Machine Learning (CS 181): 15. EM and Topic Modeling

David C. Parkes and Sasha Rush

Spring 2017

The Expectation Maximization algorithm

Initialize parameters. Then repeat:

- Expectation step: estimate class distribution $p(\mathbf{z}_i \mid \mathbf{x}_i)$ for each latent variable (given current model parameters).
- Maximization step: update parameters $\boldsymbol{\theta}$, $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, to maximize the expected complete-data log likelihood.

Until convergence. Alternate between predicting the class for each example, and updating the parameters of the model.

Today's class: A general and powerful idea not specific to mixture of Gaussians

The Expectation Maximization algorithm

Initialize parameters. Then repeat:

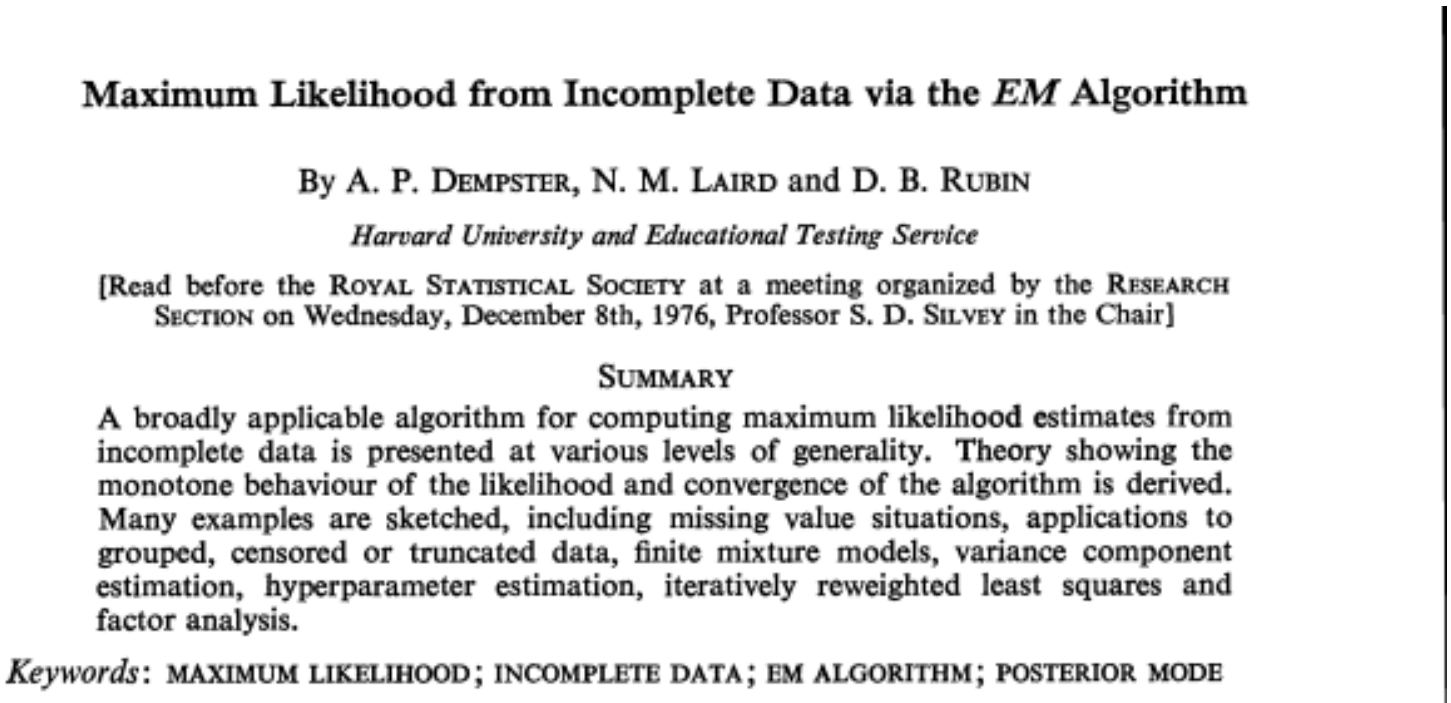
- Expectation step: estimate class distribution $p(\mathbf{z}_i \mid \mathbf{x}_i)$ for each latent variable (given current model parameters).
- Maximization step: update parameters $\boldsymbol{\theta}$, $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, to maximize the expected complete-data log likelihood.

Until convergence. Alternate between predicting the class for each example, and updating the parameters of the model.

Today's class: A general and powerful idea not specific to mixture of Gaussians

Expectation Maximization: A Little History

Dempster, Laird, and Rubin (1977) formalized and popularized this approach



Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. Dempster, N. M. Laird and D. B. Rubin

Harvard University and Educational Testing Service

[Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, December 8th, 1976, Professor S. D. Silvey in the Chair]

