Probabilistic Topic Models

Piyush Rai

Probabilistic Machine Learning (CS772A)

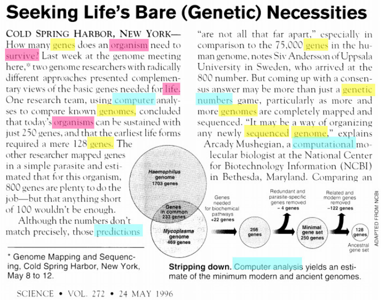
Oct 26, 2017

Remaining schedule..

- 6 more classes (+1 make up lecture for Diwali holiday; date TBD)
- Probabilistic Topic models (today)
- Deep Probabilistic Models (Bayesian neural nets, VAE, GAN)
- Models for sequential data (HMM, Linear Dynamical Systems)
- Probabilistic Graphical Models, Message Passing algorithms
- Other misc. topics
  - Semi-supervised Learning
  - Active Learning
  - Other suggestions.. ?

Topic Modeling

- Topic models is a way to model documents: assumes each document to be a mixture of topics



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

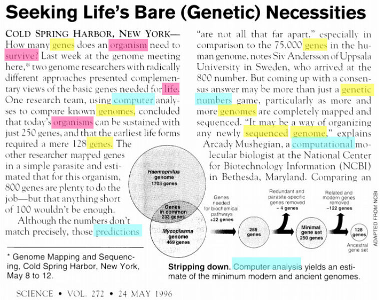
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

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

SCIENCE • VOL. 272 • 24 MAY 1996

Topic Modeling

- Topic models is a way to model documents: assumes each document to be a mixture of topics



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.

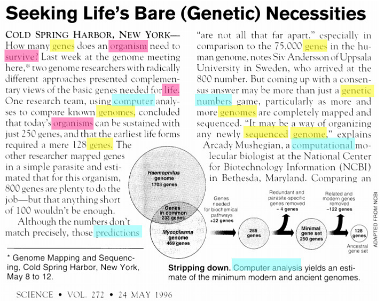
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

- The notion of "document" is abstract here (e.g., "document" may represent an internet "user" and the words may represent the textual contents of all the URLs the user visits during a session)

Topic Modeling

- Topic models is a way to model documents: assumes each document to be a mixture of topics



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

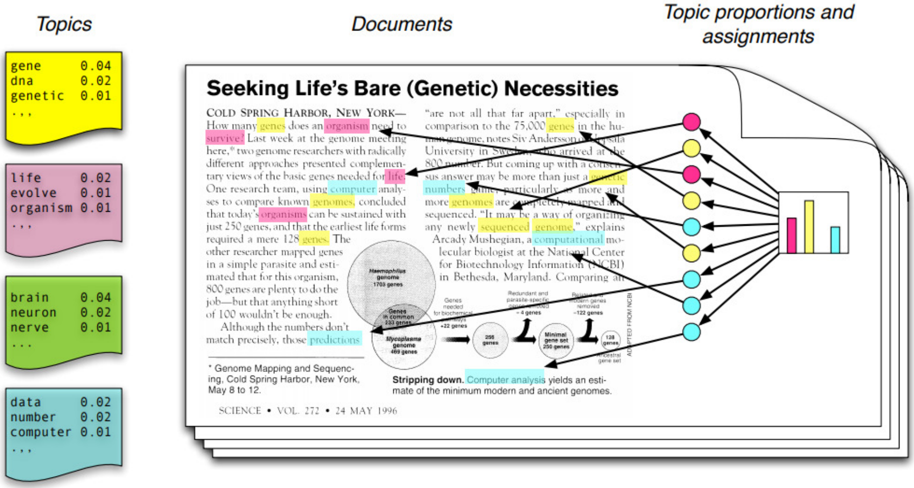
- The notion of "document" is abstract here (e.g., "document" may represent an internet "user" and the words may represent the textual contents of all the URLs the user visits during a session)

- The notion also applies for other type of data, not necessarily text (e.g., images as bag of "visual words" or speech data as phonemes) and a topic model makes sense for such data as well.

(Pic courtesy: David Blei)

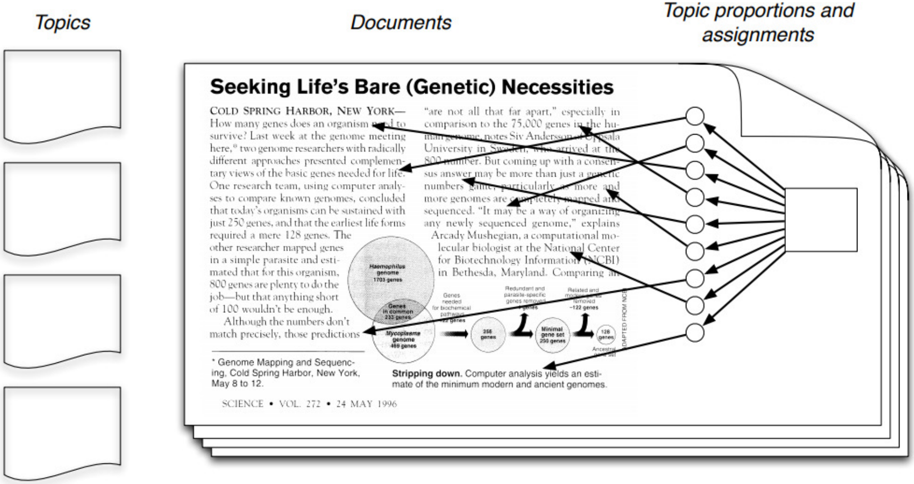
Topic Modeling

- Goal of topic modeling is to learn the topics that underlie the document collection, and represent each document in terms of these topics (fraction of various topics)



