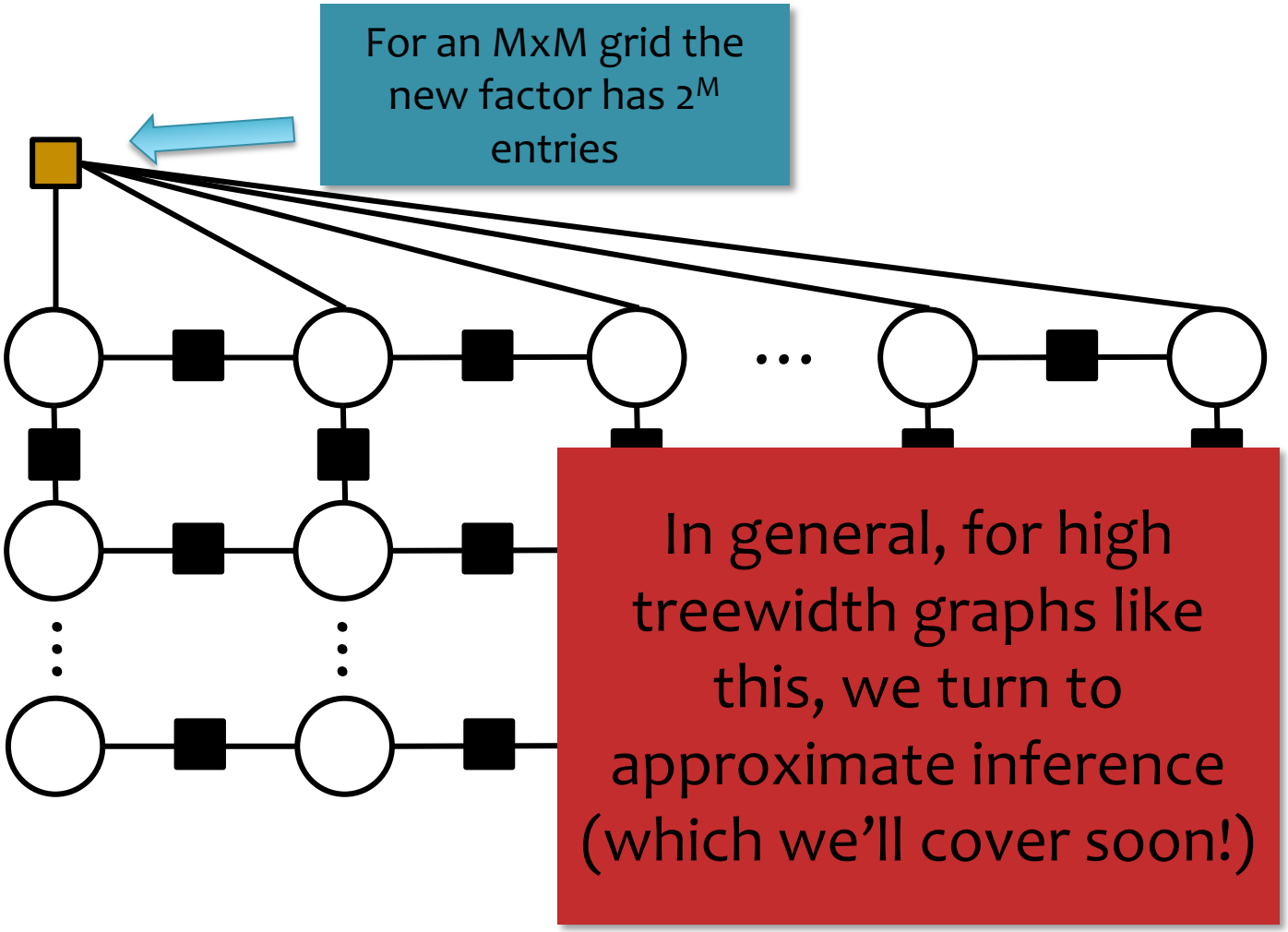
Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



Grid CRF

- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?



Grid CRF

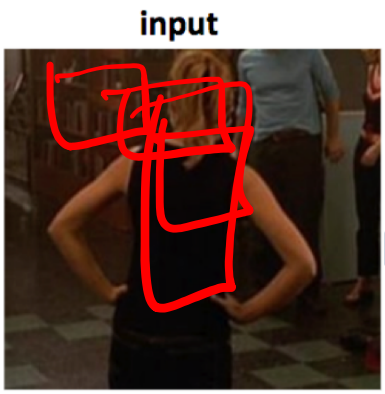
- Suppose we want to image segmentation using a grid model
- What happens when we run variable elimination?

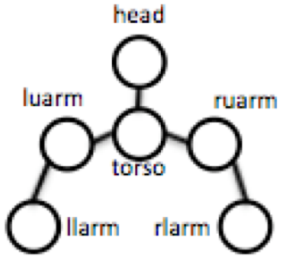
For an MxM grid the new factor has 2^M entries

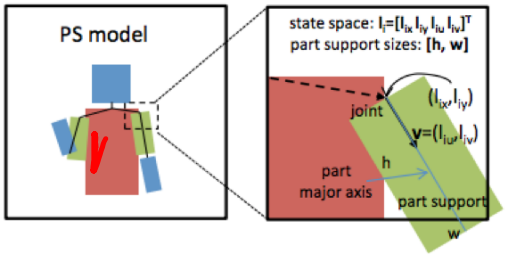
Stay Tuned!
We will discuss it soon!

In general, for high treewidth graphs like this, we turn to approximate inference (which we'll cover soon!)

Application: Pose Estimation







$$p(l|x) \propto \exp\left[\sum_{ij} \theta_{ij}^T \phi_{ij}(l_i, l_j, x) + \sum_i \theta_i^T \phi_i(l_i, x)\right] = e^{\theta^T \phi(l,x)}$$

Penalizes unrealistic poses

Local classifier for each part

$\text{argmax}_y\, p(y|x)$ is cleaned up version of local prediction

Cascaded Models for Articulated Pose Estimation, B. Sapp, A. Toshev, B. Taskar

Feature Functions for CRF in Vision

$\phi_i(y_i, x)$: local representation, high-dimensional

$\rightarrow$ $\langle w_i, \phi_i(y_i, x)\rangle$: local classifier

$\phi_{i,j}(y_i, y_j)$: prior knowledge, low-dimensional

$\rightarrow$ $\langle w_{ij}, \phi_{ij}(y_i, y_j)\rangle$: penalize outliers

learning adjusts parameters:

- unary w_i: learn local classifiers and their importance
- binary w_{ij}: learn importance of smoothing/penalization

$\text{argmax}_y\, p(y|x)$ is cleaned up version of local prediction

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

pigeon

rhinoceros

leopard

llama

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)

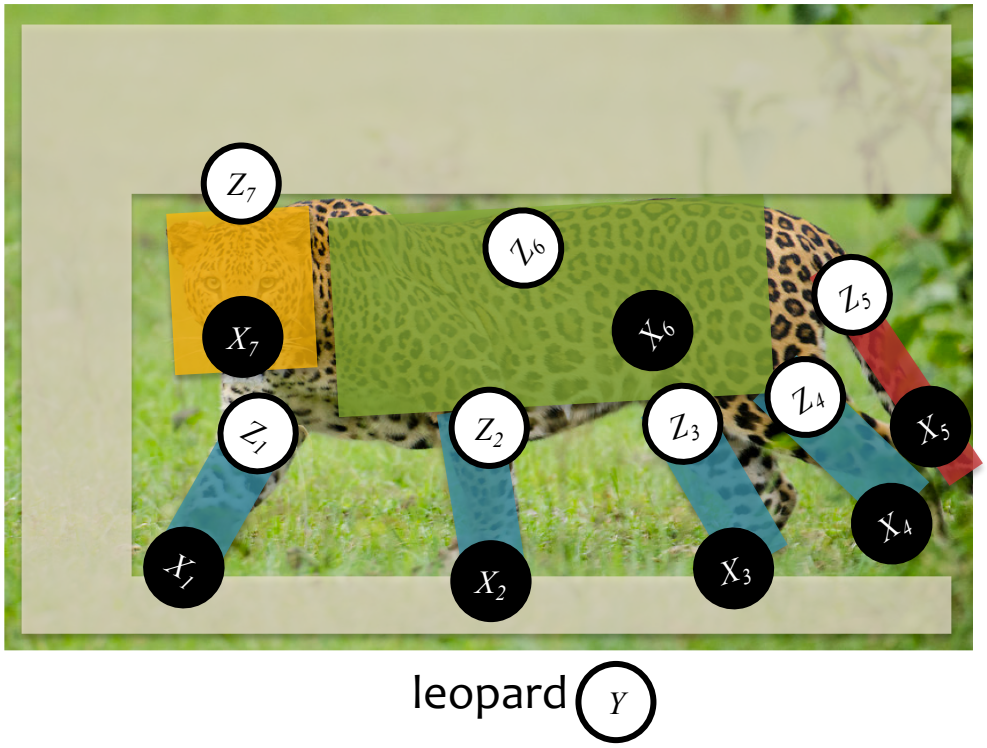
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

leopard

Case Study: Object Recognition

Data consists of images x and labels y.

- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

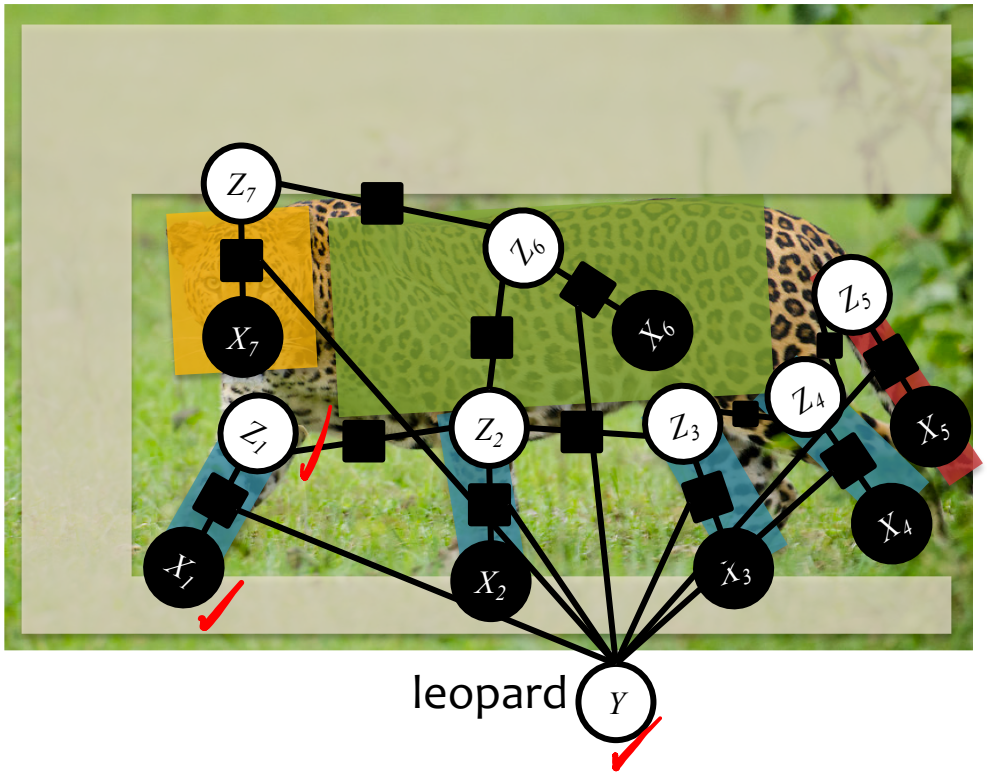


Case Study: Object Recognition

P(y|x)

Data consists of images x and labels y.

- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

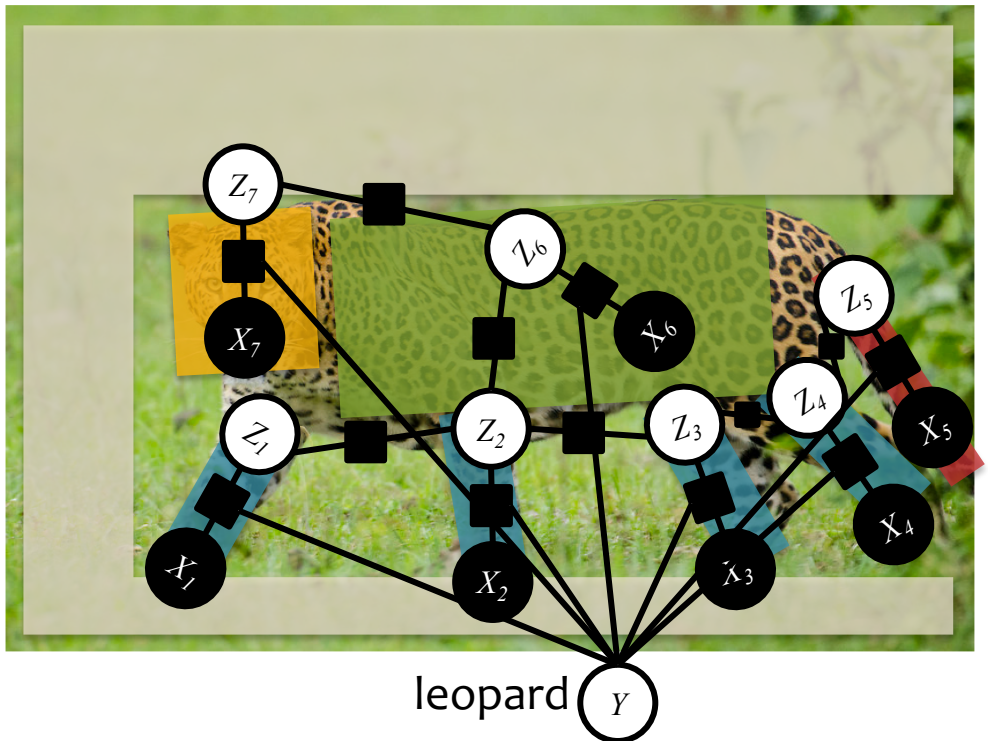


Hidden-state CRFs

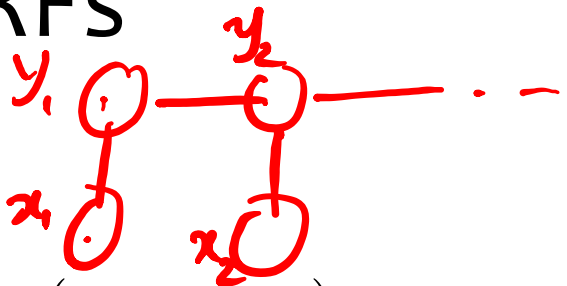
Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Joint model: $p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{\theta})} \prod_{\alpha} \psi_{\alpha}(\boldsymbol{y}_{\alpha}, \boldsymbol{z}_{\alpha}, \boldsymbol{x})$

Marginalized model: $p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x})$



Hidden-state CRFs



Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Joint model: $p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{\theta})} \prod_{\alpha} \psi_{\alpha}(\boldsymbol{y}_{\alpha}, \boldsymbol{z}_{\alpha}, \boldsymbol{x})$

Marginalized model: $p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x})$

We can train using gradient based methods:

(the values $\boldsymbol{x}$ are omitted below for clarity)

$$\frac{d\ell(\boldsymbol{\theta}|\mathcal{D})}{d\boldsymbol{\theta}} = \sum_{n=1}^{N} \left(\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{y}^{(n)})}[f_j(\boldsymbol{y}^{(n)}, \boldsymbol{z})] - \mathbb{E}_{\boldsymbol{y}, \boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\cdot,\cdot)}[f_j(\boldsymbol{y}, \boldsymbol{z})] \right)$$

$$= \sum_{n=1}^{N} \sum_{\alpha} \left(\sum_{\boldsymbol{z}_{\boldsymbol{\alpha}}} \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\alpha}} \mid \boldsymbol{y}^{(n)})}_{\text{Inference on } \textbf{clamped} \text{ factor graph}} f_{\alpha,j}(\boldsymbol{y}_{\alpha}^{(n)}, \boldsymbol{z}_{\alpha}) - \sum_{\boldsymbol{y}_{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{\alpha}}} \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{y}_{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{\alpha}})}_{\text{Inference on } \textbf{full} \text{ factor graph}} f_{\alpha,j}(\boldsymbol{y}_{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{\alpha}}) \right)$$