Summary

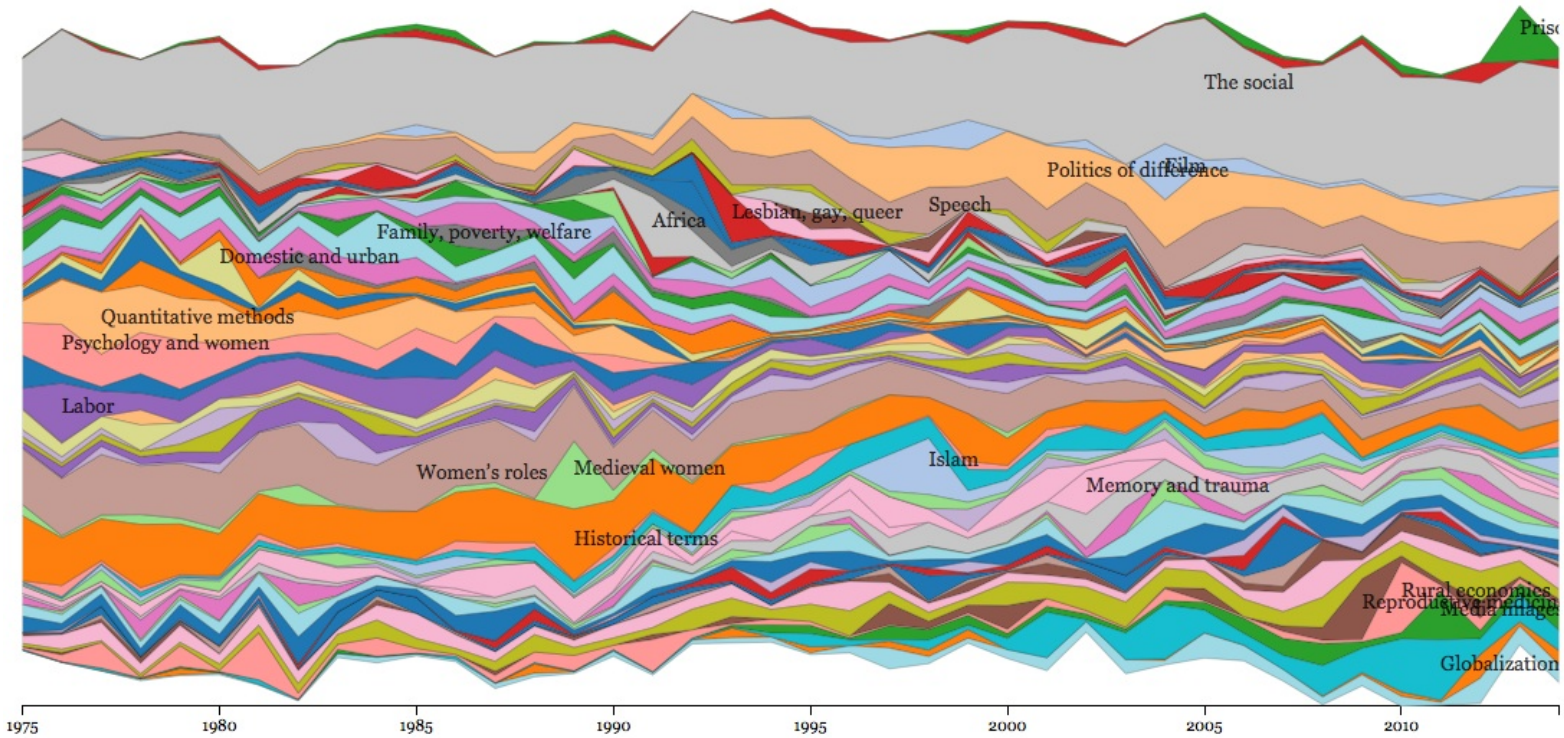
A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

Keywords: Maximum likelihood; incomplete data; EM algorithm; posterior mode

Special cases (such as Baum-Welch) in use much earlier.

Motivating Application: Topic Modeling

Exploratory analysis of *Signs: Journal of Women in Culture and Society*



[http://signsat40.signsjournal.org]

Contents

Contents

Context: Discrete Distributions

- Recall discrete distributions, assume $\mathbf{x}$ is indicators or counts.
- Examples:
  - Univariate Bernoulli case: $\mathbf{x}$ is coin-flips
  - Multinomial: $\mathbf{x}$ was counts over events.
  - Naive Bayes: $\mathbf{x}$ was counts, $\mathbf{y}$ was class indicator.
- Could often perform MLE in closed-form.

Discrete Mixture Models

- Today: continue exploring latent-variable mixture models.
- This means we do not have any $\mathbf{y}$'s, instead use latent-variable.
- High-level idea (analogous to mixture of Gaussians):
  - Latent-variable $\mathbf{z}$ determines which parameters (or combination of parameters) to use.
  - Observed data $\mathbf{x}$ is generated based on this variable.
- Our goal is to infer this mixture.

Example 1: Unsupervised Sentiment Detection

Our data $\mathbf{x}$ is movie reviews that are good or bad.

A thoughtful, provocative, insistently humanizing film.

Occasionally melodramatic, it's also extremely effective.

A sentimental mess that never rings true.

- However, we do not know the labels of the data (good/bad).
- Hope: latent variables $\mathbf{z}$ capture distinction to separate data.

Example 2: Topic Modeling

- Observe *documents* consisting of *words*.
- Determine *topics* of each document.
- Here a *topic* is a distribution over words, and each *document* has a distribution over *topics*

Example Applications:

- Group blog posts into tags based on their content.
- Group medical records into the diseases they cover.
- Group novels into the themes that are covered.

