Intelligent Agents
Topic Analysis: LDA

Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme
Summary and Agenda

• IR Agents
  – Task/goal: Information retrieval
  – Agents visit document repositories and returns doc recommendations
  – Means:
    • Vector space (bag-of-words)
      – Dimension reduction (LSI)
    • Probability based retrieval (binary)
      – Formal Foundation of TF.IDF

• Today: Language models with dimension reduction
  – Latent Dirichlet Allocation (LDA): Topic Models

• Soon:
  – What agents can take with them
  – What agents leave at the repository (win-win)
Acknowledgments

Ramesh M. Nallapati
presentation on
Generative Topic Models for Community Analysis
&
Sina Miran
presentation on
Probabilistic Latent Semantic Indexing (PLSI)
&
David M. Blei
presentation on
Probabilistic Topic Models
Topic Models

- Statistical methods that analyze the words of texts in order to:
  - Discover the themes that run through them (topics)
  - How those themes are connected to each other
  - How they change over time
Topic Modeling Scenario

- Each topic is a distribution over words
- Each document is a mixture of corpus-wide topics
- Each word is drawn from one of those topics
In reality, we only observe the documents.

The other structures are hidden variables.

Topic modeling algorithms infer these variables from data.
Plate Notation

- Naïve Bayes Model: Compact representation
  - $C$ = topic/class (name for a word distribution)
  - $N$ = number of words in document
  - $W_i$ one specific word in corpus
  - $M$ documents, $W$ now words in documents

- Idea: Generate doc from $P(W, C)$
Generative vs. Descriptive Models

- **Generative models**: Learn $P(x, y)$
  - Tasks:
    - Predict (infer) new data
    - Transform $P(x,y)$ into $P(y \mid x)$ for classification
  - Advantages
    - Assumptions and model are explicit
    - Use well-known algorithms
- **Descriptive models**: Learn $P(y \mid x)$
  - Task: Classify data
  - Advantages
    - Fewer parameters to learn
    - Better performance for classification
Forward Sampling No Evidence

Input: Bayesian network
\[ X = \{X_1, \ldots, X_N\}, \text{N- #nodes, T - # samples} \]

Output: T samples

*Process nodes in topological order – first process the ancestors of a node, then the node itself:*

1. For \( t = 0 \) to \( T \)
2. For \( i = 0 \) to \( N \)
3. \( X_i \leftarrow \text{sample } x_i^t \text{ from } P(x_i \mid \text{pa}_i) \)

M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling", Uncertainty in AI, pp. = 149-163, 1988
Sampling A Value

What does it mean to sample $x_i^t$ from $P(X_i | p_{a_i})$?

- Assume $\text{Dom}(X_i) = \{0, 1\}$
- Assume $P(X_i | p_{a_i}) = (0.3, 0.7)$

• Draw a random number $r$ from $[0,1]$  
  If $r$ falls into $[0,0.3]$, set $X_i = 0$  
  If $r$ falls into $(0.3,1]$, set $X_i = 1$
Forward Sampling (Example)

Evidence: \(X_3 = 0\)

// generate sample \(k\)
1. Sample \(x_1\) from \(P(x_1)\)
2. Sample \(x_2\) from \(P(x_2 \mid x_1)\)
3. Sample \(x_3\) from \(P(x_3 \mid x_1)\)
4. If \(x_3 \neq 0\), reject sample and start from 1, otherwise
5. sample \(x_4\) from \(P(x_4 \mid x_2, x_3)\)

Rejection sampling (rather inefficient)
Earlier Topic Models: Topics Known

- **Unigram**
  - No context information

\[
P(w_1, \ldots, w_N) = \prod_i P(w_i)
\]

fifth, an, of, futures, the, an, incorporated, a, a, the, inflation, most, dollars, quarter, in, is, mass

thrift, did, eighty, said, hard, 'm, july, bullish

that, or, limited, the

Automatically generated sentences from a unigram model
Multinomial Naïve Bayes

- How to specify Domain(C)?
  - Domain(C) = \{1, 2, \ldots, k\}
  - Domain(C) = \{0, 1\}^k

- How to specify \( P(c_d) \)?
  - Define a table
    
    | C  | P(C)  |
    |----|-------|
    | 1  | \( p_1 \) |
    | \ldots | \ldots |
    | K  | \( p_K \) |