Learning and Inference Summary

	Learning	Marginal Inference	MAP Inference
HMM	Just counting	Forward-backward	Viterbi
MEMM	Gradient based – decomposes and doesn't require inference (GLIM)	Forward-backward	Viterbi
Linear-chain CRF	Gradient based – doesn't decompose because of $Z(\boldsymbol{x})$ and requires marginal inference	Forward-backward	Viterbi
General CRF	Gradient based – doesn't decompose because of $Z(\boldsymbol{x})$ and requires (**approximate**) marginal inference	(approximate methods)	(approximate methods)
HCRF	Gradient based – same as General CRF	(approximate methods)	(approximate methods)

Summary

- HMM:
  - Pro: Easy to train
  - Con: Misses out on rich features of the observations
- MEMM:
  - Pro: Fast to train and supports rich features
  - Con: Suffers (like the HMM) from the label bias problem
- Linear-chain CRF:
  - Pro: Defeats the label bias problem with support for rich features
  - Con: Slower to train
- MBR Decoding:
  - the principled way to account for a loss function when decoding from a probabilistic model
- Generative vs. Discriminative:
  - gen. is better if the model is well-specified
  - disc. is better if the model is misspecified
- General CRFs:
  - Exact inference won't suffice for high treewidth graphs
  - More general topologies can capture intuitions about variable dependencies
- HCRF:
  - Training looks very much like CRF training
  - Incorporation of hidden variables can model domain specific knowledge

Introduction to Topic Modeling

Topic Modeling

Motivation:

Suppose you're given a massive corpora and asked to carry out the following tasks

- **Organize** the documents into **thematic categories**
- **Describe** the evolution of those categories **over time**
- Enable a domain expert to **analyze and understand** the content
- Find **relationships** between the categories
- Understand how **authorship** influences the content

Topic Modeling

Motivation:

Suppose you're given a massive corpora and asked to carry out the following tasks

- **Organize** the documents into **thematic categories**
- **Describe** the evolution of those categories **over time**
- Enable a domain expert to **analyze and understand** the content
- Find **relationships** between the categories
- Understand how **authorship** influences the content

Topic Modeling:

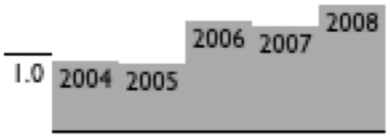
A method of (usually unsupervised) discovery of latent or hidden structure in a corpus

- Applied primarily to text corpora, but **techniques are more general**
- Provides a **modeling toolbox**
- Has prompted the exploration of a variety of new **inference methods** to accommodate **large-scale datasets**

Topic Modeling

Dirichlet-multinomial regression (DMR) topic model on ICML
(Mimno & McCallum, 2008)

Topic 0 [0.152]

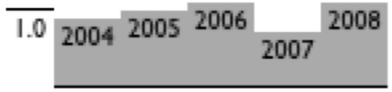


problem, optimization, problems, convex, convex optimization, linear, semidefinite programming, formulation, sets, constraints, proposed, margin, maximum margin, optimization problem, linear programming, programming, procedure, method, cutting plane, solutions

Topic 54 [0.051]

decision trees, trees, tree, decision tree, decision, tree ensemble, junction tree, decision tree learners, leaf nodes, arithmetic circuits, ensembles modts, skewing, ensembles, anytime induction decision trees, trees trees, random forests, objective decision trees, tree learners, trees grove, candidate split

Topic 99 [0.066]



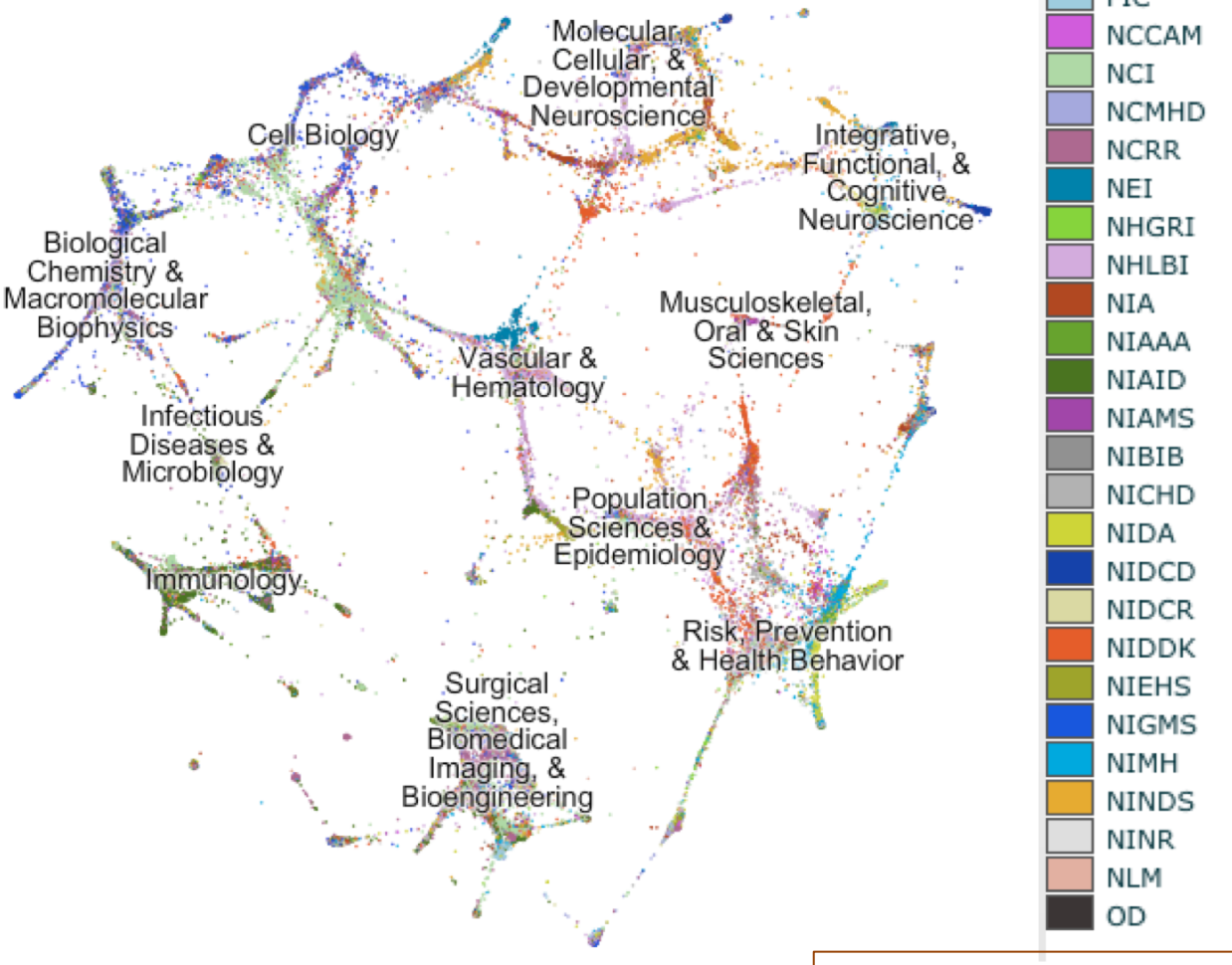
inference, approximate inference, exact inference, markov chain, models, approximate, gibbs sampling, variational, bayesian, variational inference, variational bayesian, approximation, sampling, methods, exact, bayesian inference, dynamic bayesian, process, mcmc, efficient

http:// www.cs.umass.edu/~mimno/icml100.html

Topic Modeling

- Map of NIH Grants

(Talley et al., 2011)

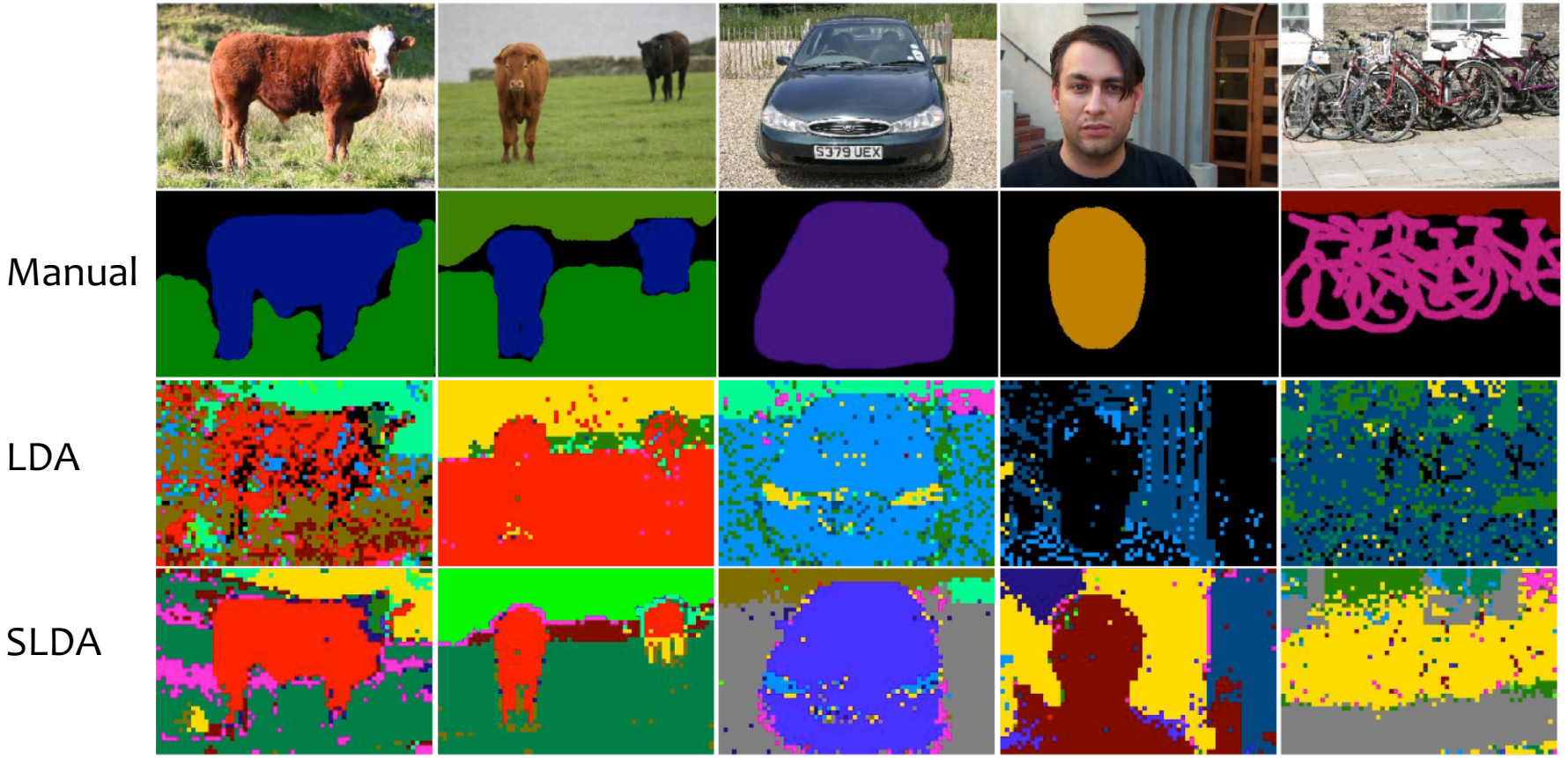


https://app.nihmaps.org/

Other Applications of Topic Models

- Spacial LDA

(Wang & Grimson, 2007)



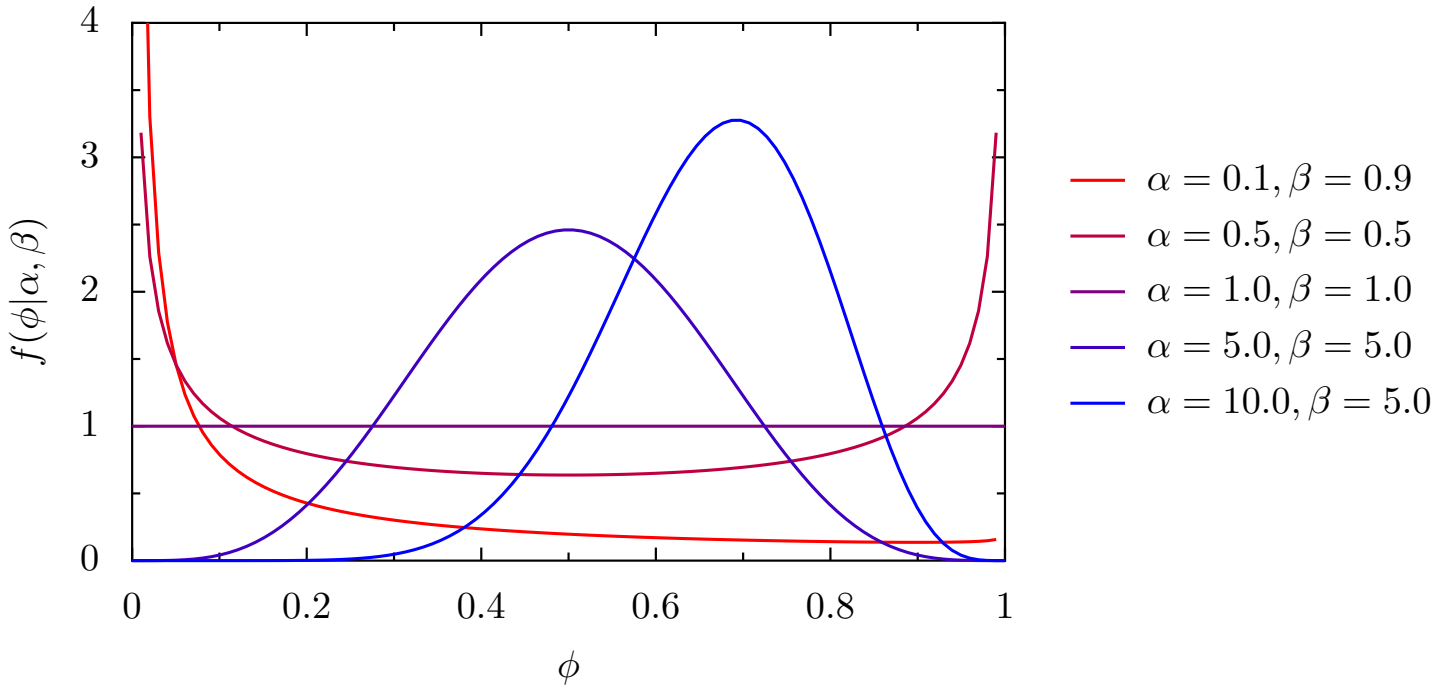
Outline

- **Applications of Topic Modeling**
- **Review: Latent Dirichlet Allocation (LDA)**
  1. Beta-Bernoulli
  2. Dirichlet-Multinomial
  3. Dirichlet-Multinomial Mixture Model
  4. LDA
- **Contrast of methods for Inference / Learning**
  - Exact inference
  - EM
  - Monte Carlo EM
  - Gibbs sampler
  - Collapsed Gibbs sampler
- **Extensions of LDA**
  - Correlated topic models
  - Dynamic topic models
  - Polylingual topic models
  - Supervised LDA

Beta-Bernoulli Model

- Beta Distribution

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



Beta-Bernoulli Model

- Generative Process

$\phi \sim \text{Beta}(\alpha, \beta)$ [*draw distribution over words*]
For each word $n \in \{1, \ldots, N\}$
$x_n \sim \text{Bernoulli}(\phi)$ [*draw word*]

