Probabilistic Topic Models

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

March 13, 2019

Topic Models

- Given a collection of objects (each object a set of discrete tokens), find topics in the collection

Pic courtesy: David Blei

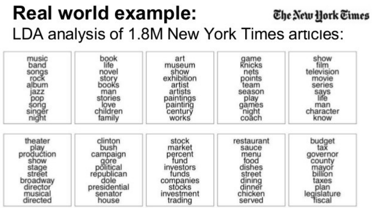
Topic Models

- Given a collection of objects (each object a set of discrete tokens), find topics in the collection
- Can also be used to represent each object as how much each topic is present in that object

Pic courtesy: David Blei

Topic Models

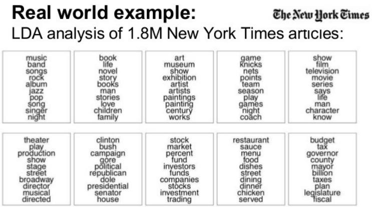
- Given a collection of objects (each object a set of discrete tokens), find topics in the collection
- Can also be used to represent each object as how much each topic is present in that object
- Some topics inferred from a collection where each object is a document (and tokens are words)



Pic courtesy: David Blei

Topic Models

- Given a collection of objects (each object a set of discrete tokens), find topics in the collection
- Can also be used to represent each object as how much each topic is present in that object
- Some topics inferred from a collection where each object is a document (and tokens are words)

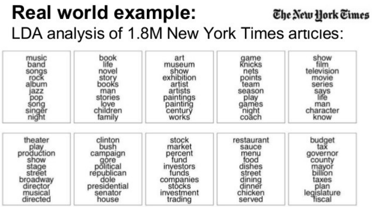


- Also applicable to other types of data, beyond text documents

Pic courtesy: David Blei

Topic Models

- Given a collection of objects (each object a set of discrete tokens), find topics in the collection
- Can also be used to represent each object as how much each topic is present in that object
- Some topics inferred from a collection where each object is a document (and tokens are words)



- Also applicable to other types of data, beyond text documents, e.g.,
  - Image collection: Each image is a "document" which is a bag of "visual words"

Pic courtesy: David Blei

Topic Models

- Given a collection of objects (each object a set of discrete tokens), find topics in the collection
- Can also be used to represent each object as how much each topic is present in that object
- Some topics inferred from a collection where each object is a document (and tokens are words)

Real world example: The New York Times

LDA analysis of 1.8M New York Times articles:

music band songs rock album jazz pop song singer night	book life novel story books man stories love children family	art museum show exhibition artist artists paintings painting century works	game knicks nets points team season play games night coach	show film television movie series says life man character know
theater play production show stage street broadway director musical directed	clinton bush campaign gore political republican dole presidential senator house	stock market percent fund investors funds companies stocks investment trading	restaurant sauce menu food dishes street dining dinner chicken served	budget tax governor county mayor billion taxes plan legislature fiscal

- Also applicable to other types of data, beyond text documents, e.g.,
  - Image collection: Each image is a "document" which is a bag of "visual words"
  - Customer-purchase data: Each customer is a "document" and items purchased are the "words"

Pic courtesy: David Blei

Topic Models

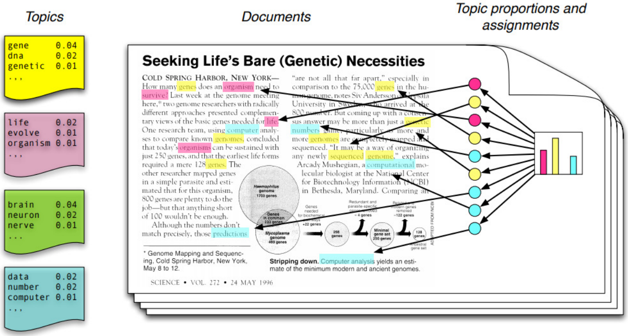
- For concreteness, let's focus our discussion on topic models for text

Topic Models

- For concreteness, let's focus our discussion on topic models for text
- Topic models are based on assigning words in each document to clusters/topics (each cluster of words represents a "topic")

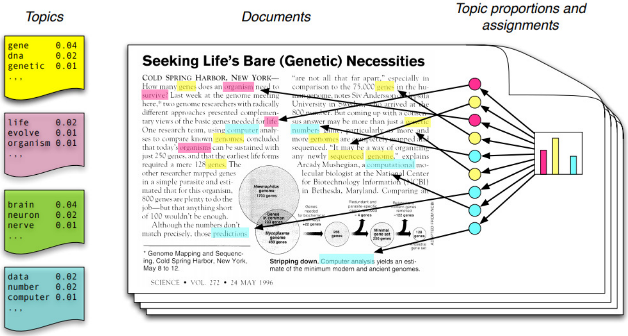
Topic Models

- For concreteness, let's focus our discussion on topic models for text
- Topic models are based on assigning words in each document to clusters/topics (each cluster of words represents a "topic"): Essentially, this is word clustering with many documents!



Topic Models

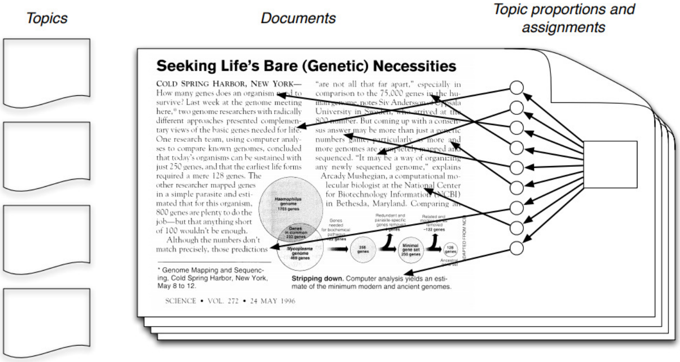
- For concreteness, let's focus our discussion on topic models for text
- Topic models are based on assigning words in each document to clusters/topics (each cluster of words represents a "topic"): Essentially, this is word clustering with many documents!



