Latent Dirichlet Allocation

BM1 Advanced Natural Language Processing

Alexander Koller

26 January 2016

with help from Christoph Teichmann and illustrations by Martín Villalba
Today’s lecture is about a method called *Latent Dirichlet Allocation (LDA)*.

We care about it for two reasons:

- It’s an unsupervised method for identifying *topics* and words that are representative of them.
- It’s a showcase for a family of statistical models called *Bayesian models* which are important in CL right now.
Let’s start simple

- You and I are playing a coin-tossing game. I see you throw 63x H, 37x T. Should I believe that the coin is fair?

- Our model of the coin has one parameter, \( p = P(H) \).

- Maximum-likelihood estimate: \( p = 0.63 \), i.e. not fair.

- But what about
  - my uncertainty about \( p \)?
  - my prior beliefs about the fairness of the coin?
Bayesian Models

- ML estimation and similar methods deliver *point estimates*: a single value for each parameter that optimizes some criterion (e.g. likelihood).

- Bayesian models: assume a *probability distribution* over parameters and estimate the shape of the pd.
  - assume a *prior* over parameters, which encodes beliefs in parameter values before making any observations
  - update prior to *posterior* after making some observations
  - uncertainty about parameter values is reflected at all times in the pd
The Dirichlet distribution

- Take the parameter $p$ itself as the value of a random variable.
  - need a probability distribution over real numbers; more specifically, over tuples of numbers that sum to one

- We use the Dirichlet distribution.

  $p_1, \ldots, p_K \sim \text{Dir}(\alpha_1, \ldots, \alpha_K)$ means:

  $$P(p_1, \ldots, p_K) = \frac{1}{B(\alpha)} (p_1^{\alpha_1-1} \cdots p_K^{\alpha_K-1})$$

$\alpha_1, \ldots, \alpha_K$ are called hyperparameters

Dir only defined if the $p_i$ sum to 1

this is the beta function (needed to normalize to 1)
Bayesian parameter estimation

- We are interested in \( P(M) \) over our model \( M = (p) \). This model is very simple; will make more complex later.

- Before we make any observations, we have a prior distribution: \( P(M) = \text{Dir}_{\alpha,\alpha}(p, 1-p) \)

- We can then update this to a posterior distribution based on observed data:

\[
P(M \mid D) = \frac{P(D \mid M) \cdot P(M)}{P(D)} \propto P(D \mid M) \cdot P(M)
\]
Calculating posteriors

prior: \[ P(p) = \text{Dir}_{\alpha,\alpha}(p, 1-p) \propto p^{\alpha-1} \cdot (1-p)^{\alpha-1} \]

likelihood: \[ P(i \times H, k \times T \mid p) = p^i \cdot (1-p)^k \]

posterior: \[ P(p \mid i \times H, k \times T) \propto P(i \times H, k \times T \mid p) \cdot P(p) \]
\[ \propto p^i \cdot (1-p)^k \cdot p^{\alpha-1} \cdot (1-p)^{\alpha-1} \]
\[ = p^{i+\alpha-1} \cdot (1-p)^{k+\alpha-1} \]

More precisely, we have:
\[ P(p \mid i \times H, k \times T) = \text{Dir}_{\alpha+i,\alpha+k}(p, 1-p) \]
The Dirichlet distribution

\[ P(p_1, \ldots, p_K) = \frac{1}{B(\alpha)} (p_1^{\alpha_1-1} \cdots p_K^{\alpha_K-1}) \]
Conjugate distributions

- Crucially, $P(M)$ and $P(M \mid D)$ have the same shape (product of Dirichlets). This is because Dirichlet and Categorical are *conjugate distributions*.
  - because $K = 2$ for the coin, we really only used the Beta (not Dirichlet) and Bernoulli (not Categorical) distributions

- This is makes the math very convenient.

- The hyperparameters of the Dirichlets are updated by adding the observed counts to the hp. of the priors.
  - priors thus perform smoothing in a very principled way
The next step

Say you come across some people who have been stabbed or poisoned. You know that each of them was killed by a pirate or a ninja. You can tell how each person died, but not by whom they were killed.
Our task

- We observe $N$ people with their causes of death.