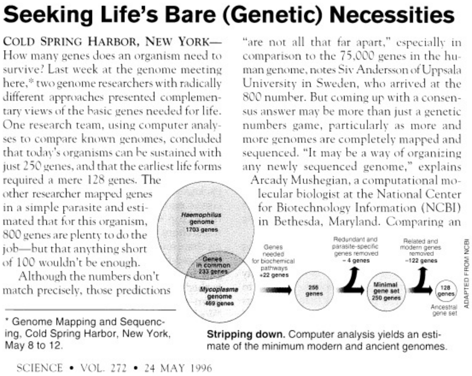
Topic Modeling

- 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 Modeling

- 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

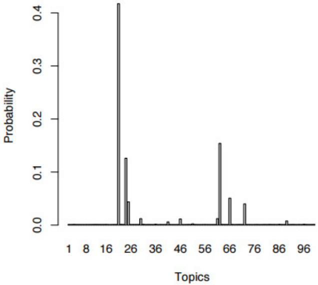
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known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

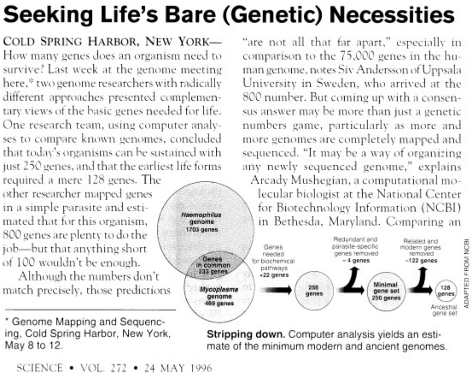
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996



Applications of Topic Modeling

- 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

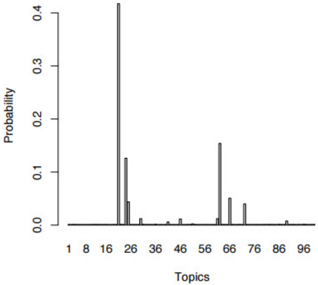
COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known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

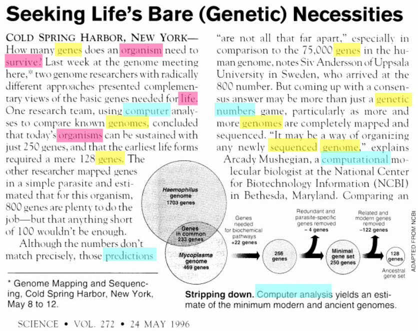
SCIENCE • VOL. 272 • 24 MAY 1996



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

Topic Models: The Basic Idea

- Topic models (especially LDA) are essentially based on assigning words in each document to clusters/topics (each cluster can be thought of as representing a "topic")
- Note: Different occurrences of the same word may be assigned to different topics



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

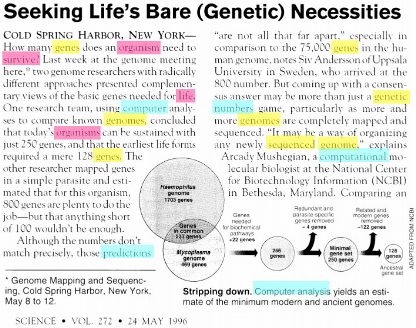
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

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

SCIENCE • VOL. 272 • 24 MAY 1996

Topic Models: The Basic Idea

- Topic models (especially LDA) are essentially based on assigning words in each document to clusters/topics (each cluster can be thought of as representing a "topic")
- Note: Different occurrences of the same word may be assigned to different topics



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

- Note that word-level topic assignments can be aggregated to get the document's topic proportions

What's a Topic?

- Suppose the vocabulary consists of V words
- Suppose there are K topics indexed by $k = 1, \ldots, K$

What's a Topic?

- Suppose the vocabulary consists of V words
- Suppose there are K topics indexed by $k = 1, \ldots, K$
- Informally, a topic represents a group of similar words representative of that topic

What's a Topic?

- Suppose the vocabulary consists of V words
- Suppose there are K topics indexed by $k = 1, \ldots, K$
- Informally, a topic represents a group of similar words representative of that topic
- Formally, a topic is a probability distribution over the V words

Topic 247

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

Topic 5

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

Topic 43

word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THINK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

Topic 56

word	prob.
DOCTOR	.074
DR.	.063
PATIENT	.061
HOSPITAL	.049
CARE	.046
MEDICAL	.042
NURSE	.031
PATIENTS	.029
DOCTORS	.028
HEALTH	.025
MEDICINE	.017
NURSING	.017
DENTAL	.015
NURSES	.013
PHYSICIAN	.012
HOSPITALS	.011

What's a Topic?

- Suppose the vocabulary consists of V words
- Suppose there are K topics indexed by $k = 1, \ldots, K$
- Informally, a topic represents a group of similar words representative of that topic
- Formally, a topic is a probability distribution over the V words

Topic 247

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

Topic 5

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

Topic 43

word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THINK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

Topic 56

word	prob.
DOCTOR	.074
DR.	.063
PATIENT	.061
HOSPITAL	.049
CARE	.046
MEDICAL	.042
NURSE	.031
PATIENTS	.029
DOCTORS	.028
HEALTH	.025
MEDICINE	.017
NURSING	.017
DENTAL	.015
NURSES	.013
PHYSICIAN	.012
HOSPITALS	.011

- Topic k can be represented by a probability vector β_k of size V

What's a Topic?