  - or use parameterized distribution \( \pi = (p_1, \ldots, p_K) \)
    
    \[ P(C=c|\pi) = \prod_{k=1}^{K} \pi_k^{z_k} \]
Recap: Binomial Distribution

- Describes the number of successes in a series of independent trials with two possible outcomes “success” or “no success”
- \( n = \text{#trials} \)
  \( p = \frac{\text{#successful trials}}{n} \)
- **Description of frequency** of having exactly \( k \) successful trials as a function
  \[
  B_{p, n}(k) = \binom{n}{k} p^k (1 - p)^{n-k}
  \]
- It holds: \( \sum_{i=0} B_{p, n}(i) = 1 \)
- If \( n=1 \): Bernoulli distribution
  \[
  \binom{n}{k} = \frac{n!}{k!(n-k)!}
  \]
Multinomial Distribution $\text{Mult}(n \mid \pi)$

- Generalization of binomial distribution
  - $K$ possible outcomes instead of 2 (success or no success)
  - Probability mass function
    - $n =$ number of trials
    - $x_j \in \{0, 1\}$ a count for how often class $j$ occurs \[ \sum_{i=1}^{k} x_i = n \]
    - $p_j =$ probability of class $j$ occurring
    - $\text{Mult}(x_1, \ldots, x_K; p_1, \ldots, p_K) = \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_{i=1}^{K} p_i^{x_i}$
      - Here, the input to $\Gamma(\cdot)$ is a positive integer, $\Gamma(n) = (n - 1)!$
      - If $n=1$: called categorical distribution ("multinoulli")
      - Often written $\text{Mult}(.; p_1, \ldots, p_K)$ or $\text{Mult}(.; | p_1, \ldots, p_K)$
      - Generates a one-hot vector
Sampling

• A variable value $a$ can be sampled from a discrete distribution
  \[ \pi = (p_1, \ldots, p_K) \]

• Notation: $a \sim \text{Mult}(\cdot | \pi)$

• Generate random number $x$ from $(0, 1]$
  • Find $l \in \{1, 2, \ldots, k\}$ such that
    \[ \sum_{i=1}^{l-1} p_i < x \leq \sum_{i=1}^{l} p_i \]
  • Return $(z_1, \ldots, z_K)$ such that $z_l = 1$ and $z_i = 0$ für $i \neq l$
Multinomial with Matrices

- Let $\beta$ be a $K \times V$ matrix ($V$ vocabulary size), each row denotes a word distribution of a topic.
- Select row $k$ before applying multinomial:
  - Notation: $\text{Mult}(\cdot | \beta_k)$ or $\text{Mult}(\cdot | \beta, k)$ or $\text{Mult}(\cdot | k, \beta)$
Mixture of Unigrams: Known Topics

- **Multinomial Naïve Bayes**
  - For each document \( d = 1, \ldots, M \)
    - Generate \( c_d \sim \text{Mult}( . \mid \pi) \)
  - For each position \( i = 1, \ldots, N_d \)
    - Generate \( w_i \sim \text{Mult}( . \mid \beta, c_d) \)

\[
\prod_{d=1}^{M} P(w_1, \ldots, w_{N_d}, c_d \mid \beta, \pi) = \prod_{d=1}^{M} \pi_{c_d} \prod_{i=1}^{N_d} \beta_{c_d,w_i}
\]

\[
\pi_{c_d} := P(c_d \mid \pi) \quad \beta_{c_d,w_i} := P(w_i \mid \beta, c_d)
\]

multinomial
Mixture of Unigrams: Unknown Topics

- Topics/classes are hidden
  - Joint probability of words and classes
    \[
    \prod_{d=1}^{M} P(w_1, \ldots, w_{N_d}, z_d | \beta, \pi) = \prod_{d=1}^{M} \pi_{z_d} \prod_{i=1}^{N_d} \beta_{z_d, w_i}
    \]
  - Sum over topics (\(K = \) number of topics)
    \[
    \prod_{d=1}^{M} P(w_1, \ldots, w_{N_d} | \beta, \pi) = \prod_{d=1}^{M} \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}
    \]
    \[
    \pi_{z_k} := P(z_k | \pi) \quad \beta_{z_k, w_i} := P(w_i | \beta, z_k)
    \]


Mixture of Unigrams: Learning

• Learn parameters $\pi$ and $\beta$

$$\arg\max_{\beta, \pi} \prod_{d=1}^{M} P(w_1, ..., w_{N_d} | \beta, \pi)$$

$$P(w_1, ..., w_{N_d} | \beta, \pi) = \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}$$

