Probabilistic Topic Models

DGM4NLP

Miguel Rios
University of Amsterdam

May 5, 2019

Outline

Introduction

- Topic modelling provides models for automatically organizing, understanding, searching, and summarizing large corpus of documents.

[top]

Introduction

- Topic modelling provides models for automatically organizing, understanding, searching, and summarizing large corpus of documents.
- Discover the hidden domains in the corpus.

[top]

Introduction

- Topic modelling provides models for automatically organizing, understanding, searching, and summarizing large corpus of documents.
- Discover the hidden domains in the corpus.
- Annotate the documents according to those domains.

[top]

Introduction

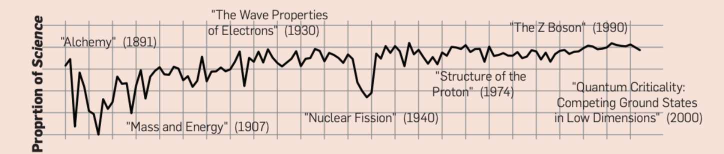
- Topic modelling provides models for automatically organizing, understanding, searching, and summarizing large corpus of documents.
- Discover the hidden domains in the corpus.
- Annotate the documents according to those domains.
- Use annotations to organise, summarise, search, and make predictions over documents.

[top]

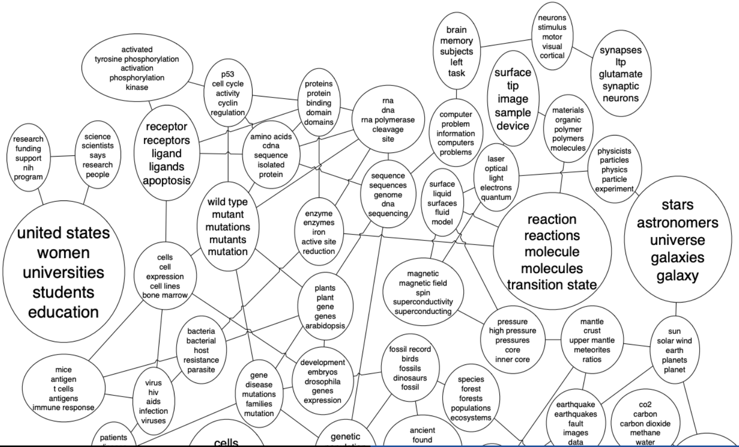
Probabilistic Topic Models

human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

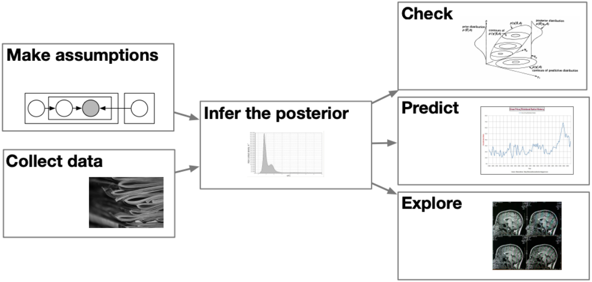
Probabilistic Topic Models



Probabilistic Topic Models



Probabilistic Models



Latent Dirichlet allocation (LDA)

- Motivation is that documents show multiple topics.

Latent Dirichlet allocation (LDA)

- Motivation is that documents show multiple topics.
- For example, in "Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense).

Latent Dirichlet allocation (LDA)

- Motivation is that documents show multiple topics.
- For example, in "Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense).
- Highlighted words related to data analysis: **computer** and **prediction**, are highlighted in blue;
  and evolutionary biology: **life** and **organism**, in pink;

Latent Dirichlet allocation (LDA)

- Motivation is that documents show multiple topics.
- For example, in "Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense).
- Highlighted words related to data analysis: **computer** and **prediction**, are highlighted in blue;
  and evolutionary biology: **life** and **organism**, in pink;
- LDA is described by its generative process, the imaginary random process by which the model assumes the documents arose.

Latent Dirichlet allocation (LDA)

- Motivation is that documents show multiple topics.
- For example, in "Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense).
- Highlighted words related to data analysis: **computer** and **prediction**, are highlighted in blue;
  and evolutionary biology: **life** and **organism**, in pink;
- LDA is described by its generative process, the imaginary random process by which the model assumes the documents arose.
- We denote a topic to be a distribution over a fixed vocabulary.

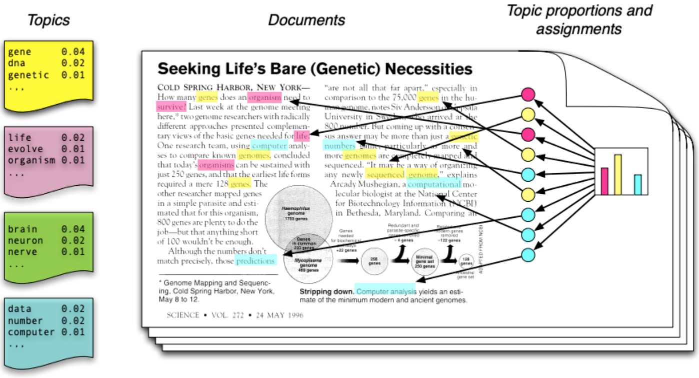
Latent Dirichlet allocation (LDA)

- Motivation is that documents show multiple topics.
- For example, in "Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense).
- Highlighted words related to data analysis: **computer** and **prediction**, are highlighted in blue;
  and evolutionary biology: **life** and **organism**, in pink;
- LDA is described by its generative process, the imaginary random process by which the model assumes the documents arose.
- We denote a topic to be a distribution over a fixed vocabulary.
- For example, the genetics topic contains words about genetics with high probability.