- Questions we are interested in:
  - Who killed each villager?
    $z_1, \ldots, z_N \in \{\text{pi, ni}\}$
  - How many were killed by pirates, how many by ninjas?
    $P(\text{pi}) = \theta_{\text{pi}}, P(\text{ni}) = \theta_{\text{ni}}$; thus, $\theta_{\text{pi}} + \theta_{\text{ni}} = 1$
  - How likely is it that a pirate chooses to stab someone?
    $P(\text{st} \mid \text{pi}) = \phi_{\text{st} \mid \text{pi}}$; thus, $P(\text{po} \mid \text{pi}) = \phi_{\text{po} \mid \text{pi}} = 1 - \phi_{\text{st} \mid \text{pi}}$
  - How likely is it that a ninja chooses to stab someone?
    $P(\text{st} \mid \text{ni}) = \phi_{\text{st} \mid \text{ni}}$; thus, $P(\text{po} \mid \text{ni}) = \phi_{\text{po} \mid \text{ni}} = 1 - \phi_{\text{st} \mid \text{ni}}$
Fundamental approach

- Goal: Bayesian model with parameters $\theta, \phi_{pi}, \phi_{ni}$.
  - maximum likelihood: try to estimate concrete values for each parameter
  - Bayesian: estimate *probability distribution* $P(\theta, \phi_{pi}, \phi_{ni})$
Generative story: Idea

\[ \theta \]

\[ \phi_{pi} \quad \phi_{ni} \]
Generative story: Idea

\[
\theta \rightarrow z_1 \rightarrow z_2 \rightarrow \ldots \rightarrow z_N
\]

\[
\phi_{pi} \quad \phi_{ni}
\]
Generative story: Idea
Generative story: Idea
Generative story

- We assume deaths are generated as follows:

$$(\theta_{pi}, \theta_{ni}) \sim \text{Dir}(\alpha, \alpha)$$

$$(\phi_{st|pi}, \phi_{po|pi}), (\phi_{st|ni}, \phi_{po|ni}) \sim \text{Dir}(\beta, \beta)$$

$z_1, \ldots, z_K \sim \text{Categorical}(\theta)$

$w_i \sim \text{Categorical}(\phi_{zi})$

- That is:
  - $P(z_i = pi) = \theta_{pi}, P(z_i = ni) = \theta_{ni}$
  - if $z_i$ came out as “pi”, then $P(w_i = st) = \phi_{st|pi}$

I abbreviate $\theta = (\theta_{pi}, \theta_{ni}), \phi_{pi} = (\phi_{st|pi}, \phi_{po|pi}), \phi_{ni} = (\phi_{st|ni}, \phi_{po|ni})$. $\alpha, \beta$ are assumed given and are called hyperparameters.
Supervised learning

If all killers are known, $P(M | D)$ is easy to compute.

$$P(M) = \text{Dir}_{\alpha, \alpha}(\theta) \cdot \text{Dir}_{\beta, \beta}(\phi_{pi}) \cdot \text{Dir}_{\beta, \beta}(\phi_{ni})$$

$$\propto \theta_{pi}^{\alpha-1} \cdot \theta_{ni}^{\alpha-1} \cdot \phi_{st|pi}^{\beta-1} \cdot \phi_{st|ni}^{\beta-1} \cdot \phi_{po|pi}^{\beta-1} \cdot \phi_{po|ni}^{\beta-1}$$

$$P(D | M) = P(z_1 = \text{pi}, w_1 = \text{st}, z_2 = \text{ni}, w_2 = \text{po})$$

$$= \theta_{pi} \cdot \phi_{st|pi} \cdot \theta_{ni} \cdot \phi_{po|ni}$$

$$P(M | D) \propto P(D | M) \cdot P(M)$$

$$\propto \theta_{pi}^{\alpha} \cdot \theta_{ni}^{\alpha} \cdot \phi_{st|pi}^{\beta} \cdot \phi_{st|ni}^{\beta} \cdot \phi_{po|pi}^{\beta} \cdot \phi_{po|ni}^{\beta}$$

$$\propto \text{Dir}_{\alpha+1, \alpha+1}(\theta) \cdot \text{Dir}_{\beta+1, \beta}(\phi_{pi}) \cdot \text{Dir}_{\beta, \beta+1}(\phi_{ni})$$

