Winter Conference in Statistics
Bayesian Machine Learning

Topic Models for Text

Mattias Villani

Department of Statistics
Stockholm University
and
Department of Computer and Information Science
Linköping University

Overview

- **Textual data**
- **Dirichlet distribution**
- **Topic models**
- **Application**: Finding software bugs from textual bug reports.

TEXT IS DATA

- **Digitalization**: text is becoming an important data source.
- The web, PDF documents (legal, political, medical, etc)
- **Unstructured** (not tables), yet **structured** (by language).
- **Big data**. 100K, 1M, 1B documents in a data set.
- **Pre-processing** to get data useful for statistical analysis.

Text applications

- **Language models** (predict the next word on smartphone)
- **Machine translation** (Google translate)
- **Document classification** (Shakespeare? Spam and blog filters. harmful EULA)
- **Sentiment analysis** (positive/negative sentiment in tweets or financial statements)
- **Information retrival** (Google search)
- **Part-of-speech tagging** (predict grammatical category)
- **Prediction models** based on text.
  - Predicting financial turbulence from economic press.
  - Finding bugs from bug reports

- **Categorical** data. y_k = #votes party k. **Multinomial model**:

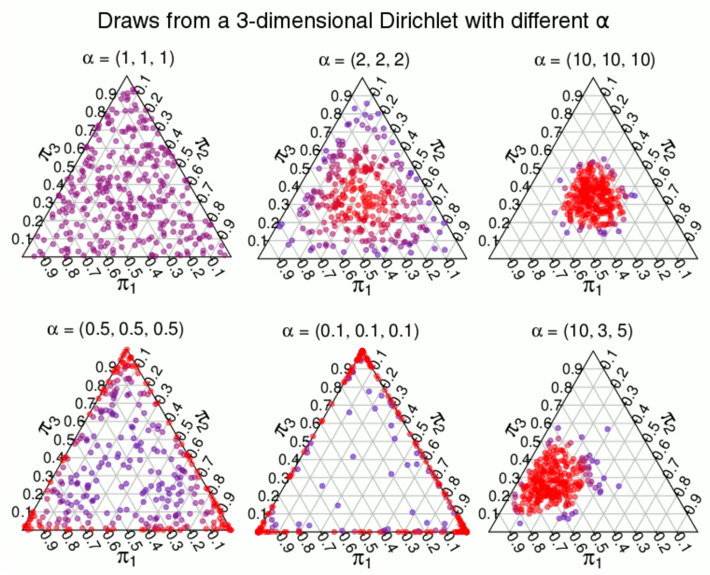
$$p(y_1, ..., y_k|\theta) \propto \prod_{k=1}^{K} \theta_k^{y_k}$$

- Needed: prior over the **unit simplex** $0 \leq \theta_k \leq 1$, $\sum_{k=1}^{K} \theta_k = 1$.
- $(\theta_1, ..., \theta_K) \sim \mathrm{Dirichlet}(\alpha_1, ..., \alpha_K)$ with density

$$p(\theta_1, ..., \theta_K) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

- Generalizes the $\mathrm{Beta}(\alpha_1, \alpha_2)$ distribution to the case $K > 2$.
- ***Prior-to-Posterior updating***

$$\begin{aligned}
\text{Model:} \quad & \mathbf{y} = (y_1, ...y_K) \sim \mathrm{Multin}(n; \theta_1, ..., \theta_K) \\
\text{Prior :} \quad & \theta = (\theta_1, ..., \theta_K) \sim \mathrm{Dirichlet}(\alpha_1, ..., \alpha_K) \\
\text{Posterior :} \quad & \theta|\mathbf{y} \sim \mathrm{Dirichlet}(\alpha_1 + y_1, ..., \alpha_K + y_K)
\end{aligned}$$

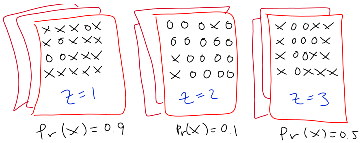


Draws from a 3-dimensional Dirichlet with different α
α = (1, 1, 1)
α = (2, 2, 2)
α = (10, 10, 10)
α = (0.5, 0.5, 0.5)
α = (0.1, 0.1, 0.1)
α = (10, 3, 5)
π1
π2
π3

- Let $\phi_1, \phi_2, \ldots, \phi_K$ be distributions over the vocabulary. **Topics**.

Topic	Word distr.	probability	dna	gene	data	distribution
1	ϕ_1	0.5	0.1	0.0	0.2	0.2
2	ϕ_2	0.0	0.5	0.4	0.1	0.0

- For each document $d = 1, \ldots, D$:
  1. Draw a **topic** z_d from a **topic distribution** $\theta = (\theta_1, \ldots, \theta_K)$.
  2. Given topic z_d, draw **words** from a **word distribution** ϕ_{z_d}.



