Introduction to Topic Models

Clint P. George

Computer and Information and Science and Engineering

October 10, 2014
Outline

1. Motivation
2. Conventional Document Modeling Methods
3. Probabilistic Topic Models
4. A Possible Real-world Application
Section 1

Motivation
Keyword-based Search

A typical Information Retrieval problem:

Example

Suppose, we search for the keyword *computers* in a document collection.

We may miss documents which do not have *computers* and contain *PC*, *laptop*, *desktop*, etc.
Keyword-based Search: Limitations

- **Synonymy**: words or phrases that have similar meanings
  - e.g., *car & automobile, hood & bonnet*
- **Polysemy**: words that have more than one distinct meaning
  - e.g., the occurrence of chair in “the chair of the board” and “the chair maker”
- When we search for documents we usually look for concepts or **topics**, not keywords.
Keyword-based Search: Limitations

- Synonymy: words or phrases that have similar meanings
  - e.g., *car* & *automobile*, *hood* & *bonnet*
- Polysemy: words that have more than one distinct meaning
  - e.g., the occurrence of *chair* in “the chair of the board” and “the chair maker”

When we search for documents we usually look for concepts or topics, not keywords.

**Question**

How do we identify the underlying topics in a corpus?
Section 2

Conventional Document Modeling Methods
Conventional Document Modeling Methods

Representing Documents

A simple approach:

- Document as **bag-of-words**—ignores any word ordering in a document
- Document’s features as the appearance frequencies of unique words

Example: Vector Space Models
Vector Space Modeling (VSM)

- A typical solution for Keyword Search
- Converts a corpus into a *term-document* matrix, each cell represents a term’s relative frequency
- Translates a *document* or *keyword query* into a vector in vector space
- Measures similarity between documents by cosine scores—*small angle* ≡ *large cosine* ≡ *similar*
Term-Frequency Inverse-Document-Frequency (TF-IDF)
Salton et al. (1975)

**TF-IDF**

\[
\text{tf-idf}_{dt} = tf_{dt} \times \log \left( \frac{D}{df_t} \right), \quad d = 1, 2, \ldots, D; \quad t = 1, 2, \ldots, V
\]

where \(tf_{dt} = \) the frequency of term \(t\) in document \(d\) and \(df_t = \) the number of documents where term \(t\) appears.
Term-Frequency Inverse-Document-Frequency (TF-IDF)
Salton et al. (1975)

**TF-IDF**

\[ \text{tf-idf}_{dt} = tf_{dt} \times \log \left( \frac{D}{df_t} \right), \quad d = 1, 2, \ldots, D; \quad t = 1, 2, \ldots, V \]

where \( tf_{dt} = \) the frequency of term \( t \) in document \( d \) and \( df_t = \) the number of documents where term \( t \) appears

**Advantages:**
- Reflects how important a word is to a document in the collection
- Helps to treat common terms in the corpus
Conventional Document Modeling Methods

TF-IDF / VSM: Limitations

- **Synonymy** may cause small cosine similarity between documents, but are related
  - leads to poor *Recall*—the fraction of all relevant documents that are retrieved
- **Polysemy** may cause large cosine similarity between documents, but are unrelated
  - leads to poor *Precision*—the fraction of retrieved documents that are relevant
- Dimensionality, $D$ and $V$, can be very large
Latent Semantic Analysis (LSA)

Deerwester et al. (1990)

- Typically works on the TF-IDF matrix $X$

An approximation based on Singular Value Decomposition:

$$X \approx X' = T' \times S' \times D'^T$$

- $T'$ contains selected Eigen vectors of $XX^T$
- $S'$ contains selected Singular Values of $X$
- $D'$ contains selected Eigen vectors of $X^TX$
Latent Semantic Analysis (LSA)

Advantages:

- Identifies a *linear subspace* in the space of TF-IDF features—significant compression in large collections
- Can achieve decent retrieval results—handles Synonymy problems to some extent
- Easy to implement
Latent Semantic Analysis (LSA)

Advantages:
- Identifies a *linear subspace* in the space of TF-IDF features—significant compression in large collections
- Can achieve decent retrieval results—handles Synonymy problems to some extent
- Easy to implement

Limitations:
- It’s a linear model—may not find nonlinear dependencies between words or documents
- It’s not a probabilistic model
- Not very readable—produces dense features
Section 3

Probabilistic Topic Models
The main goals are to

- Extract the underlying topical structure of the corpus
- Represent the documents according to these topics
- Use these representations for organizing, summarizing, and searching
Probabilistic Generative Models
Blei (2009, MLSS)

- Consider the data as observations that arise from a generative, probabilistic process that includes hidden variables
- Infer the hidden variables via posterior inference
- Use the estimated model for predicting the topic structure of newly encountered data
Latent Dirichlet Allocation (LDA, Blei et al. 2003), a generative probabilistic model, that assumes

- **Document** as a bag of words
- **Topic** as a distribution over a fixed vocabulary
- **Words** are generated from document specific topic distributions\(^1\)

\(^1\)Multinomial Sampling Experiment
Documents are formed from mixtures of topics
Each document is a \textit{mixture} of corpus-wide topics

Each word is generated from one of those \textit{topics}
In real life, we only observe the documents and their words