- Formally, a "topic" is a distribution over tokens (a prob. vector with length = # of unique tokens)

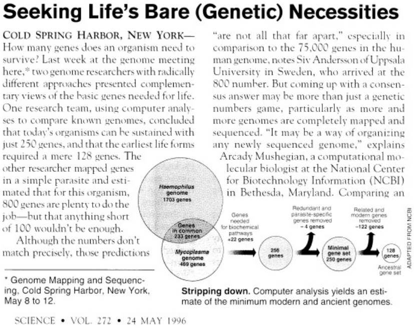
Topic Models

- The input for the topic modeling problem will be the raw contents of the documents and the goal is to infer all the unknowns of the model (topics, topic proportions for each document, etc.)



Applications of Topic Models

- Can be used to learn topic-based representation for each document
- These representation are compact (think dimensionality reduction) and semantically meaningful



Seeking Life's Bare (Genetic) Necessities

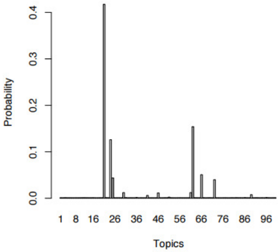
COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known 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 of Uppsala 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

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

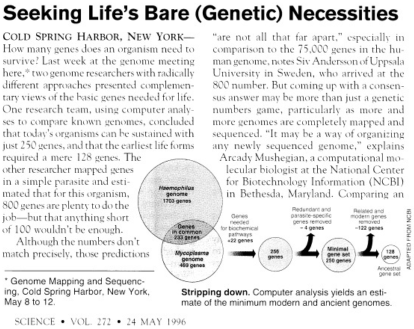
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996



Applications of Topic Models

- Can be used to learn topic-based representation for each document
- These representation are compact (think dimensionality reduction) and semantically meaningful



Seeking Life's Bare (Genetic) Necessities

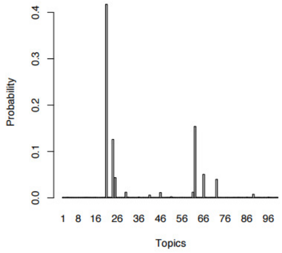
COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known 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 of Uppsala 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

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

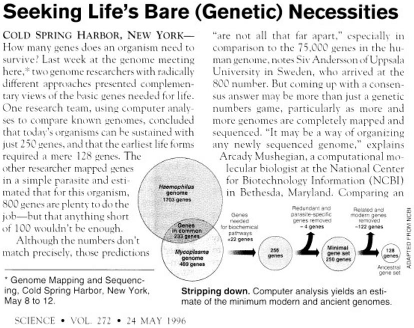
SCIENCE • VOL. 272 • 24 MAY 1996



- Also makes it easy to cluster data naturally by (learned) topics

Applications of Topic Models

- Can be used to learn topic-based representation for each document
- These representation are compact (think dimensionality reduction) and semantically meaningful



Seeking Life's Bare (Genetic) Necessities

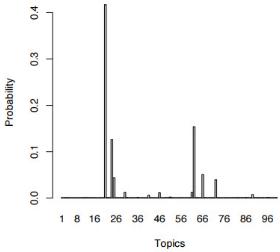
COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known 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 of Uppsala 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

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

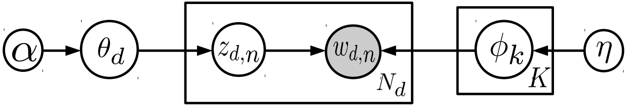
SCIENCE • VOL. 272 • 24 MAY 1996



- Also makes it easy to cluster data naturally by (learned) topics
- Can also be used to understand evolution of a corpora (e.g., how topics drift over time - "Dynamic Topic Models" - e.g., word prominent in one topic now may not be so after 10 years later)

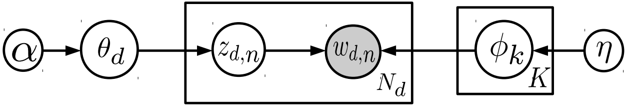
A Toy Mixture Model for Words in a Document

- Recall that topic modeling is like clustering the words of the documents
- Consider a toy K component multinoulli mixture model for the words of a single document



A Toy Mixture Model for Words in a Document

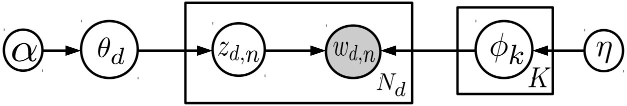
- Recall that topic modeling is like clustering the words of the documents
- Consider a toy K component multinoulli mixture model for the words of a single document



- "Multinoulli" because the observations (words here) are discrete tokens

A Toy Mixture Model for Words in a Document

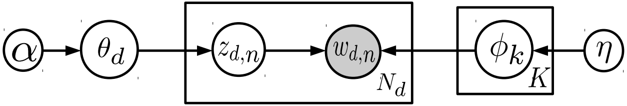
- Recall that topic modeling is like clustering the words of the documents
- Consider a toy K component multinoulli mixture model for the words of a single document



- "Multinoulli" because the observations (words here) are discrete tokens
- Assume the overall vocabulary consists of V unique words/tokens

A Toy Mixture Model for Words in a Document

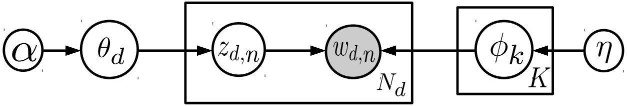
- Recall that topic modeling is like clustering the words of the documents
- Consider a toy K component multinoulli mixture model for the words of a single document



- "Multinoulli" because the observations (words here) are discrete tokens
- Assume the overall vocabulary consists of V unique words/tokens
- Assume each word $w_{d,n} \in \{1, \ldots, V\}$ belongs to a cluster $z_{d,n} \in \{1, \ldots, K\}$

A Toy Mixture Model for Words in a Document

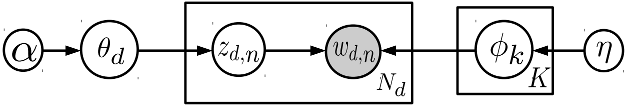
- Recall that topic modeling is like clustering the words of the documents
- Consider a toy K component multinoulli mixture model for the words of a single document