- Suppose the vocabulary consists of V words
- Suppose there are K topics indexed by $k = 1, \dots, K$
- Informally, a topic represents a group of similar words representative of that topic
- Formally, a topic is a probability distribution over the V words

Topic 247

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

Topic 5

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

Topic 43

word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THINK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

Topic 56

word	prob.
DOCTOR	.074
DR.	.063
PATIENT	.061
HOSPITAL	.049
CARE	.046
MEDICAL	.042
NURSE	.031
PATIENTS	.029
DOCTORS	.028
HEALTH	.025
MEDICINE	.017
NURSING	.017
DENTAL	.015
NURSES	.013
PHYSICIAN	.012
HOSPITALS	.011

- Topic k can be represented by a probability vector β_k of size V
  - β_{kv} denotes the proability of word v in topic k and $\sum_{v=1}^{V} \beta_{kv} = 1$

What's a Topic?

- Suppose the vocabulary consists of V words
- Suppose there are K topics indexed by $k = 1, \ldots, K$
- Informally, a topic represents a group of similar words representative of that topic
- Formally, a topic is a probability distribution over the V words

Topic 247

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

Topic 5

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

Topic 43

word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THINK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

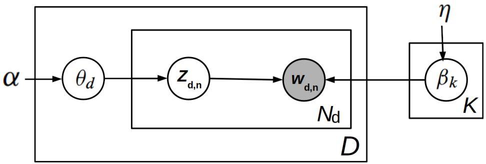
Topic 56

word	prob.
DOCTOR	.074
DR.	.063
PATIENT	.061
HOSPITAL	.049
CARE	.046
MEDICAL	.042
NURSE	.031
PATIENTS	.029
DOCTORS	.028
HEALTH	.025
MEDICINE	.017
NURSING	.017
DENTAL	.015
NURSES	.013
PHYSICIAN	.012
HOSPITALS	.011

- Topic k can be represented by a probability vector β_k of size V
  - β_{kv} denotes the proability of word v in topic k and $\sum_{v=1}^{V} \beta_{kv} = 1$
  - High-probability words are representative of that topic

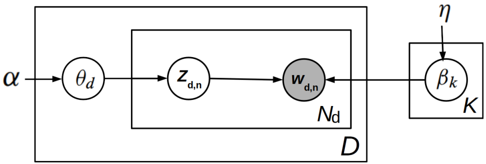
Latent Dirichlet Allocation (LDA)

Models data where each observation (document) consist of a set of discrete-valued tokens (words)



Latent Dirichlet Allocation (LDA)

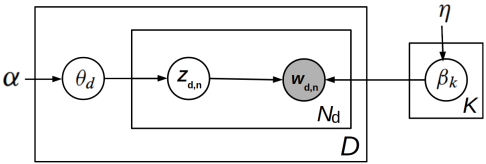
Models data where each observation (document) consist of a set of discrete-valued tokens (words)



The LDA generative model (assuming α, η known) can be summarized as follows

Latent Dirichlet Allocation (LDA)

Models data where each observation (document) consist of a set of discrete-valued tokens (words)



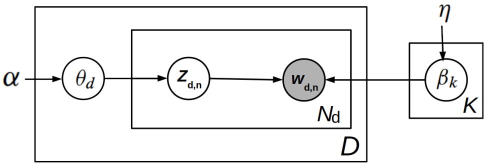
The LDA generative model (assuming α, η known) can be summarized as follows

- Draw K topics $\{\beta_k\}_{k=1}^K$ shared by all documents $d = 1, \ldots, D$

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

Latent Dirichlet Allocation (LDA)

Models data where each observation (document) consist of a set of discrete-valued tokens (words)



The LDA generative model (assuming α, η known) can be summarized as follows

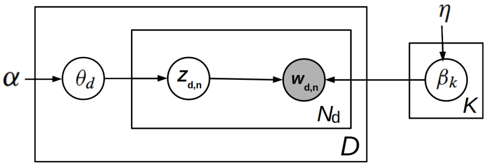
- Draw K topics $\{\beta_k\}_{k=1}^K$ shared by all documents $d = 1, \ldots, D$

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

- For each document $d = 1, \ldots, D$

Latent Dirichlet Allocation (LDA)

Models data where each observation (document) consist of a set of discrete-valued tokens (words)



The LDA generative model (assuming α, η known) can be summarized as follows

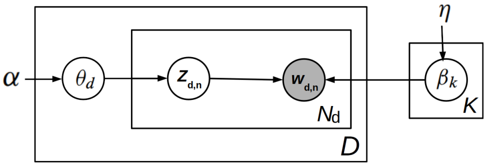
- Draw K topics $\{\beta_k\}_{k=1}^{K}$ shared by all documents $d = 1, \ldots, D$

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

- For each document $d = 1, \ldots, D$
  - Draw its K-dim topic proportion vector $\theta_d \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$

Latent Dirichlet Allocation (LDA)

Models data where each observation (document) consist of a set of discrete-valued tokens (words)



The LDA generative model (assuming α, η known) can be summarized as follows

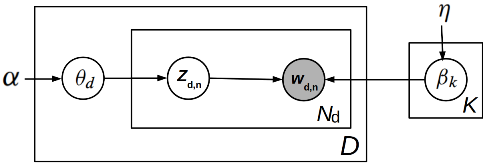
- Draw K topics $\{\beta_k\}_{k=1}^K$ shared by all documents $d = 1, \ldots, D$

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

- For each document $d = 1, \ldots, D$
  - Draw its K-dim topic proportion vector $\theta_d \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$
  - For each word $n = 1, \ldots, N_d$ in document d

