ML for Econometricians
Text as Data - Topic Models

Mattias Villani

Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

Lecture Overview

- Textual data
- Topic models
- Efficient posterior sampling in topic models

Textual data

- **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 or even more documents in a data set.
- Lots of **pre-processing** to get the data in usable form for statistical analysis.

Text applications

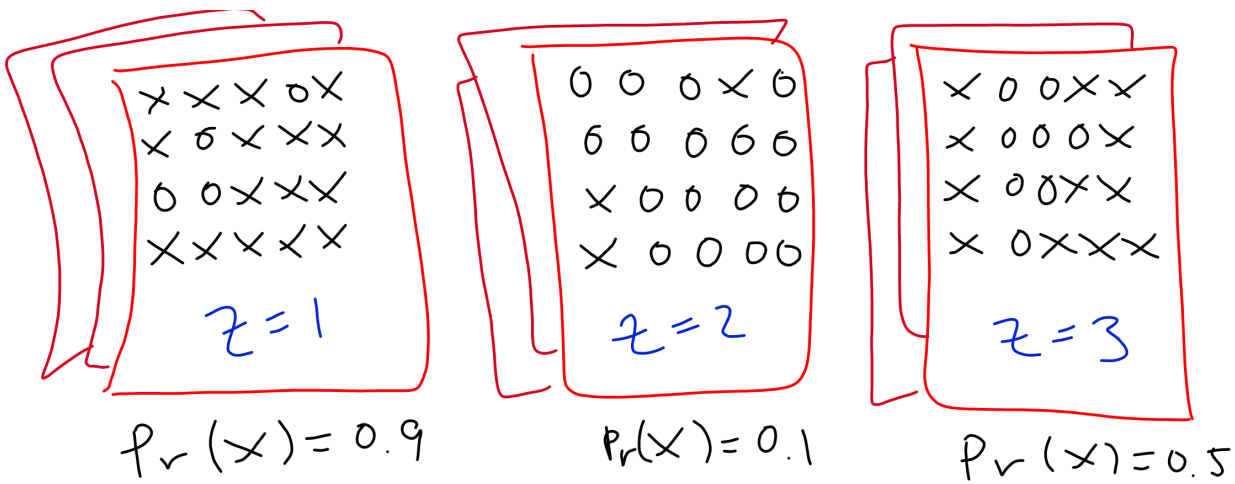
- **Language models** (predict the next word on smartphone keyboard)
- **Machine translation** (Google translate)
- **Document classification** (Shakespeare? Spam and blog filters. harmful EULA? pos/neg financial statement?)
- **Information retrival** (Google search)
- **Sentiment analysis** (positive/negative sentiment in tweets)
- **Part-of-speech tagging** (classify the grammatical category of words in a sentence)
- **Prediction models** based on text. Predicting financial turbulence from econ media. Media-Finanical markets feedback.

Topic models

- **Latent Dirichlet Allocation**.
- **Probabilistic generative** models for text documents, but also useful in other settings.
- **Very popular** model in applications and research. $>$ 12000 Google scholar citations in 11 years.
- Topic models summarizes documents into low-dimensional **topics**. **Dimension reduction**. "PCA for discrete data".
- **Many extensions** in recent years.

Mixture of unigrams

- **Mixture of unigrams**:
  1. Draw a **topic** z_d for the dth document from a topic distribution $\theta = (\theta_1, ..., \theta_K)$.
  2. Conditional on the drawn topic z_d draw words from a **word distribution** for that topic.