- Each document belong to **exactly** one topic.
- **Topic models** are **mixed-membership models**.

- Assume that we have:
  - A fixed vocabulary V
  - D documents
  - N words in each document
  - K topics

1. **For each topic** ($k = 1, ..., K$):
   a. Draw a distribution over the words $\phi_k \sim Dir(\eta, \eta, ..., \eta)$
2. **For each document** ($d = 1, ..., D$):
   a. Draw a vector of topic proportions $\theta_d \sim Dir(\alpha_1, ..., \alpha_K)$
   b. **For each word** ($i = 1, ..., N$):
      i. Draw a topic assignment $z_{di} \sim Multinomial(\theta_d)$
      ii. Draw a word $w_{di} \sim Multinomial(\phi_{z_{di}})$

Example - simulation from two topics

Topic	Word distr.	probability	dna	gene	data	distribution
1	ϕ_1	0.5	0.1	0.0	0.2	0.2
2	ϕ_2	0.0	0.5	0.4	0.1	0.0

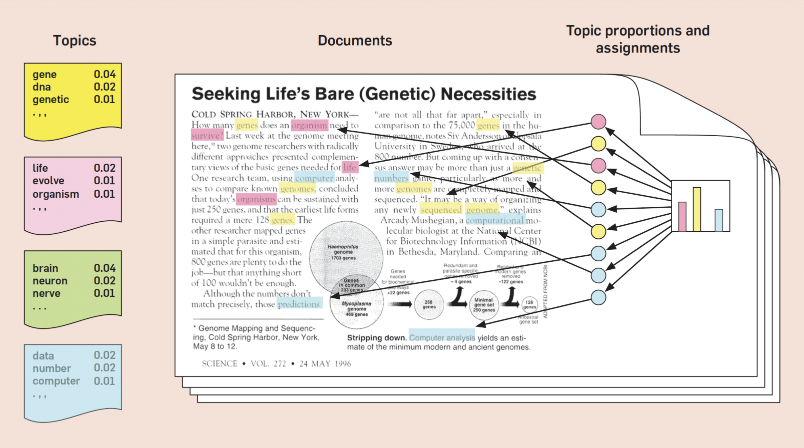
Doc 1	$\theta_1 = (0.2, 0.8)$		
	Word 1:	Topic=2	Word='gene'
	Word 2:	Topic=2	Word='gene'
	Word 3:	Topic=1	Word='data'

Doc 2	$\theta_2 = (0.9, 0.1)$		
	Word 1:	Topic=1	Word='probability'
	Word 2:	Topic=1	Word='data'
	Word 3:	Topic=1	Word='probability'

Doc 3	$\theta_2 = (0.5, 0.5)$				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

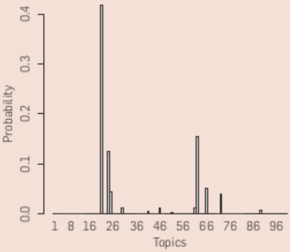
Example from Science (Blei, review paper)

Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.



Example from Science (Blei, review paper)

Figure 2. Real inference with LDA. We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left are the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in this article.



"Genetics"	"Evolution"	"Disease"	"Computers"
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

- **Posterior distribution**

$$p(\mathbf{z}, \Theta, \Phi|\mathbf{w}) \propto p(\mathbf{z}, \Theta, \Phi|\mathbf{w}) \cdot p(\mathbf{z}, \Theta, \Phi)$$

- Integrating out (**collapsing**) Θ and Φ:

$$p(\mathbf{z}|\mathbf{w}) \;=\; \int\int p(\mathbf{z}, \Theta, \Phi|\mathbf{w}) \cdot p(\mathbf{z}, \Theta, \Phi) d\Phi d\Theta$$

  will result in the following **Gibbs sampler** for the z's

$$p(z_{di} = k|w_{di} = w, \mathbf{z}_{-di}) = \underbrace{\frac{n_{k,w}^{-di} + \beta}{n_{k,\cdot}^{-di} + V\beta}}_{type-topic\ (\Phi)} \cdot \underbrace{(n_{k,d}^{-di} + \alpha)}_{topic-doc\ (\Theta)} .$$

- The rows of $\Phi|\mathbf{z}$ and $\Theta|\mathbf{z}$ are Dirichlet.
- Learned topic proportions θ_d are summaries of document content. Useful as covariates in regression/classification.