Latent Dirichlet Allocation (LDA)

Models data where each observation (document) consist of a set of discrete-valued tokens (words)



The LDA generative model (assuming α, η known) can be summarized as follows

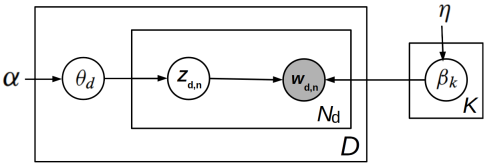
- Draw K topics $\{\beta_k\}_{k=1}^{K}$ shared by all documents $d = 1, \ldots, D$

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

- For each document $d = 1, \ldots, D$
  - Draw its K-dim topic proportion vector $\theta_d \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$
  - For each word $n = 1, \ldots, N_d$ in document d
    - First choose the topic for this word: $z_{d,n} \sim \text{multinoulli}(\theta_d)$

Latent Dirichlet Allocation (LDA)

Models data where each observation (document) consist of a set of discrete-valued tokens (words)



The LDA generative model (assuming α, η known) can be summarized as follows

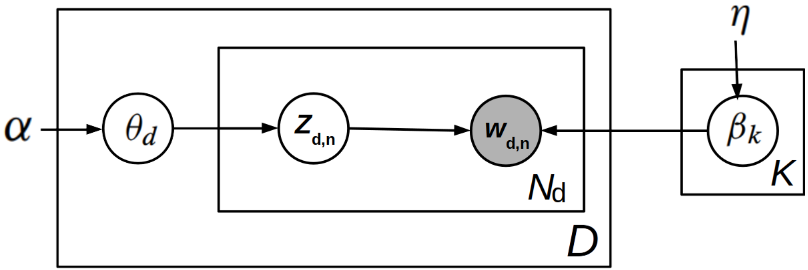
- Draw K topics $\{\beta_k\}_{k=1}^K$ shared by all documents $d = 1, \ldots, D$

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

- For each document $d = 1, \ldots, D$
  - Draw its K-dim topic proportion vector $\theta_d \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$
  - For each word $n = 1, \ldots, N_d$ in document d
    - First choose the topic for this word: $z_{d,n} \sim \text{multinoulli}(\theta_d)$
    - Now generate the word form the chosen topic: $\boldsymbol{w}_{d,n} \sim \text{multinoulli}(\beta_{z_{d,n}})$

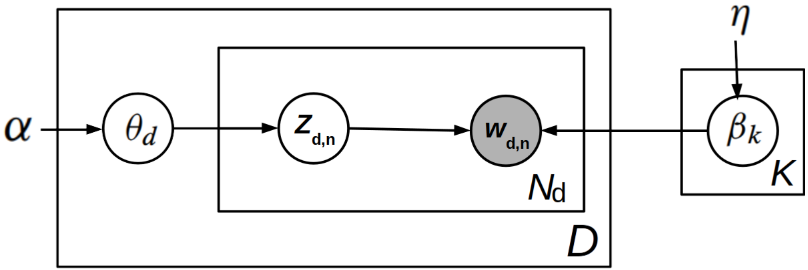
What is LDA doing?

- Note that LDA is learning D many K-component mixture models (one for each document)



What is LDA doing?

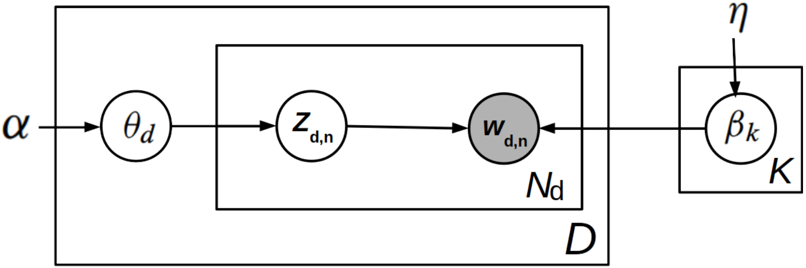
- Note that LDA is learning D many K-component mixture models (one for each document)



- These mixture models are "tied" together by the shared topics $\{\beta_k\}_{k=1}^K$ and hyperparams α, η

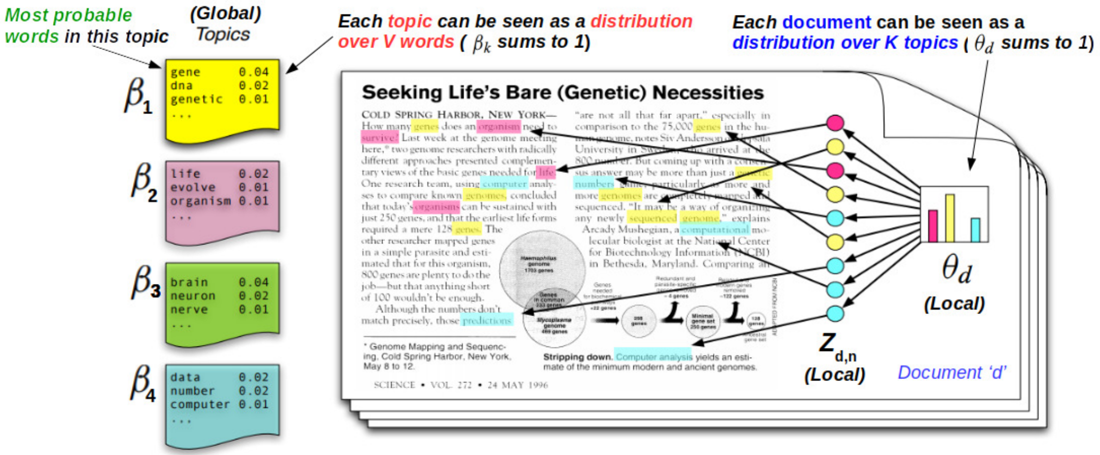
What is LDA doing?