• Use likelihood

$$\sum_{d=1}^{M} \log P(w_1, ..., w_{N_d} | \beta, \pi) = \sum_{d=1}^{M} \log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}$$

• Solve

$$\arg\max_{\beta, \pi} \sum_{d=1}^{M} \log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}$$

– Not a concave/convex function
– Note: a non-concave/non-convex function is not necessarily convex/concave
– Possibly no unique max, many saddle or turning points
  No easy way to find roots of derivative
Trick: Optimize Lower Bound

\[
\gamma, \theta
\]
Mixture of Unigrams: Learning

- The problem

\[
\arg\max_{\beta, \pi} \sum_{d=1}^{M} \log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}
\]

- Optimize w.r.t. each document

- Derive lower bound

\[
\log \sum_{i} \gamma_i x_i \geq \sum_{i} \gamma_i \log x_i \text{ where } \gamma_i \geq 0 \land \sum_{i} \gamma_i = 1
\]

Jensen's inequality

\[
\log(a \cdot b) \geq a \cdot \log b
\]

\[
\log \sum_{i} x_i = \log \sum_{i} \gamma_i \frac{x_i}{\gamma_i} \geq \sum_{i} (\gamma_i \log x_i - \gamma_i \log \gamma_i)
\]

Entropy of \( \gamma \)

Sometimes called \( I(.) \)

\[
\log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i} \geq \sum_{k=1}^{K} \left( \gamma_k \log \left( \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i} \right) \right) + H(\gamma)
\]
The model

\[ \pi \]

\[ P(z_k | \pi) \]

\[ \gamma \]

\[ P(w_i | \beta, z_k) \]

\[ \beta \]

\[ N \]

\[ M \]

\[ \pi_{z_k} := P(z_k | \pi) \]

\[ \beta_{z_k, w_i} := P(w_i | \beta, z_k) \]
Mixture of Unigrams: Learning

- Optimization problem for each document

\[ \arg\max_{\beta, \pi} \sum_{k=1}^{K} \left( \gamma_k \log(\pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}) \right) + H(\gamma) \]

- We have introduced a new latent variable \( \gamma \) to approximate the original functional to be optimized

- Each document is assumed to be associated with a latent variable \( \gamma \in [0,1]^K, \Sigma_k \gamma_k = 1 \) independent of other random variables

- Can be seen as a class in the new space \( \gamma_k, \pi_{z_k}, \beta_{z_k, w_i} \)
Mixture of Unigrams: Learning

- New optimization problem:
  \[
  \arg \max_{\beta, \pi} \sum_{k=1}^{K} \left( \gamma_k \log(\pi_{z_k}) \prod_{i=1}^{N_d} \beta_{z_k, w_i} \right) + H(\gamma)
  \]

- Solution: Expectation Maximization
  - Iterative algorithm to find local optimum
  - Guess values of \( \gamma_k, \pi_{z_k}, \beta_{z_k, w_i} \)
  - Compute \( \gamma_k = P(\gamma_k | \pi_{z_k}, \beta_{z_k, w_i}) \) according to model
  - Use maximum-likelihood estimation to optimize \( \pi_{z_k}, \beta_{z_k, w_i} \) until no further improvement

- Guaranteed to maximize a lower bound on the log-likelihood of the observed data

- Use \( \pi_{z_k}, \beta_{z_k, w_i} \) to estimate \( P(z_k | \pi), P(w_i | \beta, z_k) \), respectively
Graphical Idea of the EM Algorithm

\[ \theta = (\pi_K, \beta_K, w_i) \]

Log-likelihood with latent variable

\[ \mathcal{L} (\gamma_k, \theta) \]

\[ \ln p(X | \theta) \]
The model

\[ P(z_k | \pi) \]

\[ \pi_{z_k} := P(z_k | \pi) \]

\[ \beta_{z_k, w_i} := P(w_i | \beta, z_k) \]
Mixture of Unigrams: Learning

- **EM solution**
  - E step (compute $\gamma_k = P(\gamma_k | \pi_{z_k}, \beta_{z_k, w_i})$)
    $$\gamma_k^{(t+1)} = \frac{\gamma_k^{(t)} \pi_{z_k}^{(t)} \prod_{i=1}^{N d} \beta_{z_k, w_i}^{(t)}}{\sum_{j=1}^{K} \gamma_{z_j}^{(t)} \pi_{z_j}^{(t)} \prod_{i=1}^{N d} \beta_{z_j, w_i}^{(t)}}$$
  - M step (maximum likelihood optimization: use frequencies)
    $$\pi_{z_k}^{(t+1)} = \frac{\sum_{d=1}^{M} \gamma_{d k}^{(t)}}{M}$$
    $$\beta_{z_k, w_i}^{(t+1)} = \frac{\sum_{d=1}^{M} \gamma_{d k}^{(t)} n(d, w_i)}{\sum_{d=1}^{M} \gamma_{d k}^{(t)} \sum_{j=1}^{N d} n(d, w_j)}$$