Topic models

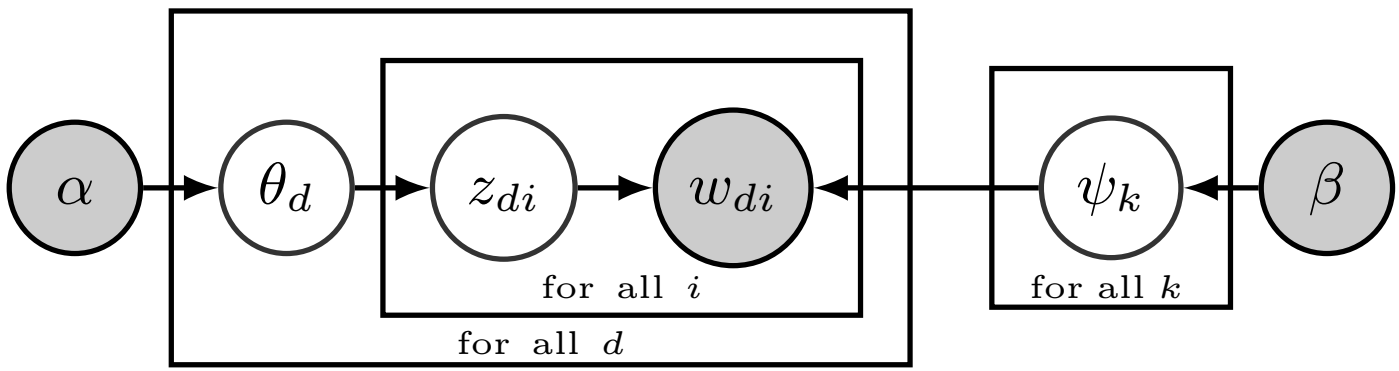
- Mixture of unigrams: each document belong to exactly one topic.
- **Topic models** are **mixed-membership models**: each document can belong to **several topics simultaneously**.
- More general than text. Individuals' behavior is determined by a mixture of influences.

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** $(k = 1, ..., K)$:
   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 $w_{di} \sim Multinomial(\phi_{z_{di}})$

GENERATING A CORPUS FROM A TOPIC MODEL



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

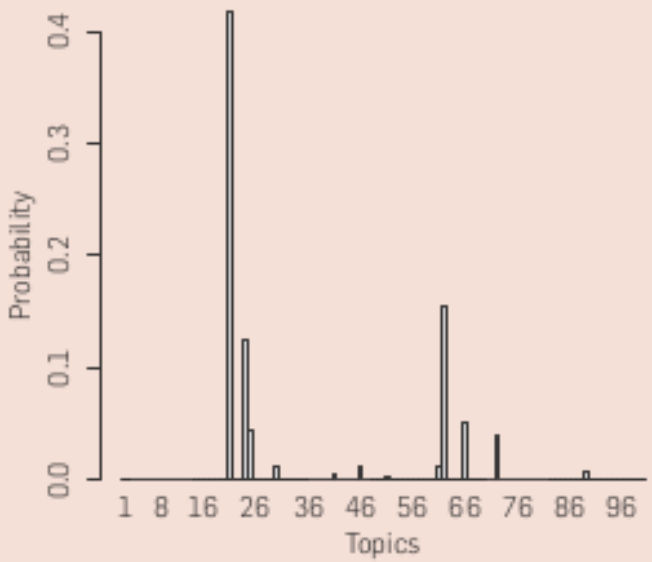
Doc 1	$\theta_1 = (0.2, 0.8)$		
	Word 1:	Topic=2	Word='gene'
	Word 2:	Topic=2	Word='gene'
	Word 3:	Topic=1	Word='data'

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.



"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

Gibbs sampler

- **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.
- The inferred topic proportions θ_d are useful summaries of document content that can be used as covariates in regression/classification.
- Example: predicting financial events from news articles.

(Naive) algorithm

```
# Initialization
Sample all topic indicators randomly
Calculate n^(w) and n^(d) # topic-word and topic-document
                          # matrices with counts


# Gibbs sampler
for each gibbs iteration do
    for each token w_i do
        remove z_i from n^(w) and n^(d)
        for each k in 1 to K do
            prob_k[k] = (n^(w)_{k,v_i} + β) / (n^(w)_{k,·} + Vβ) · (n^(d)_{k,d_i} + α)
        end for
        z_i <- draw multinomial(prob_k)
        add z_i to n^(w) and n^(d)
    end for
end for
```

Speeding up MCMC 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**.
- Lesson: with a huge number of parameters, everything matters (`log(x)` is costly ....).
- Needed: **Efficient data structures**, **sparsity**, **efficient search**, **efficient sort** etc.