- Note that LDA is learning D many K-component mixture models (one for each document)



- These mixture models are "tied" together by the shared topics $\{\beta_k\}_{k=1}^K$ and hyperparams α, η
- Word co-occurrences (within/across document) influences the topics $\{\beta_k\}_{k=1}^K$ learned by the model

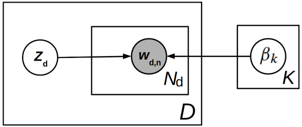
What LDA Output Looks Like, Qualitatively



(Figure: Modified from the original by David Blei)

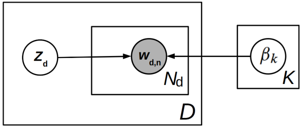
Some LDA Precursors

- A naïve mixture model: Each document having a single topic, all words from that topic

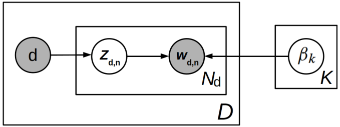


Some LDA Precursors

- A naïve mixture model: Each document having a single topic, all words from that topic

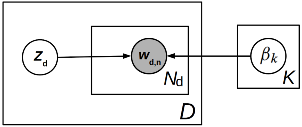


- Each word having its own topic: Probabilistic Latent Semantic Analysis (PLSA), Hofmann (2001)

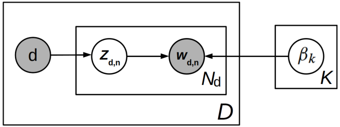


Some LDA Precursors

- A naïve mixture model: Each document having a single topic, all words from that topic



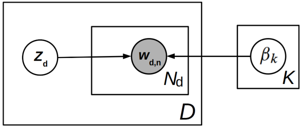
- Each word having its own topic: Probabilistic Latent Semantic Analysis (PLSA), Hofmann (2001)



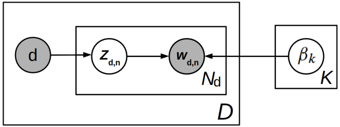
- PLSA has too many parameters $p(w,d) = p(d)p(w|d) = p(d)\sum_{k=1}^{K} p(w|z=k)p(z=k|d)$

Some LDA Precursors

- A naïve mixture model: Each document having a single topic, all words from that topic



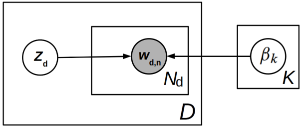
- Each word having its own topic: Probabilistic Latent Semantic Analysis (PLSA), Hofmann (2001)



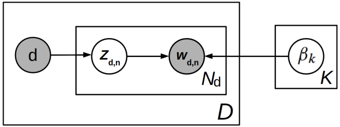
- PLSA has too many parameters $p(w,d) = p(d)p(w|d) = p(d)\sum_{k=1}^{K} p(w|z=k)p(z=k|d)$
  - VK params for $p(w|z=k)$, KD params for $p(z=k|d)$. Grows in the number of documents

Some LDA Precursors

- A naïve mixture model: Each document having a single topic, all words from that topic



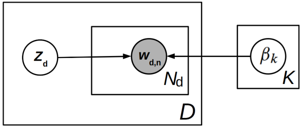
- Each word having its own topic: Probabilistic Latent Semantic Analysis (PLSA), Hofmann (2001)



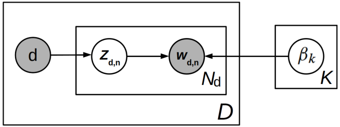
- PLSA has too many parameters $p(w, d) = p(d)p(w|d) = p(d)\sum_{k=1}^{K} p(w|z = k)p(z = k|d)$
    - VK params for $p(w|z = k)$, KD params for $p(z = k|d)$. Grows in the number of documents
    - LDA however models $p(z = k|d)$ via random variables θ_d, not tied to only training data, so can extend to previously unseen documents unlike PLSA. A subtle but important difference (Blei et al, 2003)

Some LDA Precursors

- A naïve mixture model: Each document having a single topic, all words from that topic

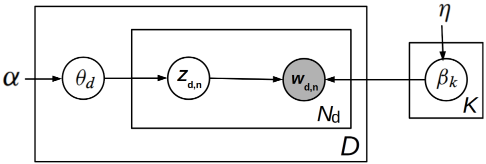


- Each word having its own topic: Probabilistic Latent Semantic Analysis (PLSA), Hofmann (2001)



- PLSA has too many parameters $p(w,d) = p(d)p(w|d) = p(d)\sum_{k=1}^{K} p(w|z=k)p(z=k|d)$
  - VK params for $p(w|z=k)$, KD params for $p(z=k|d)$. Grows in the number of documents
  - LDA however models $p(z=k|d)$ via random variables θ_d, not tied to only training data, so can extend to previously unseen documents unlike PLSA. A subtle but important difference (Blei et al, 2003)
- Also, unlike LDA, PLSA is not a Bayesian model

Inference for LDA

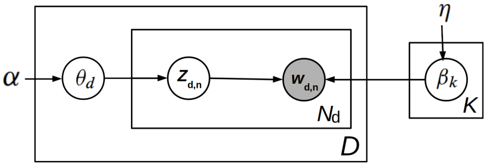


- The goal is to infer the posterior distribution over all the latent variables

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

- $\mathbf{Z}$: 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
- $\beta = \{\beta_1, \ldots, \beta_K\}$: The K topics; each β_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

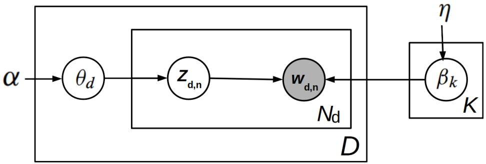
Inference for LDA



- A wide variety of inference methods exist for LDA (MCMC, VB, EP, batch, online, ...)