  Independence assumption

  $\pi_{z_k} := P(z_k | \pi)$
  $\beta_{z_k, w_i} := P(w_i | \beta, z_k)$

  $n(d, w_i)$ number of times word $w_i$ occurs in document $d$
Back to Topic Modeling Scenario

- Documents are associated with a single topic
- Words do not depend on context
  - Bag-of-words model
Probabilistic LSI

- Select a document \(d\) with probability \(P(d)\)
- For each word of \(d\) in the training set
  - Choose a topic \(z\) with probability \(P(z \mid d)\)
  - Generate a word with probability \(P(w \mid z)\)
- Documents can have multiple topics

\[
P(d, w_i) = P(d) \sum_{k=1}^{K} P(w_i \mid z_k) P(z_k \mid d)
\]

Prior Distribution for Topic Mixture

- **Goal:** topic mixture proportions for each document drawn from some distribution
  - Distribution on multinomials
    (k-tuples of non-negative numbers that sum to one)
- **The space of all of these multinomials can be interpreted geometrically as a (k-1)-simplex**
  - K-1 independent values
  - Simplex = Generalization of a triangle to (k-1) dimensions
- **Criteria for selecting our prior:**
  - It needs to be defined for a (k-1)-simplex
  - Should have nice properties

[Wikipedia]
Model – Parameters

\[ \alpha \] → Proportions parameter
(k-dimensional vector of real numbers)

\[ \theta \] → Per-document topic distribution
(k-dimensional vector of probabilities summing up to 1)

\[ z \] → Per-word topic assignment
(number from 1 to \( k \))

\[ w \] → Observed word
(number from 1 to \( v \), where \( v \) is the number of words in the vocabulary)

\[ \beta \] → Word “prior”
(\( v \)-dimensional)
LDA Model
Latent Dirichlet Allocation

- Document = mixture of topics according to a Dirichlet prior
Dirichlet Distributions

\[ p(\theta | \alpha) = \frac{\Gamma(\sum \alpha_i)}{\prod \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \]

- Defined over a (k-1)-simplex
  - Takes K non-negative arguments which sum to one.
  - Consequently it is a natural distribution to use over multinomial distributions.
- The Dirichlet parameter \( \alpha_i \) can be thought of as a prior count of the \( i^{th} \) class

\[
\text{Dir}(x_1, \ldots, x_K; p_1, \ldots, p_K) = \frac{\Gamma(\sum x_i + 1)}{\prod \Gamma(x_i + 1)} \prod_{i=1}^{K} p_i^{x_i}
\]
A panel illustrating probability density functions of a few Dirichlet distributions over a 2-simplex, for the following α vectors (clockwise, starting from the upper left corner): (1.3, 1.3, 1.3), (3,3,3), (7,7,7), (2,6,11), (14, 9, 5), (6,2,6). [Wikipedia]
LDA Model – Plate Notation

- For each document $d$,
  - Generate $\theta_d \sim \text{Dirichlet}(\alpha)$
  - For each position $i = 1, \ldots, N_d$
    - Generate a topic $z_i \sim \text{Mult}(\cdot | \theta_d)$
    - Generate a word $w_i \sim \text{Mult}(\cdot | z_i, \beta)$

$$P(\beta, \theta, z_1, \ldots, z_{N_d}, w_1, \ldots, w_{N_d})$$

$$= \prod_{d=1}^{M} P(\theta_d | \alpha) \prod_{i=1}^{N_d} P(z_i | \theta_d) P(w_i | \beta, z_i)$$
Corpus-level Parameter $\alpha$ (uniform: $\alpha_i = \alpha_j$)

- Let $\alpha = 1$
- Per-document topic distribution: $K = 10$, $D = 15$
Corpus-level Parameter $\alpha$

- $\alpha = 10$

- $\alpha = 100$
Corpus-level Parameter $\alpha$

- $\alpha = 0.1$
- $\alpha = 0.01$
What are the words’ topics and word distribs of topics?