<table>
<thead>
<tr>
<th>i</th>
<th>$z_i$</th>
<th>$w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>![Character 1]</td>
<td>![Character 2]</td>
</tr>
<tr>
<td>2</td>
<td>![Character 3]</td>
<td>![Character 4]</td>
</tr>
</tbody>
</table>

$\alpha = (1, 1)$

$\alpha = (2, 2)$

$\alpha = (2, 1)$
Unsupervised learning

- In the original scenario, we can only observe deaths, not killers. Then $P(D \mid M)$ is less convenient:

\[
P(D \mid M) = P(w_1 = st, w_2 = po) = \sum_{k_1, k_2 \in \{pi, ni\}} P(z_1 = k_1, w_1 = st, z_2 = k_2, w_2 = po)
\]

- This sums over a number of terms that is exponential in $N$, and thus infeasible to compute.

- In practice, we compute only *expected values* under $P(M \mid D)$, and only *approximately*, using *sampling*.
Let’s extend our model a bit: \( M = (\theta, \phi_{pi}, \phi_{ni}, z_1, \ldots, z_N) \). Data now only consists of \( D = (w_1, \ldots, w_N) \).

Useful expected values of functions \( f(M, D) \):

- Expected value of pirate/ninja mixing proportion
  \[
  E_{P(M|D)}[\theta_{pi}] = \int P(M|D) \cdot \theta_{pi}(M) \, dM
  \]

- Expected value of pirate habits
  \[
  E_{P(M|D)}[\phi_{st|pi}] = \int P(M|D) \cdot \phi_{st|pi}(M) \, dM
  \]

- Expected value \( \approx \) probability that first villager was killed by a pirate
  \[
  E_{P(M|D)}[z_1 = pi] = \int P(M \mid D) \cdot \|z_1(M) = pi\| \, dM
  \]
Gibbs Sampling

- *Gibbs sampling* is a Markov Chain Monte Carlo (MCMC) method for estimating such expectations.

- At any time $t$, we are in a *state* and make a random transition into some other state.
  - state in Gibbs sampler is guess of entire model

\[
\begin{align*}
M_1 & \rightarrow M_2 & \rightarrow M_4 \\
\downarrow & & \downarrow \\
M_3 & \rightarrow M_5
\end{align*}
\]

\[
t = 1 \quad t = 2 \quad t = 3
\]
Gibbs Sampling

- Fundamental idea of Gibbs sampling:
  - split state into smaller blocks
  - in each step, resample one block based on all others
Gibbs Sampling

- Fundamental idea of Gibbs sampling:
  - split state into smaller blocks
  - in each step, resample one block based on all others
Gibbs Sampling

- Fundamental idea of Gibbs sampling:
  - split state into smaller blocks
  - in each step, resample one block based on all others

\[
P(z_1 = \pi_i | w, z_2 = n_i, M)
\]

|   | \(z_1\) | \(z_2\) | \(\theta_{\pi_i}\) | \(\phi_{st|\pi_i}\) | \(\phi_{st|n_i}\) |
|---|---------|---------|-------------------|-------------------|-------------------|
|   | 0.5     | 0.8     | 0.4               |                   |                   |
Gibbs Sampling

- Fundamental idea of Gibbs sampling:
  - split state into smaller blocks
  - in each step, resample one block based on all others

\[
P(z_1 = \text{pi} \mid w, z_2 = \text{ni}, M)
\]

\[
P(z_1 = \text{ni} \mid w, z_2 = \text{ni}, M)
\]
Gibbs Sampling

- Fundamental idea of Gibbs sampling:
  - split state into smaller blocks
  - in each step, resample one block based on all others

\[
P(z_1 = \text{pi} \mid w, z_2 = \text{ni}, M)
\]

\[
P(z_1 = \text{ni} \mid w, z_2 = \text{ni}, M)
\]

\[
P(z_2 = \text{pi} \mid w, z_1 = \text{ni}, M)
\]

\[
P(z_2 = \text{ni} \mid w, z_1 = \text{ni}, M)
\]
**Gibbs Sampling**

- Fundamental idea of Gibbs sampling:
  - split state into smaller blocks
  - in each step, resample one block based on all others

\[
P(z_1 = \pi | w, z_2 = \text{ni}, M)
\quad \text{and}
\quad P(z_2 = \pi | w, z_1 = \text{ni}, M)
\quad \text{and}
\quad P(z_2 = \text{ni} | w, z_1 = \text{ni}, M)
\quad \text{and}
\quad P(z_1 = \text{ni} | w, z_2 = \text{ni}, M)
\]

- Parameters:
  - \(\theta_{\pi i}\)
  - \(\phi_{st|\pi}\)
  - \(\phi_{st|\text{ni}}\)
Gibbs Sampling

- Transition probabilities must be the true conditional probabilities $P(z_i \mid w, z_{-i}, M)$.