Latent Dirichlet allocation (LDA)

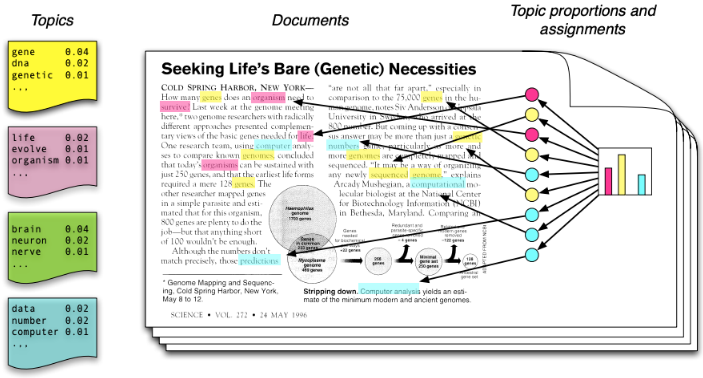
- Motivation is that documents show multiple topics.
- For example, in "Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense).
- Highlighted words related to data analysis: **computer** and **prediction**, are highlighted in blue;
  and evolutionary biology: **life** and **organism**, in pink;
- LDA is described by its generative process, the imaginary random process by which the model assumes the documents arose.
- We denote a topic to be a distribution over a fixed vocabulary.
- For example, the genetics topic contains words about genetics with high probability.
- We assume that these topics are specified before any data has

LDA



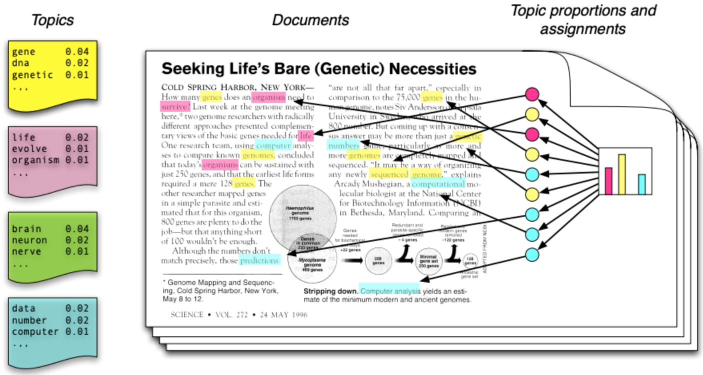
- Each topic is a distribution over words

LDA



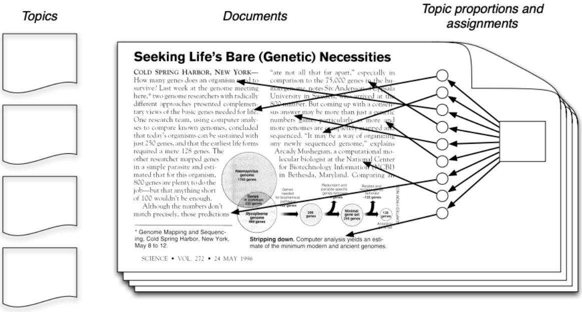
- Each topic is a distribution over words
- Each document is a mixture of corpus-wide topics

LDA



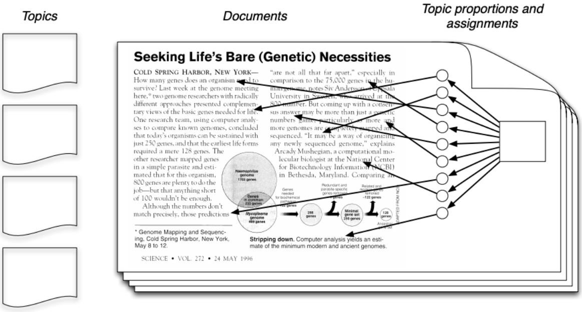
- Each topic is a distribution over words
- Each document is a mixture of corpus-wide topics

LDA Objective



- We only observe the documents

LDA Objective



- We only observe the documents
- The conditional distribution of the topic structure given the observed documents

LDA

For each document :

1. Randomly choose a distribution over topics.

LDA

For each document :

1. Randomly choose a distribution over topics.
2. For each word in the document:

LDA

For each document :

1. Randomly choose a distribution over topics.
2. For each word in the document:
   a. Randomly choose a topic from the distribution over topics in step 1.

LDA

For each document :

1. Randomly choose a distribution over topics.
2. For each word in the document:
   a. Randomly choose a topic from the distribution over topics in step 1.
   b. Randomly choose a word from the corresponding distribution over the vocabulary

LDA

For each document :

1. Randomly choose a distribution over topics.
2. For each word in the document:
   a. Randomly choose a topic from the distribution over topics in step 1.
   b. Randomly choose a word from the corresponding distribution over the vocabulary

- Each document exhibits the topics in different proportion (step1); each word in each document is drawn from one of the topics (step 2b), where the selected topic is chosen from the per-document distribution over topics (step 2a)

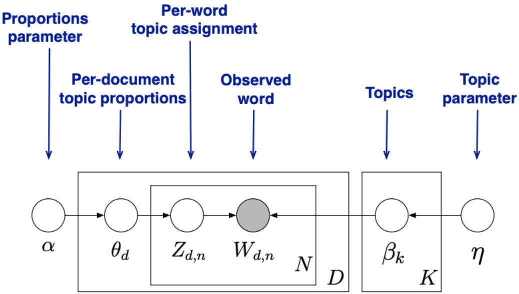
LDA

For each document :

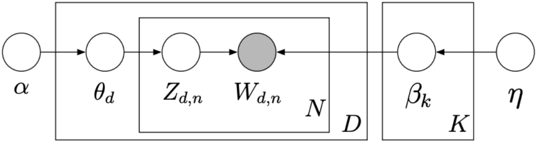
1. Randomly choose a distribution over topics.
2. For each word in the document:
   a. Randomly choose a topic from the distribution over topics in step 1.
   b. Randomly choose a word from the corresponding distribution over the vocabulary

- Each document exhibits the topics in different proportion (step1); each word in each document is drawn from one of the topics (step 2b), where the selected topic is chosen from the per-document distribution over topics (step 2a)
- From the example article, the distribution over topics would place probability on genetics, data analysis, and evolutionary biology, and each word is drawn from one of those three topics

LDA PGM

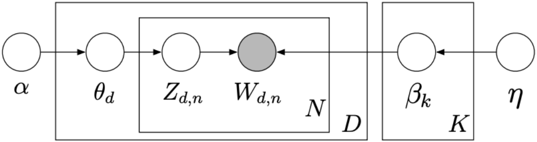

Proportions parameter
Per-word topic assignment
Per-document topic proportions
Observed word
Topics
Topic parameter
α
θ_d
$Z_{d,n}$
$W_{d,n}$
N
D
β_k
K
η

LDA PGM



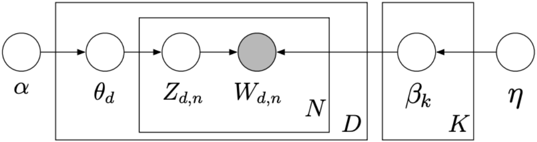
- This joint defines a posterior, $p(\theta, z, \beta|w)$.