- "Multinoulli" because the observations (words here) are discrete tokens
- Assume the overall vocabulary consists of V unique words/tokens
- Assume each word $w_{d,n} \in \{1, \ldots, V\}$ belongs to a cluster $z_{d,n} \in \{1, \ldots, K\}$
- θ_d is the mixture proportion vector (summing to 1) for the document

A Toy Mixture Model for Words in a Document

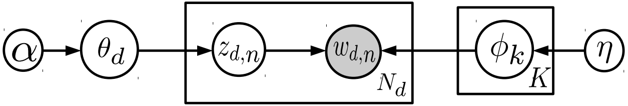
- Recall that topic modeling is like clustering the words of the documents
- Consider a toy K component multinoulli mixture model for the words of a single document



- "Multinoulli" because the observations (words here) are discrete tokens
- Assume the overall vocabulary consists of V unique words/tokens
- Assume each word $w_{d,n} \in \{1, \ldots, V\}$ belongs to a cluster $z_{d,n} \in \{1, \ldots, K\}$
- θ_d is the mixture proportion vector (summing to 1) for the document
- ϕ_k denotes the parameters of mixture component k

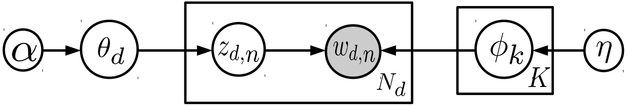
A Toy Mixture Model for Words in a Document

- Recall that topic modeling is like clustering the words of the documents
- Consider a toy K component multinoulli mixture model for the words of a single document

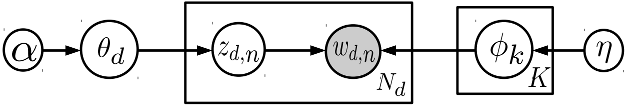


- "Multinoulli" because the observations (words here) are discrete tokens
- Assume the overall vocabulary consists of V unique words/tokens
- Assume each word $w_{d,n} \in \{1, \ldots, V\}$ belongs to a cluster $z_{d,n} \in \{1, \ldots, K\}$
- θ_d is the mixture proportion vector (summing to 1) for the document
- ϕ_k denotes the parameters of mixture component k
- In a Bayesian version, θ_d and ϕ_k will have priors with hyperparams α and η, respectively

A Toy Mixture Model for Words in a Document

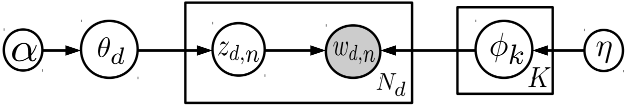


A Toy Mixture Model for Words in a Document



- The generative story will be

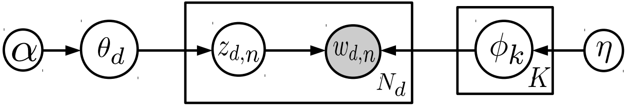
A Toy Mixture Model for Words in a Document



- The generative story will be
  - Draw the K topic vectors $\{\phi_k\}_{k=1}^{K}$ from a V-dim Dirichlet

$$\phi_k \sim \text{Dirichlet}(\eta, \ldots, \eta) \qquad k = 1, \ldots, K$$

A Toy Mixture Model for Words in a Document



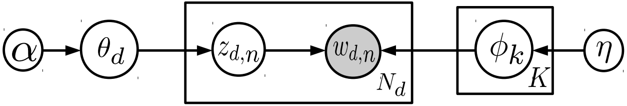
- The generative story will be
  - Draw the K topic vectors $\{\phi_k\}_{k=1}^K$ from a V-dim Dirichlet

$$\phi_k \sim \text{Dirichlet}(\eta, \ldots, \eta) \qquad k = 1, \ldots, K$$

  - Draw the mixing proportion vector θ_d from a K-dim Dirichlet

$$\theta_d \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

A Toy Mixture Model for Words in a Document



- The generative story will be
  - Draw the K topic vectors $\{\phi_k\}_{k=1}^K$ from a V-dim Dirichlet

$$\phi_k \sim \text{Dirichlet}(\eta, \ldots, \eta) \qquad k = 1, \ldots, K$$

  - Draw the mixing proportion vector θ_d from a K-dim Dirichlet

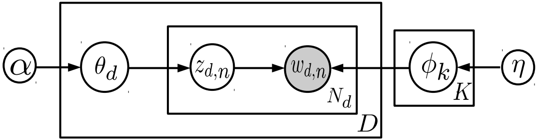
$$\theta_d \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

  - For each word $w_{d,n}$, $n = 1, \ldots, N_d$

$$\begin{aligned} z_{d,n} &\sim \text{multinoulli}(\theta_d) \\ w_{d,n} &\sim \text{multinoulli}(\phi_{z_{d,n}}) \end{aligned}$$

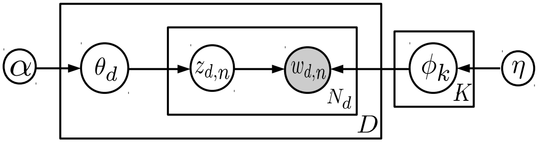
Latent Dirichlet Allocation (LDA)

- Let's consider a more "real" case. We have a collection of $D > 1$ documents
- The previous toy model can be extended easily to handle this



Latent Dirichlet Allocation (LDA)

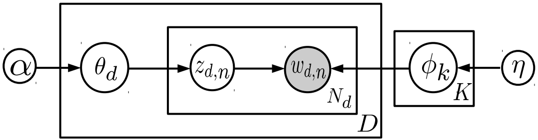
- Let's consider a more "real" case. We have a collection of $D > 1$ documents
- The previous toy model can be extended easily to handle this



- All docs share the same set of topics $\phi_1, \ldots, \phi_K$ but have different topic proportions $\theta_1, \ldots, \theta_D$

Latent Dirichlet Allocation (LDA)

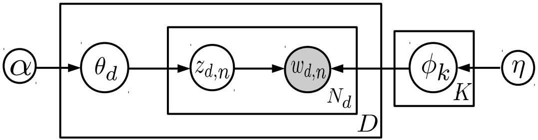
- Let's consider a more "real" case. We have a collection of $D > 1$ documents
- The previous toy model can be extended easily to handle this