Speeding up Gibbs sampling for topic models

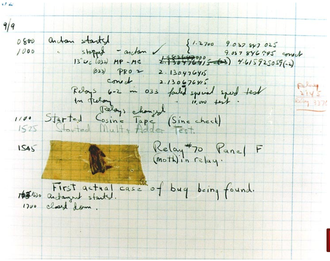
- **Gibbs sampling** from $p(z_{di} = k | w_{di} = w, \mathbf{z}_{-di})$ is **slow** since it runs serially over all tokens in the corpus.
- Example: PubMed abstracts. 10% of the data: **78.5M tokens** from **820K docs**.
- Big data lesson: with a huge number of parameters, everything matters (`log(x)` is costly ...).
- Needed: **Efficient data structures**, **sparsity**, **efficient search, efficient sort** etc.
- See my previous student **Måns Magnusson**'s PhD thesis. 2019 Cramér prize winner!

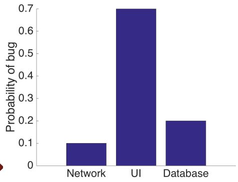
Predicting bug location from bug reports

- Predicting the location of bugs in computer code. Ericsson.
- Prediction machine: **Bug report** $\Rightarrow \Pr($**location of bug**$)$
- New **multi-class** model for high-dimensional data.
  Many covariates, **many classes**.
- **Diagonal Orthant Multinomial Probit**. No reference class.
- **DOLDA** - Diagonal Orthant Latent Dirichlet Allocation
- **Interpretable predictions** via semantical topics and aggressive **horseshoe regularization**.

Predicting bug location from bug reports

```python
#-----------------------------------------

def Network(networkInputs):

    # CODE
    # MORE CODE
    # TOO MUCH CODE

    return(networkOutputs)

#-----------------------------------------

def UI(UIinputs):

    # CODE
    # MORE CODE
    # TOO MUCH CODE

    return(UIOutputs)

#-----------------------------------------

def Database(DBinputs):

    # CODE
    # MORE CODE
    # TOO MUCH CODE

    return(DBOutputs)

#-----------------------------------------
```





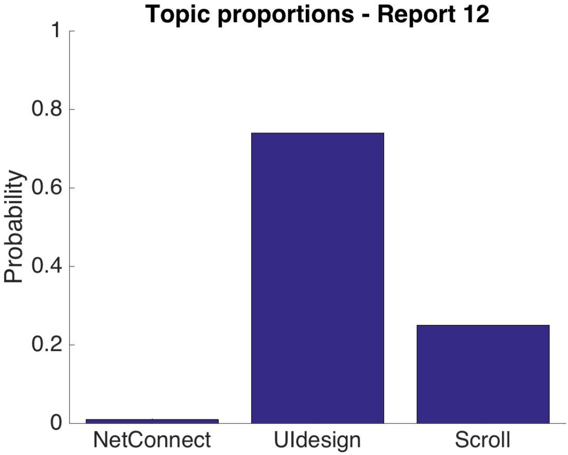
Example data

Dataset	No. Bug reports	No. classes	Vocabulary size
Mozilla	15,000	118	3505
Eclipse	15,000	49	3367
Telecom	9,778	26	5286

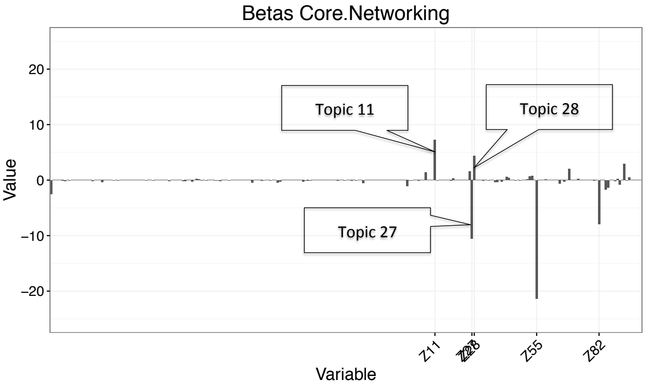
- Automatically summarize a bug report by topics.

Topic	Topic label	Top 10 words in topic
11	HTTP	proxy server http network connection request connect error www host
27	Layout	div style px background color border css width height element html
28	Connection Headers	http cache accept en public localhost gmt max modified alive
55	Search	search google bar results box type find engine enter text
82	Scrolling	scroll page scrolling mouse scrollbar bar left bottom click content

Topic proportions θ_d



- **Horseshoe regularization prior** for the $\beta_{\text{topic,class}}$:



- **DOLDA - Diagonal Orthant Latent Dirichlet Allocation**. Supervised LDA. Topics are directly related to classes.

- System:

  *- I am **very certain** that class UI contains the bug*

  *- **because** report talks a lot about UIdesign and Scroll and very little about NetConnect.*

  - Sending the bug report to the UI-team.

- System:

  *- I am **very uncertain** where the bug is*

  *- **because** bug report contains a jumble of topics.*

  - Don't trust me. Please ask human.

Interpretable prediction without loss of accuracy

Dataset	# Classes	DOLDA	StackingLDA
Mozilla	118	45%	39%
Eclipse	49	61%	55%
Telecom	26	71%	75%