TTIC 31210:
Advanced Natural Language Processing

Kevin Gimpel

Spring 2017

Lecture 12:
Bayesian Inference, Unsupervised NLP

Generative Story Template

1: Draw a set of parameters θ from $p(\theta \mid \alpha)$

2: Draw a latent structure z from $p(z \mid \theta)$

3: Draw the observed data x from $p(x \mid z, \theta)$

$$p(x, z, \theta \mid \alpha) = p(\theta \mid \alpha)\, p(z \mid \theta)\, p(x \mid z, \theta)$$

Key Quantities

$$p(x, z, \theta \mid \alpha) = p(\theta \mid \alpha)\, p(z \mid \theta)\, p(x \mid z, \theta)$$

Our data is a set of samples: $x^{(1)}, x^{(2)}, ..., x^{(n)}$

joint: $p(x^{(1)}, ..., x^{(n)}, z^{(1)}, ..., z^{(n)}, \theta \mid \alpha)$

$$= p(\theta \mid \alpha) \left(\prod_{i=1}^{n} p(z^{(i)} \mid \theta)\, p(x^{(i)} \mid z^{(i)}, \theta) \right)$$

posterior: $p(z^{(1)}, ..., z^{(n)}, \theta \mid x^{(1)}, ..., x^{(n)}, \alpha)$

collapsed posterior: $p(z^{(1)}, ..., z^{(n)} \mid x^{(1)}, ..., x^{(n)}, \alpha)$