LDA PGM



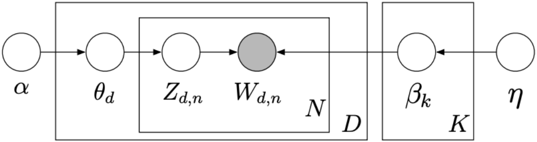
- This joint defines a posterior, $p(\theta, z, \beta | w)$.
- From a collection of documents, infer

LDA PGM



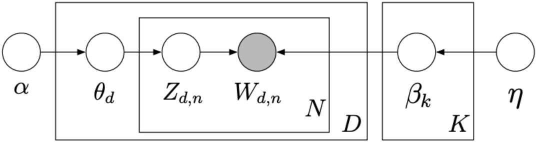
- This joint defines a posterior, $p(\theta, z, \beta | w)$.
- From a collection of documents, infer
- Per-word topic assignment $z_{d,n}$

LDA PGM



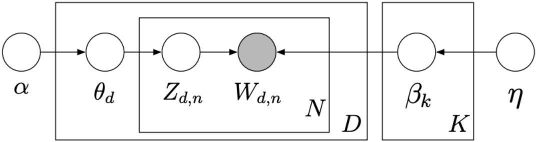
- This joint defines a posterior, $p(\theta, z, \beta | w)$.
- From a collection of documents, infer
- Per-word topic assignment $z_{d,n}$
- Per-document topic proportions θ_d

LDA PGM



- This joint defines a posterior, $p(\theta, z, \beta | w)$.
- From a collection of documents, infer
- Per-word topic assignment $z_{d,n}$
- Per-document topic proportions θ_d
- Per-corpus topic distributions β_k

LDA PGM



- This joint defines a posterior, $p(\theta, z, \beta | w)$.
- From a collection of documents, infer
- Per-word topic assignment $z_{d,n}$
- Per-document topic proportions θ_d
- Per-corpus topic distributions β_k
- Then use posterior expectations to perform the task at hand: information retrieval, document similarity, exploration, and others.

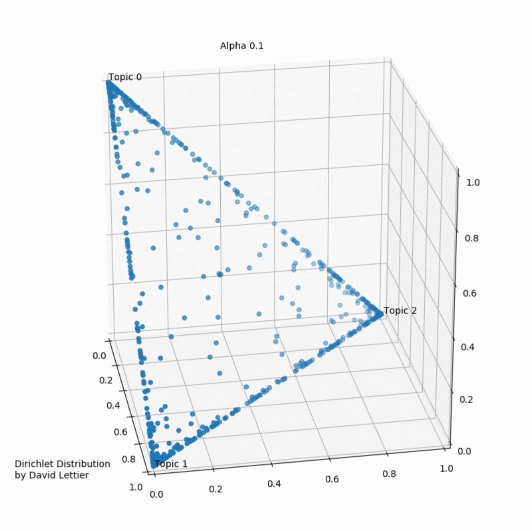
Dirichlet distribution

- The Dirichlet distribution is an exponential family distribution over the simplex, i.e., positive vectors that sum to one

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma\left(\alpha_i\right)} \prod_i \theta_i^{\alpha_i - 1} \tag{1}$$

- It is conjugate to the multinomial. Given a multinomial observation, the posterior distribution of θ is a Dirichlet.
- The parameter α controls the mean shape and sparsity of θ .
- The topic proportions are a K dimensional Dirichlet. The topics are a V dimensional Dirichlet.
- The alpha controls the mixture of topics for any given document.

Dirichlet distribution



Dirichlet distribution

- At low alpha values (less than one), most of the topic distribution samples are in the corners (near the topics).

Dirichlet distribution

- At low alpha values (less than one), most of the topic distribution samples are in the corners (near the topics).
- At alpha equal to one, any space on the surface of the triangle (3-simplex) is fair game (uniformly distributed). You could equally likely end up with a sample favoring only one topic, a sample that gives an even mixture of all the topics, or something in between.

Dirichlet distribution

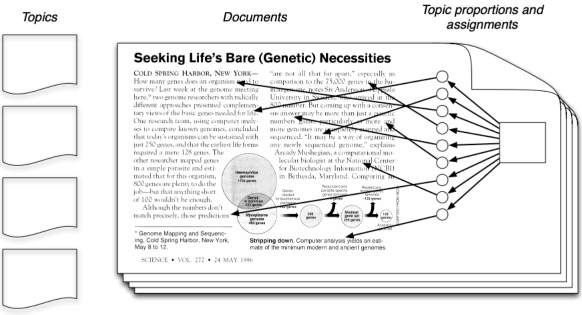
- At low alpha values (less than one), most of the topic distribution samples are in the corners (near the topics).
- At alpha equal to one, any space on the surface of the triangle (3-simplex) is fair game (uniformly distributed). You could equally likely end up with a sample favoring only one topic, a sample that gives an even mixture of all the topics, or something in between.
- For alpha values greater than one, the samples start to congregate to the center. This means that as alpha gets bigger, your samples will more likely be uniform or an even mixture of all the topics.

Working

LDA trades off two goals.