- \( P(\beta, \theta, z|w, \alpha) \)

### 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 average genome can be sustained with just 350 genes, and that the oldest life forms required a mere 128 genes. The other researcher, using gene and organismatic reasoning, 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


Stripping down, computer analysis yields an estimate of the minimum modern and ancient genomes.
Topic-specific Words: “Smoothed” LDA Model

- Give a different word distribution to each topic
  - $\beta$ is $K \times V$ matrix ($V$ vocabulary size), each row denotes word distribution of a topic

- For each document $d$
  - Choose $\theta_d \sim \text{Dirichlet}(\alpha)$
  - Choose $\beta_k \sim \text{Dirichlet}(\eta)$
  - For each position $i = 1, \ldots, N_d$
    - Generate a topic $z_k \sim \text{Mult}(\cdot | \theta_d)$
    - Generate a word $w_i \sim \text{Mult}(\cdot | z_k, \beta_{zk})$
But why does LDA actually work?

• **Trade-off between 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

• **These goals are at odds**
  – Putting a document in a single topic makes #2 hard:
    All of its words must have non-negligible 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**
Query Answering Problem (non-smoothed version)

To which topics does a given document belong?

\[
P(\theta, z|w, \alpha, \beta) = \frac{P(\theta, z, w|\alpha, \beta)}{P(w|\alpha, \beta)}
\]

\[
P(\theta, z, w|\alpha, \beta) = P(\theta|\alpha) \prod_{i=1}^{N} P(z_i|\theta)P(w_i|z_i, \beta)
\]

\[
P(w|\alpha, \beta) = \int \sum_{k=1}^{K} P(w, \theta, z |\alpha, \beta) \, d\theta = \int \sum_{k=1}^{K} P(\theta|\alpha) \prod_{i=1}^{N} P(z_i|\theta)P(w_i|z_i, \beta) \, d\theta = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left( \prod_{k=1}^{K} \theta_k^{\alpha_k-1} \right) \left( \prod_{i=1}^{N} \sum_{k=1}^{K} \prod_{j=1}^{V} (\theta_k \beta_{kj})^{w_i^j} \right) \, d\theta
\]

This not only looks awkward, but is as well \textit{computationally intractable} in general. Coupling between $\theta$ and $\beta_{ij}$. Solution: \textit{Approximations}.

\[
p(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i-1}
\]
LDA Learning

- Parameter learning:
  - Variational Inference / EM
    - Numerical approximation using lower-bounds
    - Results in biased solutions
    - Convergence has numerical guarantees
  - Gibbs Sampling
    - Stochastic simulation
    - Unbiased solutions
    - Stochastic convergence


We have a lecture on Approximation Algorithms for Probabilistic Models!
Back to Agents

• Agents **not only** use models
• Agents *build* models that are appropriate to fulfil the agents’ goals …
  – … or maximize the utilities derived from preference structures and goals
• Agents need to *derive approximation algorithms* for query answering on the models they find appropriate

Agents also “learn” QA strategies
LDA Application: Reuters Data

• Setup
  – 100-topic LDA trained on a 16,000 documents corpus of news articles by Reuters
  – Some standard stop words removed

• Top-7 words from some of the $P(w|z)$

<table>
<thead>
<tr>
<th>“Arts”</th>
<th>“Budgets”</th>
<th>“Children”</th>
<th>“Education”</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>million</td>
<td>children</td>
<td>school</td>
</tr>
<tr>
<td>film</td>
<td>tax</td>
<td>women</td>
<td>students</td>
</tr>
<tr>
<td>show</td>
<td>program</td>
<td>people</td>
<td>schools</td>
</tr>
<tr>
<td>music</td>
<td>budget</td>
<td>child</td>
<td>education</td>
</tr>
<tr>
<td>movie</td>
<td>billion</td>
<td>years</td>
<td>teachers</td>
</tr>
<tr>
<td>play</td>
<td>federal</td>
<td>families</td>
<td>high</td>
</tr>
<tr>
<td>musical</td>
<td>year</td>
<td>work</td>
<td>public</td>
</tr>
</tbody>
</table>
LDA Application: Reuters Data

• Result

Again: “Arts”, “Budgets”, “Children”, “Education”.