- All docs share the same set of topics $\phi_1, \ldots, \phi_K$ but have different topic proportions $\theta_1, \ldots, \theta_D$
- Equivalent to jointly learning D multinoulli mixture models (one per document)

Latent Dirichlet Allocation (LDA)

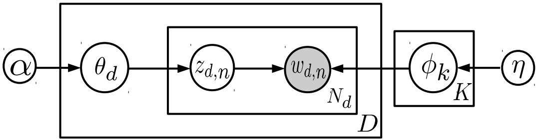
- Let's consider a more "real" case. We have a collection of $D > 1$ documents
- The previous toy model can be extended easily to handle this



- All docs share the same set of topics $\phi_1, \ldots, \phi_K$ but have different topic proportions $\theta_1, \ldots, \theta_D$
- Equivalent to jointly learning D multinoulli mixture models (one per document)
  - Information shared acorss documents via global variables $\{\phi_k\}_{k=1}^K$, and global hyperparams α and η

Latent Dirichlet Allocation (LDA)

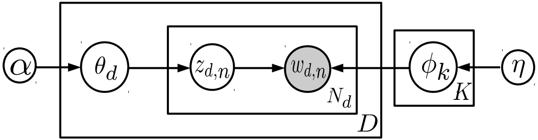
- Let's consider a more "real" case. We have a collection of $D > 1$ documents
- The previous toy model can be extended easily to handle this



- All docs share the same set of topics $\phi_1, \ldots, \phi_K$ but have different topic proportions $\theta_1, \ldots, \theta_D$
- Equivalent to jointly learning D multinoulli mixture models (one per document)
  - Information shared acorss documents via global variables $\{\phi_k\}_{k=1}^K$, and global hyperparams α and η
- Each topic is a distribution over V tokens (ϕ_k), each document is a distrib. over K topics (θ_d)

Latent Dirichlet Allocation (LDA)

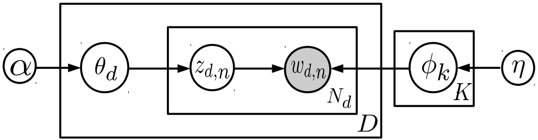
- Let's consider a more "real" case. We have a collection of $D > 1$ documents
- The previous toy model can be extended easily to handle this



- All docs share the same set of topics $\phi_1, \ldots, \phi_K$ but have different topic proportions $\theta_1, \ldots, \theta_D$
- Equivalent to jointly learning D multinoulli mixture models (one per document)
  - Information shared acorss documents via global variables $\{\phi_k\}_{k=1}^K$, and global hyperparams α and η
- Each topic is a distribution over V tokens (ϕ_k), each document is a distrib. over K topics (θ_d)
- This is basically the Latent Dirichlet Allocation (LDA) model (Blei *et al*, 2003)

Latent Dirichlet Allocation (LDA)

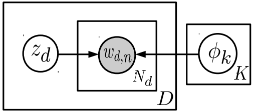
- Let's consider a more "real" case. We have a collection of $D > 1$ documents
- The previous toy model can be extended easily to handle this



- All docs share the same set of topics $\phi_1, \ldots, \phi_K$ but have different topic proportions $\theta_1, \ldots, \theta_D$
- Equivalent to jointly learning D multinoulli mixture models (one per document)
  - Information shared acorss documents via global variables $\{\phi_k\}_{k=1}^K$, and global hyperparams α and η
- Each topic is a distribution over V tokens (ϕ_k), each document is a distrib. over K topics (θ_d)
- This is basically the Latent Dirichlet Allocation (LDA) model (Blei *et al*, 2003)
- Generative story is identical to the single doc. toy example but now θ_d for each doc.

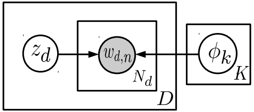
Some LDA Precursors

- **Mixture of unigrams models:** Each document having a single topic, all words from that topic



Some LDA Precursors

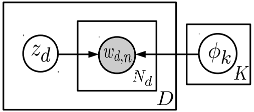
- **Mixture of unigrams models:** Each document having a single topic, all words from that topic



- Unlike the above mixture of unigrams model, LDA is an example of "admixture model" (all words of the document not drawn from the same topic but from a mixture of topics)

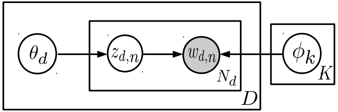
Some LDA Precursors

- **Mixture of unigrams models:** Each document having a single topic, all words from that topic



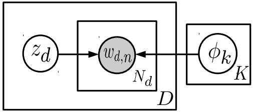
- Unlike the above mixture of unigrams model, LDA is an example of "admixture model" (all words of the document not drawn from the same topic but from a mixture of topics)

- **Probabilistic Latent Semantic Analysis:** Like LDA, each word in a doc has its own topic



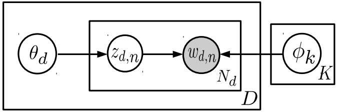
Some LDA Precursors

- **Mixture of unigrams models:** Each document having a single topic, all words from that topic



- Unlike the above mixture of unigrams model, LDA is an example of "admixture model" (all words of the document not drawn from the same topic but from a mixture of topics)

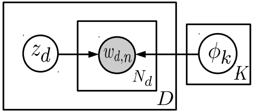
- **Probabilistic Latent Semantic Analysis:** Like LDA, each word in a doc has its own topic



- However, unlike LDA, PLSA (Hofmann (2001) is not a Bayesian model

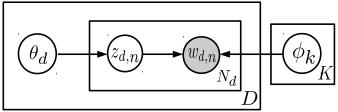
Some LDA Precursors

- **Mixture of unigrams models:** Each document having a single topic, all words from that topic



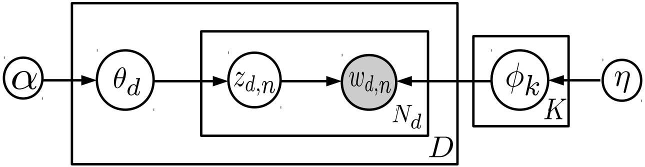
- Unlike the above mixture of unigrams model, LDA is an example of "admixture model" (all words of the document not drawn from the same topic but from a mixture of topics)