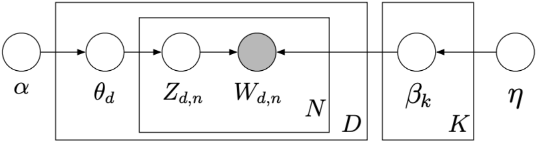
- **(1)** For each document, allocate its words to as few topics as possible.
  (2) For each topic, assign high probability to as few terms as possible.
- Putting a document in a single topic makes 2 hard: All of its words must have probability under that topic.
- Putting very few words in each topic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

Posterior Inference



- Our goal is to compute the distribution of the hidden variables conditioned on the documents
  p(topics, proportions, assignments—documents)

Posterior Inference

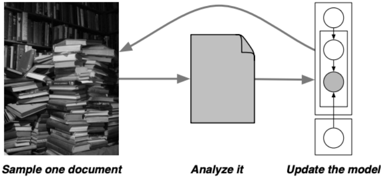


- The joint distribution of the latent variables and documents is $\prod_{i=1}^{K} p\left(\beta_i|\eta\right) \prod_{d=1}^{D} p\left(\theta_d|\alpha\right) \left(\prod_{n=1}^{N} p\left(z_{d,n}|\theta_d\right) p\left(w_{\alpha,n}|\beta_{1:k,z_{d,n}}\right)\right)$
- The posterior of the latent variables given the documents is $p(\beta, \theta, \mathbf{z}|\mathbf{w})$

Posterior Inference

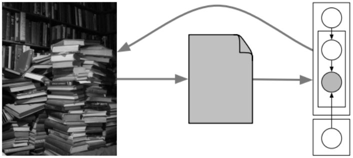
- $p(\beta, \theta, \mathbf{z} | \mathbf{w}) = \frac{p(\beta,\theta,\mathbf{z},\mathbf{w})}{\int_\beta \int_\theta \sum_\mathbf{z} p(\beta,\theta,\mathbf{z},\mathbf{w})}$
- The denominator, the marginal $p(w)$ is intractable

Posterior Inference



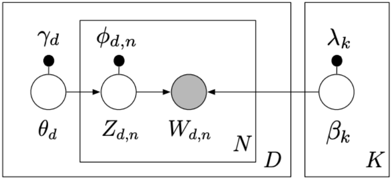
- Condition on large data sets and approximate the posterior.
- Variational inference, we optimize over a family of distributions to find the member closest in KL divergence to the posterior.

Posterior Inference



1. Sample a document w_d from the collection
2. Infer how w_d exhibits the current topics
3. Create intermediate topics, formed as though the w_d is the only document.
4. Adjust the current topics according to the intermediate topics.
5. Repeat.

Mean-field variational inference for LDA

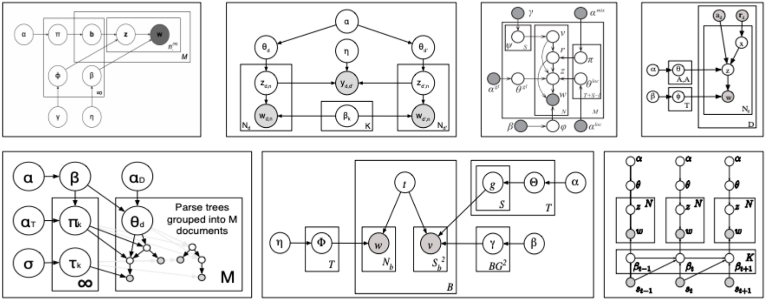


1. Document variables: Topic proportions θ and topic assignments $z_{1:N}$.
2. Corpus variables: Topics $\beta_{1:K}$
3. The variational approximation is:
   $q(\boldsymbol{\beta}, \boldsymbol{\theta}, z) = \prod_{k=1}^{K} q(\beta_k|\lambda_k) \prod_{d=1}^{D} q(\theta_d|\gamma_d) \prod_{n=1}^{N} q(z_{d,n}|\phi_{d,n})$

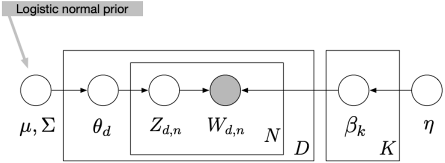
Mean-field variational inference for LDA

```
Initialize topics randomly.
repeat
    for each document do
        repeat
            Update the topic assignment variational parameters.
            Update the topic proportions variational parameters.
        until document objective converges
    end for
    Update the topics from aggregated per-document parameters.
until corpus objective converges.
```

LDA Extensions

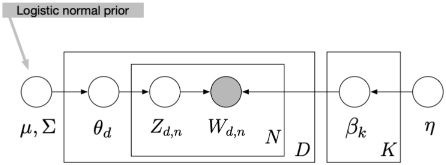


Correlated topic models



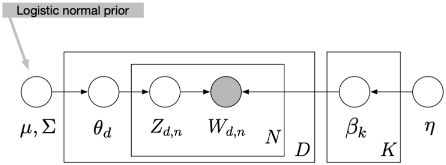
- Draw topic proportions from a logistic normal

Correlated topic models



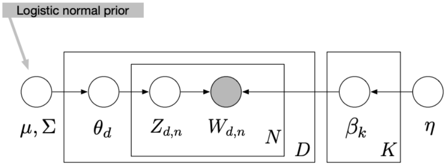
- Draw topic proportions from a logistic normal
- Allows topic occurrences to have correlation.

Correlated topic models



- Draw topic proportions from a logistic normal
- Allows topic occurrences to have correlation.
- Gives a map of topics and how they are related

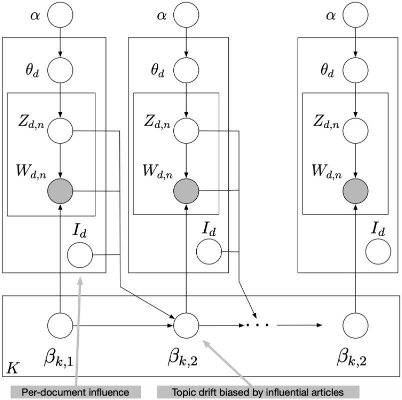
Correlated topic models



- Draw topic proportions from a logistic normal
- Allows topic occurrences to have correlation.
- Gives a map of topics and how they are related
- Better fit for observed data, but computation is more complex

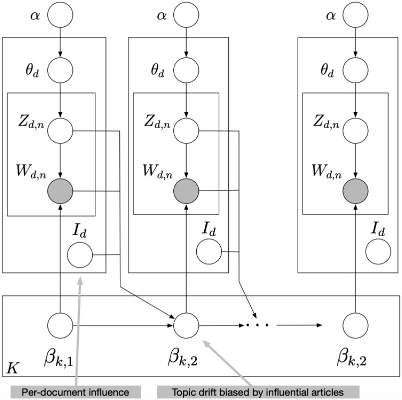
Dynamic topic models

- LDA assumes that the order of documents does not matter.