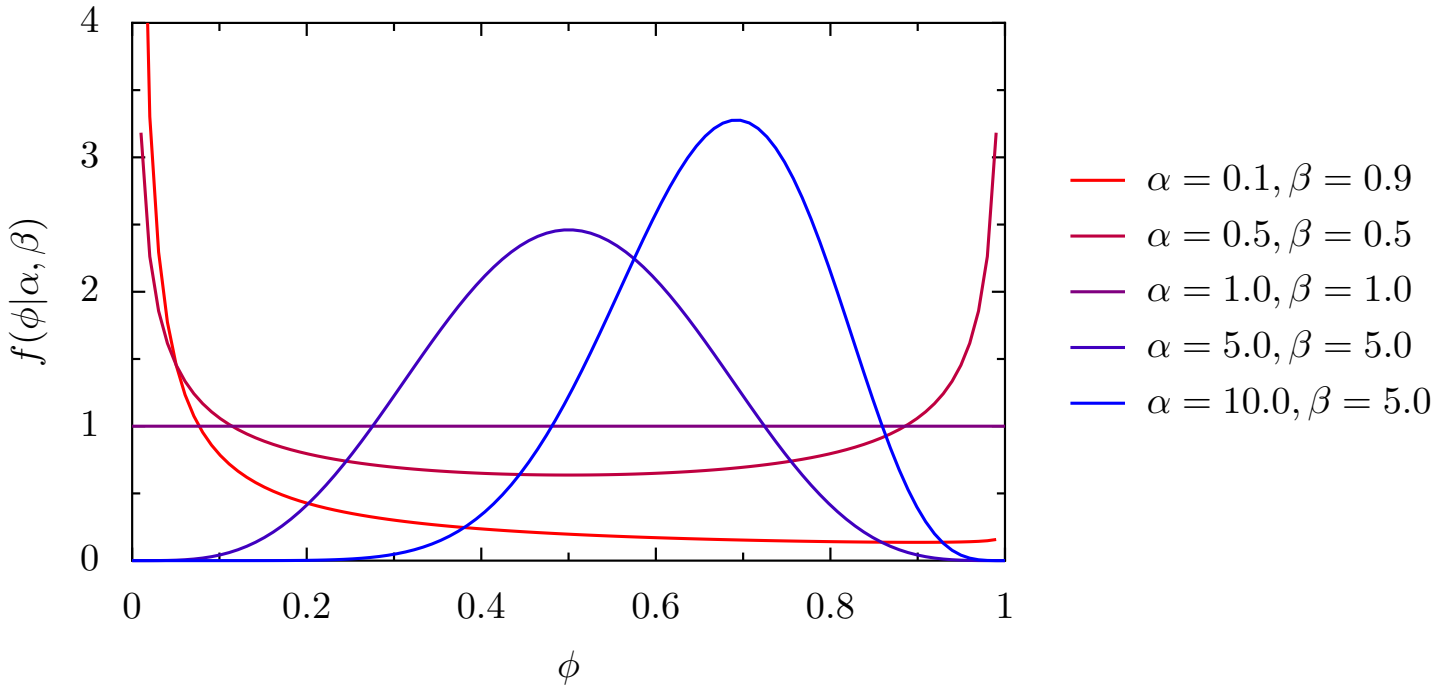
- Example corpus (heads/tails)

H	T	T	H	H	T	T	H	H	H
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}

Dirichlet-Multinomial Model

- Dirichlet Distribution

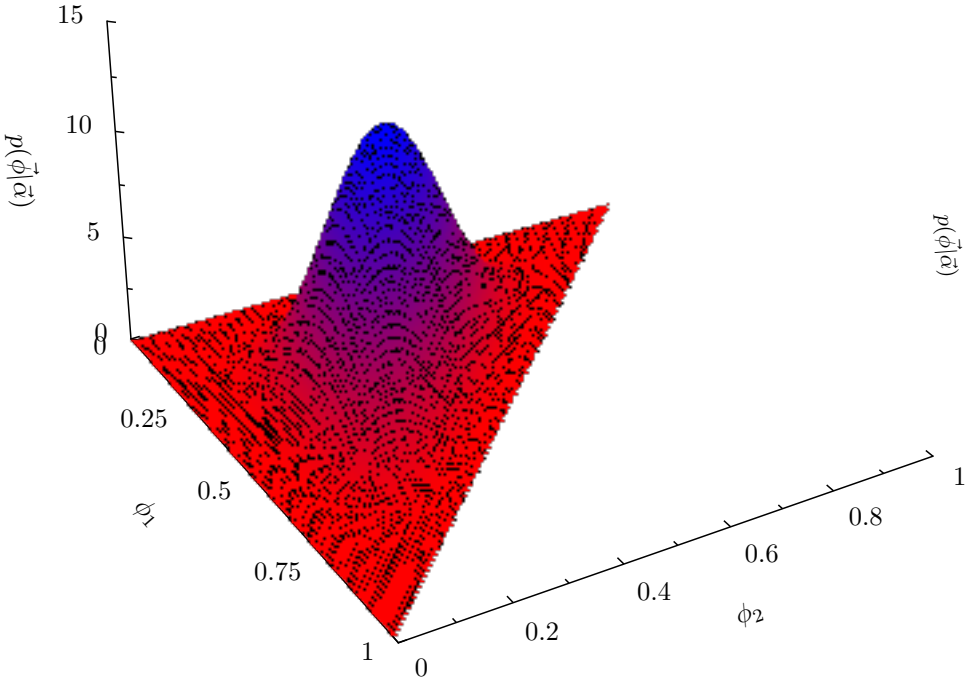
$$f(\phi|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

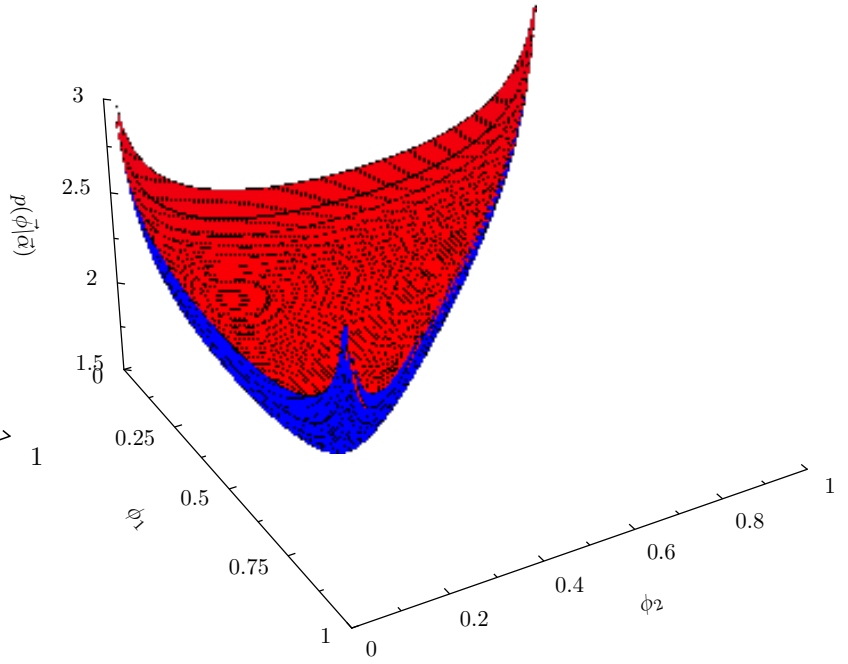


Dirichlet-Multinomial Model

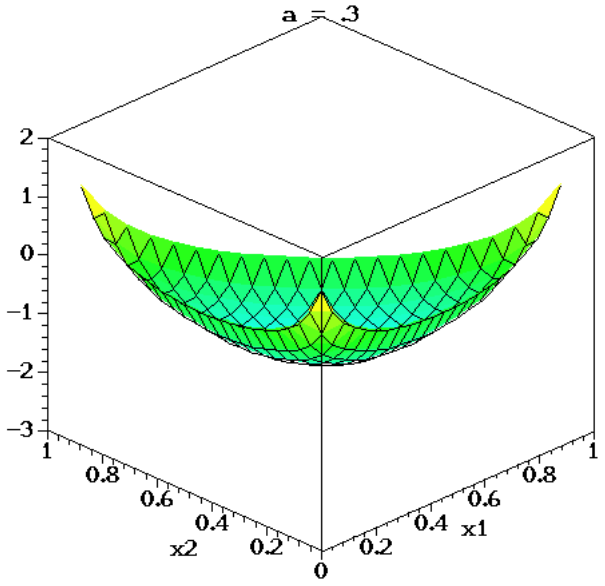
- Dirichlet Distribution

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





Dirichlet-Multinomial Model



Dirichlet-Multinomial Model

- Generative Process

$\phi \in \mathbb{R}^6$

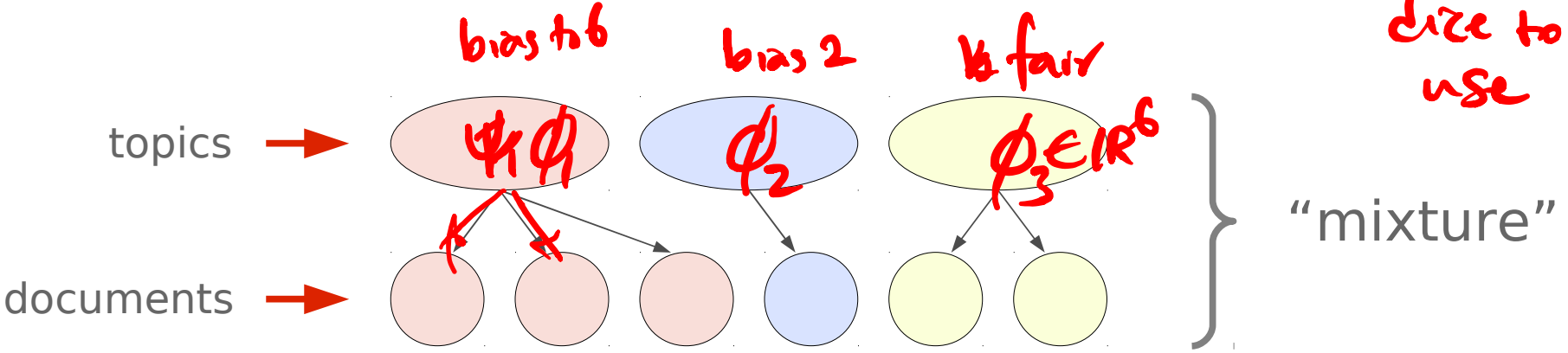
$\phi \sim \text{Dir}(\beta)$ [*draw distribution over words*]
For each word $n \in \{1, \ldots, N\}$
$x_n \sim \text{Mult}(1, \phi)$ [*draw word*]

- Example corpus

the	he	is	the	and	the	she	she	is	is
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}

Dirichlet-Multinomial Mixture Model

- Generative Process



- Example corpus

the	he	is
x_{11}	x_{12}	x_{13}

Document 1

the	and	the
x_{21}	x_{22}	x_{23}

Document 2

she	she	is	is
x_{31}	x_{32}	x_{33}	x_{34}

Document 3

Dirichlet-Multinomial Mixture Model

- Generative Process

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

- Example corpus

the	he	is
x_{11}	x_{12}	x_{13}

Document 1

the	and	the
x_{21}	x_{22}	x_{23}

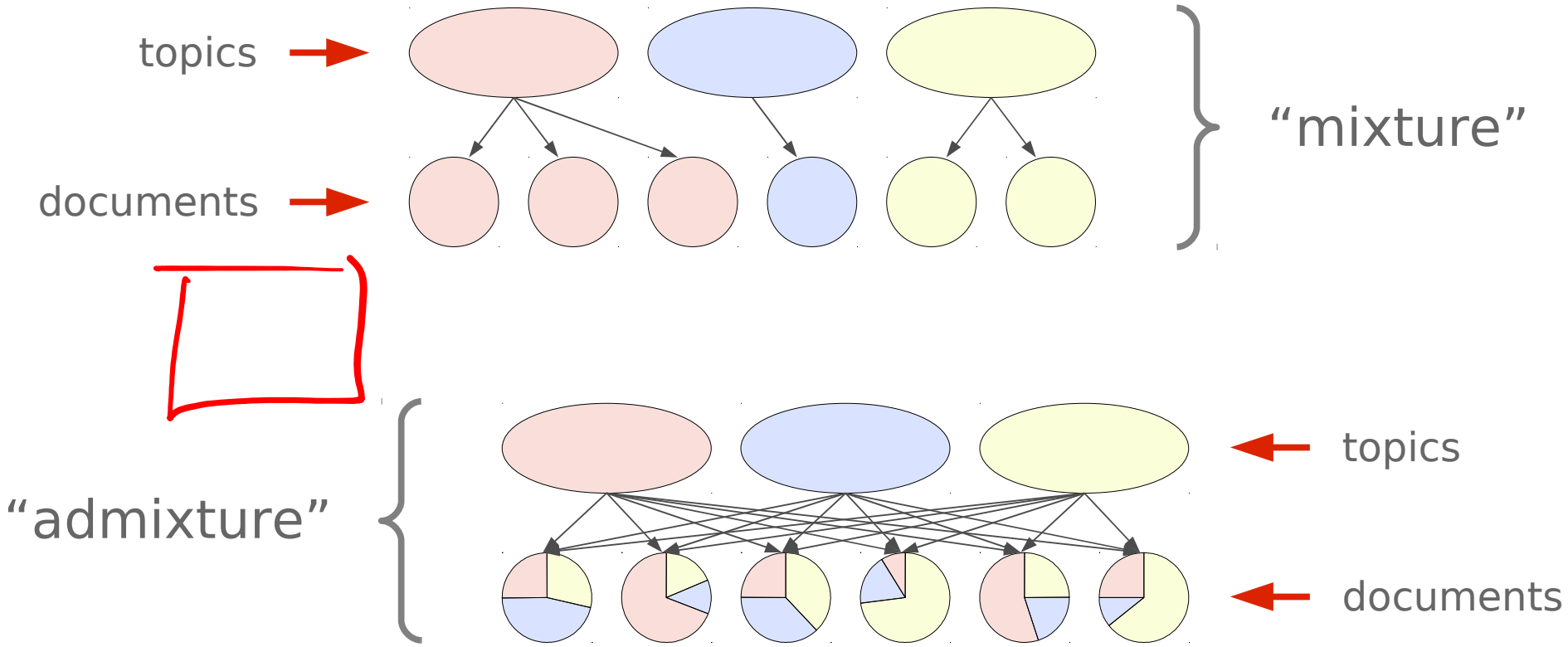
Document 2

she	she	is	is
x_{31}	x_{32}	x_{33}	x_{34}

Document 3

Mixture vs. Admixture (LDA)