Gibbs Sampling Template

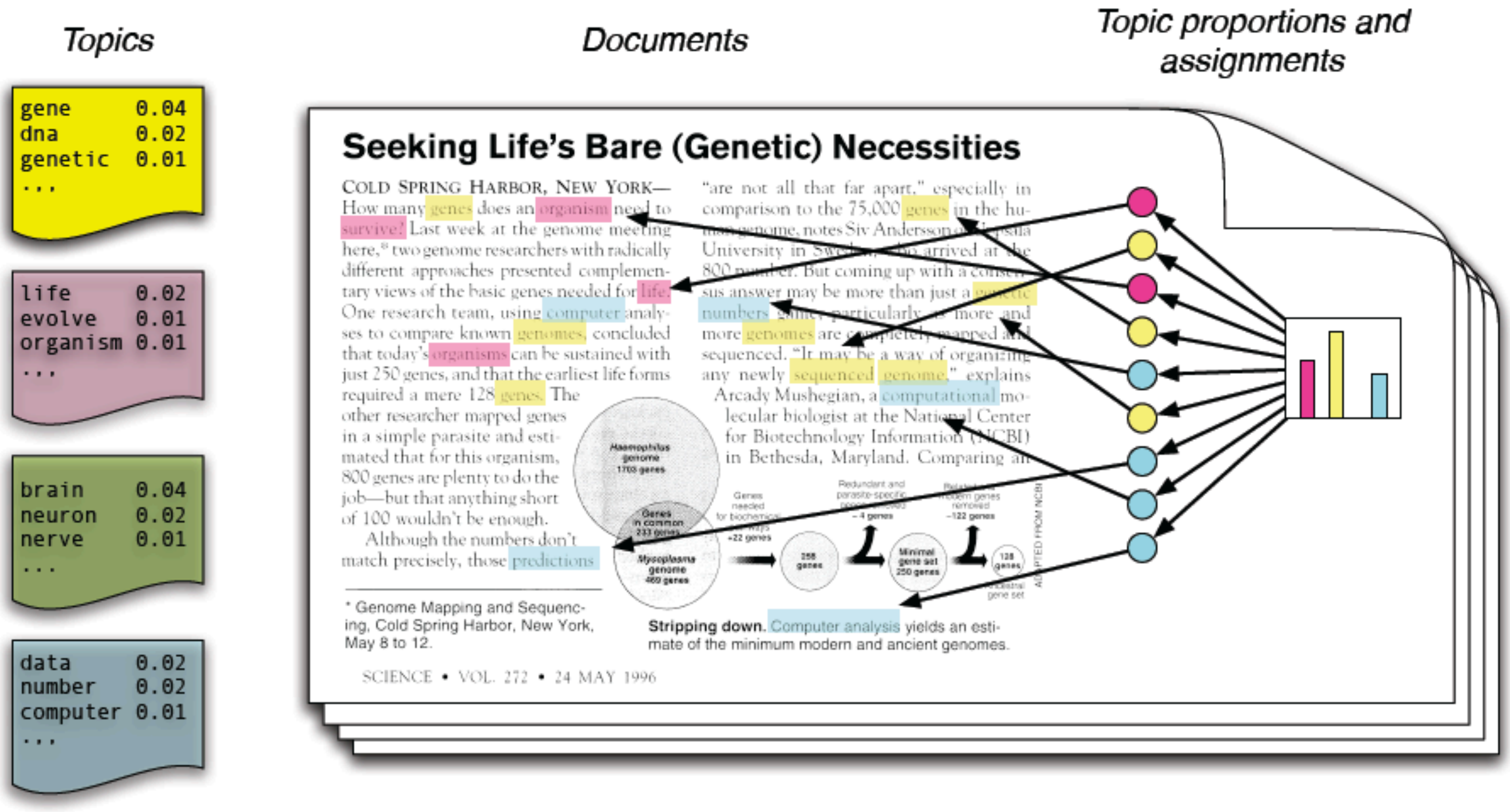
$U_1, ..., U_p =$ latent variables

$U_{-i} =$ all latent variables other than U_i

$\boldsymbol{X} =$ all observed data and hyperparameters

```
Gibbs sampling:
    initialize all $U_i$ to values $u_i$
    repeat until convergence:
        sample $u$ from $p(U_i \mid u_{-i}, \boldsymbol{X})$
        set $U_i \leftarrow u$
```

Topic Modeling



Topics

```
gene     0.04
dna      0.02
genetic  0.01
...
```

```
life     0.02
evolve   0.01
organism 0.01
...
```

```
brain    0.04
neuron   0.02
nerve    0.01
...
```

```
data     0.02
number   0.02
computer 0.01
...
```

Documents

Topic proportions and assignments

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

Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed
–4 genes
Minimal gene set 250 genes
Related to modern genes removed
–122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

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

Blei et al. (2003)

LDA Generative Story

```
1: For each topic $k = 1...K$, draw multinomial word distribution $\beta_k \sim \text{Dirichlet}(\psi)$
2: For each document $i$:
  a: Draw a multinomial topic distribution $\theta^{(i)} \sim \text{Dirichlet}(\alpha)$
  b: For each position $j$ in document $i$:
    i: Draw a topic $z^{(i,j)} \sim \text{Multinomial}(\theta^{(i)})$
    ii: Draw a word $w^{(i,j)} \sim \text{Multinomial}(\beta_{z^{(i,j)}})$
```

K = # topics
N = # documents
M = # words in each document
V = # words in vocabulary

Gibbs Sampling for LDA

$$Z^{(i,j)} \mid \text{everything else} \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$$

$$\theta^{(i)} \in \mathbb{R}^K$$

$$\beta \in \mathbb{R}^{K \times V}$$

Gibbs Sampling for LDA

$$Z^{(i,j)} \mid \text{everything else} \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$$

$$\theta^{(i)} \mid \text{everything else} \sim \text{Dirichlet}(\alpha + m^{(i)})$$

$$\beta_k \mid \text{everything else} \sim \text{Dirichlet}(\psi + n_k)$$

$\theta^{(i)} \in \mathbb{R}^K$

$\beta \in \mathbb{R}^{K \times V}$

$m_k^{(i)} = \#$ words in doc i from topic k

$n_{k,v} = \#$ of times word v appears with topic k in any document

- we now have a way to generate samples from the posterior for the LDA model
- how should we do the following?
  - get topic assignments for each word in the document collection?
  - get topic distribution for a document?
  - get estimates of topic-word distributions for each topic?

LDA

Generative Story:

$\beta_k \sim \text{Dirichlet}(\psi)$

$\theta^{(i)} \sim \text{Dirichlet}(\alpha)$

$Z^{(i,j)} \sim \text{Multinomial}(\theta^{(i)})$

Posteriors:

$\beta_k \mid \text{ everything else} \sim \text{Dirichlet}(\psi + n_k)$

$\theta^{(i)} \mid \text{ everything else} \sim \text{Dirichlet}(\alpha + m^{(i)})$

$Z^{(i,j)} \mid \text{ everything else} \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot,w^{(i,j)}})$

Conjugate Priors

- Dirichlet is (simplest) conjugate prior to multinomial
  - Dirichlet hyperparameters are like “pseudo-observations”

- definition: “posterior obtained from a given prior in the prior family and a given likelihood function belongs to the same prior family”

- direct result of “algebraic similarity” between prior family and likelihood

- often leads to tractability & closed-form analytic solutions for posterior

Key Quantities

$$p(x, z, \theta \mid \alpha) = p(\theta \mid \alpha)\, p(z \mid \theta)\, p(x \mid z, \theta)$$

