

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

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

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

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

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

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

- Generalizes the $\text{Beta}(\alpha_1, \alpha_2)$ distribution to the case $K > 2$.
- Conjugate prior for **Multinomial** data

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

■ **Prior-to-Posterior updating**

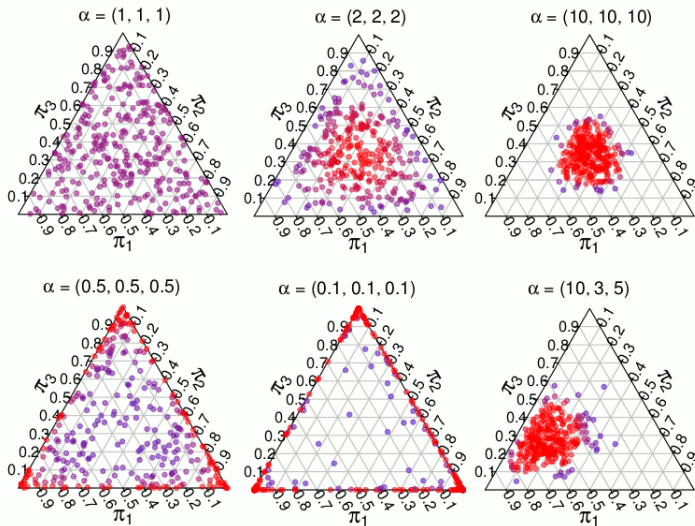
Model: $\mathbf{y} = (y_1, \dots, y_K) \sim \text{Multin}(n; \theta_1, \dots, \theta_K)$

Prior: $\theta = (\theta_1, \dots, \theta_K) \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_K)$

Posterior: $\theta | \mathbf{y} \sim \text{Dirichlet}(\alpha_1 + y_1, \dots, \alpha_K + y_K)$

DIRICHLET DISTRIBUTION

Draws from a 3-dimensional Dirichlet with different α

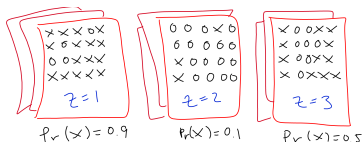


MIXTURE OF UNIGRAMS

- Let $\phi_1, \phi_2, \dots, \phi_K$ 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 $d = 1, \dots, D$:
 - Draw a **topic** z_d from a **topic distribution** $\theta = (\theta_1, \dots, \theta_K)$.
 - 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**.

GENERATING A CORPUS FROM A TOPIC MODEL

■ Assume that we have:

- A fixed vocabulary V
- D documents
- N words in each document
- K topics

1. For each **topic** ($k = 1, \dots, K$):

- a. Draw a distribution over the words $\phi_k \sim \text{Dir}(\eta, \eta, \dots, \eta)$

2. For each **document** ($d = 1, \dots, D$):

- a. Draw a vector of topic proportions $\theta_d \sim \text{Dir}(\alpha_1, \dots, \alpha_K)$

b. For each **word** ($i = 1, \dots, N$):

- i. Draw a topic assignment $z_{di} \sim \text{Multinomial}(\theta_d)$
- ii. Draw a word $w_{di} \sim \text{Multinomial}(\phi_{z_{di}})$

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

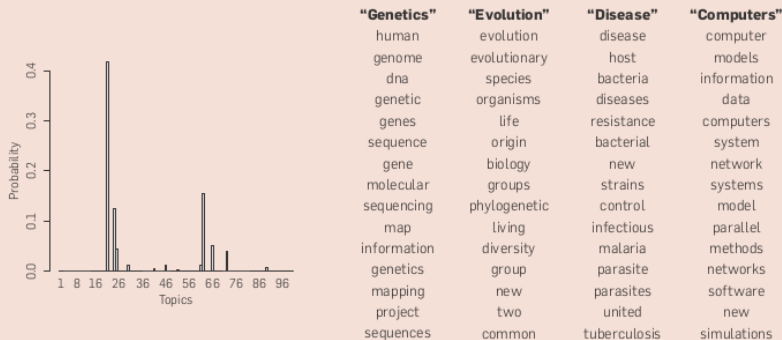
Doc 1	$\theta_1 = (0.2, 0.8)$				
	Word 1:	Topic=2	Word='gene'		
	Word 2:	Topic=2	Word='gene'		
	Word 3:	Topic=1	Word='data'		

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	$\theta_2 = (0.5, 0.5)$				
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

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.