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants.
Measuring Performance

- **Perplexity** of a probability model
- Describe **how well a probability distribution** or probability model **predicts** a sample
  - \( q \): Model of an unknown probability distribution \( p \) based on a training sample drawn from \( p \)
  - Evaluate \( q \) by asking how well it predicts a separate test sample \( x_1, \ldots, x_N \) also drawn from \( p \)
  - Perplexity of \( q \) w.r.t. sample \( x_1, \ldots, x_N \) defined as
    \[
    2^{-\frac{1}{N} \sum_{i=1}^{N} \log_2 q(x_i)}
    \]
  - A better model \( q \) will tend to assign higher probabilities to \( q(x_i) \) → lower perplexity ("less surprised by sample")
Perplexity of Various Models

- Unigram
- Mixture of Unigrams
- PLSA
- LDA

The graph shows the perplexity plotted against the number of topics for different models. The x-axis represents the number of topics, ranging from 0 to 100, and the y-axis represents the perplexity, ranging from 1400 to 3400. The models are compared based on how well they predict the perplexity as the number of topics increases.
Use of LDA

- A widely used topic model (Griffiths, Steyvers, 04)
- Complexity is an issue
- Use in IR:
  - Ad hoc retrieval (Wei and Croft, SIGIR 06: TREC benchmarks)
  - Improvements over traditional LM (and LSI techniques)
  - But no consensus on whether there is any improvement over a relevance model, i.e., model with relevance feedback (relevance feedback part of the TREC tests)


Acknowledgements

Topic modelling

Tomoharu Iwata
Social annotation services

- Delicious, Flickr, CiteULike, youtube, Last.fm, Technorati, Hatena
- Users can attach annotations freely to objects, and share the annotations.
Derive content-unrelated annotations

- manufacturer of camera that took the photo
  - ‘nikon’, ‘canon’
- when they were taken
  - ‘2008’, ‘november’
- remind the annotator
  - ‘toread’
- qualities
  - ‘great’, ‘*****’
- ownership
Proposed model

• generative model for contents (words) and annotations with relevance based on topic models
• infer relevance to the content for each annotation

machine-learning toread bayes ***** neuroscience

Content-related: machine-learning bayes, neuroscience

Content-unrelated: toread *****

group engineering brain develop theory learning human research systems modelling
(bag-of-words)
Latent Dirichlet allocation

[Latent Dirichlet Allocation, JMLR2003]
Correspondence LDA

Proposed model

- N: #words, M: #annotations, D: #documents, K: #topics
- each annotation is associated with a latent variable r, r=1 if content-related, r=0 otherwise
# Topics in Delicious

<table>
<thead>
<tr>
<th>Annotation</th>
<th>content word</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>money</td>
</tr>
<tr>
<td>web</td>
<td>finance</td>
</tr>
<tr>
<td>imported</td>
<td>economics</td>
</tr>
<tr>
<td>design</td>
<td>business</td>
</tr>
<tr>
<td>internet</td>
<td>economy</td>
</tr>
<tr>
<td>online</td>
<td>Finance</td>
</tr>
<tr>
<td>cool</td>
<td>financial</td>
</tr>
<tr>
<td>toread</td>
<td>investing</td>
</tr>
<tr>
<td>tools</td>
<td>bailout</td>
</tr>
<tr>
<td>blog</td>
<td>finances</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic1</th>
<th>Topic2</th>
<th>Topic3</th>
<th>Topic4</th>
<th>Topic5</th>
</tr>
</thead>
<tbody>
<tr>
<td>money</td>
<td>video</td>
<td>opensource</td>
<td>food</td>
<td>windows</td>
</tr>
<tr>
<td>financial</td>
<td>music</td>
<td>software</td>
<td>recipes</td>
<td>linux</td>
</tr>
<tr>
<td>credit</td>
<td>videos</td>
<td>programming</td>
<td>recipe</td>
<td>sysadmin</td>
</tr>
<tr>
<td>market</td>
<td>fun</td>
<td>development</td>
<td>cooking</td>
<td>Windows</td>
</tr>
<tr>
<td>economic</td>
<td>entertainment</td>
<td>linux</td>
<td>Food</td>
<td>security</td>
</tr>
<tr>
<td>october</td>
<td>funny</td>
<td>tools</td>
<td>Recipes</td>
<td>computer</td>
</tr>
<tr>
<td>economy</td>
<td>movies</td>
<td>rails</td>
<td>baking</td>
<td>microsoft</td>
</tr>
<tr>
<td>banks</td>
<td>media</td>
<td>ruby</td>
<td>health</td>
<td>network</td>
</tr>
<tr>
<td>government</td>
<td>video</td>
<td>webdev</td>
<td>vegetarian</td>
<td>Linux</td>
</tr>
<tr>
<td>bank</td>
<td>film</td>
<td>rubyonrails</td>
<td>diy</td>
<td>ubuntu</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>content word</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>money</td>
<td>music</td>
<td>project</td>
<td>recipe</td>
<td>windows</td>
</tr>
<tr>
<td>financial</td>
<td>video</td>
<td>code</td>
<td>food</td>
<td>system</td>
</tr>
<tr>
<td>credit</td>
<td>link</td>
<td>server</td>
<td>recipes</td>
<td>microsoft</td>
</tr>
<tr>
<td>market</td>
<td>tv</td>
<td>ruby</td>
<td>make</td>
<td>linux</td>
</tr>
<tr>
<td>economic</td>
<td>movie</td>
<td>rails</td>
<td>wine</td>
<td>software</td>
</tr>
<tr>
<td>october</td>
<td>itunes</td>
<td>source</td>
<td>made</td>
<td>file</td>
</tr>
<tr>
<td>economy</td>
<td>film</td>
<td>file</td>
<td>add</td>
<td>server</td>
</tr>
<tr>
<td>banks</td>
<td>amazon</td>
<td>version</td>
<td>love</td>
<td>user</td>
</tr>
<tr>
<td>government</td>
<td>play</td>
<td>files</td>
<td>eat</td>
<td>files</td>
</tr>
<tr>
<td>bank</td>
<td>interview</td>
<td>development</td>
<td>good</td>
<td>ubuntu</td>
</tr>
</tbody>
</table>
### Topics in Flickr