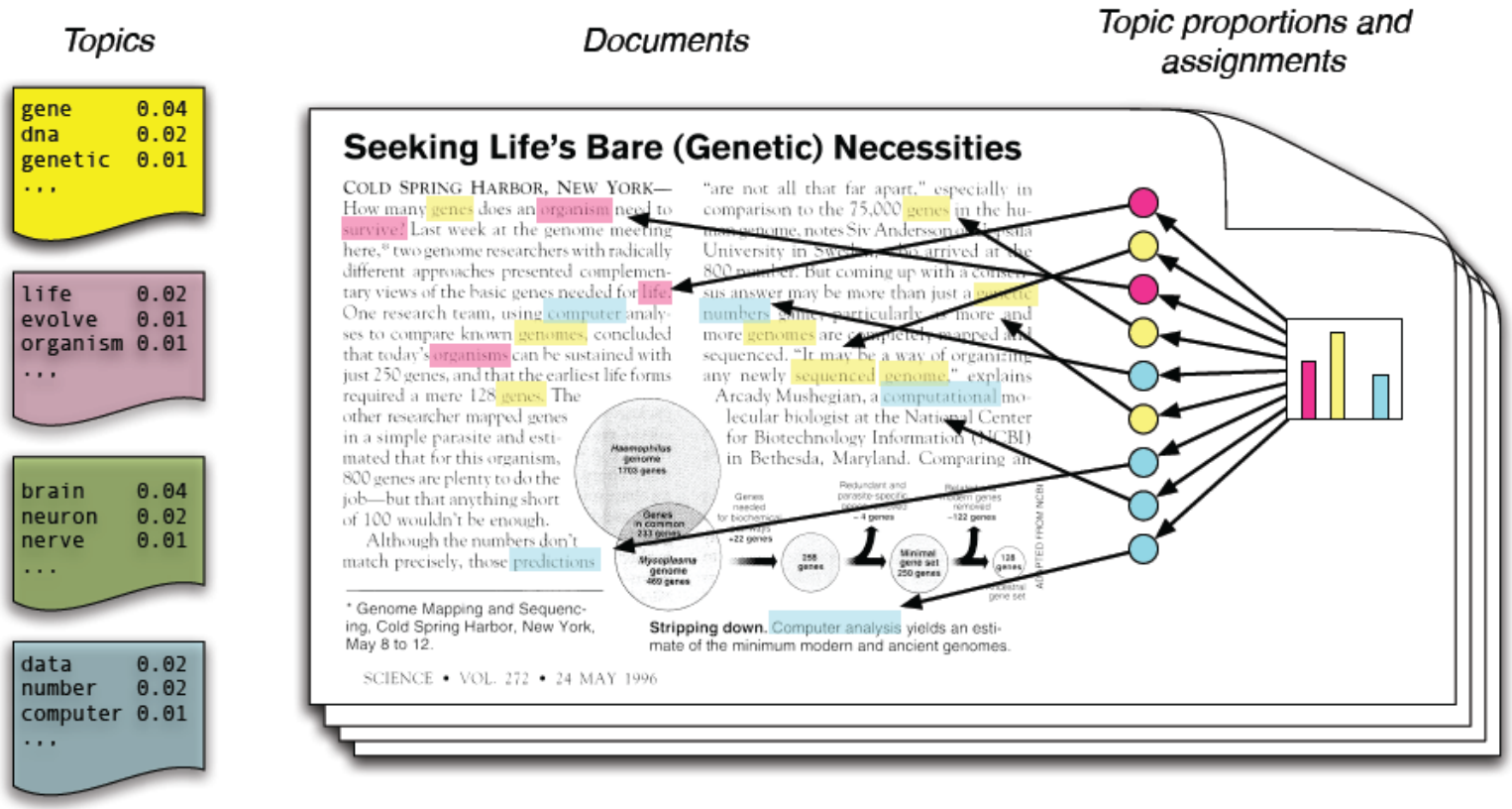
Example 2: Topic Modeling

- Observe *documents* consisting of *words*.
- Determine *topics* of each document.
- Here a *topic* is a distribution over words, and each *document* has a distribution over *topics*

Example Applications:

- Group blog posts into tags based on their content.
- Group medical records into the diseases they cover.
- Group novels into the themes that are covered.

Topic Model in Practice



[Blei, 2011]

Contents

Detour: Joint Distributions

- Consider binary-class multinomial Naive Bayes

$$p(\mathbf{x}, \mathbf{y} \mid \mathbf{w})$$

- Recall in this model
  1. Generate using class distribution $p(\mathbf{y} \mid \theta)$
  2. Then generate data using class conditional $p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$
- This is a specific factorization of the joint distribution $p(\mathbf{x}, \mathbf{y} \mid \mathbf{w})$
- (Here we are taking a Bayesian view and conditioning on our parameters as random variables.)

Detour: Joint Distributions

- Consider binary-class multinomial Naive Bayes

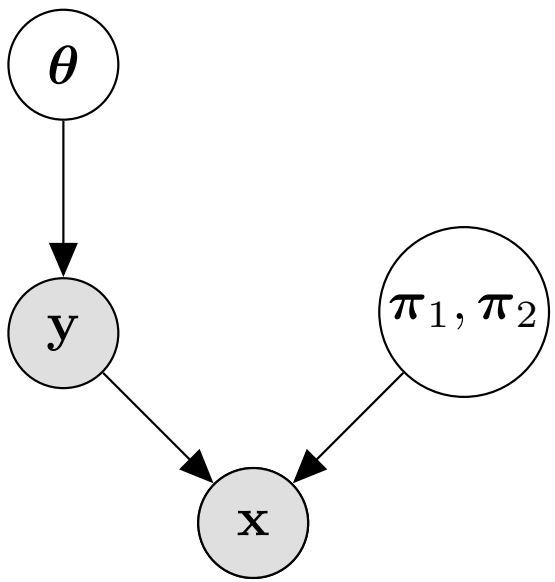
$$p(\mathbf{x}, \mathbf{y} \mid \mathbf{w})$$

- Recall in this model
  1. Generate using class distribution $p(\mathbf{y} \mid \theta)$
  2. Then generate data using class conditional $p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$
- This is a specific factorization of the joint distribution $p(\mathbf{x}, \mathbf{y} \mid \mathbf{w})$
- (Here we are taking a Bayesian view and conditioning on our parameters as random variables.)

Directed Graphical Model: Naive Bayes (Single Point)

- Directed graphical models provide a language for describing factorizations of distributions.

$$p(\mathbf{x}, \mathbf{y} \mid \mathbf{w}) = p(\mathbf{y} \mid \theta) p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$$

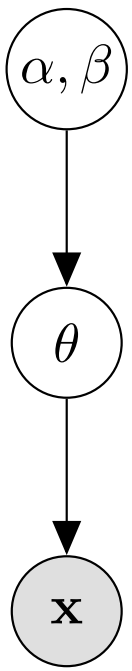


Graphical model of Naive Bayes. Each node is conditioned on its parents, gray nodes are observed at training.

Directed Graphical Model: Beta-Bernoulli

- Can use similar notation to describe other distributions, and priors over parameters.

$$p(\mathbf{x} \mid \alpha, \beta) = \int p(\theta \mid \alpha, \beta) p(\mathbf{x} \mid \theta) d\boldsymbol{\theta}$$

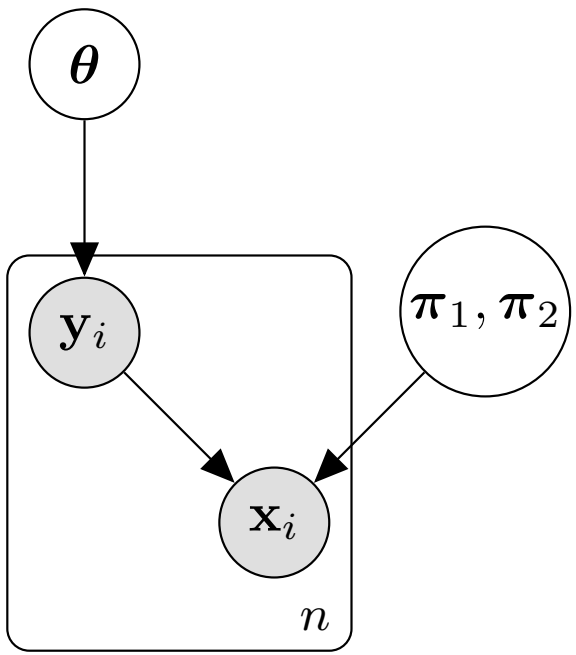


Graphical model of Beta-Bernoulli. Here hyperparameters α, β and parameters θ are directly represented in the model.