† Latent Dirichlet Allocation (Blei et al, 2003)

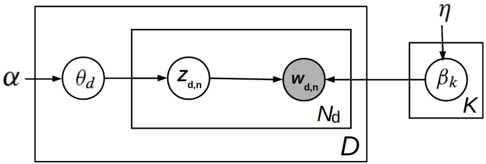
Inference for LDA



- A wide variety of inference methods exist for LDA (MCMC, VB, EP, 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)

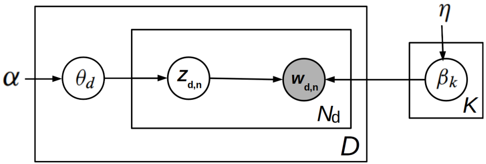
Inference for LDA



- A wide variety of inference methods exist for LDA (MCMC, VB, EP, 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)

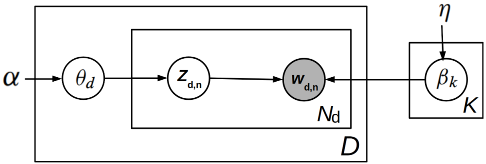
Inference for LDA



- A wide variety of inference methods exist for LDA (MCMC, VB, EP, 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 (MCMC, VB, EP, 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)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as[†]

$$p(z_{d,n} = k|\mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(z_{d,n} = k|\mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } z_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n}|z_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

[†] Finding Scientific Topics (Griffiths and Steyvers, 2004)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as[†]

$$p(z_{d,n} = k | \mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(z_{d,n} = k | \mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } z_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n} | z_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

- $\mathbf{Z}_{-(d,n)}$ = topic assignments of all words except $\boldsymbol{w}_{d,n}$, $\mathbf{W}_{-(d,n)}$ = all words except $\boldsymbol{w}_{d,n}$

[†] Finding Scientific Topics (Griffiths and Steyvers, 2004)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as†

$$p(z_{d,n} = k|\mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(z_{d,n} = k|\mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } z_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n}|z_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

- $\mathbf{Z}_{-(d,n)}$ = topic assignments of all words except $\boldsymbol{w}_{d,n}$, $\mathbf{W}_{-(d,n)}$ = all words except $\boldsymbol{w}_{d,n}$
- The terms on RHS can be further simplified, e.g.,

$$p(z_{d,n} = k|\mathbf{Z}_{-(d,n)}) = \int \underbrace{p(z_{d,n} = k|\theta_d)}_{\theta_{dk}} \underbrace{p(\theta_d|\mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\theta = \frac{N_d^{(k)} + \alpha}{N_d - 1 + K\alpha} \qquad \text{(verify)}$$

† Finding Scientific Topics (Griffiths and Steyvers, 2004)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as[†]

$$p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } \boldsymbol{z}_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n}|\boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

- $\mathbf{Z}_{-(d,n)}$ = topic assignments of all words except $\boldsymbol{w}_{d,n}$, $\mathbf{W}_{-(d,n)}$ = all words except $\boldsymbol{w}_{d,n}$
- The terms on RHS can be further simplified, e.g.,

$$p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{z}_{d,n} = k|\theta_d)}_{\theta_{dk}} \underbrace{p(\theta_d|\mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\theta = \frac{N_d^{(k)} + \alpha}{N_d - 1 + K\alpha} \qquad \text{(verify)}$$

$$p(\boldsymbol{w}_{d,n} = v|\boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{w}_{d,n} = v|\boldsymbol{z}_{d,n} = k, \beta_k)}_{\beta_{kv}} \underbrace{p(\beta_k|\mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\beta_k$$

[†] Finding Scientific Topics (Griffiths and Steyvers, 2004)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as[†]

$$p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } \boldsymbol{z}_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n}|\boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

- $\mathbf{Z}_{-(d,n)}$ = topic assignments of all words except $\boldsymbol{w}_{d,n}$, $\mathbf{W}_{-(d,n)}$ = all words except $\boldsymbol{w}_{d,n}$
- The terms on RHS can be further simplified, e.g.,

$$p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{z}_{d,n} = k|\theta_d)}_{\theta_{dk}} \underbrace{p(\theta_d|\mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\theta = \frac{N_d^{(k)} + \alpha}{N_d - 1 + K\alpha} \qquad \text{(verify)}$$

$$p(\boldsymbol{w}_{d,n} = v|\boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{w}_{d,n} = v|\boldsymbol{z}_{d,n} = k, \beta_k)}_{\beta_{kv}} \underbrace{p(\beta_k|\mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\beta_k = \text{(?? try yourself!)}$$

[†] Finding Scientific Topics (Griffiths and Steyvers, 2004)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as†

$$p(\boldsymbol{z}_{d,n} = k | \mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(\boldsymbol{z}_{d,n} = k | \mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } \boldsymbol{z}_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n} | \boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