Our goal is to **infer** the underlying topic structure
Graphical Model (Overview)
Blei (2009, MLSS)

- Nodes are random variables
- Shaded nodes represent observed words
- Edges indicate dependence
- Plates indicate replicated structure.
Graphical Model (Overview)
Blei (2009, MLSS)

- Represents a pattern of conditional dependence between the random variables
- E.g. the above graph is for the mathematical expression

$$p(y, X_1, X_2, \ldots, X_N) = p(y) \prod_{n=1}^{N} p(X_n|y)$$
LDA Graphical Model

Blei (2009, MLSS)

Nodes are random variables
LDA Graphical Model

Blei (2009, MLSS)

\[
\prod_{k=1}^{K} p_\eta(\beta_k) \prod_{d=1}^{D} p_\alpha(\theta_d) \left( \prod_{n=1}^{N_d} p(Z_{d,n} | \theta_d) p(W_{d,n} | \beta_{1:K}, Z_{d,n}) \right)
\]
What can we do with the LDA model?

Blei (2011, KDD)

- Given the corpus, infer the hidden structures:
  - Per-word topic assignment
  - Per-document topic proportions—dimensionality reduction
  - Per-corpus topic distributions

- Use them to perform information retrieval, document clustering, exploration, ...
The Computational Problem

- Computing the conditional distribution of the topic structure given the observed words

\[ p(\beta_{1:K}, \theta_{1:D}, Z_{1:D} | W_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, Z_{1:D}, W_{1:D})}{p(W_{1:D})} \]  

  - The numerator is the joint distribution of all the random variables
  - The denominator is the marginal likelihood

- Approximate posterior inference algorithms—sampling-based algorithms and variational algorithms
Real Inference with LDA
Blei (2011, KDD)

- Corpus: OCR’d collection of Science articles (1990-2000)
  - 17,000 documents
  - 20,000 unique words in the vocabulary
- Model: 100-topic LDA model using the Variational Inference algorithm (Blei et al. 2003)
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, these predictions are not all that far apart,” especially in comparison to the 75,000 genes in the human genome, notes Siv Anderson 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 Araceli Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an
15 most probable words from the most frequent topics found in the article

- **“Genetics”**
  - human
  - genome
  - dna
  - genetic
  - genes
  - sequence
  - gene
  - molecular
  - sequencing
  - map
  - information
  - genetics
  - mapping
  - project
  - sequences

- **“Evolution”**
  - evolution
  - evolutionary
  - species
  - organisms
  - life
  - origin
  - biology
  - groups
  - phylogenetic
  - living
  - diversity
  - group
  - new
  - two
  - common

- **“Disease”**
  - disease
  - host
  - bacteria
  - diseases
  - resistance
  - bacterial
  - new
  - strains
  - control
  - infectious
  - malaria
  - parasite
  - parasites
  - united
  - tuberculosis

- **“Computers”**
  - computer
  - models
  - information
  - data
  - computers
  - system
  - network
  - systems
  - model
  - parallel
  - methods
  - networks
  - software
  - new
  - simulations
Section 4

A Possible Real-world Application
Electronic Legal Discovery

- Management of electronically stored information in litigation, dispute resolution proceedings, and investigations

(flowchart)

requesting party

search, review, and produce responsive and non-privileged documents

responding party

a request for production, given a document collection
Electronic Legal Discovery

- Management of electronically stored information in litigation, dispute resolution proceedings, and investigations

A sample request

“Discover all documents or communications that describe, discuss, refer to, report on, or relate to the company’s engagement in structured commodity transactions known as prepay transactions.” —TREC 2010 Legal Track, based on the Enron dataset
Document Retrieval and Ranking Methods

Employing LDA for retrieval:

- Infer the LDA model of the corpus, using the variational Bayes algorithm (Hoffman et al. 2011)
- Let $\hat{\theta}_d$ and $\hat{\theta}_{\text{query}}$ be the estimates of $\theta_d$ and $\theta_{\text{query}}$ obtained from the VB algorithm, where $\theta_d$ and $\theta_{\text{query}}$ are the distributions on the topics for document $d$ and the query keywords.
- Compute similarity scores based on (inverse) cosine distance between $\hat{\theta}_{\text{query}}$ and $\hat{\theta}_d$s, and sort them to rank documents on relevance.
Lucene-based retrieval (use as a baseline):

- Index the corpus after parsing all documents
- Present the query keywords to the Lucene search algorithm and use its relevance response as the document’s relevance index.
Corpus: from the TREC 2010 Legal Track—query-201 dataset, 66 relevant emails out of 278 emails

Classification performance: ROCs are created using documents’ ranking scores from different methods and their true labels

Token normalization, e.g., stemming, improves performance
Evaluating Document Ranking methods

(a) ROCs with raw word tokens

(b) ROCs with normalized word tokens
Acknowlegments & Resources

- Many of the slides are taken from Dr. David Blei’s talks: MLSS-2009, KDD-2011
- Dr. David Blei’s web page
- R LDA implementation
  http://cran.r-project.org/web/packages/lda/index.html
- Python TF-IDF, LSI, LDA, and HDP implementations
  http://radimrehurek.com/gensim/index.html