Directed Graphical Model: Naive Bayes (All Data)

- Can also describe full data $D = \{\mathbf{x}_i, \mathbf{y}_i\}$ using sums.

$$p(D \,|\, \mathbf{w}) = \prod_i p(\mathbf{x}_i, \mathbf{y}_i \,|\, \mathbf{w}) = \prod_i p(\mathbf{y}_i \,|\, \boldsymbol{\theta}) p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$$



Graphical model of complete data. *Plate* notation indicates a repeated copies of the latent variables within.

Review: Latent Variable Notation for Unsupervised

- Observed variables (given)

$$D = \{\mathbf{x}_i\}_{i=1}^n \ \mathbf{x}_i \in \mathbb{R}^m$$

- Latent variables (not given)

$$\{\mathbf{z}_i\}_{i=1}^n$$

Complete data negative log-likelihood:

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n} \ln p(\mathbf{x}_i, \mathbf{z}_i \,|\, \mathbf{w})$$

The Gaussian Mixture Model

- Observed data (given): $\mathbf{x}_i \in \mathbb{R}^m$
- Latent variable (not given): $\mathbf{z}_i \in \{C_k\}_{k=1}^c$, for c clusters
- Class distribution (categorical / generalized Bernoulli):

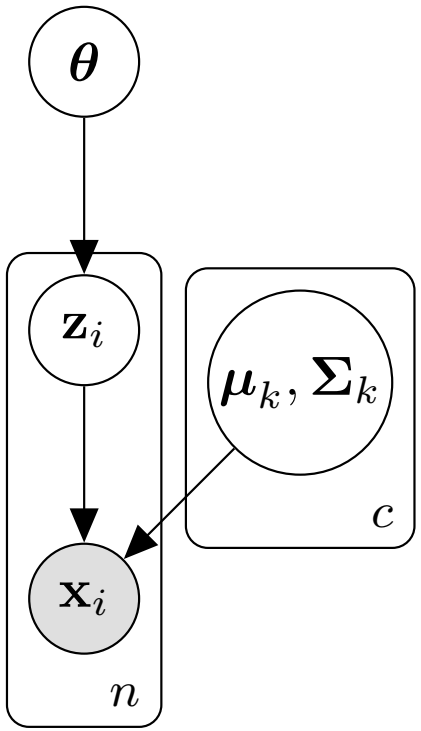
$$p(\mathbf{z} = C_k) = \theta_k, \quad \text{for } k \in \{1, \ldots, c\}$$

- Class conditional distribution (Normal):

$$p(\mathbf{x}|\mathbf{z} = C_k) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k), \quad \text{for } k \in \{1, \ldots, c\}$$

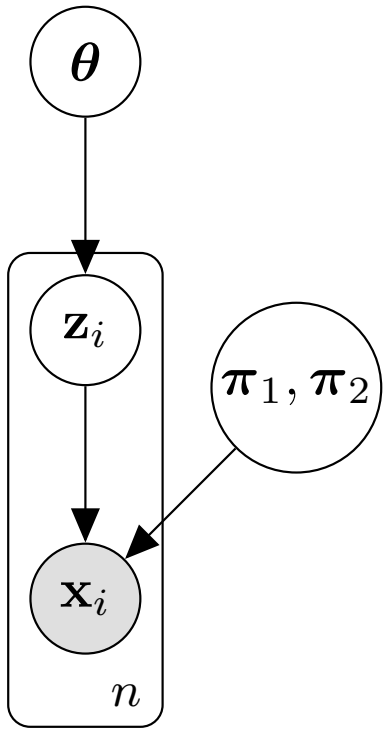
- Parameters of the model: $\boldsymbol{\theta}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^c$

Graphical Model: Gaussian Mixture (last class)



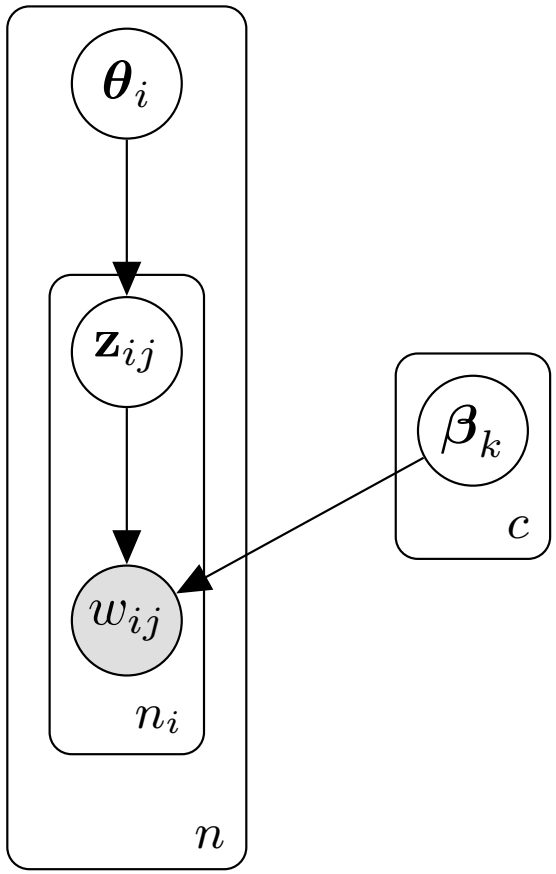
Graphical model from GMM. Note that there is one non-observed $\mathbf{z}_i$ (latent class) for each input data, and one unobserved class mean $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ covariance for each class.

Today: Mixture of Multinomials (Model 1)



Latent variable version of binary-class Naive bayes.