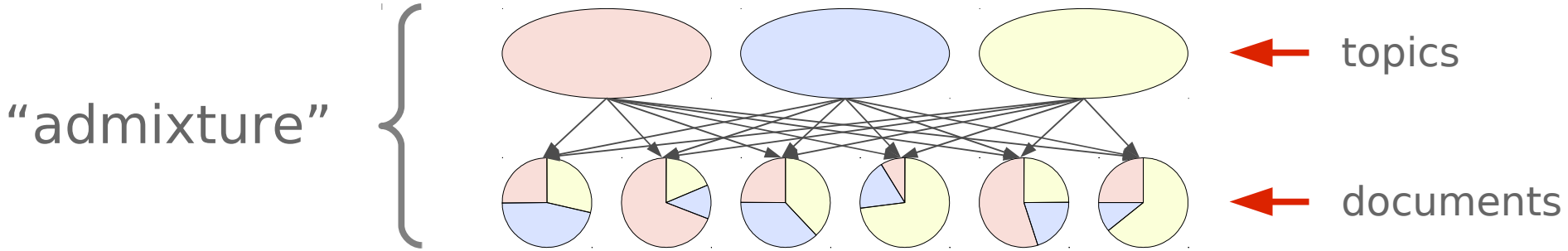
$\sum \pi_k N(x; \mu_k, \Sigma_k)$



Diagrams from Wallach, JHU 2011, slides

Latent Dirichlet Allocation

- Generative Process



- Example corpus

the	he	is
x_{11}	x_{12}	x_{13}

Document 1

the	and	the
x_{21}	x_{22}	x_{23}

Document 2

she	she	is	is
x_{31}	x_{32}	x_{33}	x_{34}

Document 3

Latent Dirichlet Allocation

- Generative Process

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

- Example corpus

the	he	is
x_{11}	x_{12}	x_{13}

Document 1

the	and	the
x_{21}	x_{22}	x_{23}

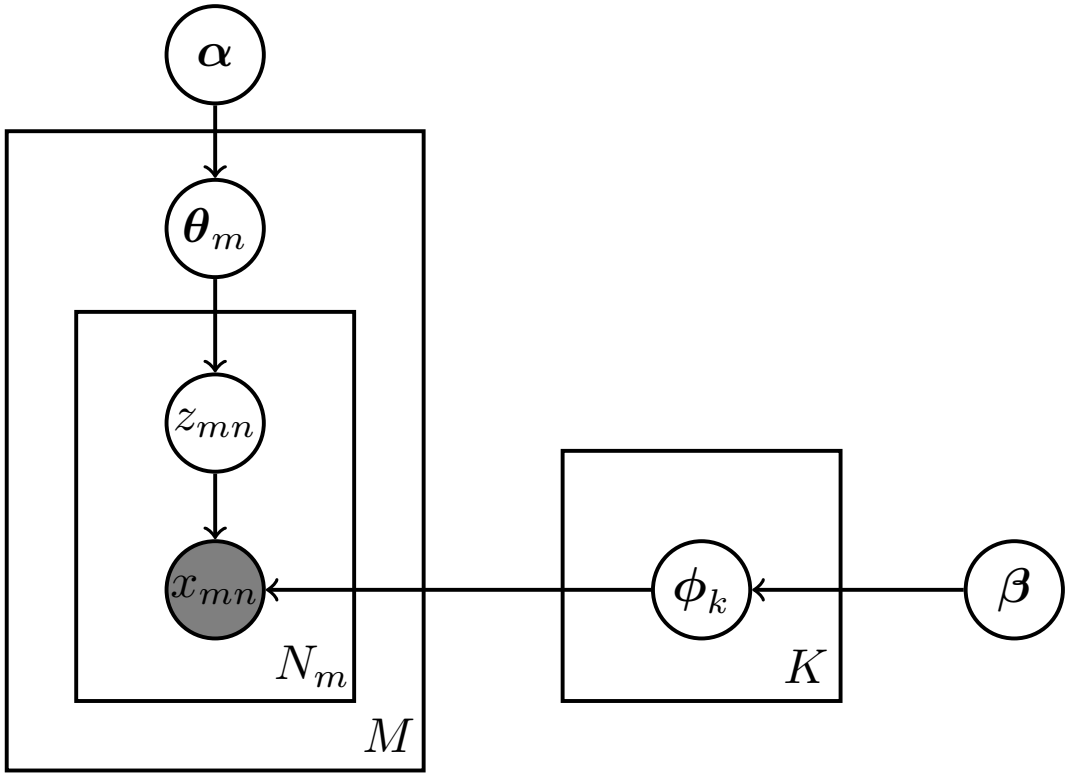
Document 2

she	she	is	is
x_{31}	x_{32}	x_{33}	x_{34}

Document 3

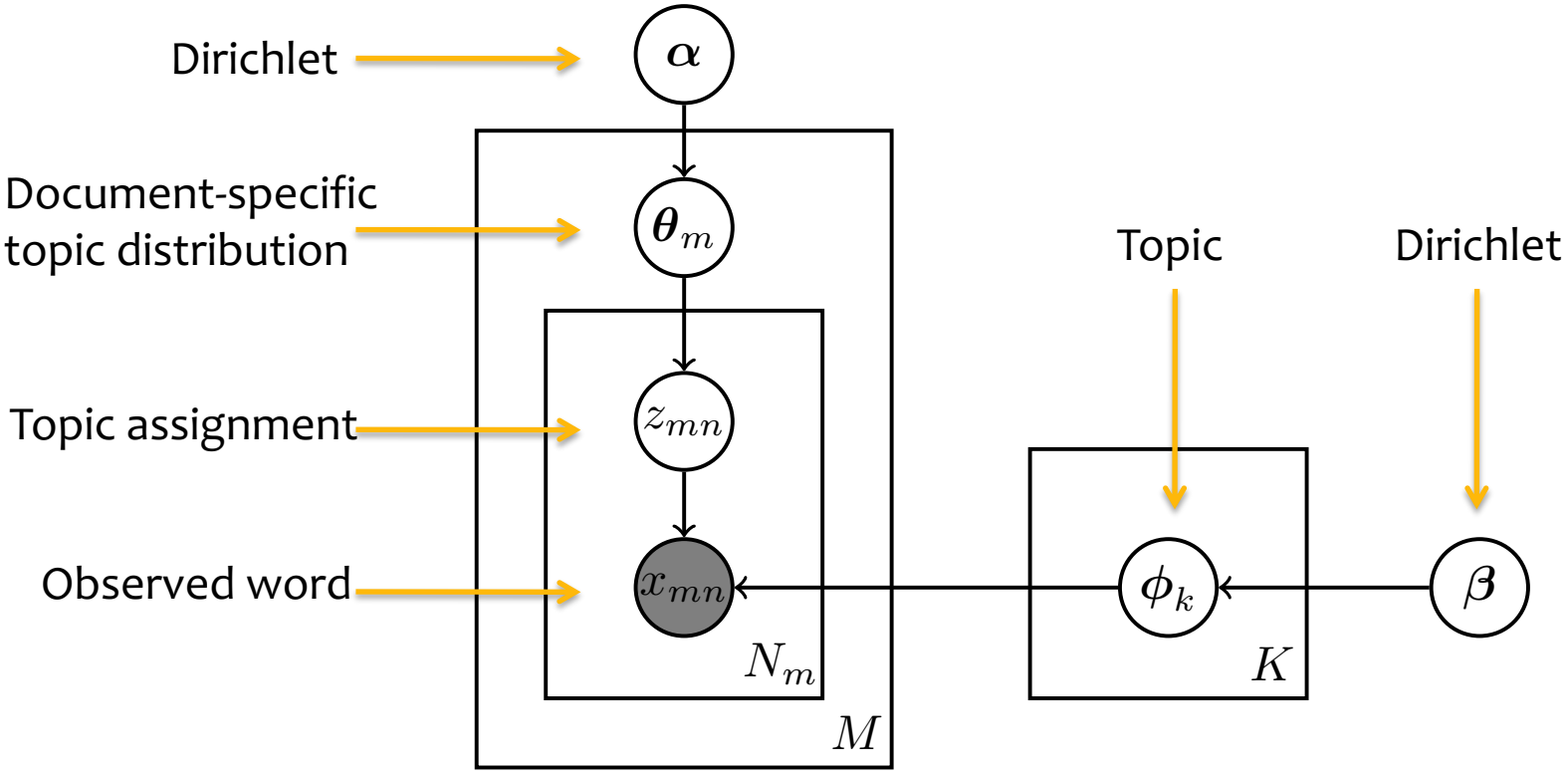
Latent Dirichlet Allocation

- Plate Diagram



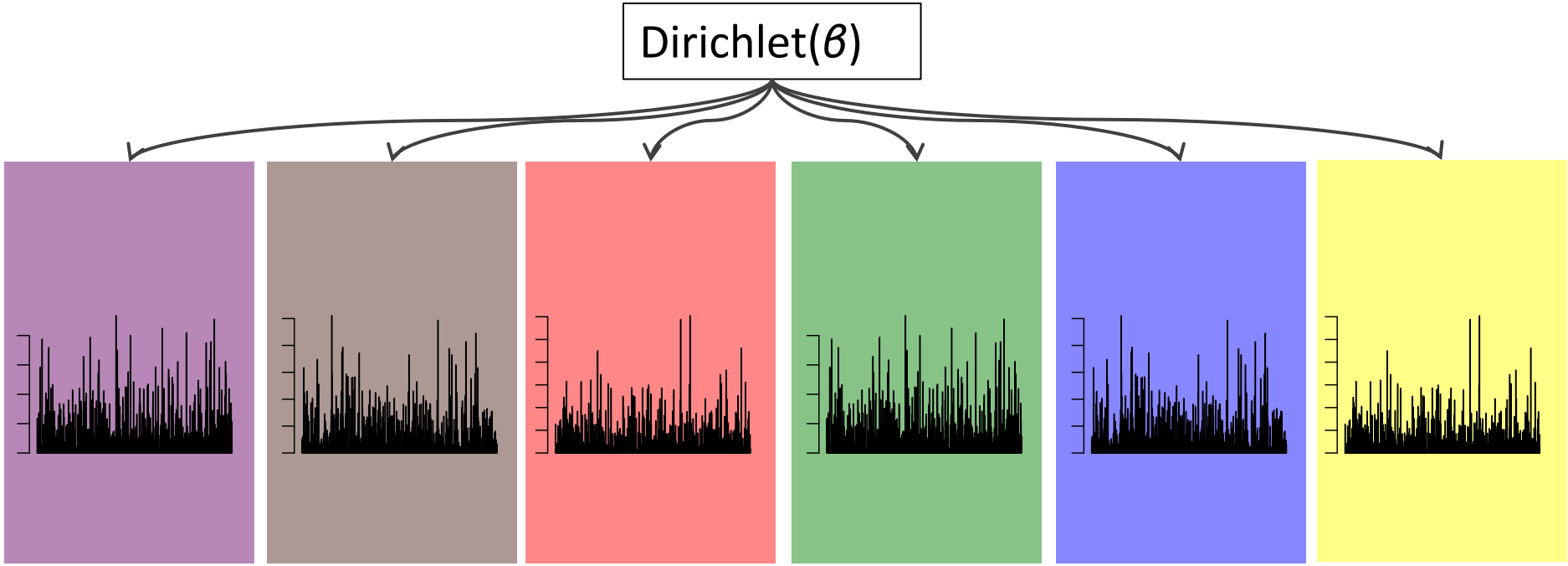
Latent Dirichlet Allocation

- Plate Diagram



(Blei, Ng, & Jordan, 2003)

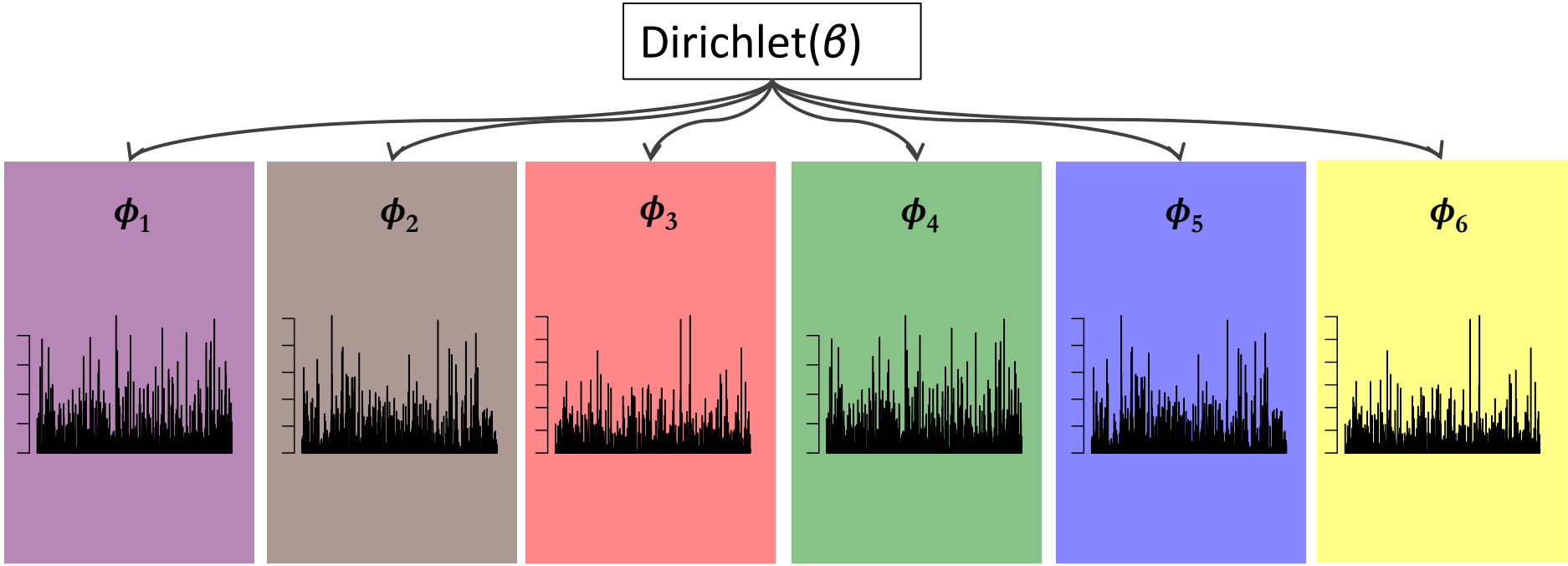
LDA for Topic Modeling



- 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)

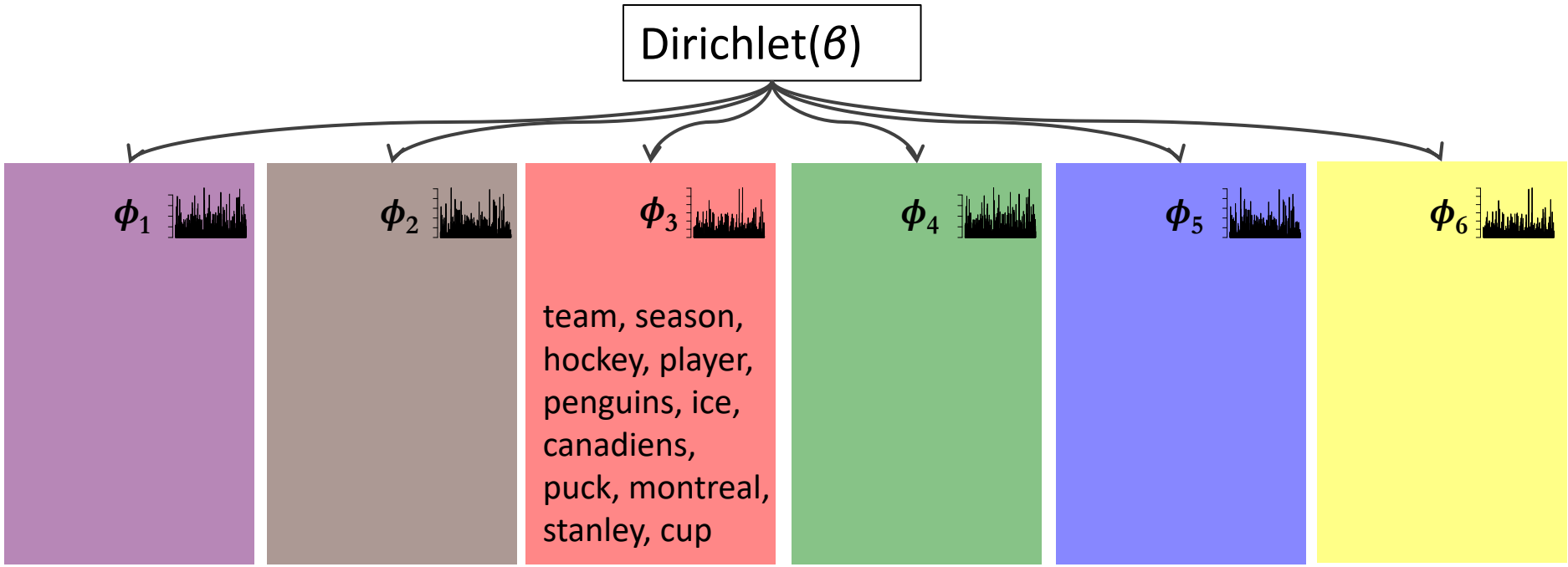
LDA for Topic Modeling



- 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)