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>unrelated</td>
<td>dance</td>
<td>sea</td>
<td>autumn</td>
<td>rock</td>
<td>beach</td>
</tr>
<tr>
<td>2008</td>
<td>bar</td>
<td>sunset</td>
<td>trees</td>
<td>house</td>
<td>travel</td>
</tr>
<tr>
<td>nikon</td>
<td>dc</td>
<td>sky</td>
<td>tree</td>
<td>party</td>
<td>vacation</td>
</tr>
<tr>
<td>canon</td>
<td>digital</td>
<td>clouds</td>
<td>mountain</td>
<td>park</td>
<td>camping</td>
</tr>
<tr>
<td>white</td>
<td>concert</td>
<td>mountains</td>
<td>fall</td>
<td>inn</td>
<td>landscape</td>
</tr>
<tr>
<td>yellow</td>
<td>bands</td>
<td>ocean</td>
<td>garden</td>
<td>coach</td>
<td>texas</td>
</tr>
<tr>
<td>red</td>
<td>music</td>
<td>panorama</td>
<td>bortescristian</td>
<td>creature</td>
<td>lake</td>
</tr>
<tr>
<td>photo</td>
<td>washington dc</td>
<td>south</td>
<td>geotagged</td>
<td>halloween</td>
<td>cameraphone</td>
</tr>
<tr>
<td>italy</td>
<td>dancing</td>
<td>ireland</td>
<td>mud</td>
<td>mallory</td>
<td>md</td>
</tr>
<tr>
<td>california</td>
<td>work</td>
<td>oregon</td>
<td>natura</td>
<td>night</td>
<td>sun</td>
</tr>
<tr>
<td>color</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Probable Image

- **Topic 1**: Images with dance, bar, dc, digital, concert, bands, music, washington dc, dancing, work.
- **Topic 2**: Images with sea, sunset, sky, clouds, mountains, ocean, panorama, south, ireland, oregon.
- **Topic 3**: Images with autumn, trees, tree, mountain, fall, garden, bortescristian, geotagged, mud, natura.
- **Topic 4**: Images with rock, house, party, park, inn, coach, creature, halloween, mallory, night.
- **Topic 5**: Images with beach, travel, vacation, camping, landscape, texas, lake, cameraphone, md, sun.
Perplexity

The proposed method performed better than Corr-LDA in the case of noisy social annotation data.

CorrLDA

Proposed

x-axis: #topics

y-axis: perplexity

an example of data without content-unrelated tags. This data consist of patents, to which IPC code were attached by experts according to their content.
Acknowledgements

Generative Topic Models for Community Analysis

Ramesh Nallapati
http://www.cs.cmu.edu/~wcohen/10-802/lda-sep-18.ppt

&

Arthur Asuncion, Qiang Liu, Padhraic Smyth:
Statistical Approaches to Joint Modeling of Text and Network Data
What if the corpus has network structure?