Today: Topic Model (Latent Dirichlet Allocation)

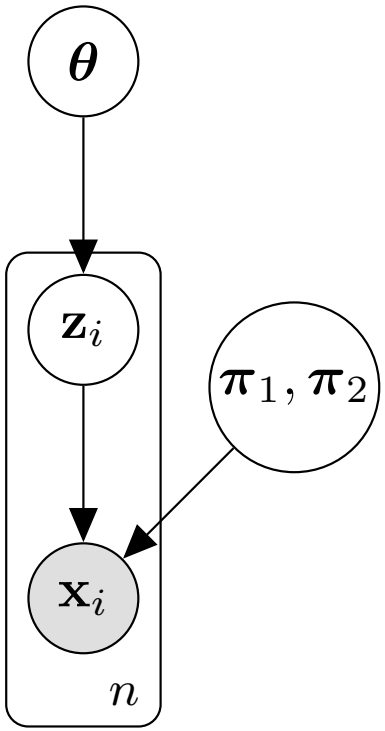


Final Note: Inference

- Inference is the process of determining the values of unobserved nodes in these graphs.
- Sometimes the inference can be done in closed-form (Naive Bayes), other times it is requires algorithms like EM (GMM).
- Inference procedure is separate from graphical model, and there are often many different choices, e.g.
  - Particle Fitering
  - Gibbs Sampling
  - Message Passing (coming soon)
  - Variational Inference

Contents

Mixture of Multinomials



Data log-likelihood with $c = 2$ (again not easy to solve):

$$\sum_{i=1}^{n} \ln p(\mathbf{x}_i; \mathbf{w}) = \sum_{i=1}^{n} \ln \left(\sum_{k=1}^{c} \theta_k \, p(\mathbf{x}_i \,|\, \mathbf{z}_i = C_k, \boldsymbol{\pi}_k) \right),$$

Expected Data Log-Likelihood

Recall $\mathcal{L}(\mathbf{w})$ is complete-data log likelihood. Define the expected complete-data log likelihood using q

$$\begin{aligned}\mathbf{E}_{\mathbf{Z}}[\mathcal{L}(\mathbf{w})] &= \mathbf{E}_{\mathbf{Z}}[\sum_{i=1}^{n} \ln(p(\mathbf{x}_i, \mathbf{z}_i; \mathbf{w}))] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c} q_{ik} \ln \theta_k + \sum_{i=1}^{n}\sum_{k=1}^{c} q_{ik} \ln p(\mathbf{x}_i|\mathbf{z}_i = C_k, \boldsymbol{\pi}_k).\end{aligned}$$

Where we use class probs

$$\mathbf{q}_i = p(\mathbf{z}_i \,|\, \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$$

(Very similar process as last lecture. Gaussian replaced by multinomial)

Expected Data Log-Likelihood

Recall $\mathcal{L}(\mathbf{w})$ is complete-data log likelihood. Define the expected complete-data log likelihood using q

$$\begin{aligned}\mathbf{E}_{\mathbf{Z}}[\mathcal{L}(\mathbf{w})] &= \mathbf{E}_{\mathbf{Z}}[\sum_{i=1}^{n} \ln(p(\mathbf{x}_i, \mathbf{z}_i; \mathbf{w}))] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c} q_{ik} \ln \theta_k + \sum_{i=1}^{n}\sum_{k=1}^{c} q_{ik} \ln p(\mathbf{x}_i|\mathbf{z}_i = C_k, \boldsymbol{\pi}_k).\end{aligned}$$

Where we use class probs

$$\mathbf{q}_i = p(\mathbf{z}_i \,|\, \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$$

(Very similar process as last lecture. Gaussian replaced by multinomial)

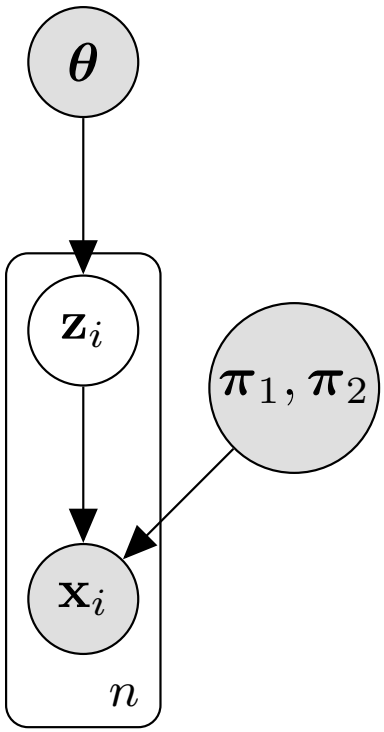
Expectation-Maximization

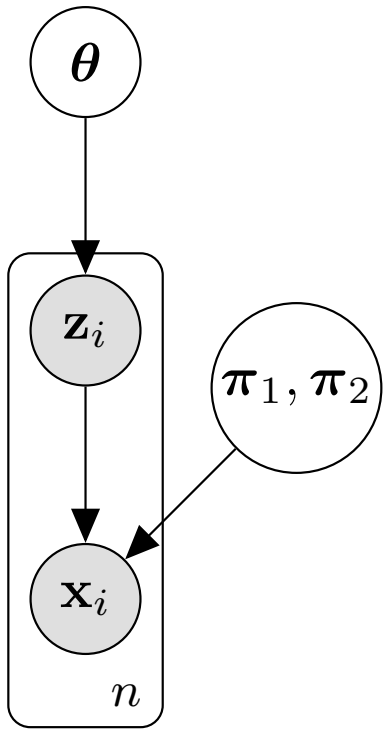
- We will focus on using Expectation-Maximization (EM).
  - Expectation step: Estimates class given observed for all i,

$$\mathbf{q}_i = p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$$

  - Maximization step: Re-estimate $\boldsymbol{\theta}$ and $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2$ using the $\mathbf{q}$'s

Expectation-Maximization





Loop

1. Compute $\mathbf{q}_{ik} \propto \theta_k \prod_j \pi_{kj}^{x_{ik}}$ using Bayes' rule (see Naive Bayes lecture.)

2. Reestimate θ and $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$ by maximizing expected likelihood.

Maximization Step

Recall for Naive bayes we had the following count-based updates:

$$\begin{aligned} \theta_1 &= \frac{\sum_{i=1}^{n} y_{i1}}{n} \\ \boldsymbol{\pi}_1 &= \frac{\sum_{i=1}^{n} y_{i1}\mathbf{x}_i}{\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i1}x_{ij}} \end{aligned}$$