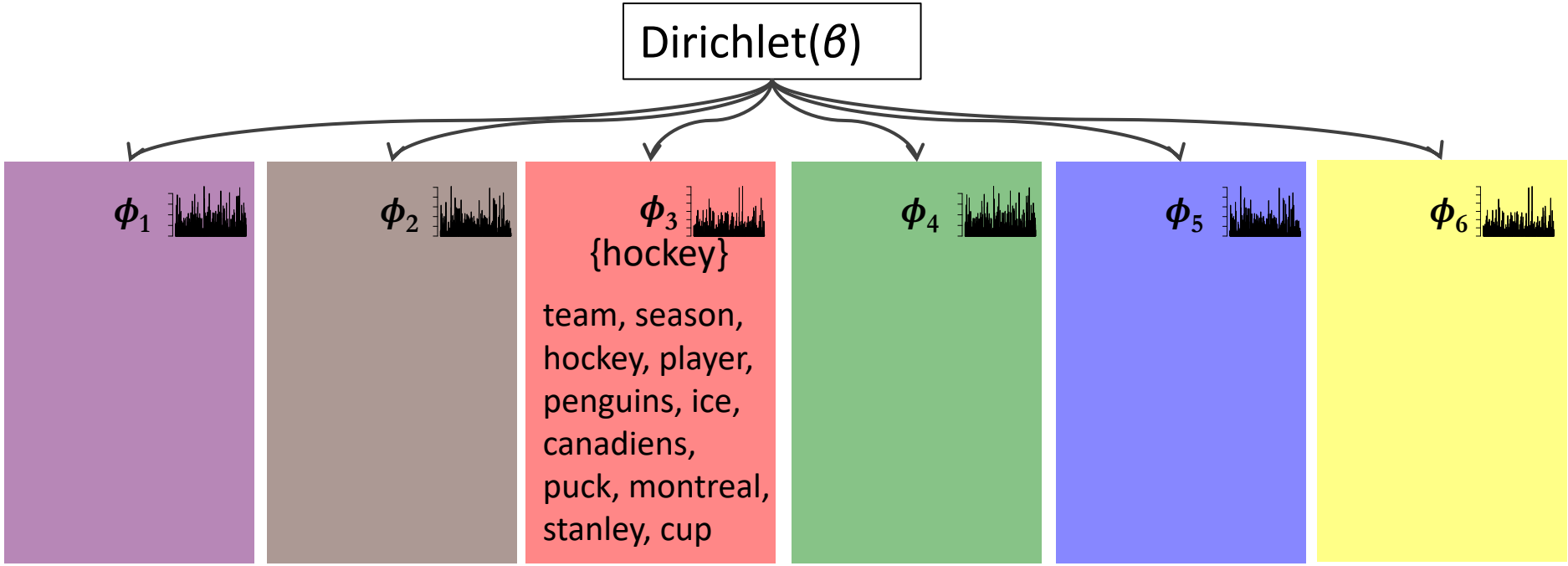
LDA for Topic Modeling



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

(Blei, Ng, & Jordan, 2003)

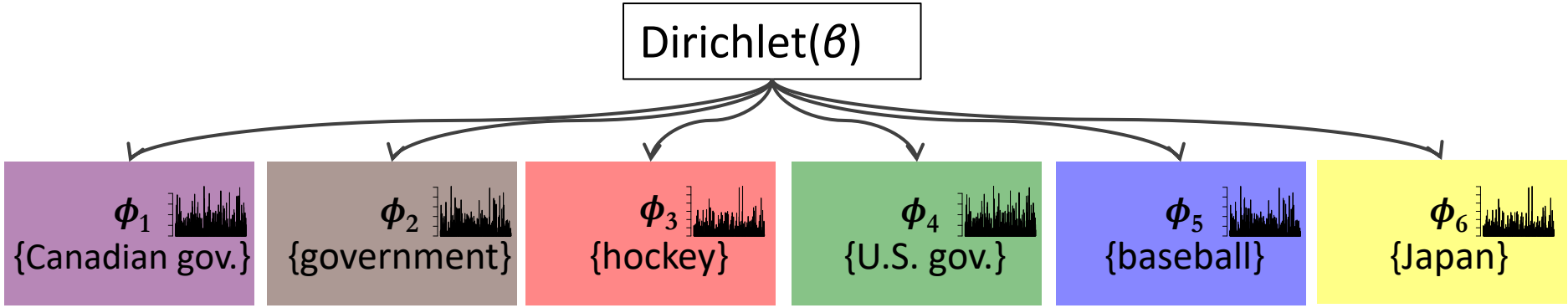
LDA for Topic Modeling



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

(Blei, Ng, & Jordan, 2003)

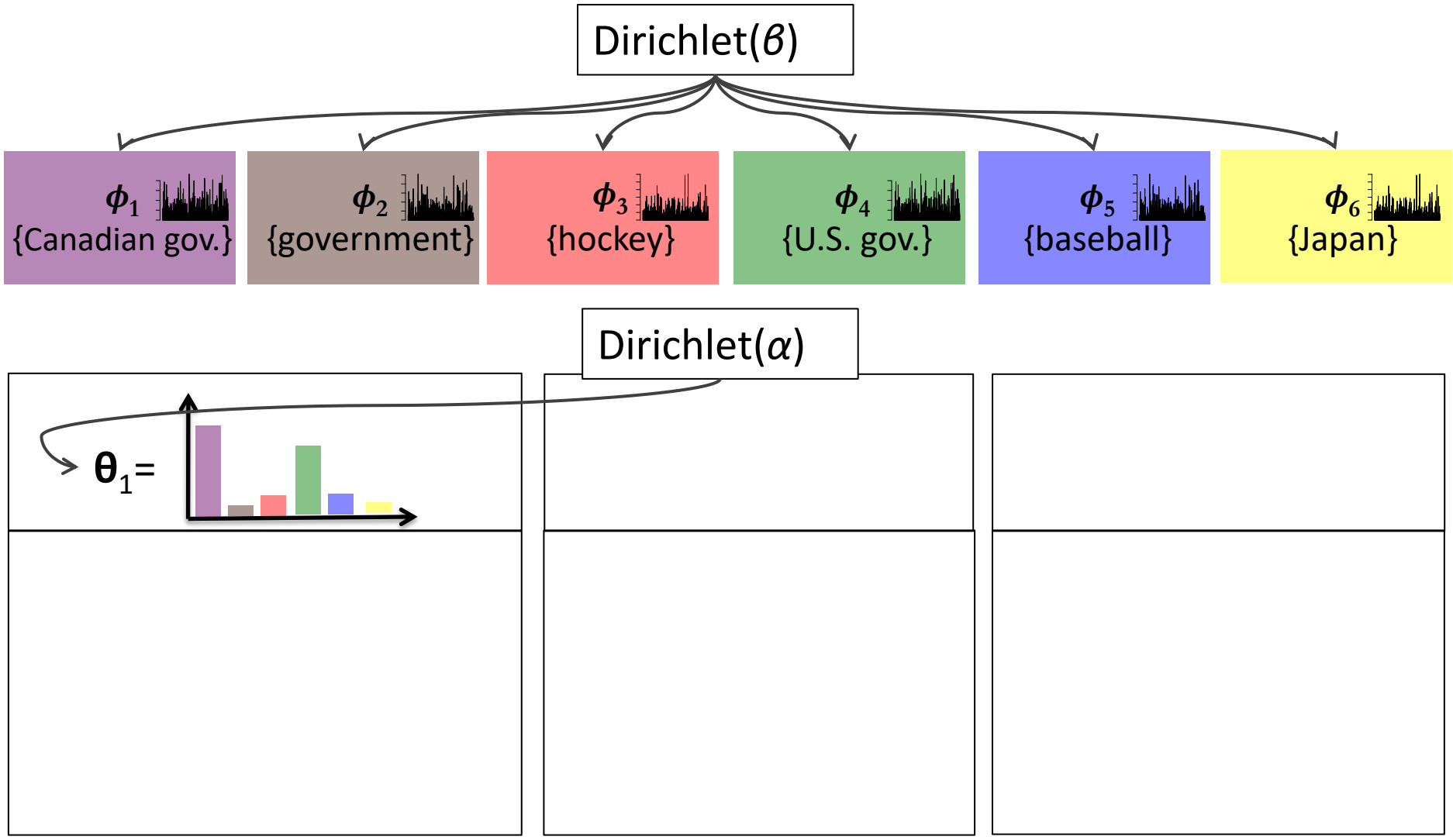
LDA for Topic Modeling



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

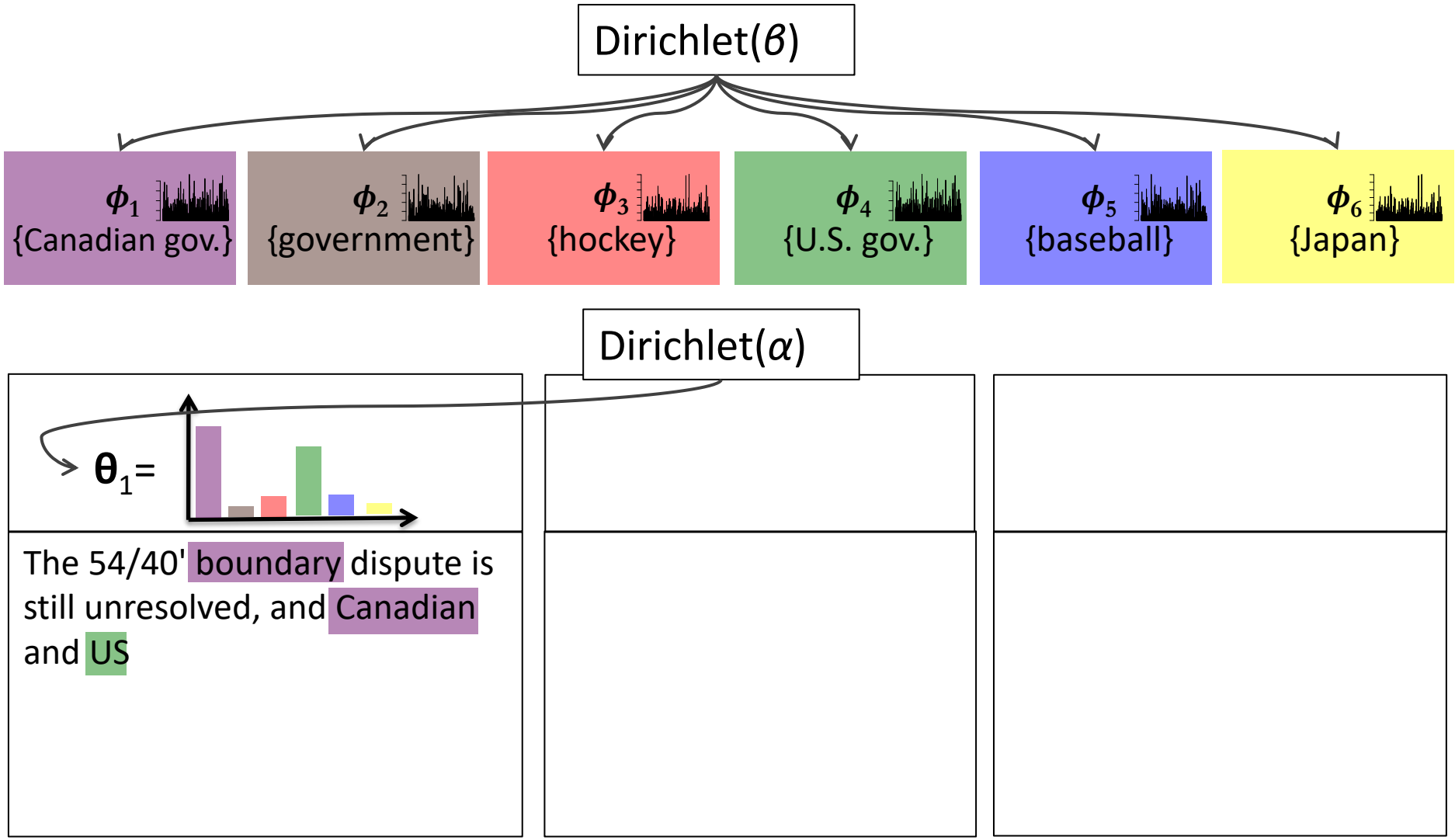
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



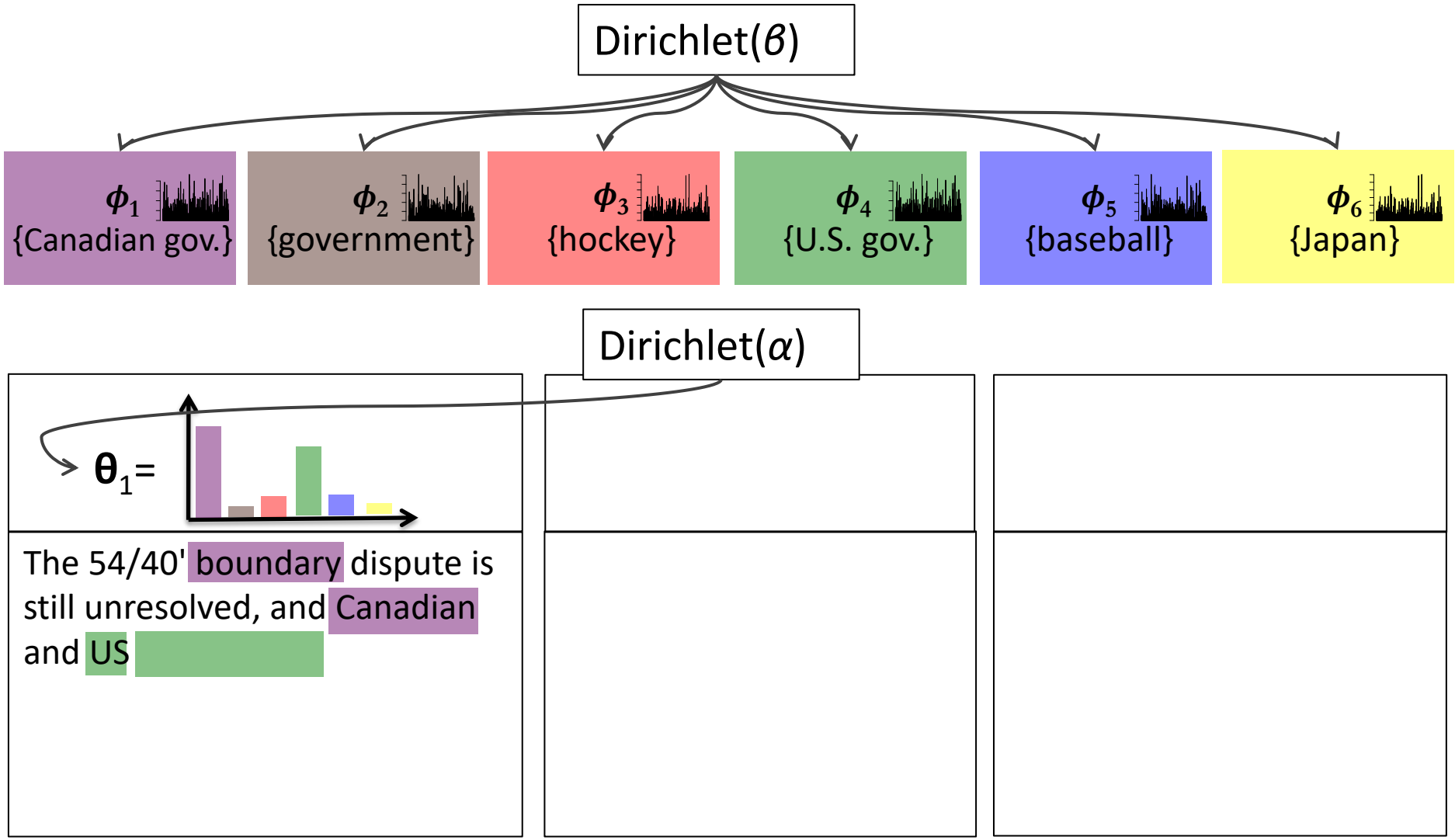
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