CORA citation network. Figure from [Chang, Blei, AISTATS 2009]

Hyperlink modeling using LDA

- For each document $d$,
  - Generate $\theta_d \sim \text{Dirichlet}(\alpha)$
  - For each position $i = 1, \ldots, N_d$
    - Generate a topic $z_i \sim \text{Mult}(\cdot | \theta_d)$
    - Generate a word $w_i \sim \text{Mult}(\cdot | \beta_{z_i})$
  - For each citation $j = 1, \ldots, L_c$
    - Generate $z_i \sim \text{Mult}(\theta_d)$
    - Generate $c_i \sim \text{Mult}(\cdot | \gamma_{z_j})$

- Learning using variational EM, Gibbs sampling

Topic Influence in Blogs

Modeling Citation Influences - Copycat Model

- Each topic in a citing document is drawn from one of the topic mixtures of cited publications

Modeling Citation Influences

- Citation influence model: Combination of LDA and Copycat model

Modeling Citation Influences

- Citation influence graph for LDA paper
Modeling Citation Influences

- Words in LDA paper assigned to citations

<table>
<thead>
<tr>
<th>Cited Title</th>
<th>Associated Words</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probabilistic Latent Semantic Indexing</td>
<td>text(0.04), latent(0.04), modeling(0.02), model(0.02), indexing(0.01), semantic(0.01), document(0.01), collections(0.01)</td>
<td>0.49</td>
</tr>
<tr>
<td>Modelling heterogeneity with and without the Dirichlet process</td>
<td>dirichlet(0.02), mixture(0.02), allocation(0.01), context(0.01), variable(0.0135), bayes(0.01), continuous(0.01), improves(0.01), model(0.01), proportions(0.01)</td>
<td>0.25</td>
</tr>
<tr>
<td>Introduction to Variational Methods for Graphical Methods</td>
<td>variational(0.01), inference(0.01), algorithms(0.01), including(0.01), each(0.01), we(0.01), via(0.01)</td>
<td>0.22</td>
</tr>
</tbody>
</table>
Relational Topic Model (RTM) [ChangBlei 2009]

- Same setup as LDA, except now we have observed network information across documents

\[ y_{d,d'} \sim \psi(y_{d,d'} | z_d, z_{d'}, \eta) \]

“Link probability function”

Documents with similar topics are more likely to be linked.

Relational Topic Model (RTM) [ChangBlei 2009]

- For each document $d$
  - Draw topic proportions
    $\theta_d | \alpha \sim \text{Dir}(\alpha)$
  - For each word $w_{d,n}$
    - Draw assignment
      $z_{d,n} | \theta_d \sim \text{Mult}(\theta_d)$
    - Draw word
      $w_{d,n} | z_{d,n}, \beta_{1:K} \sim \text{Mult}(\beta_{z_{d,n}})$
  - For each pair of documents $d, d'$
    - Draw binary link indicator
      $y | z_d, z_{d'} \sim \psi(\cdot | z_d, z_{d'}, \eta)$
Document networks

<table>
<thead>
<tr>
<th></th>
<th># Docs</th>
<th># Links</th>
<th>Ave. Doc-Length</th>
<th>Vocab-Size</th>
<th>Link Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORA</td>
<td>4,000</td>
<td>17,000</td>
<td>1,200</td>
<td>60,000</td>
<td>Paper citation (undirected)</td>
</tr>
<tr>
<td>Netflix Movies</td>
<td>10,000</td>
<td>43,000</td>
<td>640</td>
<td>38,000</td>
<td>Common actor/director</td>
</tr>
<tr>
<td>Enron (Undirected)</td>
<td>1,000</td>
<td>16,000</td>
<td>7,000</td>
<td>55,000</td>
<td>Communication between person i and person j</td>
</tr>
<tr>
<td>Enron (Directed)</td>
<td>2,000</td>
<td>21,000</td>
<td>3,500</td>
<td>55,000</td>
<td>Email from person i to person j</td>
</tr>
</tbody>
</table>
Conclusion

• Topic Modeling

• Relational topic modeling provides a useful start for combining text and network data in a single statistical framework

• RTM can improve over simpler approaches for link prediction

• Opportunities for future work:
  – Faster algorithms for larger data sets
  – Better understanding of non-edge modeling
  – Extended models