Computing our m-step will lead to using *expected* counts from q.

$$\begin{aligned} \theta_1 &= \frac{\sum_{i=1}^{n} q_{i1}}{n} \\ \boldsymbol{\pi}_1 &= \frac{\sum_{i=1}^{n} q_{i1}\mathbf{x}_i}{\sum_{i=1}^{n}\sum_{j=1}^{m} q_{i1}x_{ij}} \end{aligned}$$

Intuition

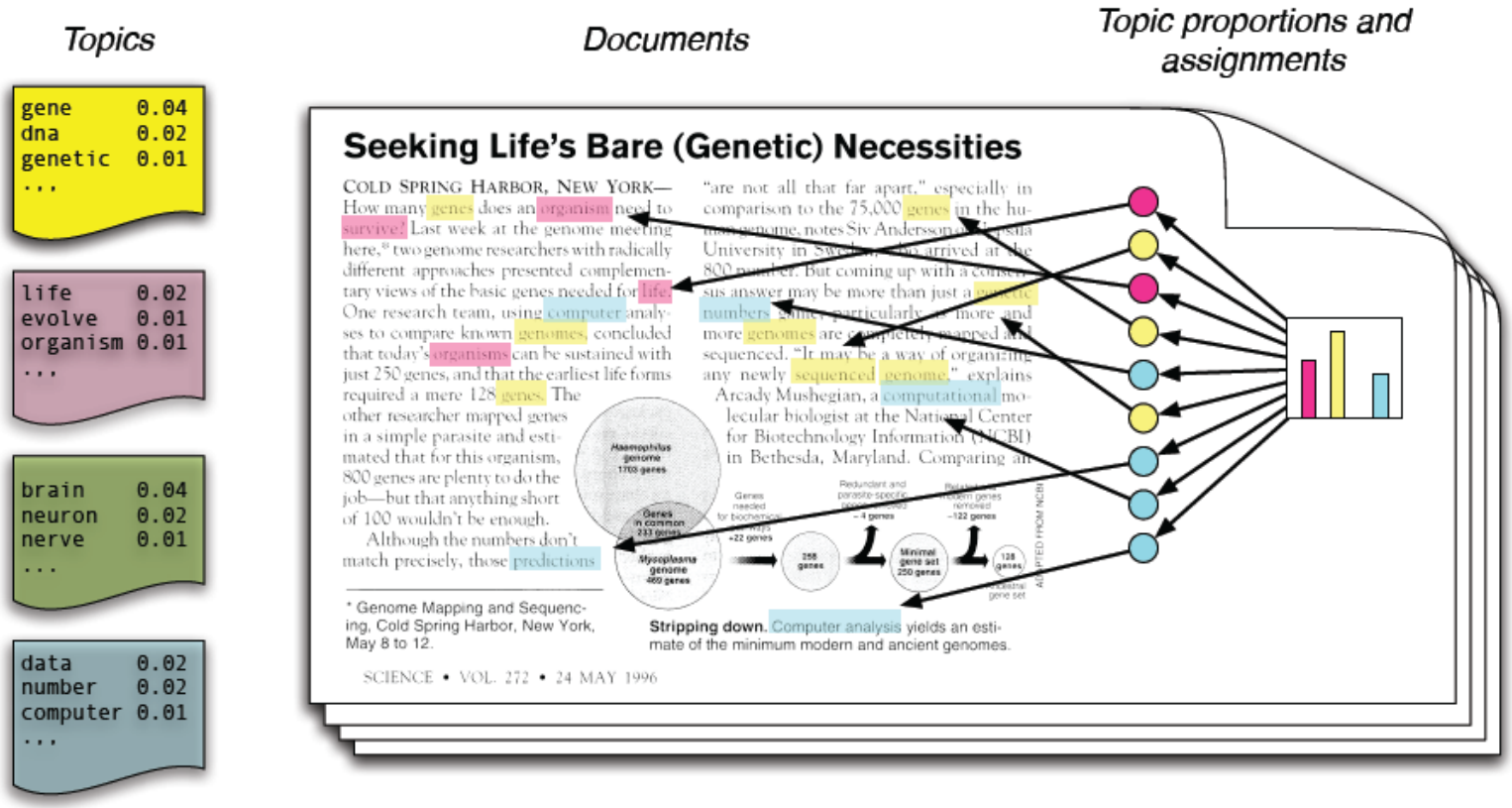
A thoughtful, provocative, insistently humanizing film.

Occasionally melodramatic, it's also extremely effective.

A sentimental mess that never rings true.

- As we run EM, $\mathbf{z}_i$ may begin to indicate sentiment of sentence.
- $\boldsymbol{\pi}_1$ would give higher prob to bad words: *mess, joke, terrible*
- $\boldsymbol{\pi}_2$ would give higher prob to good words: *thoughtful, humanizing, effective*

Contents


Topics
Documents
Topic proportions and assignments
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
Seeking Life's Bare (Genetic) Necessities
COLD SPRING HARBOR, NEW YORK—
How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.
Although the numbers don't match precisely, those predictions
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson
University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 689 genes
Genes needed for biochemical tasks +22 genes
Redundant and parasite-specific genes removed ~4 genes
Recent modern genes removed ~122 genes
256 genes
Minimal gene set 250 genes
128 genes
Minimal gene set
ADAPTED FROM NCBI
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
SCIENCE • VOL. 272 • 24 MAY 1996

Topic Model (High-Level)

Observe:

- a set of documents, each with a sequence of words.

Choose:

- Number of *topics* to learn. Each topic has a distribution over words likely in the topic.
- For example words: "TF, concentration, yard" might be common in a documents about Harvard.

Estimate:

- Topics associated with each document.
- Words associated with each topic.