Dynamic topic models

- LDA assumes that the order of documents does not matter.
- Corpora span hundreds of years



Dynamic topic models

- Each document has an influence score Id.

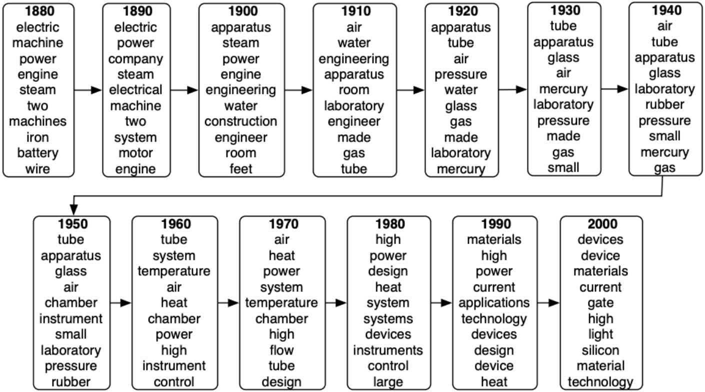
Dynamic topic models

- Each document has an influence score Id.
- Each topic is biased with the documents with high influence.

Dynamic topic models

- Each document has an influence score Id.
- Each topic is biased with the documents with high influence.
- The posterior of the influence scores could find articles that best explain the changes in language.

Dynamic topic models



Outline

1. Probabilistic Topic Models
2. Neural Variational Inference for Text Processing
3. Discovering Discrete Latent Topics
4. LDA VAE

Neural Variational Inference for Text Processing

- Neural variational framework for generative models of documents based on the variational auto-encoder.

[Miao et al., 2016]

Neural Variational Inference for Text Processing

- Neural variational framework for generative models of documents based on the variational auto-encoder.
- NVDM is a generative model of text which aims to extract a continuous semantic latent variable for each document.

[Miao et al., 2016]

Neural Variational Inference for Text Processing

- Neural variational framework for generative models of documents based on the variational auto-encoder.
- NVDM is a generative model of text which aims to extract a continuous semantic latent variable for each document.
- Model is denoted by variational auto-encoder:

[Miao et al., 2016]

Neural Variational Inference for Text Processing

- Neural variational framework for generative models of documents based on the variational auto-encoder.
- NVDM is a generative model of text which aims to extract a continuous semantic latent variable for each document.
- Model is denoted by variational auto-encoder:
- MLP encoder (inference) compresses the bag-of-words document representation into a continuous latent distribution,

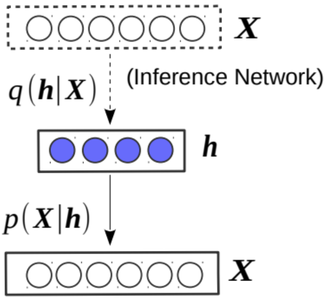
[Miao et al., 2016]

Neural Variational Inference for Text Processing

- Neural variational framework for generative models of documents based on the variational auto-encoder.
- NVDM is a generative model of text which aims to extract a continuous semantic latent variable for each document.
- Model is denoted by variational auto-encoder:
- MLP encoder (inference) compresses the bag-of-words document representation into a continuous latent distribution,
- Softmax decoder (generative model) reconstructs the document by generating the words independently.

[Miao et al., 2016]

Neural Variational Inference for Text Processing



Neural Variational Inference for Text Processing

- Let $XinR^{|V|}$ be the bag-of-words representation of a document and $x_i inR^{|V|}$ be the one-hot representation of the word at position i.

Neural Variational Inference for Text Processing

- Let $XinR^{|V|}$ be the bag-of-words representation of a document and $x_i inR^{|V|}$ be the one-hot representation of the word at position i.
- MLP encoder $q(z|x)$ compresses document representations into continuous hidden vectors

Neural Variational Inference for Text Processing

- Let $XinR^{|V|}$ be the bag-of-words representation of a document and $x_i inR^{|V|}$ be the one-hot representation of the word at position i.
- MLP encoder $q(z|x)$ compresses document representations into continuous hidden vectors
- Softmax decoder $p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{z})$ reconstructs the documents by independently generating the words.

Neural Variational Inference for Text Processing

- Let $XinR^{|V|}$ be the bag-of-words representation of a document and $x_i inR^{|V|}$ be the one-hot representation of the word at position i.
- MLP encoder $q(z|x)$ compresses document representations into continuous hidden vectors
- Softmax decoder $p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{N} p\left(\boldsymbol{x}_i|\boldsymbol{z}\right)$ reconstructs the documents by independently generating the words.
- We derive the lower bound:
  $\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{i=1}^{N} \log p_\theta\left(\boldsymbol{x}_i|\boldsymbol{z}\right)\right] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$
  where N is the number of words in the document

Data

- Standard news corpora:

Data

- Standard news corpora:
- 20NewsGroups is a collection of newsgroup documents, consisting of 11,314 training and 7,531 test articles.

Data

- Standard news corpora:
- 20NewsGroups is a collection of newsgroup documents, consisting of 11,314 training and 7,531 test articles.
- Reuters RCV1-v2 is a large collection from Reuters newswire stories with 794,414 training and 10,000 test cases.

Data

- Standard news corpora:
- 20NewsGroups is a collection of newsgroup documents, consisting of 11,314 training and 7,531 test articles.
- Reuters RCV1-v2 is a large collection from Reuters newswire stories with 794,414 training and 10,000 test cases.
- The vocabulary size of these two datasets are set as 2,000 and 10,000

Results

Model	Dim	20News	RCV1
LDA	50	1091	1437
LDA	200	1058	1142
NVDM	50	**836**	563
NVDM	200	852	**550**

- perplexity is computed $ppl = \exp\left(-\frac{1}{D}\sum_n^{N_d}\frac{1}{N_d}\log p(\boldsymbol{x}_d)\right)$, where D is the number of documents, N_d represents the length of the dth document.