- **Probabilistic Latent Semantic Analysis:** Like LDA, each word in a doc has its own topic



- However, unlike LDA, PLSA (Hofmann (2001) is not a Bayesian model
  - As a result, LDA is less prone to overfitting

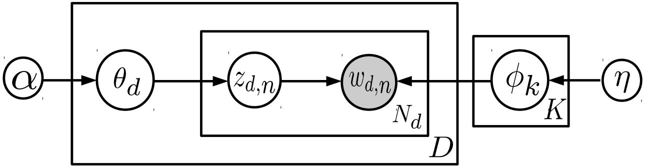
Inference for LDA



- The goal is to infer the posterior distribution over all the latent variables

$$p(\mathbf{Z}, \Theta, \Phi | \mathbf{W}, \alpha, \eta) = \frac{p(\mathbf{W}|\Phi, \mathbf{Z})p(\mathbf{Z}|\Theta)p(\Phi|\eta)p(\Theta|\alpha)}{p(\mathbf{W}|\alpha, \eta)} \qquad \text{(assuming hyperparams } \alpha, \eta \text{ are fixed)}$$

Inference for LDA

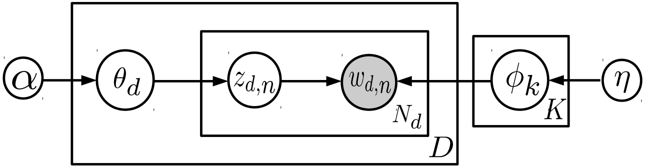


- The goal is to infer the posterior distribution over all the latent variables

$$p(\mathbf{Z}, \Theta, \Phi | \mathbf{W}, \alpha, \eta) = \frac{p(\mathbf{W}|\Phi, \mathbf{Z})p(\mathbf{Z}|\Theta)p(\Phi|\eta)p(\Theta|\alpha)}{p(\mathbf{W}|\alpha, \eta)} \qquad \text{(assuming hyperparams } \alpha, \eta \text{ are fixed)}$$

- **Z**: Topic assignments of all words across all documents

Inference for LDA

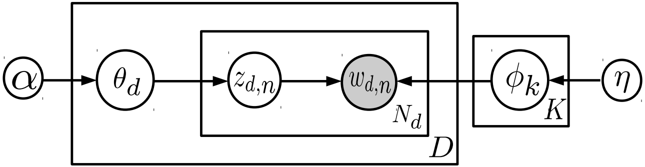


- The goal is to infer the posterior distribution over all the latent variables

$$p(\mathbf{Z}, \Theta, \Phi | \mathbf{W}, \alpha, \eta) = \frac{p(\mathbf{W}|\Phi, \mathbf{Z})p(\mathbf{Z}|\Theta)p(\Phi|\eta)p(\Theta|\alpha)}{p(\mathbf{W}|\alpha, \eta)} \qquad \text{(assuming hyperparams } \alpha, \eta \text{ are fixed)}$$

- **Z**: Topic assignments of all words across all documents
- $\Theta = \{\theta_1, \ldots, \theta_D\}$: Topic proportion vectors; each θ_d is a K-dim vector

Inference for LDA

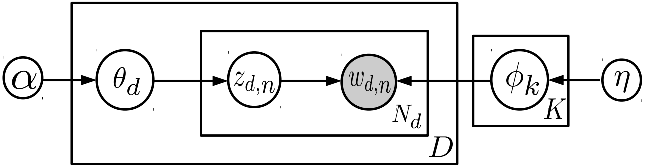


- The goal is to infer the posterior distribution over all the latent variables

$$p(\mathbf{Z}, \Theta, \Phi | \mathbf{W}, \alpha, \eta) = \frac{p(\mathbf{W}|\Phi, \mathbf{Z})p(\mathbf{Z}|\Theta)p(\Phi|\eta)p(\Theta|\alpha)}{p(\mathbf{W}|\alpha, \eta)} \qquad \text{(assuming hyperparams } \alpha, \eta \text{ are fixed)}$$

- **Z**: Topic assignments of all words across all documents
- $\Theta = \{\theta_1, \ldots, \theta_D\}$: Topic proportion vectors; each θ_d is a K-dim vector
- $\Phi = \{\phi_1, \ldots, \phi_K\}$: The K topics; each ϕ_k is a V-dim vector

Inference for LDA

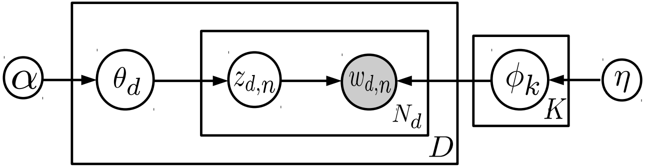


- The goal is to infer the posterior distribution over all the latent variables

$$p(\mathbf{Z}, \Theta, \Phi | \mathbf{W}, \alpha, \eta) = \frac{p(\mathbf{W}|\Phi, \mathbf{Z})p(\mathbf{Z}|\Theta)p(\Phi|\eta)p(\Theta|\alpha)}{p(\mathbf{W}|\alpha, \eta)} \qquad \text{(assuming hyperparams } \alpha, \eta \text{ are fixed)}$$

- **Z**: Topic assignments of all words across all documents
- $\Theta = \{\theta_1, \ldots, \theta_D\}$: Topic proportion vectors; each θ_d is a K-dim vector
- $\Phi = \{\phi_1, \ldots, \phi_K\}$: The K topics; each ϕ_k is a V-dim vector
- $\mathbf{W} = \{\boldsymbol{w}_{d,n}\}, d = 1, \ldots, D, n = 1, \ldots, N_d$: Collection of all words in all the documents

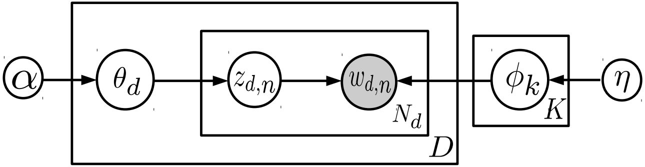
Inference for LDA



- A wide variety of inference methods exist for LDA (VI, Variational EM, MCMC, batch, online, ...)