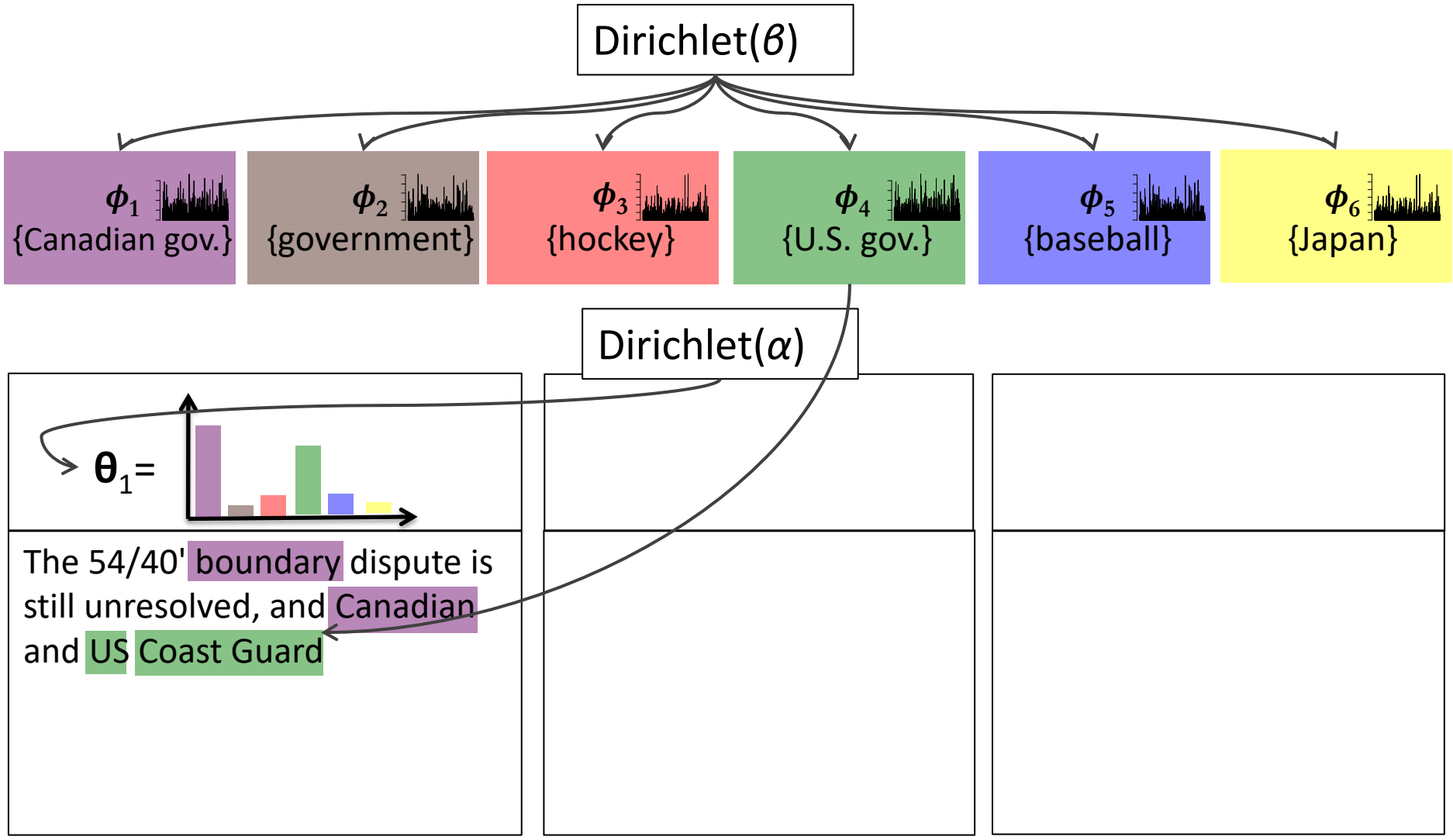
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



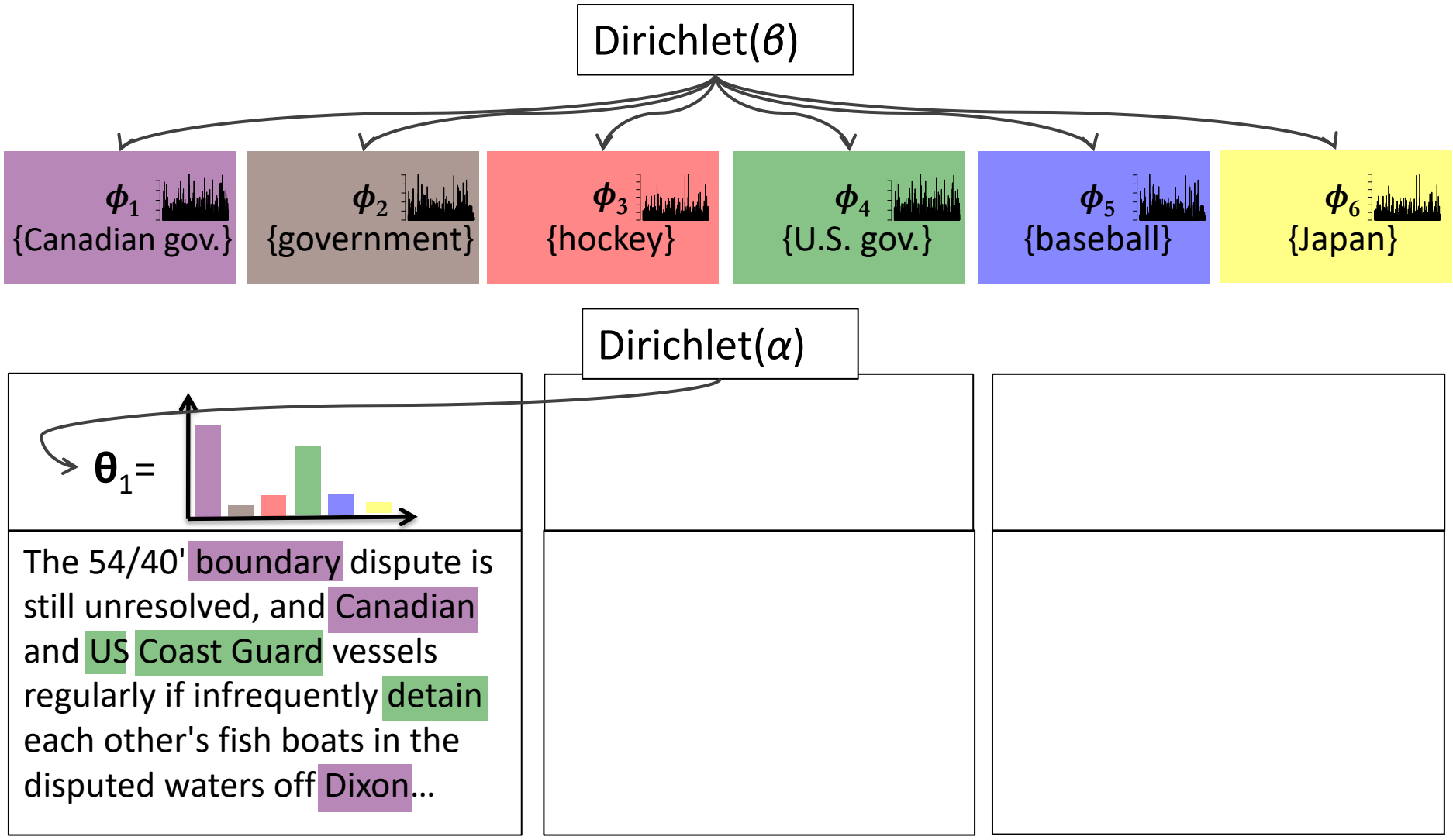
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



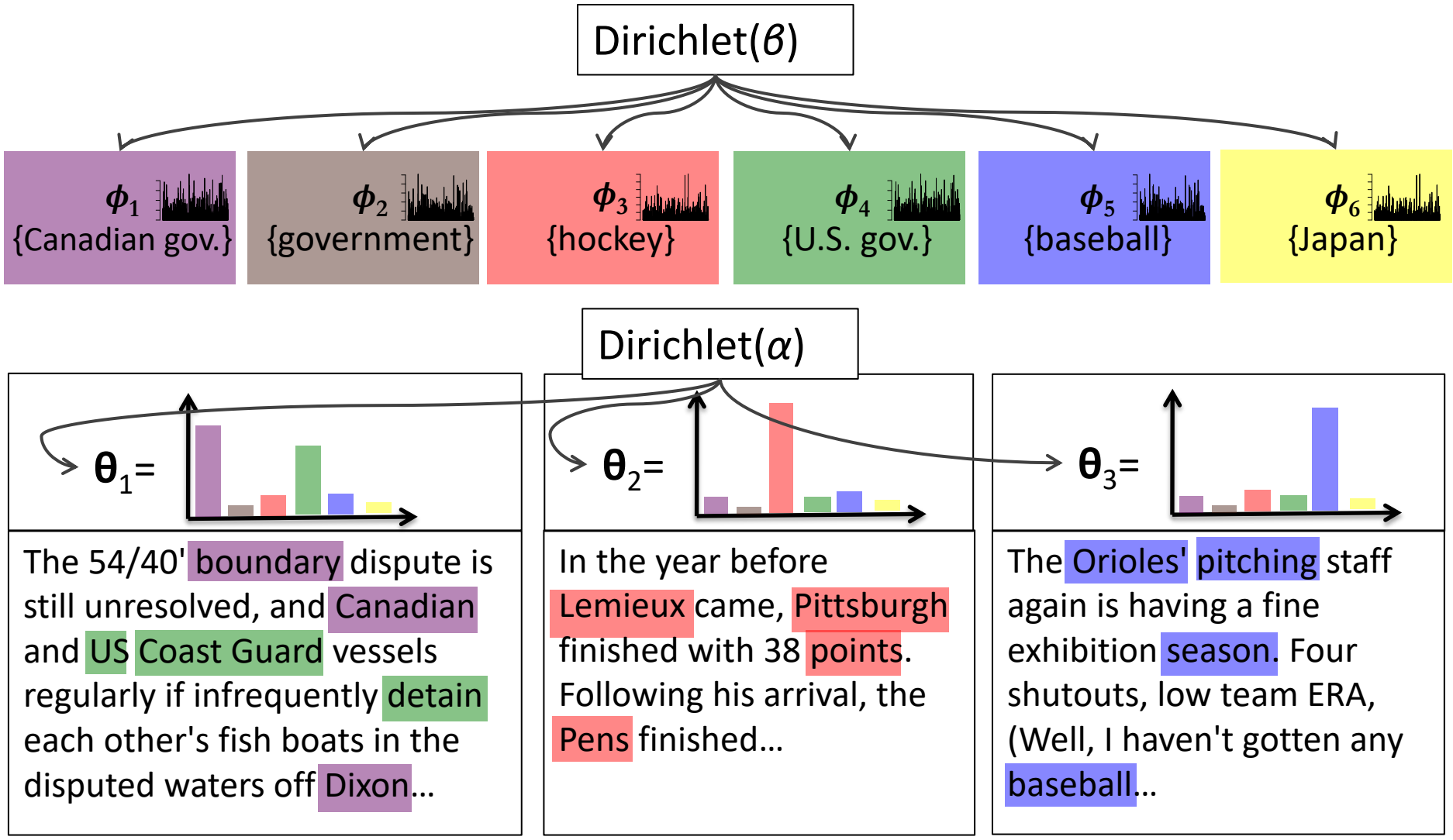
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



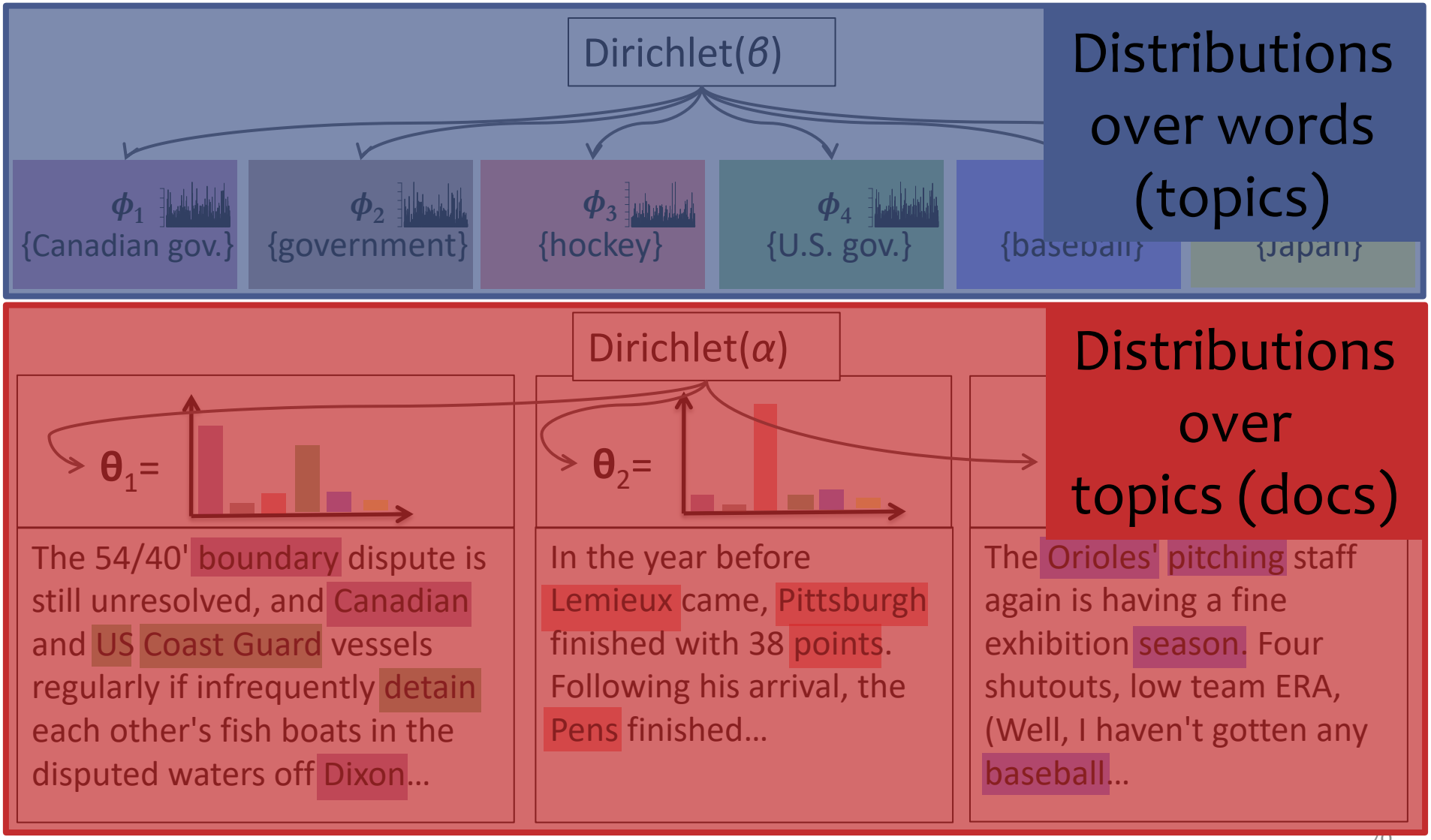
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