Results

Model	Dim	20News	RCV1
LDA	50	1091	1437
LDA	200	1058	1142
NVDM	50	**836**	563
NVDM	200	852	**550**

- perplexity is computed $ppl = \exp\left(-\frac{1}{D}\sum_{n}^{N_d}\frac{1}{N_d}\log p\left(\boldsymbol{x}_d\right)\right)$, where D is the number of documents, N_d represents the length of the dth document.
- Since $logp(x)$ in the NVDM is the variational lower bound (which is an upper bound on perplexity).

Results

The topics learned by NVDM on 20News

Space	*Religion*	*Encryption*	*Sport*	*Policy*
orbit	muslims	rsa	goals	bush
lunar	worship	cryptography	pts	resources
solar	belief	crypto	teams	charles
shuttle	genocide	keys	league	austin
moon	jews	pgp	team	bill
launch	islam	license	players	resolution
fuel	christianity	secure	nhl	mr
nasa	atheists	key	stats	misc
satellite	muslim	escrow	min	piece
japanese	religious	trust	buf	marc

Outline

1. Probabilistic Topic Models
2. Neural Variational Inference for Text Processing
3. Discovering Discrete Latent Topics
4. LDA VAE

Discovering Discrete Latent Topics

- Introduce a neural network to parameterise the multinomial topic distribution

$$
\begin{aligned}
\theta_d &\sim \mathrm{G}\left(\mu_0, \sigma_0^2\right), \text{ for } d \in D \\
z_n &\sim \text{Multi}\left(\theta_d\right), \text{ for } n \in [1, N_d] \\
w_n &\sim \text{Multi}\left(\beta_{z_n}\right), \text{ for } n \in [1, N_d]
\end{aligned}
\tag{2}
$$

Discovering Discrete Latent Topics

- Introduce a neural network to parameterise the multinomial topic distribution

$$\begin{aligned} \theta_d &\sim \mathrm{G}\left(\mu_0, \sigma_0^2\right), \text{ for } d \in D \\ z_n &\sim \text{Multi}\left(\theta_d\right), \text{ for } n \in [1, N_d] \\ w_n &\sim \text{Multi}\left(\beta_{z_n}\right), \text{ for } n \in [1, N_d] \end{aligned} \tag{2}$$

- $G(\mu_0, \sigma_0^2)$ is composed of a NN $\theta = g(x)$ conditioned on a isotropic Gaussian $x \sim \mathcal{N}(\mu_0, \sigma_0^2 0)$

Discovering Discrete Latent Topics

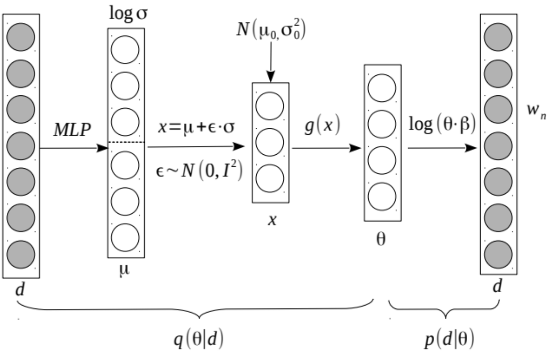
- Introduce a neural network to parameterise the multinomial topic distribution

$$\begin{aligned} \theta_d &\sim \mathrm{G}\left(\mu_0, \sigma_0^2\right), \text{ for } d \in D \\ z_n &\sim \text{Multi}\left(\theta_d\right), \text{ for } n \in [1, N_d] \\ w_n &\sim \text{Multi}\left(\beta_{z_n}\right), \text{ for } n \in [1, N_d] \end{aligned} \quad (2)$$

- $G(\mu_0, \sigma_0^2)$ is composed of a NN $\theta = g(x)$ conditioned on a isotropic Gaussian $x \sim \mathcal{N}(\mu_0, \sigma_0^2 0)$
- Gaussian Softmax Construction pass a Gaussian random vector through a softmax function to parameterise the multinomial document topic distributions.

$$\begin{aligned} x &\sim \mathcal{N}\left(\mu_0, \sigma_0^2\right) \\ \theta &= \text{softmax}\left(W_1^T x\right) \end{aligned} \quad (3)$$

Discovering Discrete Latent Topics



[Miao et al., 2017]

Discovering Discrete Latent Topics

- Neural Topic Models with a finite number of topics K.

Discovering Discrete Latent Topics

- Neural Topic Models with a finite number of topics K.
- The topic distribution over words given a topic assignment z_n is
  $p(w_n|\beta, z_n) = \text{Multi}(\beta_{z_n})$.

Discovering Discrete Latent Topics

- Neural Topic Models with a finite number of topics K.
- The topic distribution over words given a topic assignment z_n is
  $p(w_n|\beta, z_n) = \text{Multi}(\beta_{z_n})$.
- Introduce topic vectors $t \in R^{K \times H}$

Discovering Discrete Latent Topics

- Neural Topic Models with a finite number of topics K.
- The topic distribution over words given a topic assignment z_n is
  $p(w_n|\beta, z_n) = \text{Multi}(\beta_{z_n})$.
- Introduce topic vectors $t \in R^{K \times H}$
- word vectors $v \in R^{V \times H}$

Discovering Discrete Latent Topics

- Neural Topic Models with a finite number of topics K.
- The topic distribution over words given a topic assignment z_n is
  $p(w_n|\beta, z_n) = \text{Multi}(\beta_{z_n})$.
- Introduce topic vectors $t \in R^{K \times H}$
- word vectors $v \in R^{V \times H}$
- and generate the topic distributions over words by:
  $\beta_k = \text{softmax}\left(v \cdot t_k^T\right)$
  $\beta \in R^{K \times V}$ is the semantic similarity between topics and words.