† Latent Dirichlet Allocation (Blei et al, 2003)

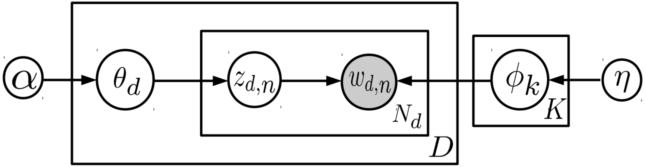
Inference for LDA



- A wide variety of inference methods exist for LDA (VI, Variational EM, MCMC, batch, online, ...)
- LDA is a locally conjugate model (recall that the model uses Dirichlet/multinoulli distr.)

† Latent Dirichlet Allocation (Blei et al, 2003)

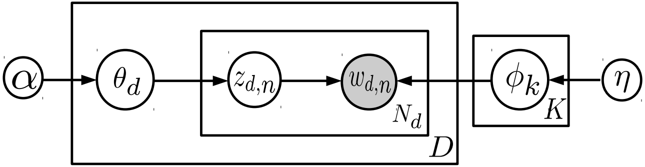
Inference for LDA



- A wide variety of inference methods exist for LDA (VI, Variational EM, MCMC, batch, online, ...)
- LDA is a locally conjugate model (recall that the model uses Dirichlet/multinoulli distr.)
  - The likelihood and priors are all exponential family distributions

† Latent Dirichlet Allocation (Blei et al, 2003)

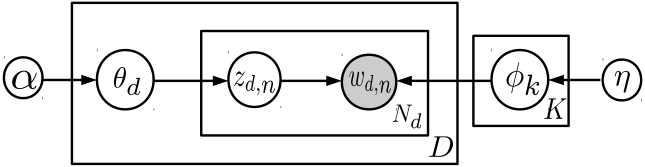
Inference for LDA



- A wide variety of inference methods exist for LDA (VI, Variational EM, MCMC, batch, online, ...)
- LDA is a locally conjugate model (recall that the model uses Dirichlet/multinoulli distr.)
  - The likelihood and priors are all exponential family distributions
- MCMC, in particular Gibbs sampling, is very easy to derive due to local conjugacy

† Latent Dirichlet Allocation (Blei et al, 2003)

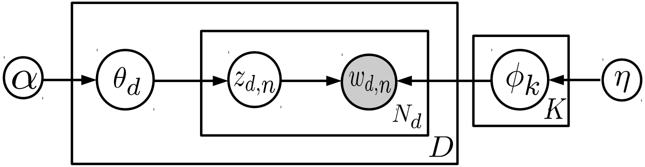
Inference for LDA



- A wide variety of inference methods exist for LDA (VI, Variational EM, MCMC, batch, online, ...)
- LDA is a locally conjugate model (recall that the model uses Dirichlet/multinoulli distr.)
  - The likelihood and priors are all exponential family distributions
- MCMC, in particular Gibbs sampling, is very easy to derive due to local conjugacy
- Local conjugacy also allow deriving a simple VB algorithm (the original LDA paper† used VB)

† Latent Dirichlet Allocation (Blei et al, 2003)

Inference for LDA



- A wide variety of inference methods exist for LDA (VI, Variational EM, MCMC, batch, online, ...)
- LDA is a locally conjugate model (recall that the model uses Dirichlet/multinoulli distr.)
  - The likelihood and priors are all exponential family distributions
- MCMC, in particular Gibbs sampling, is very easy to derive due to local conjugacy
- Local conjugacy also allow deriving a simple VB algorithm (the original LDA paper[†] used VB)
- Note: Can even collapse some variables and do collapsed Gibbs or collapsed VB

[†] Latent Dirichlet Allocation (Blei et al, 2003)

LDA Evaluation

- Many ways to test how well LDA performs on some data

LDA Evaluation

- Many ways to test how well LDA performs on some data
- Extrinsic measures: Perform LDA and measure performance on some external task (e.g., accuracy of a document classification model using the topic based document representation)

LDA Evaluation

- Many ways to test how well LDA performs on some data
- Extrinsic measures: Perform LDA and measure performance on some external task (e.g., accuracy of a document classification model using the topic based document representation)
- Perplexity is another intrinsic measure

LDA Evaluation

- Many ways to test how well LDA performs on some data
- Extrinsic measures: Perform LDA and measure performance on some external task (e.g., accuracy of a document classification model using the topic based document representation)
- Perplexity is another intrinsic measure
- Given a test collection of M document, the perplexity is defined as

$$perplexity(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{M}\log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\right\}$$

.. where $\boldsymbol{w}_d$ collectively denotes all words in a test document d

LDA Evaluation

- Many ways to test how well LDA performs on some data
- Extrinsic measures: Perform LDA and measure performance on some external task (e.g., accuracy of a document classification model using the topic based document representation)
- Perplexity is another intrinsic measure
- Given a test collection of M document, the perplexity is defined as

$$perplexity(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{M}\log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\right\}$$

.. where $\boldsymbol{w}_d$ collectively denotes all words in a test document d

- We set aside some documents as a test set and compute the average marginal likelihood $p(\boldsymbol{w}_d)$ the model assigns to the words in these held-out documents

LDA Evaluation

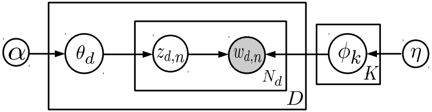
- Many ways to test how well LDA performs on some data
- Extrinsic measures: Perform LDA and measure performance on some external task (e.g., accuracy of a document classification model using the topic based document representation)
- Perplexity is another intrinsic measure
- Given a test collection of M document, the perplexity is defined as

$$perplexity(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\right\}$$

.. where $\boldsymbol{w}_d$ collectively denotes all words in a test document d

- We set aside some documents as a test set and compute the average marginal likelihood $p(\boldsymbol{w}_d)$ the model assigns to the words in these held-out documents
- A high average probability (low perplexity) implies a good fit to the data