Formal Vocabulary for Topic Models

Problem setup:

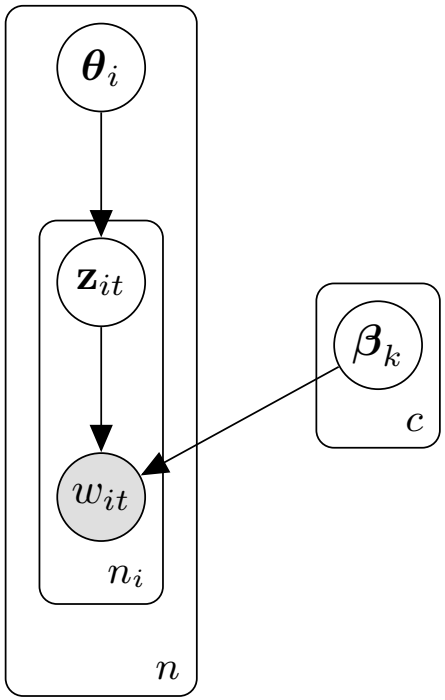
- Vocabulary ($\mathcal{V}$): a set of word/token types
- Document ($\mathbf{w}_i$): a sequence of n_i-tokens $w_{i,t} \in \mathcal{V}$
- Topics ($1, \ldots, c$): elements of the mixture.
- Topic-Word Distribution ($\boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_c$): a distribution over word types $\mathcal{V}$ associated with a topic
- Document-Topic Distribution ($\boldsymbol{\theta}_i$): a distribution over topics $1, \ldots, c$ associated with a document
- Latent Topic for Word ($\mathbf{z}_{it}$): one-hot vector representing the topic for a word.

Formal Vocabulary for Topic Models

Problem setup:

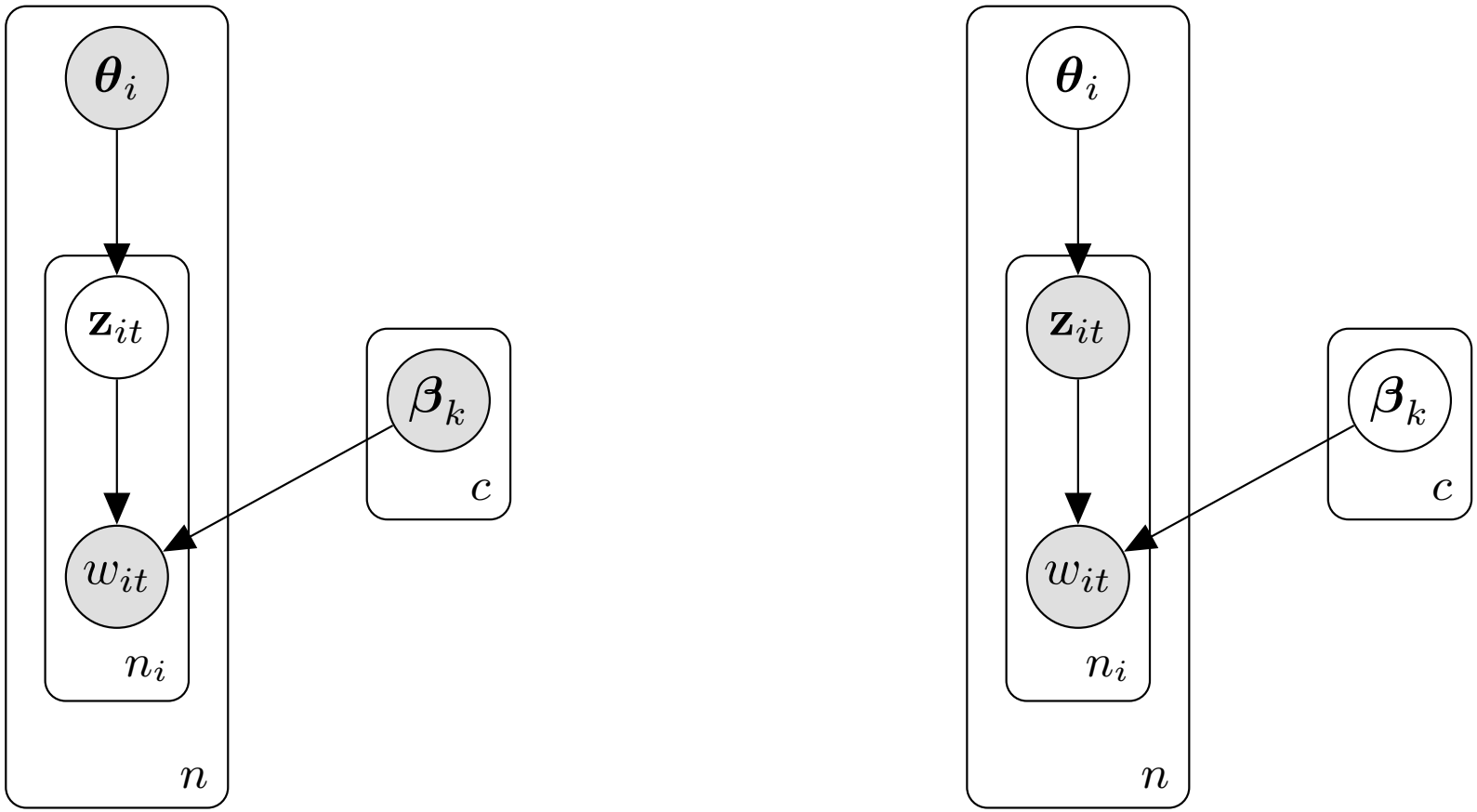
- Vocabulary ($\mathcal{V}$): a set of word/token types
- Document ($\mathbf{w}_i$): a sequence of n_i-tokens $w_{i,t} \in \mathcal{V}$
- Topics $(1, \ldots, c)$: elements of the mixture.
- Topic-Word Distribution ($\boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_c$): a distribution over word types $\mathcal{V}$ associated with a topic
- Document-Topic Distribution ($\boldsymbol{\theta}_i$): a distribution over topics $1, \ldots, c$ associated with a document
- Latent Topic for Word ($\mathbf{z}_{it}$): one-hot vector representing the topic for a word.