■ 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 \mathbf{z} 's

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

- 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.

- Predicting the location of bugs in computer code. Ericsson.
- Prediction machine: **Bug report** $\Rightarrow \Pr(\text{location of bug})$
- New **multi-class** model for high-dimensional data.
- **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

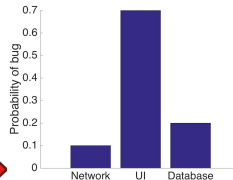
```
#-----  
def Network(networkInputs):  
    # CODE  
    # MORE CODE  
    # TOO MUCH CODE  
    return(networkOutputs)  
#-----  
def UI(UIinputs):  
    # CODE  
    # MORE CODE  
    # TOO MUCH CODE  
    return(UIOutputs)  
#-----  
def Database(DBinputs):  
    # CODE  
    # MORE CODE  
    # TOO MUCH CODE  
    return(DBOutputs)  
#-----
```

9/9

0800 Action started
1000 - stopped - action ✓
1500 low HP - MC
2000 890 - 2.13091095
2500 crash
Relays 602 - 022 field speed speed test
for relay
Relays changed
Started Cosine Tape (Sine check)
1525 Started Multi Padder Test
1545 Relay #70 Panel F
(Moth) in relay.
First actual case of bug being found.
1600 relay started.
1700 closed down.

Relay #70 Panel F
(Moth) in relay.

Relay #70 Panel F
(Moth) in relay.

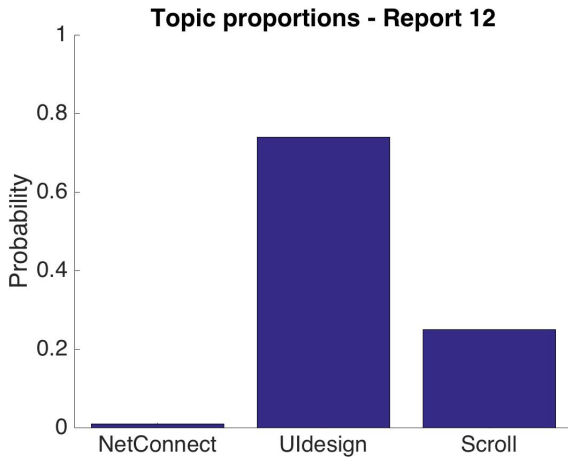


EXAMPLE DATA

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

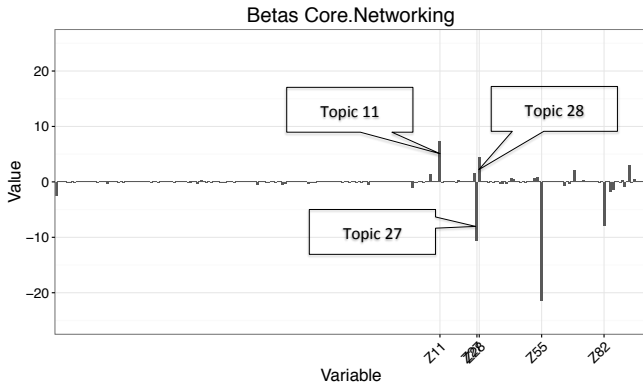
- Automatically summarize a bug report by topics.

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



IDEAL: FEW TOPICS FOR EACH COMPONENT

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



■ DOLDA - Diagonal Orthant Latent Dirichlet Allocation.

Supervised LDA. Topics are directly related to components.

■ System:

- *I am **very certain** that component UI contains the bug*
- ***because** report talks a lot about UI design 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	Dataset	DOLDA	StackingLDA
Mozilla	118	45%	39%
Eclipse	49	61%	55%
Telecom	26	71%	75%