LDA as (Bayesian) Non-negative/Poisson Matrix Factorization

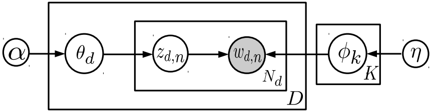


- Assume $\mathbf{X}$ is $V \times D$ word-document count matrix ($X_{v,n}$ = no of times word v appears in doc d)

---

† Sec 4 and 5 of "Beta-Negative Binomial Process and Poisson Factor Analysis" (Zhou et al, 2012)

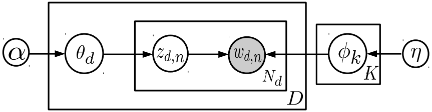
LDA as (Bayesian) Non-negative/Poisson Matrix Factorization



- Assume $\mathbf{X}$ is $V \times D$ word-document count matrix ($X_{v,n}$ = no of times word v appears in doc d)
- Assume $\Phi = [\phi_1, \ldots, \phi_K]$ is $V \times K$, and $\Theta = [\theta_1, \ldots, \theta_D]$ is $K \times D$, both are non-neg matrices

† Sec 4 and 5 of "Beta-Negative Binomial Process and Poisson Factor Analysis" (Zhou et al, 2012)

LDA as (Bayesian) Non-negative/Poisson Matrix Factorization

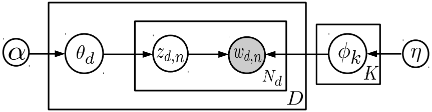


- Assume $\mathbf{X}$ is $V \times D$ word-document count matrix ($X_{v,n}$ = no of times word v appears in doc d)
- Assume $\Phi = [\phi_1, \ldots, \phi_K]$ is $V \times K$, and $\Theta = [\theta_1, \ldots, \theta_D]$ is $K \times D$, both are non-neg matrices
- Then LDA can be seen as doing a Poisson matrix factorization† of the matrix $\mathbf{X}$

$$\mathbf{X} \sim \text{Poisson}(\Phi\Theta)$$

† Sec 4 and 5 of "Beta-Negative Binomial Process and Poisson Factor Analysis" (Zhou et al, 2012)

LDA as (Bayesian) Non-negative/Poisson Matrix Factorization



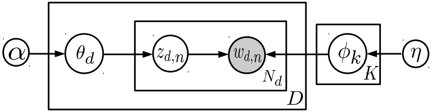
- Assume $\mathbf{X}$ is $V \times D$ word-document count matrix ($X_{v,n}$ = no of times word v appears in doc d)
- Assume $\Phi = [\phi_1, \ldots, \phi_K]$ is $V \times K$, and $\Theta = [\theta_1, \ldots, \theta_D]$ is $K \times D$, both are non-neg matrices
- Then LDA can be seen as doing a Poisson matrix factorization[†] of the matrix $\mathbf{X}$

$$\mathbf{X} \sim \text{Poisson}(\Phi\Theta)$$

- Notation here: Poisson applied element-wise on $\Phi\Theta$ rate matrix

[†] Sec 4 and 5 of "Beta-Negative Binomial Process and Poisson Factor Analysis" (Zhou et al, 2012)

LDA as (Bayesian) Non-negative/Poisson Matrix Factorization



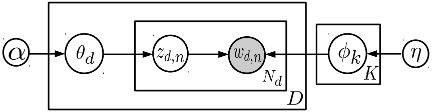
- Assume $\mathbf{X}$ is $V \times D$ word-document count matrix ($X_{v,n}$ = no of times word v appears in doc d)
- Assume $\Phi = [\phi_1, \ldots, \phi_K]$ is $V \times K$, and $\Theta = [\theta_1, \ldots, \theta_D]$ is $K \times D$, both are non-neg matrices
- Then LDA can be seen as doing a Poisson matrix factorization[†] of the matrix $\mathbf{X}$

$$\mathbf{X} \sim \text{Poisson}(\Phi\Theta)$$

- Notation here: Poisson applied element-wise on $\Phi\Theta$ rate matrix
- Note that word-to-topic assignments $z_{d,n}$ don't appear in this formulation

[†] Sec 4 and 5 of "Beta-Negative Binomial Process and Poisson Factor Analysis" (Zhou et al, 2012)

LDA as (Bayesian) Non-negative/Poisson Matrix Factorization



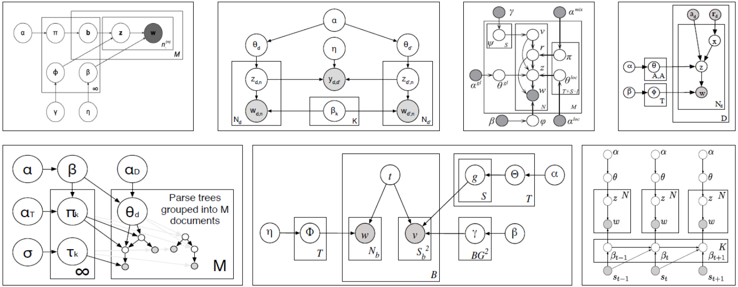
- Assume $\mathbf{X}$ is $V \times D$ word-document count matrix ($X_{v,n}$ = no of times word v appears in doc d)
- Assume $\Phi = [\phi_1, \ldots, \phi_K]$ is $V \times K$, and $\Theta = [\theta_1, \ldots, \theta_D]$ is $K \times D$, both are non-neg matrices
- Then LDA can be seen as doing a Poisson matrix factorization[†] of the matrix $\mathbf{X}$

$$\mathbf{X} \sim \text{Poisson}(\Phi\Theta)$$

- Notation here: Poisson applied element-wise on $\Phi\Theta$ rate matrix
- Note that word-to-topic assignments $z_{d,n}$ don't appear in this formulation
- Note: Even if priors on ϕ_k and θ_d are not Dirichlet (e.g., each entry of vectors ϕ_k and/or θ_d has some other non-neg prior[†], such as gamma, the Bayesian Poisson NMF equivalence still holds)

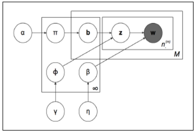
[†] Sec 4 and 5 of "Beta-Negative Binomial Process and Poisson Factor Analysis" (Zhou et al, 2012)