Discovering Discrete Latent Topics

- With lower bound:
$\mathcal{L}_d = \sum_{n=1}^{N} \left[\log p \left(w_n | \beta, \hat{\theta} \right) \right] - D_{KL}[q(x|d) \| p(x)]$

Discovering Discrete Latent Topics

- With lower bound:
  $\mathcal{L}_d = \sum_{n=1}^{N} \left[\log p \left(w_n | \beta, \hat{\theta} \right) \right] - D_{KL}[q(x|d) \| p(x)]$
- $$\begin{aligned} \log p \left(w_n | \beta, \hat{\theta} \right) &= \log \sum_{z_n} \left[p \left(w_n | \beta_{z_n} \right) p \left(z_n | \hat{\theta} \right) \right] \\ &= \log(\hat{\theta} \cdot \beta) \end{aligned} \tag{4}$$

Discovering Discrete Latent Topics

- With lower bound:
  $\mathcal{L}_d = \sum_{n=1}^{N} \left[\log p \left(w_n | \beta, \hat{\theta} \right) \right] - D_{KL}[q(x|d) \| p(x)]$
-
$$\begin{aligned} \log p \left(w_n | \beta, \hat{\theta} \right) &= \log \sum_{z_n} \left[p \left(w_n | \beta_{z_n} \right) p \left(z_n | \hat{\theta} \right) \right] \\ &= \log(\hat{\theta} \cdot \beta) \end{aligned} \tag{4}$$

CON addition of topic diversity regularisation to the objective

Discovering Discrete Latent Topics

- Unbounded neural topic models the topics $t \in R^{\infty \times H}$ are dynamically generated by $\text{RNN}_{\text{Topic}}$ The generation of β is as follows:

$$\begin{aligned} t_k &= \text{RNN}_{\text{ Topic }}(t_{k-1}) \\ \beta_k &= \text{softmax}\left(v \cdot t_k^T\right) \end{aligned} \tag{5}$$

Discovering Discrete Latent Topics

- Unbounded neural topic models the topics $t \in R^{\infty \times H}$ are dynamically generated by $\mathrm{RNN}_{\mathrm{Topic}}$ The generation of β is as follows:

$$\begin{aligned} t_k &= \mathrm{RNN}_{\text{ Topic }}\left(t_{k-1}\right) \\ \beta_k &= \operatorname{softmax}\left(v \cdot t_k^T\right) \end{aligned} \tag{5}$$

- where v represents the word vectors, t_k is the kth topic generated by RNN

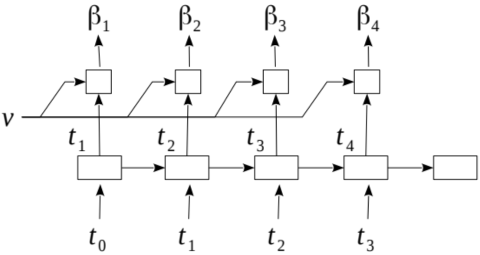
Discovering Discrete Latent Topics

- Unbounded neural topic models the topics $t \in R^{\infty \times H}$ are dynamically generated by $\text{RNN}_{\text{Topic}}$ The generation of β is as follows:

$$\begin{aligned} t_k &= \text{RNN}_{\text{ Topic }}(t_{k-1}) \\ \beta_k &= \text{softmax}\left(v \cdot t_k^T\right) \end{aligned} \tag{5}$$

- where v represents the word vectors, t_k is the kth topic generated by RNN
- If $I > \gamma$, we increase the active number of topics i by 1, $\mathcal{I} = \sum_d^D \left[\mathcal{L}_d^i - \mathcal{L}_d^{i-1}\right] / \sum_d^D \left[\mathcal{L}_d^i\right]$

Discovering Discrete Latent Topics



- $$\begin{aligned} t_k &= \text{RNN}_{\text{Topic}}\left(t_{k-1}\right) \\ \beta_k &= \text{softmax}\left(v \cdot t_k^T\right) \end{aligned} \tag{6}$$

Results

Finite Topic Model	MXM		20News		RCV1	
	50	200	50	200	50	200
GSM	**306**	**272**	**822**	830	**717**	**602**
GSB	309	296	838	826	788	634
RSB	311	297	835	**822**	750	628
OnlineLDA (Hoffman et al., 2010)	312	342	893	1015	1062	1058
NVLDA (Srivastava & Sutton, 2016)	330	357	1073	993	791	797

Unbounded Topic Model	MXM	20News	RCV1
RSB-TF	**303**	**825**	**622**
HDP (Wang et al., 2011)	370	937	918

- *MXM the Million Song Dataset with 210,519 training and

Results

Space	*Religion*	*Encryption*	*Sport*	*Science*
space	god	encryption	player	science
satellite	atheism	device	hall	theory
april	exist	technology	defensive	scientific
sequence	atheist	protect	team	universe
launch	moral	americans	average	experiment
president	existence	chip	career	observation
station	marriage	use	league	evidence
radar	system	privacy	play	exist
training	parent	industry	bob	god
committee	murder	enforcement	year	mistake

Outline

1. Probabilistic Topic Models
2. Neural Variational Inference for Text Processing
3. Discovering Discrete Latent Topics
4. LDA VAE

LDA VAE

- Effective VAE based model for LDA

[Srivastava and Sutton, 2017]

LDA VAE

- Effective VAE based model for LDA
- Dirichlet within VAE is difficult to develop an effective reparameterisation function
  Solve by constructing a Laplace approximation to the Dirichlet prior.

[Srivastava and Sutton, 2017]

LDA VAE

- Effective VAE based model for LDA
- Dirichlet within VAE is difficult to develop an effective reparameterisation function
  Solve by constructing a Laplace approximation to the Dirichlet prior.
- This approximation to the Dirichlet results in the distribution over the softmax variables

[Srivastava and Sutton, 2017]

Laplace approximation

- Approximation in the softmax basis instead of the simplex.