- Then can be shown that after a certain point, prob of visiting a state $M$ is close to true probability $P(M \mid D)$.

- Thus, can approximate expected value of some function $f(M, D)$ under $P(M \mid D)$ by sampling $M$’s and taking mean of $f(M, D)$ in visited states.
Collapsed Gibbs sampler for LDA

- Standard sampler for LDA is a *collapsed* Gibbs sampler: some components of the model have been integrated out.

<table>
<thead>
<tr>
<th></th>
<th>Z₁</th>
<th>Z₂</th>
<th>θₚᵢ</th>
<th>φₜₛ</th>
<th>ₚᵢ</th>
<th>φₜₛ</th>
<th>ₙᵢ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
<td>0.8</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Collapsed Gibbs sampler for LDA

- Standard sampler for LDA is a *collapsed* Gibbs sampler: some components of the model have been integrated out.

| Z₁ | Z₂ | \( \theta_{pi} \) | \( \phi_{st|pi} \) | \( \phi_{st|ni} \) |
|-----|----|-----------------|-----------------|-----------------|
| 0.5 | 0.8 | 0.4             |                 |                 |
Collapsed Gibbs sampler for LDA

- Standard sampler for LDA is a *collapsed* Gibbs sampler: some components of the model have been integrated out.
Collapsed Gibbs sampler for LDA

- Standard sampler for LDA is a *collapsed* Gibbs sampler: some components of the model have been integrated out.
Collapsed Gibbs sampler for LDA

- Standard sampler for LDA is a *collapsed* Gibbs sampler: some components of the model have been integrated out.

\[
P(z_1 = \pi \mid w, z_2 = n) \quad P(z_1 = n \mid w, z_2 = n)
\]
Collapsed Gibbs sampler for LDA

- Standard sampler for LDA is a *collapsed* Gibbs sampler: some components of the model have been integrated out.

\[
P(z_1 = \pi_i | w, z_2 = n_i)
\]

\[
P(z_1 = n_i | w, z_2 = n_i)
\]
Collapsed Gibbs sampler for LDA

- Standard sampler for LDA is a *collapsed* Gibbs sampler: some components of the model have been integrated out.

\[
P(z_1 = \text{pi} \mid w, z_2 = \text{ni})
\]

\[
P(z_1 = \text{ni} \mid w, z_2 = \text{ni})
\]
It remains to determine the transition probabilities $P(z_i \mid w, z_{-i})$.

Formula turns out to be remarkably simple:

$$P(z_i = p_i \mid w, z_{-i}) \propto P(w, z_{-i}, z_i = p_i)$$

$$= \int \int P(w, z_{-i}, z_i = p_i, \theta, \phi) \, d\theta \, d\phi$$

$$= \ldots$$

$$\propto (n_{p_i}^{(-i)} + \alpha_{p_i}) \frac{n_{p_i, w_i}^{(-i)} + \beta_{w_i | p_i}}{\sum_{w'} n_{p_i, w'}^{(-i)} + \beta_{w' | p_i}}$$

# people other than i that were killed by pirates

# people other than i that were killed by pirates using method $w'$
Topic models

given: raw documents
learn: word probs.
for (abstract) topics

learn: topic mixture
in each document

Topic proportions and assignments

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

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 Dr. Anderson. A team from University of Sydney, 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 genomes," explains Araceli Musheget, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing araceli Musheget, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

(Blei, Comm. ACM 12)
Topic modeling is almost the same problem as the pirate/ninja problem:

- abstract topics = \{pirate, ninja\}
- words in document = \{stabbed, poisoned\}

Full LDA makes two changes:

- can have K topics instead of just two, and also more than two different words
- there are M > 1 documents, and each document can have its own mixture $\theta_d$ of topics
Examples

(Blei 2012)

<table>
<thead>
<tr>
<th>“Genetics”</th>
<th>“Evolution”</th>
<th>“Disease”</th>
<th>“Computers”</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>evolution</td>
<td>disease</td>
<td>computer</td>
</tr>
<tr>
<td>genome</td>
<td>evolutionary</td>
<td>host</td>
<td>models</td>
</tr>
<tr>
<td>dna</td>
<td>species</td>
<td>bacteria</td>
<td>information</td>
</tr>
<tr>
<td>genetic</td>
<td>organisms</td>
<td>diseases</td>
<td>data</td>
</tr>
<tr>
<td>genes</td>
<td>life</td>
<td>resistance</td>
<td>computers</td>
</tr>
<tr>
<td>sequence</td>
<td>origin</td>
<td>bacterial</td>
<td>system</td>
</tr>
<tr>
<td>gene</td>
<td>biology</td>
<td>new</td>
<td>network</td>
</tr>
<tr>
<td>molecular</td>
<td>groups</td>
<td>strains</td>
<td>systems</td>
</tr>
<tr>
<td>sequencing</td>
<td>phylogentic</td>
<td>control</td>
<td>model</td>
</tr>
<tr>
<td>map</td>
<td>living</td>
<td>infectious</td>
<td>parallel</td>
</tr>
<tr>
<td>information</td>
<td>diversity</td>
<td>malaria</td>
<td>methods</td>
</tr>
<tr>
<td>genetics</td>
<td>group</td>
<td>parasite</td>
<td>networks</td>
</tr>
<tr>
<td>mapping</td>
<td>new</td>
<td>parasites</td>
<td>software</td>
</tr>
<tr>
<td>project</td>
<td>two</td>
<td>united</td>
<td>new</td>
</tr>
<tr>
<td>sequences</td>
<td>common</td>
<td>tuberculosis</td>
<td>simulations</td>
</tr>
</tbody>
</table>

15 words with highest $\phi_{k,w}$ for each topic over whole corpus
(with made-up topic label)

topic mixture for one article in *Science*
Using our notation, the posterior is mentioned, this is called the distribution of the topic structure problem, computing the conditional dependent on the where we construct a for topic modeling is commonly used sampling algorithm for topic modeling is to develop efficient methods.

Gibbs sampling is to develop efficient methods for approximating the posterior adaptations of general-purpose methods for approximating the posterior. Topic modeling algorithms form the hidden topic structure to be adapting an alternative distribution an approximation of Equation 2 by distribution.

More technically, the sum is over all possible contribution over every possible instantiation of the hidden variables. The denominator, which is known close to the true posterior. Topic modeling algorithms form a large; this sum is intractable to compute.

collections usually contain observed words at selection of the corpus, and the algorithm is to run the hidden topic variables for a particular to create Figures 1 and 3—are often to many modern probabilistic models.

To pic score

Some examples of topics from Science over time (1880-2002)

<table>
<thead>
<tr>
<th>Year</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1880</td>
<td>European</td>
</tr>
<tr>
<td>1900</td>
<td>Nuclear</td>
</tr>
<tr>
<td>1920</td>
<td>War</td>
</tr>
<tr>
<td>1940</td>
<td>Energy</td>
</tr>
<tr>
<td>1970</td>
<td>States</td>
</tr>
</tbody>
</table>