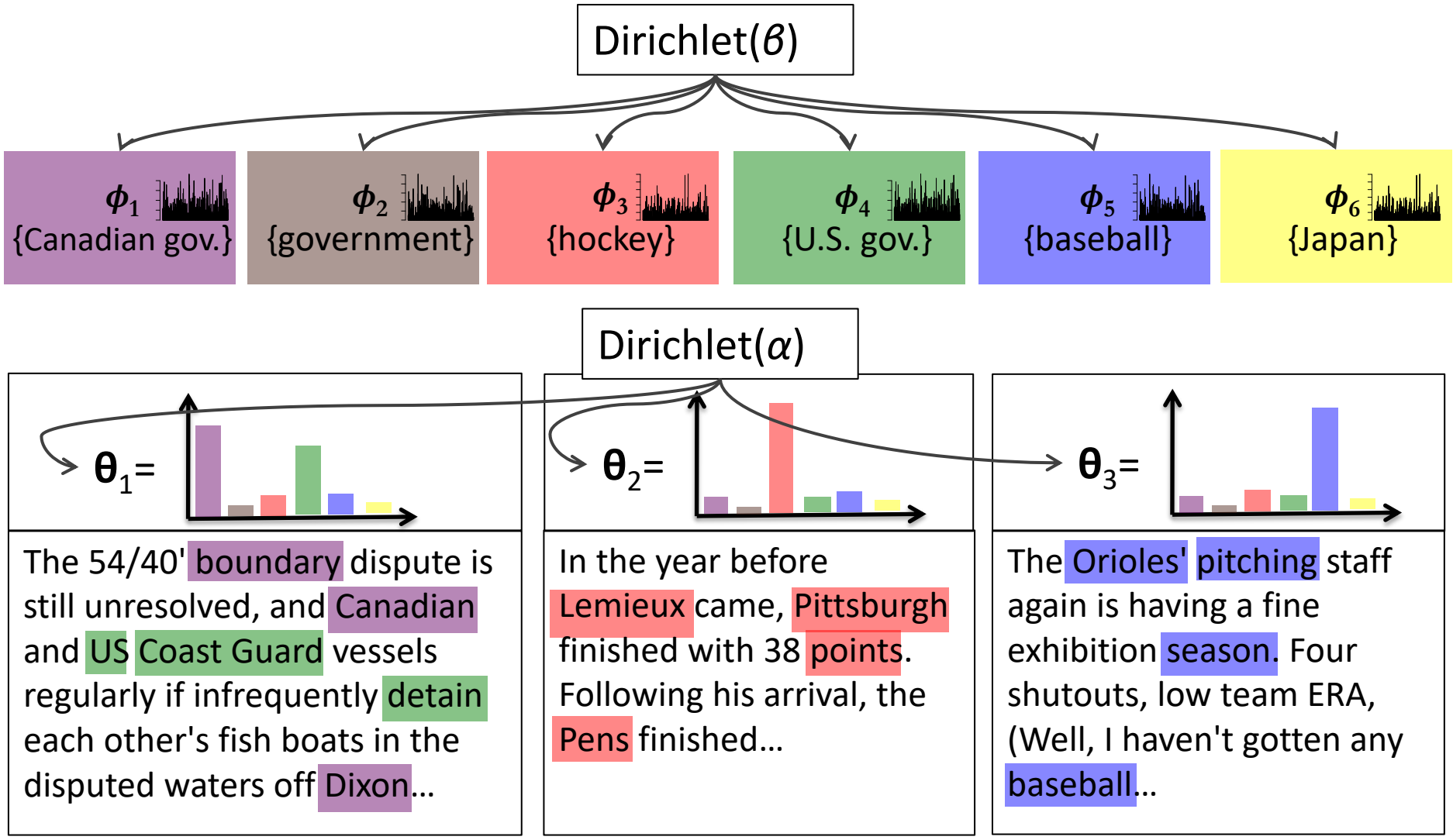
(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling



(Blei, Ng, & Jordan, 2003)

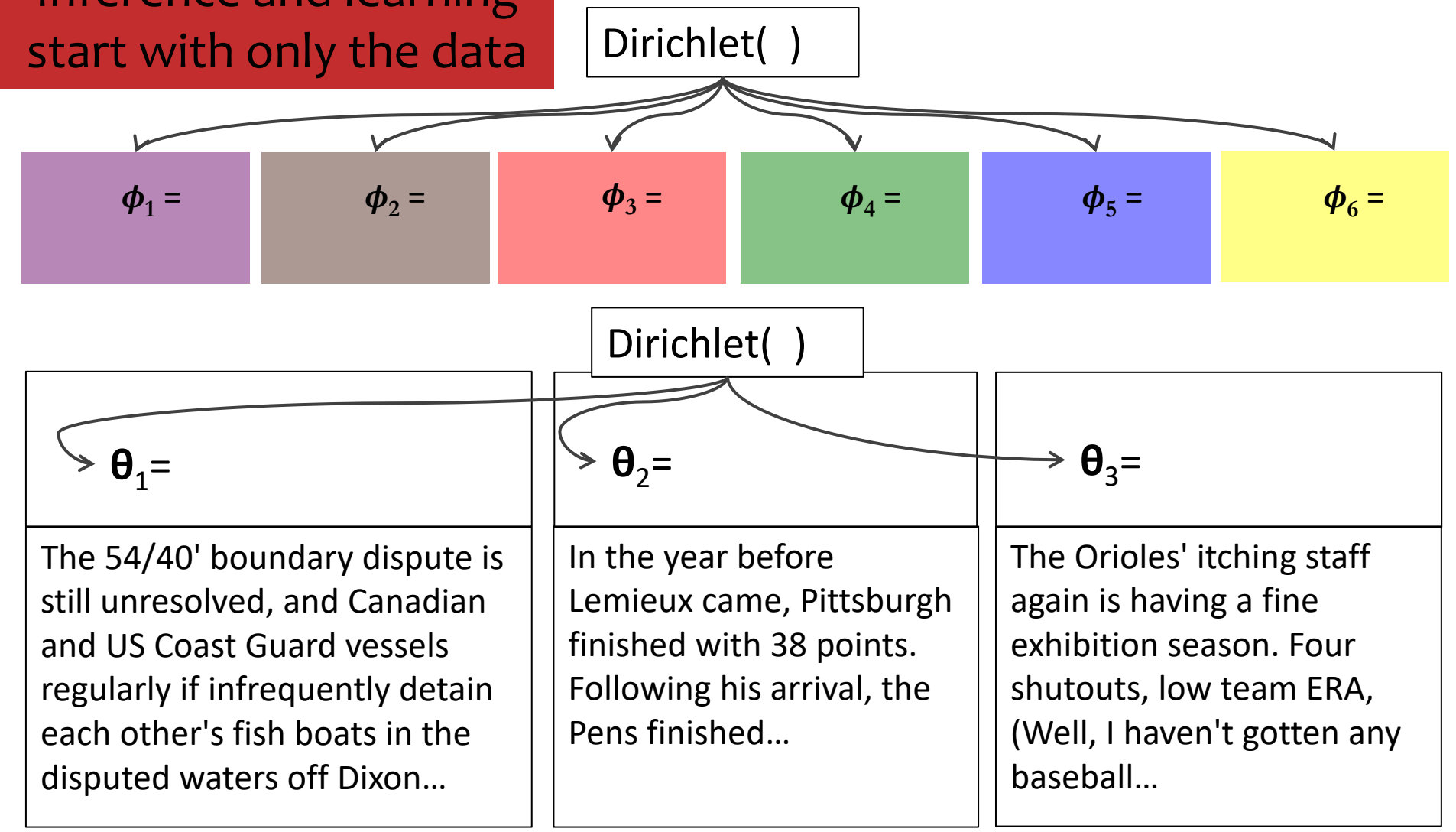
LDA for Topic Modeling



(Blei, Ng, & Jordan, 2003)

LDA for Topic Modeling

Inference and learning start with only the data



Latent Dirichlet Allocation

Questions:

- Is this a believable story for the generation of a corpus of documents?

- Why might it work well anyway?

Latent Dirichlet Allocation

Why does LDA "work"?

- LDA trades off two goals.
  1. For each document, allocate its words to as few topics as possible.
  2. For each topic, assign high probability to as few terms as possible.
- These goals are at odds.
  - Putting a document in a single topic makes #2 hard:
    All of its words must have probability under that topic.
  - Putting very few words in each topic makes #1 hard:
    To cover a document's words, it must assign many topics to it.
- Trading off these goals finds groups of tightly co-occurring words.

Slide from David Blei, MLSS 2012

Latent Dirichlet Allocation

How does this relate to my other favorite model for capturing low-dimensional representations of a corpus?

- Builds on latent semantic analysis (Deerwester et al., 1990; Hofmann, 1999)
- It is a mixed-membership model (Erosheva, 2004).
- It relates to PCA and matrix factorization (Jakulin and Buntine, 2002)
- Was independently invented for genetics (Pritchard et al., 2000)

Slide from David Blei, MLSS 2012

Outline

- **Applications of Topic Modeling**
- **Review: Latent Dirichlet Allocation (LDA)**
  1. Beta-Bernoulli
  2. Dirichlet-Multinomial
  3. Dirichlet-Multinomial Mixture Model
  4. LDA
- **Contrast of methods for Inference / Learning**
  - Exact inference
  - EM
  - Monte Carlo EM
  - Gibbs sampler
  - Collapsed Gibbs sampler
- **Extensions of LDA**
  - Correlated topic models
  - Dynamic topic models
  - Polylingual topic models
  - Supervised LDA

Unsupervised Learning

Three learning paradigms:

1. Maximum likelihood

$$\arg\max_{\theta} p(X|\theta)$$

1. Maximum a posteriori (MAP)

$$\arg\max_{\theta} p(\theta|X) \propto p(X|\theta)p(\theta)$$

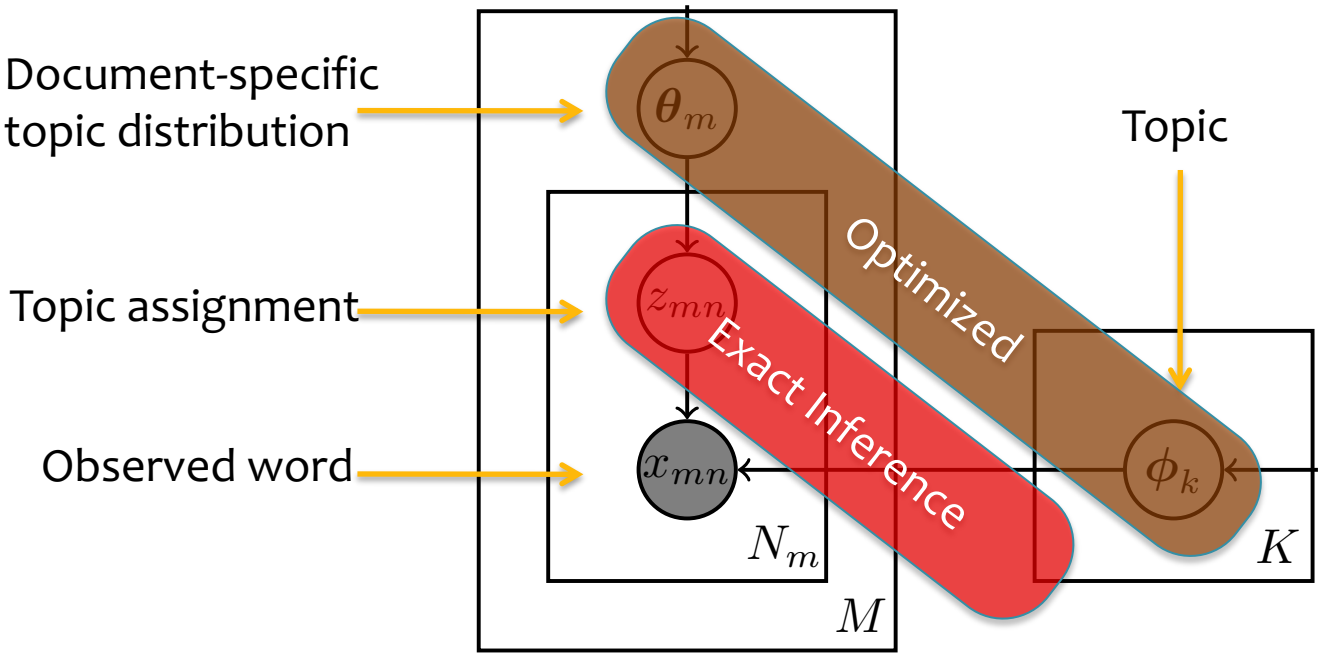
1. Bayesian approach

Estimate the posterior:

$$p(\theta|X) = \ \ldots$$

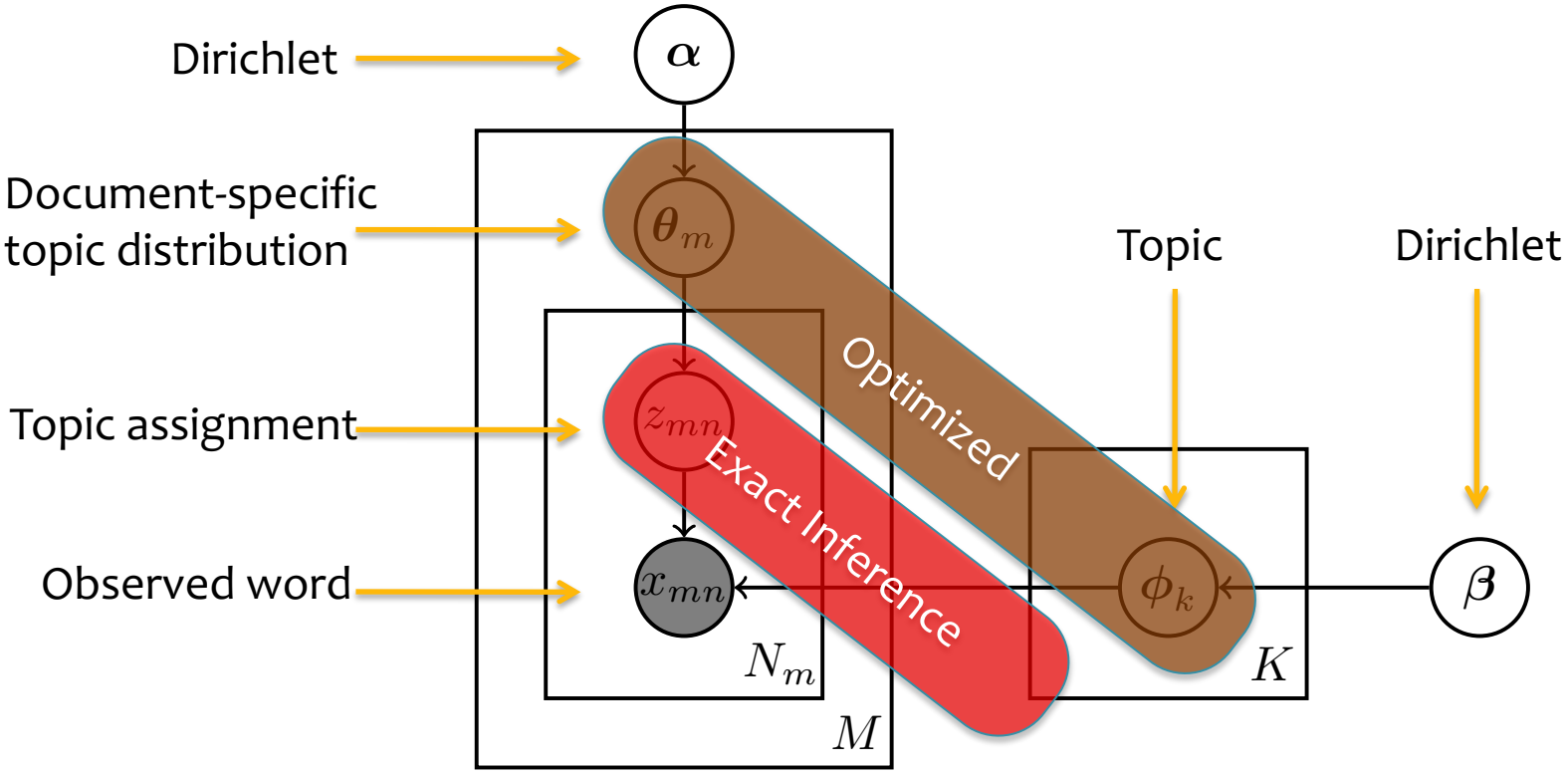
LDA Inference

- Standard EM (Maximum Likelihood)