- $\mathbf{Z}_{-(d,n)}$ = topic assignments of all words except $\boldsymbol{w}_{d,n}$, $\mathbf{W}_{-(d,n)}$ = all words except $\boldsymbol{w}_{d,n}$
- The terms on RHS can be further simplified, e.g.,

$$p(\boldsymbol{z}_{d,n} = k | \mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{z}_{d,n} = k | \theta_d)}_{\theta_{dk}} \underbrace{p(\theta_d | \mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\theta = \frac{N_d^{(k)} + \alpha}{N_d - 1 + K\alpha} \quad \text{(verify)}$$

$$p(\boldsymbol{w}_{d,n} = v | \boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{w}_{d,n} = v | \boldsymbol{z}_{d,n} = k, \beta_k)}_{\beta_{kv}} \underbrace{p(\beta_k | \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\beta_k = (\text{?? try yourself!})$$

- $N_d^{(k)}$ is the number of words in document d assigned to topic k (excluding the n^{th} word)

† Finding Scientific Topics (Griffiths and Steyvers, 2004)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as[†]

$$p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } \boldsymbol{z}_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n}|\boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

- $\mathbf{Z}_{-(d,n)}$ = topic assignments of all words except $\boldsymbol{w}_{d,n}$, $\mathbf{W}_{-(d,n)}$ = all words except $\boldsymbol{w}_{d,n}$
- The terms on RHS can be further simplified, e.g.,

$$p(\boldsymbol{z}_{d,n} = k|\mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{z}_{d,n} = k|\theta_d)}_{\theta_{dk}} \underbrace{p(\theta_d|\mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\theta = \frac{N_d^{(k)} + \alpha}{N_d - 1 + K\alpha} \quad \text{(verify)}$$

$$p(\boldsymbol{w}_{d,n} = v|\boldsymbol{z}_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{w}_{d,n} = v|\boldsymbol{z}_{d,n} = k, \beta_k)}_{\beta_{kv}} \underbrace{p(\beta_k|\mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\beta_k = \text{(?? try yourself!)}$$

- $N_d^{(k)}$ is the number of words in document d assigned to topic k (excluding the n^{th} word)
- Note: Can compute the document topic proportions vector θ_d and topic β_k using the inferred $\mathbf{Z}$

$$\theta_{dk} = \frac{\sum_{n=1}^{N_d} 1[\boldsymbol{z}_{d,n} = k]}{N_d}$$

[†] Finding Scientific Topics (Griffiths and Steyvers, 2004)

A Collapsed Gibbs Sampler for LDA

- Collapse θ_d's and β_k's (due to Dirichlet-multinomial conjugacy) and only infer $\mathbf{Z}$ as[†]

$$p(z_{d,n} = k|\mathbf{Z}_{-(d,n)}, \mathbf{W}) \propto \underbrace{p(z_{d,n} = k|\mathbf{Z}_{-(d,n)})}_{\text{collapsed prior for } z_{d,n}} \underbrace{p(\boldsymbol{w}_{d,n}|z_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{collapsed likelihood for } \boldsymbol{w}_{d,n}}$$

- $\mathbf{Z}_{-(d,n)}$ = topic assignments of all words except $\boldsymbol{w}_{d,n}$, $\mathbf{W}_{-(d,n)}$ = all words except $\boldsymbol{w}_{d,n}$
- The terms on RHS can be further simplified, e.g.,

$$p(z_{d,n} = k|\mathbf{Z}_{-(d,n)}) = \int \underbrace{p(z_{d,n} = k|\theta_d)}_{\theta_{dk}} \underbrace{p(\theta_d|\mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\theta = \frac{N_d^{(k)} + \alpha}{N_d - 1 + K\alpha} \qquad \text{(verify)}$$

$$p(\boldsymbol{w}_{d,n} = v|z_{d,n} = k, \mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)}) = \int \underbrace{p(\boldsymbol{w}_{d,n} = v|z_{d,n} = k, \beta_k)}_{\beta_{kv}} \underbrace{p(\beta_k|\mathbf{W}_{-(d,n)}, \mathbf{Z}_{-(d,n)})}_{\text{a Dirichlet}} d\beta_k = \text{(?? try yourself!)}$$

- $N_d^{(k)}$ is the number of words in document d assigned to topic k (excluding the n^{th} word)
- Note: Can compute the document topic proportions vector θ_d and topic β_k using the inferred $\mathbf{Z}$

$$\theta_{dk} = \frac{\sum_{n=1}^{N_d} 1[z_{d,n} = k]}{N_d} \qquad \text{and} \qquad \beta_{kv} = \frac{\sum_{d=1}^{D} \sum_{n=1}^{N_d} 1[z_{d,n} = k, \boldsymbol{w}_{d,n} = v]}{\sum_{d=1}^{D} N_d}$$

† Finding Scientific Topics (Griffiths and Steyvers, 2004)

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

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

.. where $\boldsymbol{w}_d$ collectively denotes the set of words in 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

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

.. where $\boldsymbol{w}_d$ collectively denotes the set of words in document d

- We set aside some documents as a test set and compute the average proability 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

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

.. where $\boldsymbol{w}_d$ collectively denotes the set of words in document d

- We set aside some documents as a test set and compute the average proabillity 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 vs Matrix Factorization

- LDA is sort of equivalent to a non-negative matrix factorization model for discrete data
- Input: $V \times N$ word x document matrix $\mathbf{X}$ (of discrete values)

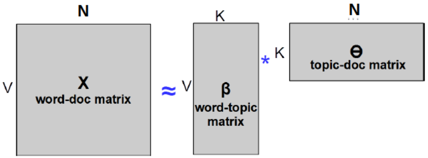
	Docs																			
Terms	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
data	1	1	0	0	2	0	0	0	0	0	1	2	1	1	1	0	1	0	0	0
examples	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
introduction	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
mining	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
network	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
package	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