Our data is a set of samples: $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$

$$\text{posterior:} \quad p(z^{(1)}, \ldots, z^{(n)}, \theta \mid x^{(1)}, \ldots, x^{(n)}, \alpha)$$

$$\boxed{\text{collapsed posterior:} \quad p(z^{(1)}, \ldots, z^{(n)} \mid x^{(1)}, \ldots, x^{(n)}, \alpha)}$$

Collapsed Gibbs Sampling for LDA

$$\text{Posterior: } Z^{(i,j)} \mid Z^{-(i,j)}, \theta, \beta, \boldsymbol{w}, \alpha, \psi \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$$

$$\text{Collapsed: } Z^{(i,j)} \mid Z^{-(i,j)}, \boldsymbol{w}, \alpha, \psi \sim ?$$

- the collapsed posterior is tricky to work with because all latent variables become coupled
- i.e., we now have fewer independence assumptions to help us simplify things
- [on board]

Collapsed Gibbs Sampling for LDA

Posterior: $Z^{(i,j)} \mid Z^{-(i,j)}, \theta, \beta, \boldsymbol{w}, \alpha, \psi \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$

Collapsed: $p(Z^{(i,j)} \mid Z^{-(i,j)}, \boldsymbol{w}, \alpha, \psi) = \dfrac{\psi + [n_{-(i,j),k}]_{w^{(i,j)}}}{V\psi + \sum_v [n_{-(i,j),k}]_v} \times \dfrac{\alpha + [m_{-(i,j)}]_k}{K\alpha + \sum_{k'} [m_{-(i,j)}]_{k'}}$

Expectation Maximization (EM)

$$\max_{\theta} \prod_{i} \sum_{z} p(x^{(i)}, z \mid \theta)$$

- EM is an algorithmic template that finds a local maximum of the marginal likelihood of the observed data

Expectation Maximization (EM)

$$\max_\theta \prod_i \sum_z p(x^{(i)}, z \mid \theta)$$

- working instead with the log-likelihood:

$$\sum_i \log \sum_z p(x^{(i)}, z \mid \theta)$$

$$= \sum_i \log \sum_z q_i(z) \frac{p(x^{(i)}, z \mid \theta)}{q_i(z)}$$

- where q_i is some distribution over values for *z*

Expectation Maximization (EM)

$$\max_\theta \prod_i \sum_z p(x^{(i)}, z \mid \theta)$$

- working instead with the log-likelihood:

$$\sum_i \log \sum_z p(x^{(i)}, z \mid \theta)$$

$$= \sum_i \log \sum_z q_i(z) \frac{p(x^{(i)}, z \mid \theta)}{q_i(z)}$$

via Jensen's inequality $$\geq \sum_i \sum_z q_i(z) \log \frac{p(x^{(i)}, z \mid \theta)}{q_i(z)}$$

Expectation Maximization (EM)

$$\max_{\theta} \prod_{i} \sum_{z} p(x^{(i)}, z \mid \theta)$$

- maximize lower bound of the log-likelihood:

$$\sum_{i} \sum_{z} q_i(z) \log \frac{p(x^{(i)}, z \mid \theta)}{q_i(z)}$$

- alternate between optimizing wrt *q* and theta

EM

- "E" step:
  - compute posteriors over latent variables:

$$\text{for each } i,\ q_i(z) = p(z \mid x^{(i)}, \theta)$$

EM

- "E" step:
  - – compute posteriors over latent variables:

$$\text{for each } i,\ q_i(z) = p(z \mid x^{(i)}, \theta)$$

- "M" step:
  - – update parameters given posteriors:

$$\theta = \operatorname*{argmax}_{\theta'} \sum_i \sum_z q_i(z) \log \frac{p(x^{(i)}, z \mid \theta')}{q_i(z)}$$

EM for LDA

- "E" step:
  - compute posteriors over latent variables:

for each document i and word position j,

$$q_{(i,j)}(z) = p(z^{(i,j)} \mid \theta, \beta, \boldsymbol{w}, \alpha, \psi)$$

EM for LDA

- "E" step:
  - compute posteriors over latent variables:

    for each document i and word position j,

    $$q_{(i,j)}(z) = p(z^{(i,j)} \mid \theta, \beta, \boldsymbol{w}, \alpha, \psi)$$

- "M" step:
  - update parameters given posteriors:

$$\langle \theta, \beta \rangle = \operatorname*{argmax}_{\langle \theta', \beta' \rangle} \sum_{(i,j)} \sum_{z} q_{(i,j)}(z) \log \frac{p(x^{(i,j)}, z^{(i,j)} \mid \theta', \beta', \alpha, \psi)}{q_{(i,j)}(z)}$$