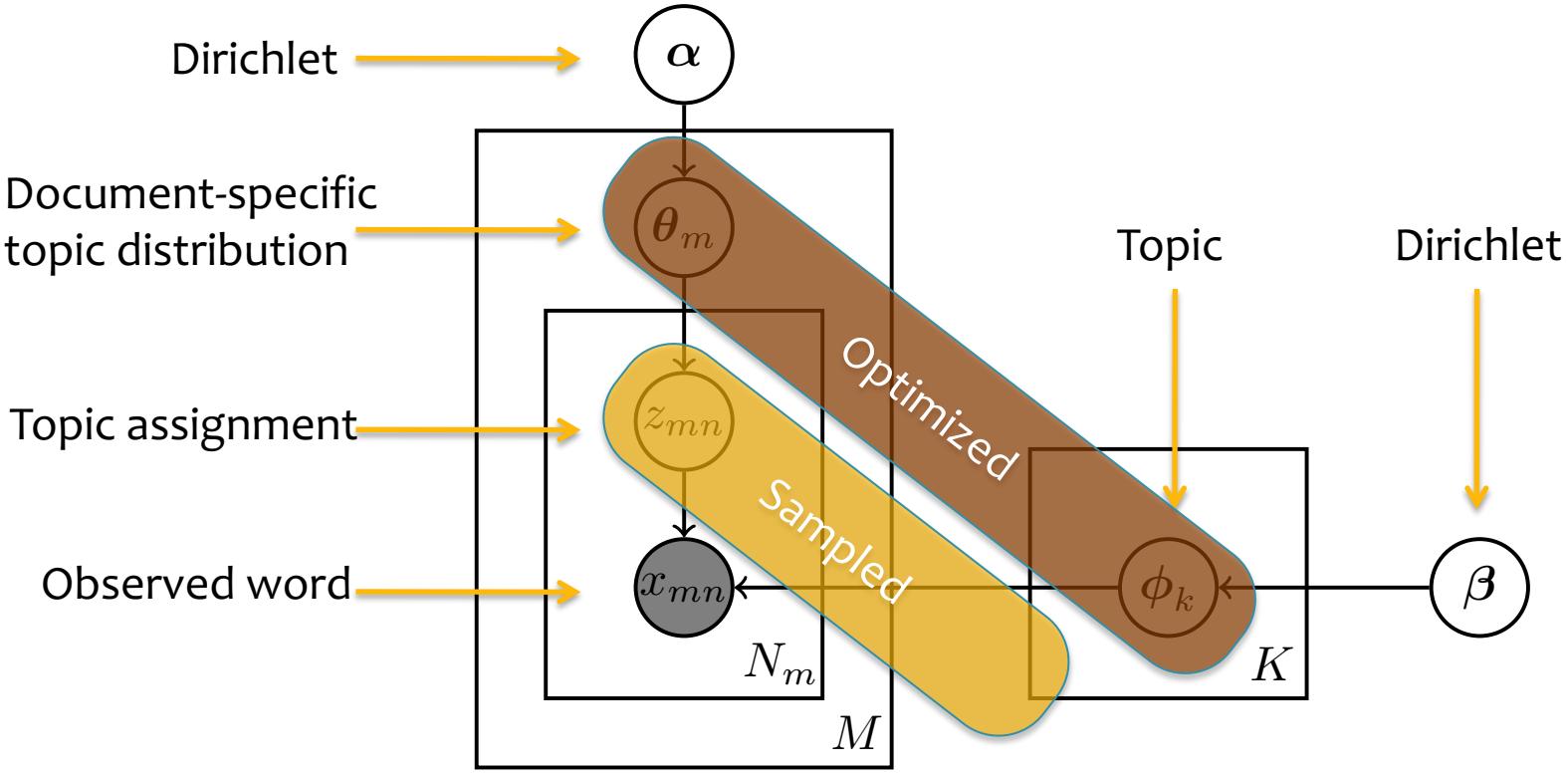
LDA Inference

- Standard EM (MAP)



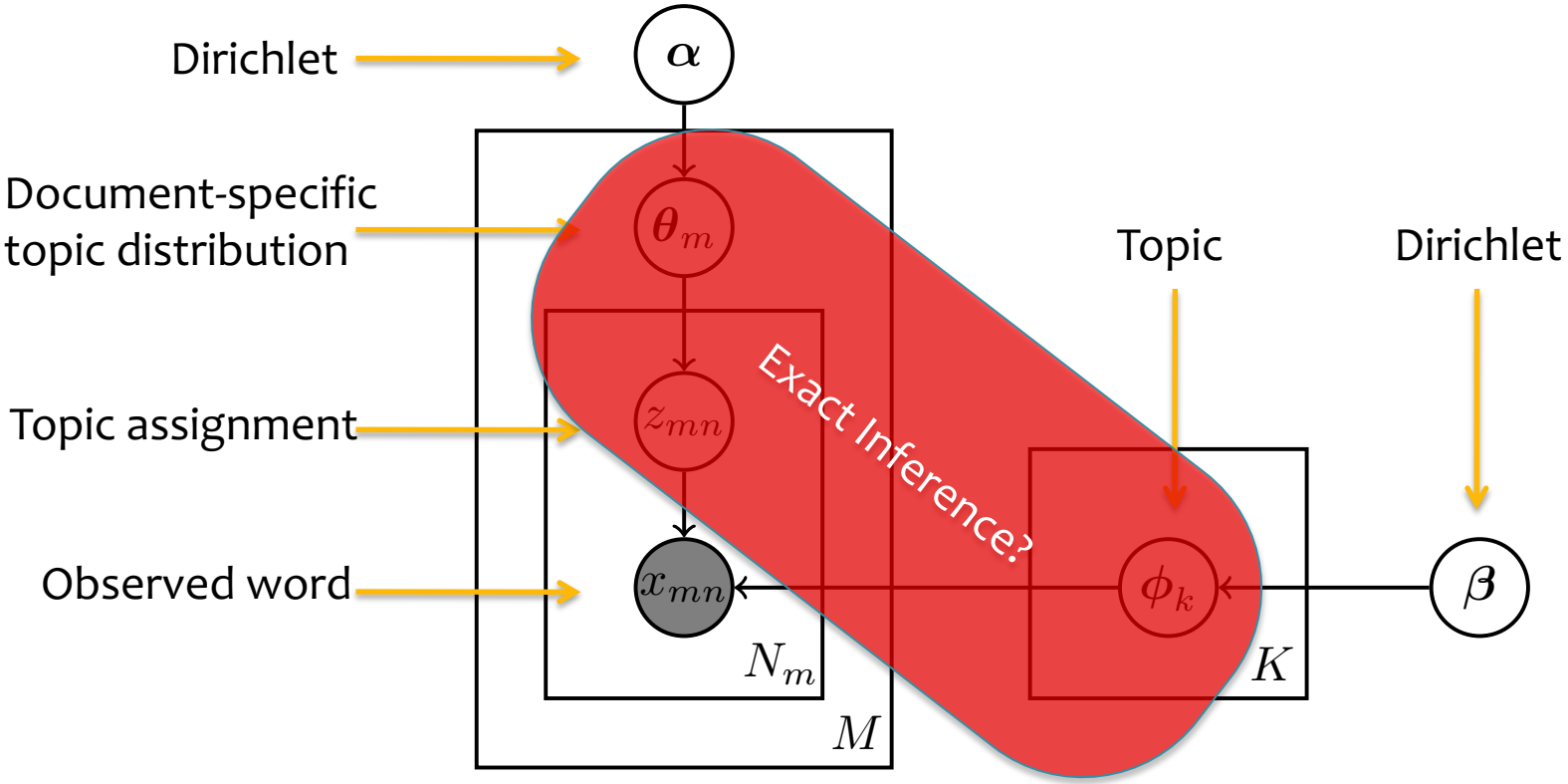
LDA Inference

- Monte Carlo EM



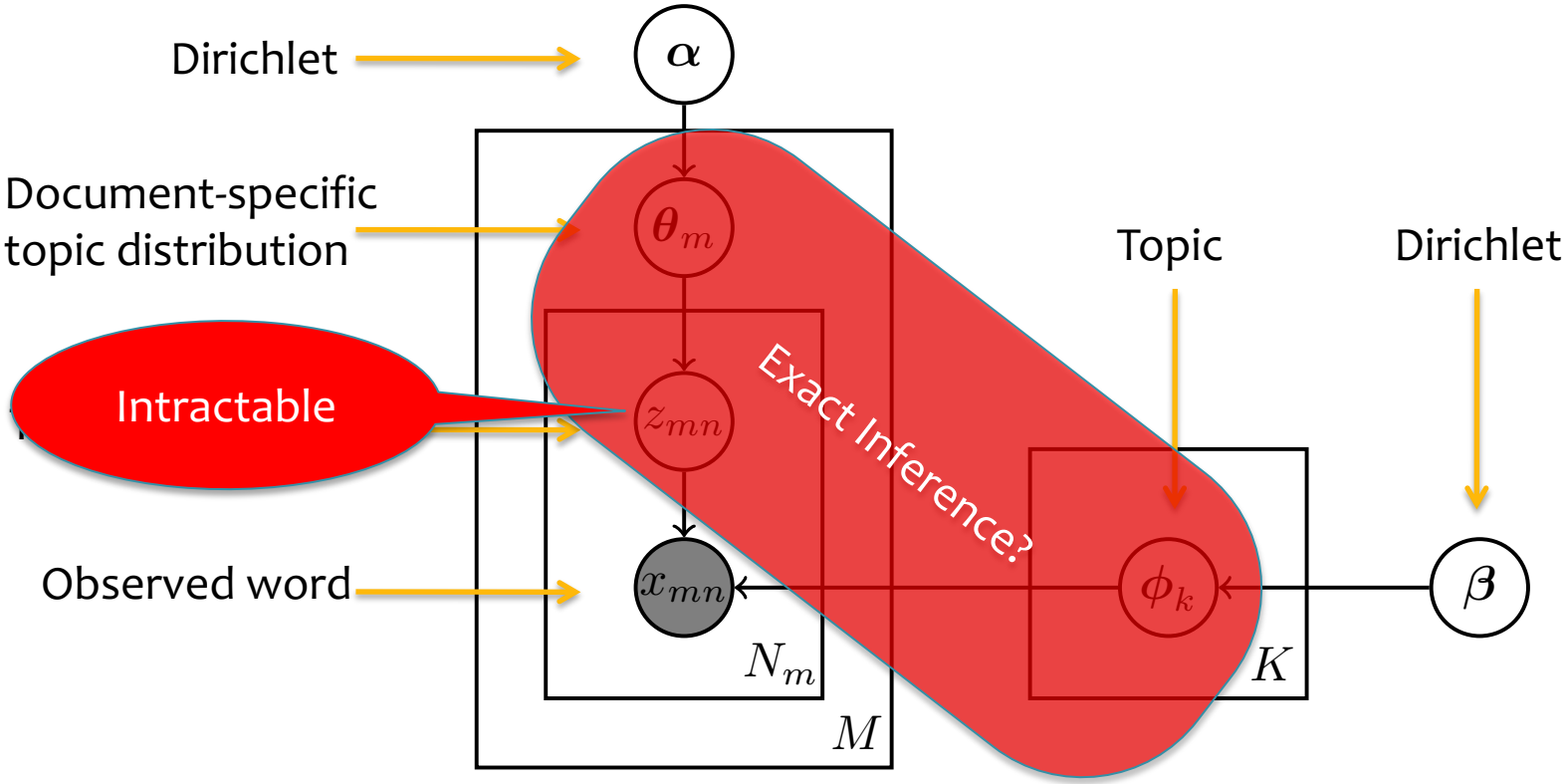
LDA Inference

- Bayesian Approach



LDA Inference

- Bayesian Approach

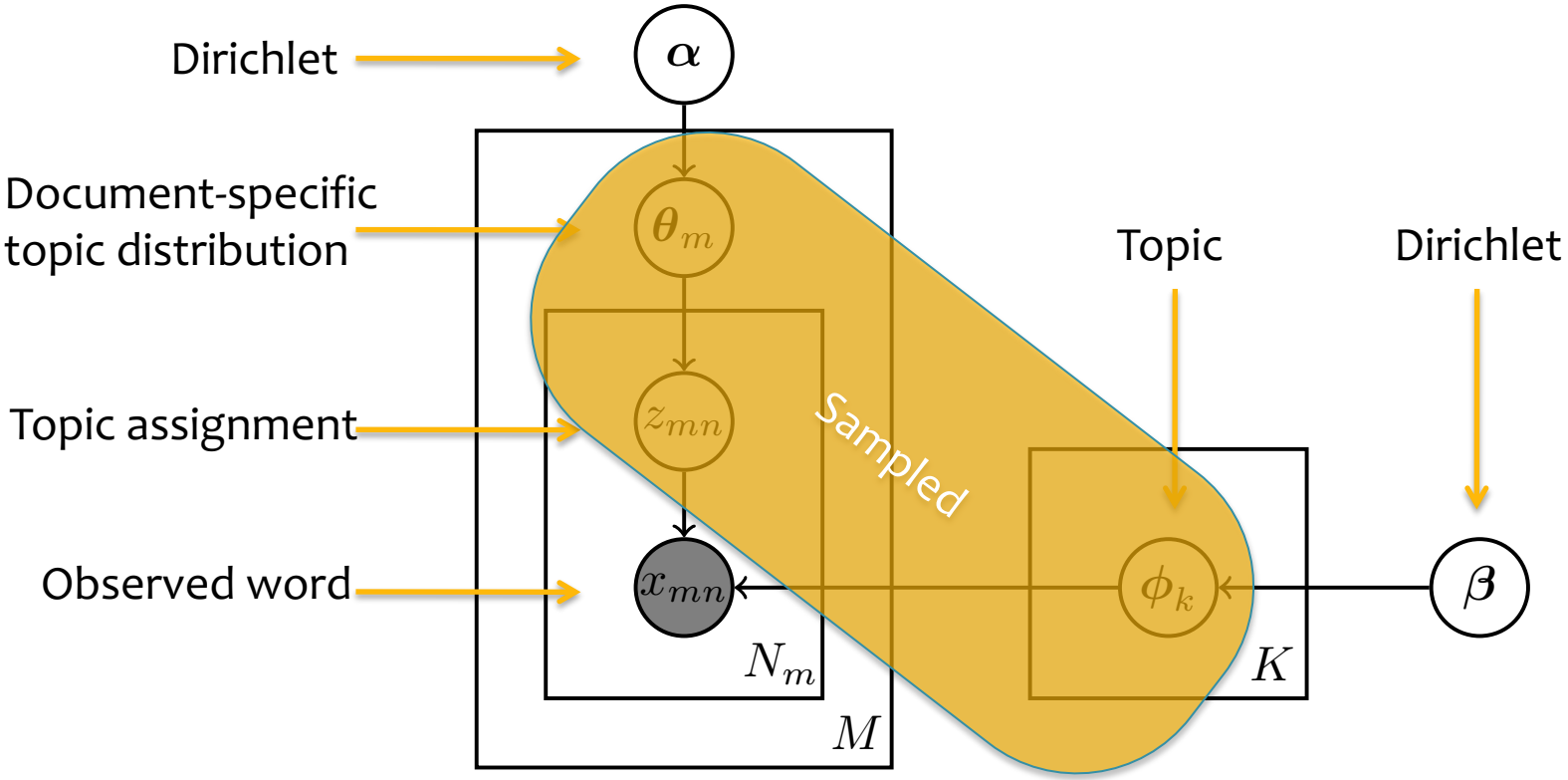


Exact Inference in LDA

- Exactly computing the posterior is intractable in LDA
  - Junction tree algorithm: exact inference in general graphical models
    1. "moralization" converts directed to undirected
    2. "triangulation" breaks 4-cycles by adding edges
    3. Cliques arranged into a junction tree
  - Time complexity is exponential in size of cliques
  - LDA cliques will be large (at least O(# topics)), so complexity is $O(2^{\#\ \text{topics}})$
- Exact MAP inference in LDA is NP-hard for a large number of topics (Sontag & Roy, 2011)

LDA Inference

- Explicit Gibbs Sampler



LDA Inference

- Collapsed Gibbs Sampler