LDA vs Matrix Factorization

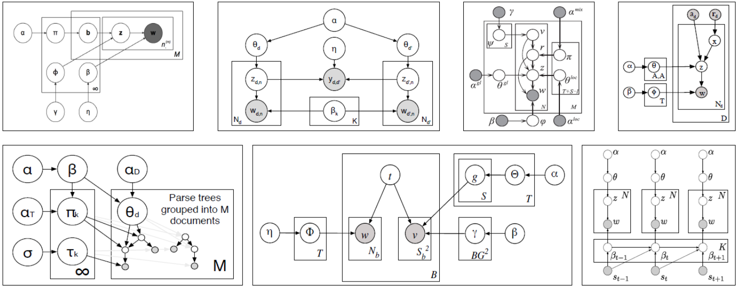
- LDA is sort of equivalent to a non-negative matrix factorization model for discrete data
- Input: $V \times N$ word x document matrix $\mathbf{X}$ (of discrete values)

Terms	Docs 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
data	1	1	0	0	2	0	0	0	0	0	1	2	1	1	1	0	1	0	0	0
examples	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
introduction	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
mining	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
network	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
package	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

- Output: $V \times K$ word-topic matrix β, $K \times N$ topic-document matrix Θ

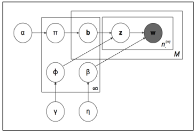


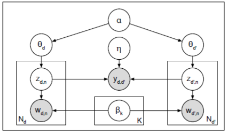
LDA is easy to extend

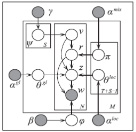


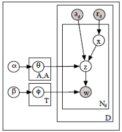
Several extensions of LDA (too many to list here :)). Some include

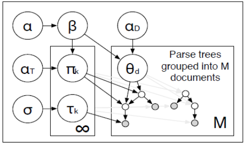
LDA is easy to extend

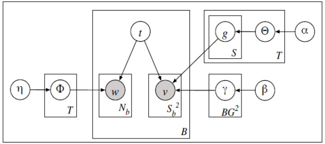


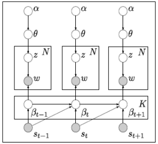








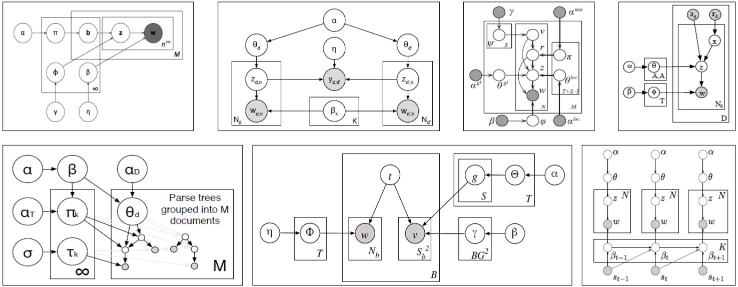




Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)

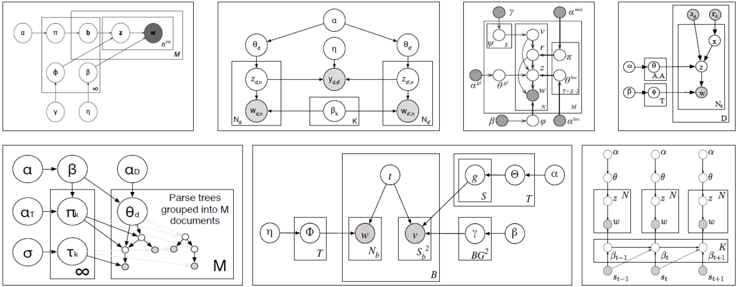
LDA is easy to extend



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

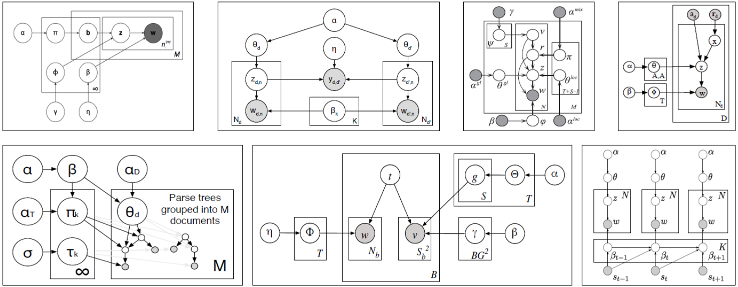
LDA is easy to extend



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

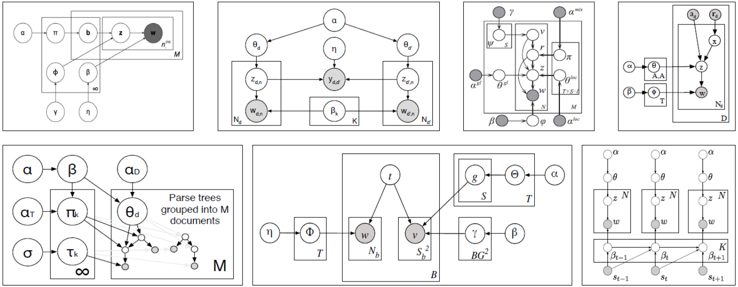
LDA is easy to extend



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), and many others..

Summary

- LDA is a simple but powerful model for discrete data (e.g., text documents)
- Easy to extend to more sophisticated models
- A variety of inference algorithms can be developed for doing inference for LDA based models
- Connections to matrix factorization methods, e.g., Poisson likelihood for the document-word count matrix with gamma priors on the latent factors (recall HW3 problem)
- Some recent work on "Deep Topic Models"