Laplace approximation

- Approximation in the softmax basis instead of the simplex.
- Dirichlet probability density function over the softmax variable h is:

$$P(\theta(\mathbf{h})|\alpha) = \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma\left(\alpha_k\right)} \prod_k \theta_k^{\alpha_k} g\left(\mathbf{1}^T \mathbf{h}\right) \quad (7)$$

Laplace approximation

- Approximation in the softmax basis instead of the simplex.
- Dirichlet probability density function over the softmax variable h is:

$$P(\theta(\mathbf{h})|\alpha) = \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma\left(\alpha_k\right)} \prod_k \theta_k^{\alpha_k} g\left(\mathbf{1}^T \mathbf{h}\right) \tag{7}$$

- Here $\theta = \sigma(h)$, where $\sigma(\cdot)$ represents the softmax function

LDA VAE

- Approximation to the Dirichlet results in the distribution over the softmax variables h as a multivariate normal with mean μ_1 and covariance matrix Σ_1 where:

$$\begin{aligned} \mu_{1k} &= \log \alpha_k - \frac{1}{K} \sum_i \log \alpha_i \\ \Sigma_{1kk} &= \frac{1}{\alpha_k} \left(1 - \frac{2}{K}\right) + \frac{1}{K^2} \sum_i \frac{1}{\alpha_k} \end{aligned} \tag{8}$$

LDA VAE

- Approximate of $p(\theta|\alpha)$ with $\hat{p}\left(\theta|\mu_1, \Sigma_1\right) = \mathcal{LN}\left(\theta|\mu_1, \Sigma_1\right)$

LDA VAE

- Approximate of $p(\theta|\alpha)$ with $\hat{p}\left(\theta|\mu_1, \Sigma_1\right) = \mathcal{LN}\left(\theta|\mu_1, \Sigma_1\right)$
- where LN is a logistic normal distribution with parameters μ_1, Σ_1 for k (number of topics).

LDA VAE

- Approximate of $p(\theta|\alpha)$ with $\hat{p}(\theta|\mu_1, \Sigma_1) = \mathcal{LN}(\theta|\mu_1, \Sigma_1)$
- where LN is a logistic normal distribution with parameters μ_1, Σ_1 for k (number of topics).
- and ELBO:

$$L(\boldsymbol{\Theta}) = \sum_{d=1}^{D} \Big[- \Big(\tfrac{1}{2} \Big\{ \mathrm{tr} \left(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{\Sigma}_0 \right) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_1^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - K + \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} \Big\} \\ + \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,I)} \Big[\mathbf{w}_d^{\top} \log \Big(\sigma(\boldsymbol{\beta}) \sigma \Big(\boldsymbol{\mu}_0 + \boldsymbol{\Sigma}_0^{1/2} \boldsymbol{\epsilon} \Big) \Big) \tag{9}$$

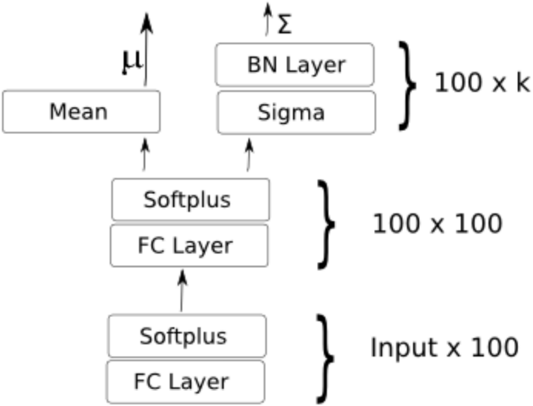
LDA VAE

- Approximate of $p(\theta|\alpha)$ with $\hat{p}\left(\theta|\mu_1, \Sigma_1\right) = \mathcal{LN}\left(\theta|\mu_1, \Sigma_1\right)$
- where LN is a logistic normal distribution with parameters μ_1, Σ_1 for k (number of topics).
- and ELBO:

$$L(\boldsymbol{\Theta}) = \sum_{d=1}^{D} \left[-\left(\frac{1}{2} \left\{ \mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{\Sigma}_0 \right) + \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 \right)^T \boldsymbol{\Sigma}_1^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 \right) - K + \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} \right\} \right. \right. \\ \left. \left. + \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,I)} \left[\mathbf{w}_d^\top \log \left(\sigma(\boldsymbol{\beta}) \sigma \left(\boldsymbol{\mu}_0 + \boldsymbol{\Sigma}_0^{1/2} \boldsymbol{\epsilon} \right) \right) \right] \right) \right] \tag{9}$$

- with $\mu_0 = f_\mu(\mathbf{w}, \boldsymbol{\delta})$ and $\boldsymbol{\Sigma}_0 = \mathrm{diag}\left(f_\Sigma(\mathbf{w}, \boldsymbol{\delta})\right)$

Architecture



Results

# topics	**ProdLDA VAE**	**LDA VAE**	**LDA DMFVI**	**LDA Collapsed Gibbs**	**NVDM**
50	1172	1059	1046	**728**	837
200	1168	1128	1195	**688**	884

ppl 20 Newsgroups

Results

Model	Topics
ProdLDA	motherboard meg printer quadra hd windows processor vga mhz connector armenian genocide turks turkish muslim massacre turkey armenians armenia greek voltage nec outlet circuit cable wiring wire panel motor install season nhl team hockey playoff puck league flyers defensive player israel israeli lebanese arab lebanon arabs civilian territory palestinian militia
LDA NVLDA	db file output program line entry write bit int return drive disk get card scsi use hard ide controller one game team play win year player get think good make use law state health file gun public issue control firearm people say one think life make know god man see
LDA DMFVI	write article dod ride right go get night dealer like gun law use drug crime government court criminal firearm control lunar flyers hitter spacecraft power us existence god go mean stephanopoulos encrypt spacecraft ripem rsa cipher saturn violate lunar crypto file program available server version include software entry ftp use
LDA Collapsed Gibbs	get right back light side like see take time one list mail send post anonymous internet file information user message thanks please know anyone help look appreciate get need email jesus church god law say christian one christ day come bike dod ride dog motorcycle write article bmw helmet get
NVDM	light die burn body life inside mother tear kill christian insurance drug different sport friend bank owner vancouver buy prayer input package interface output tape offer component channel level model price quadra hockey slot san playoff jose deal market dealer christian church gateway catholic christianity homosexual resurrection modem mouse sunday

References I

topic-models. URL http://www.cs.columbia.edu/~blei/papers/BleiLafferty2009.pdf.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1727–1736, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/miao16.html.

References II

Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering discrete latent topics with neural variational inference. *CoRR*, abs/1706.00359, 2017. URL http://arxiv.org/abs/1706.00359.

Akash Srivastava and Charles A. Sutton. Autoencoding variational inference for topic models. In *ICLR*, 2017.