The graphical model for latent Dirichlet allocation.
Examples

some topics related to *play*

<table>
<thead>
<tr>
<th>Topic 77</th>
<th>word</th>
<th>prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUSIC</td>
<td>.090</td>
<td></td>
</tr>
<tr>
<td>DANCE</td>
<td>.034</td>
<td></td>
</tr>
<tr>
<td>SONG</td>
<td>.033</td>
<td></td>
</tr>
<tr>
<td><strong>PLAY</strong></td>
<td>.030</td>
<td></td>
</tr>
<tr>
<td>SING</td>
<td>.026</td>
<td></td>
</tr>
<tr>
<td>SINGING</td>
<td>.026</td>
<td></td>
</tr>
<tr>
<td>BAND</td>
<td>.026</td>
<td></td>
</tr>
<tr>
<td>PLAYED</td>
<td>.023</td>
<td></td>
</tr>
<tr>
<td>SANG</td>
<td>.022</td>
<td></td>
</tr>
<tr>
<td>SONGS</td>
<td>.021</td>
<td></td>
</tr>
<tr>
<td>DANCING</td>
<td>.020</td>
<td></td>
</tr>
<tr>
<td>PIANO</td>
<td>.017</td>
<td></td>
</tr>
<tr>
<td>PLAYING</td>
<td>.016</td>
<td></td>
</tr>
<tr>
<td>RHYTHM</td>
<td>.015</td>
<td></td>
</tr>
<tr>
<td>ALBERT</td>
<td>.013</td>
<td></td>
</tr>
<tr>
<td>MUSICAL</td>
<td>.013</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 82</th>
<th>word</th>
<th>prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LITERATURE</td>
<td>.031</td>
<td></td>
</tr>
<tr>
<td>POEM</td>
<td>.028</td>
<td></td>
</tr>
<tr>
<td>POETRY</td>
<td>.027</td>
<td></td>
</tr>
<tr>
<td>POET</td>
<td>.020</td>
<td></td>
</tr>
<tr>
<td>PLAYS</td>
<td>.019</td>
<td></td>
</tr>
<tr>
<td>POEMS</td>
<td>.019</td>
<td></td>
</tr>
<tr>
<td><strong>PLAY</strong></td>
<td>.015</td>
<td></td>
</tr>
<tr>
<td>LITERARY</td>
<td>.013</td>
<td></td>
</tr>
<tr>
<td>WRITERS</td>
<td>.013</td>
<td></td>
</tr>
<tr>
<td>DRAMA</td>
<td>.012</td>
<td></td>
</tr>
<tr>
<td>WROTE</td>
<td>.012</td>
<td></td>
</tr>
<tr>
<td>POETS</td>
<td>.011</td>
<td></td>
</tr>
<tr>
<td>WRITER</td>
<td>.011</td>
<td></td>
</tr>
<tr>
<td>SHAKEPHERE</td>
<td>.010</td>
<td></td>
</tr>
<tr>
<td>WRITTEN</td>
<td>.009</td>
<td></td>
</tr>
<tr>
<td>STAGE</td>
<td>.009</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 166</th>
<th>word</th>
<th>prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLAY</strong></td>
<td>.136</td>
<td></td>
</tr>
<tr>
<td>BALL</td>
<td>.129</td>
<td></td>
</tr>
<tr>
<td>GAME</td>
<td>.065</td>
<td></td>
</tr>
<tr>
<td>PLAYING</td>
<td>.042</td>
<td></td>
</tr>
<tr>
<td>HIT</td>
<td>.032</td>
<td></td>
</tr>
<tr>
<td>PLAYED</td>
<td>.031</td>
<td></td>
</tr>
<tr>
<td>BASEBALL</td>
<td>.027</td>
<td></td>
</tr>
<tr>
<td>GAMES</td>
<td>.025</td>
<td></td>
</tr>
<tr>
<td>BAT</td>
<td>.019</td>
<td></td>
</tr>
<tr>
<td>RUN</td>
<td>.019</td>
<td></td>
</tr>
<tr>
<td>THROW</td>
<td>.016</td>
<td></td>
</tr>
<tr>
<td>BALLS</td>
<td>.015</td>
<td></td>
</tr>
<tr>
<td>TENNIS</td>
<td>.011</td>
<td></td>
</tr>
<tr>
<td>HOME</td>
<td>.010</td>
<td></td>
</tr>
<tr>
<td>CATCH</td>
<td>.010</td>
<td></td>
</tr>
<tr>
<td>FIELD</td>
<td>.010</td>
<td></td>
</tr>
</tbody>
</table>

(Steyvers & Griffiths 2007)
Conclusion

- LDA and extensions for topic modeling.
  - Topics interesting in their own right, also useful in various applications.
  - Simplest useful Bayesian model in NLP.

- We used Gibbs sampling to approximate integral.
  - Alternative is Variational Bayes: approximate $P(M|D)$ on paper, then solve integral exactly.

- Limitation: Number K of topics must be given. We will fix this next time.