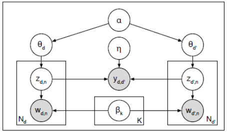
LDA is easy to extend

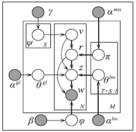


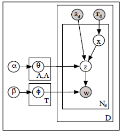
Several extensions of LDA (too many to list here :)). Some include

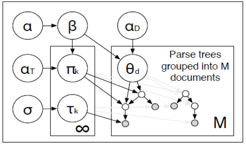
LDA is easy to extend

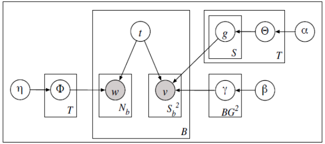


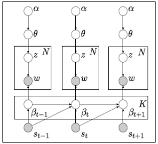








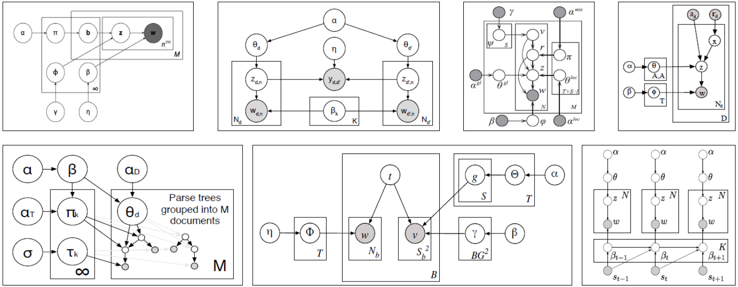




Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)

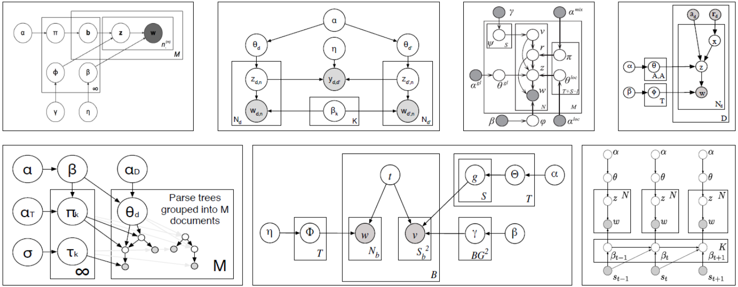
LDA is easy to extend



Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)
- LDA for multimodal data (images and text) and multilingual data

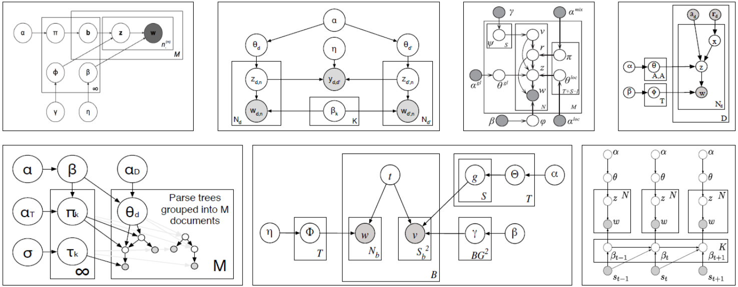
LDA is easy to extend



Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)
- LDA for multimodal data (images and text) and multilingual data
- LDA with topic hierarchies/correlations

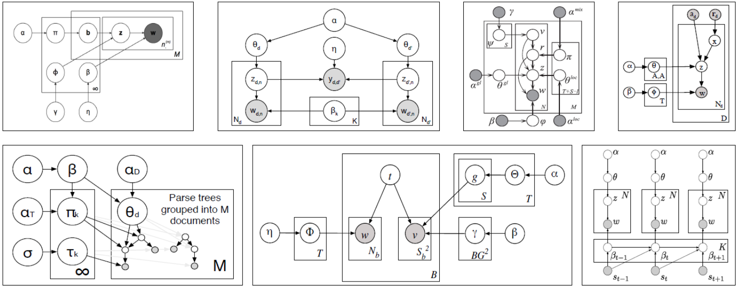
LDA is easy to extend



Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)
- LDA for multimodal data (images and text) and multilingual data
- LDA with topic hierarchies/correlations
- LDA with time-evolving topics

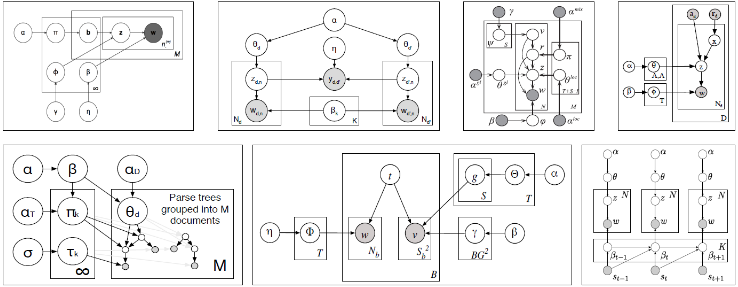
LDA is easy to extend



Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)
- LDA for multimodal data (images and text) and multilingual data
- LDA with topic hierarchies/correlations
- LDA with time-evolving topics
- LDA with HMM (takes into account word order)

LDA is easy to extend



Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)
- LDA for multimodal data (images and text) and multilingual data
- LDA with topic hierarchies/correlations
- LDA with time-evolving topics
- LDA with HMM (takes into account word order)
- LDA for graphs

LDA: Some Other Improvements

- Learning the number of topics using nonparametric Bayesian models, e.g.,
  - Hierarchical Dirichlet Process (HDP) topic models

LDA: Some Other Improvements

- Learning the number of topics using nonparametric Bayesian models, e.g.,
  - Hierarchical Dirichlet Process (HDP) topic models
- Faster inference methods, e.g., online (SVI), amortized VI, SGLD, etc

LDA: Some Other Improvements

- Learning the number of topics using nonparametric Bayesian models, e.g.,
  - Hierarchical Dirichlet Process (HDP) topic models
- Faster inference methods, e.g., online (SVI), amortized VI, SGLD, etc
- Deep Topic Models (e.g., using variational autoencoders for document-word count vectors)