Topic Model



1. Draw a document-topic distribution $\boldsymbol{\theta}_i$.
2. For each token t,
   - Draw a topic $\mathbf{z}_{it}$ from $\boldsymbol{\theta}_i$
   - Draw next word from topic-word distribution $\boldsymbol{\beta}_{\mathbf{z}_{it}}$

Inference with EM



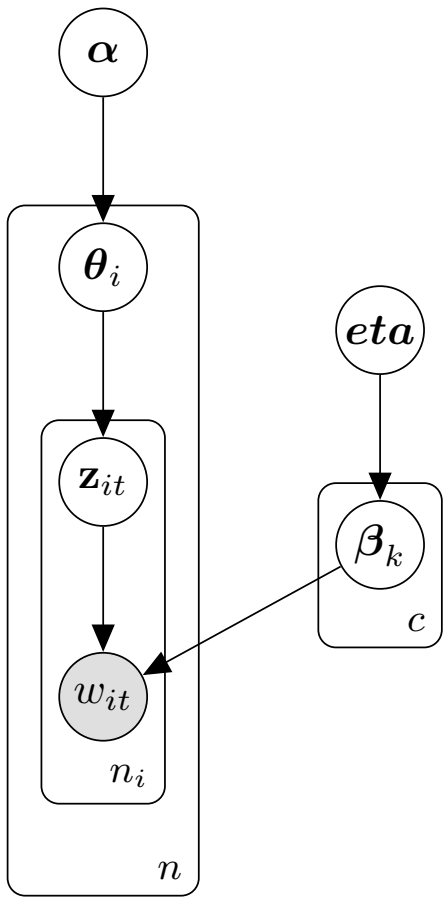
Alternate between predicting expected topic for each word and the estimating parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$.

Topic Modeling in Practice

- Many other inference techniques used. Many use sampling-based methods (Gibbs sampling), some use other approaches that utilize SGD.
- Other approaches try to find sparse topics, only contain small number of non-zero prob words.

Hyperparameters: Latent Dirichlet Allocation

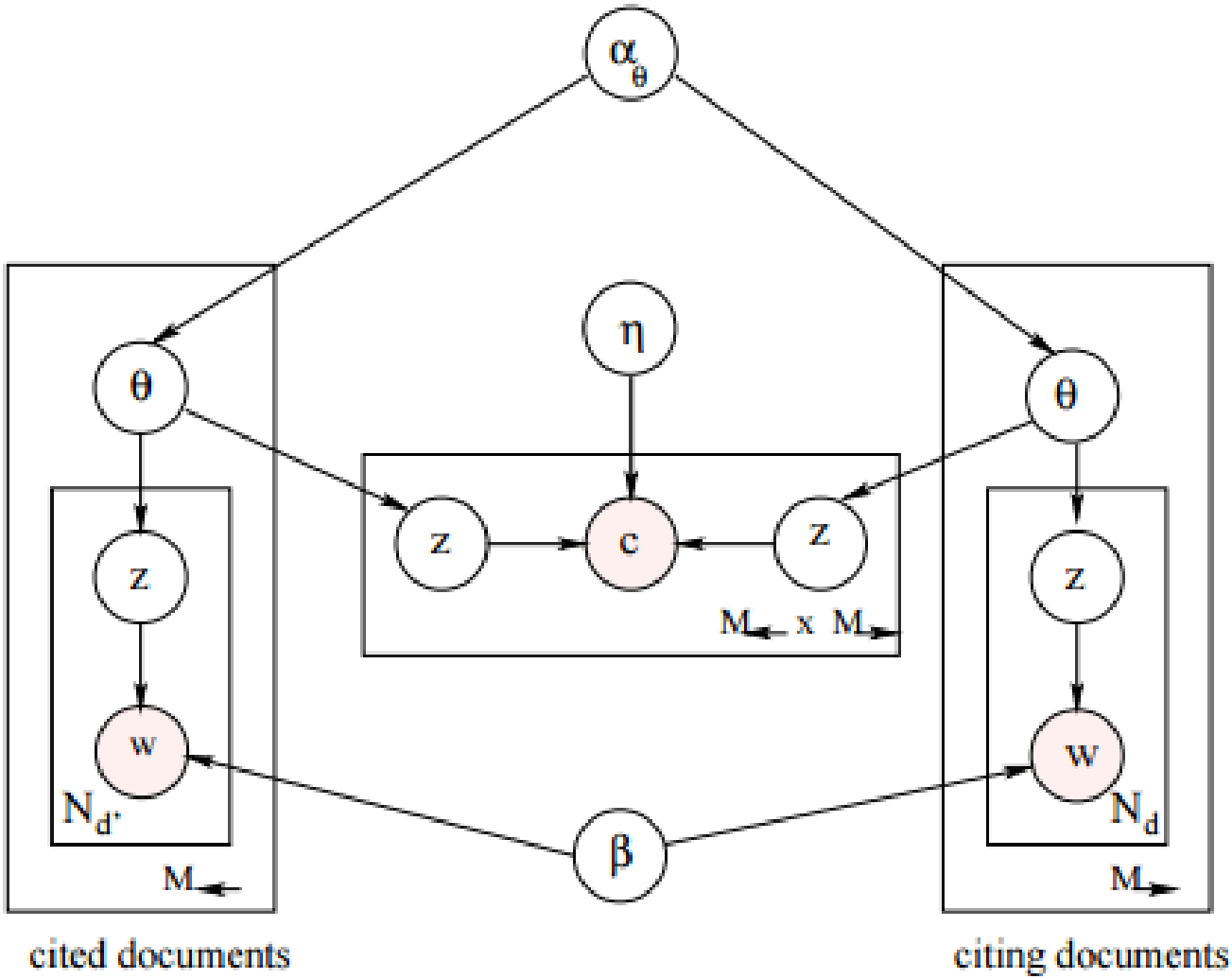
- Latent Dirichlet Allocation (LDA) is popular method. Specifies Dirichlet-Multinomial model with priors over $\boldsymbol{\theta}_i$ and $\boldsymbol{\beta}$.



Evaluation

- Tricky problem. How do you evaluate unsupervised topic modeling?
- Approach 1: Held-Out Likelihood
  - Check how well the model explains unseen data.
- Approach 2: Human interpretability
  - How well do the topics correspond to human judgment.

Topic Modeling Extensions Example



[Nallapati et al 2008]

Example