Speeding up MCMC for topic models

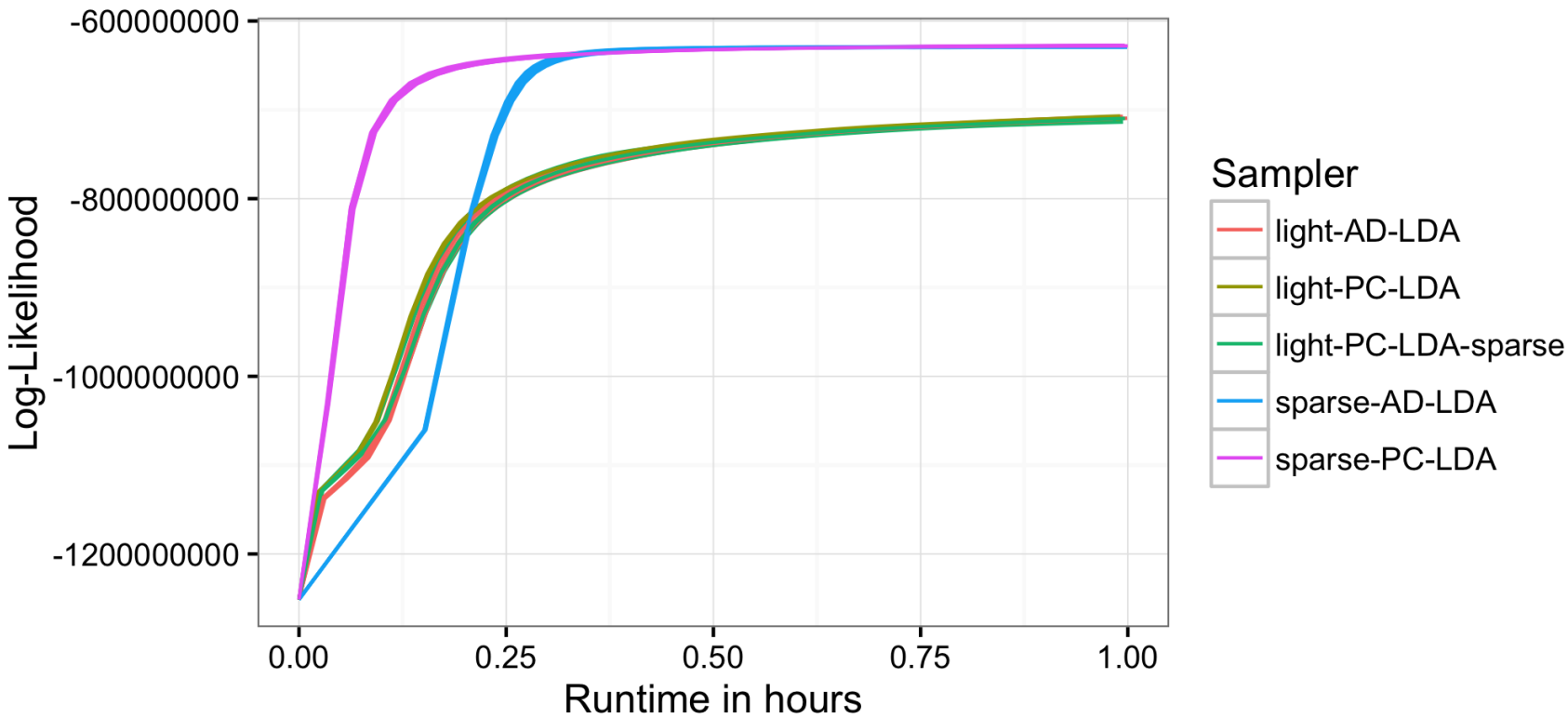
1. **Efficient sparse data structures**. Lots of binaries. Only store non-zero positions.
2. **Partially collapsing**. Integrate out Θ, but not Φ. Documents are now conditionally independent: $p(\mathbf{z}_1, ..., \mathbf{z}_D|\Phi, \mathbf{w}) = \prod_{d=1}^{D} p(\mathbf{z}_d|\Phi, \mathbf{w})$. **Document-parallel sampling** of the z's. Very little efficiency loss from not integrating out Φ [1]
3. **Exploit sparsity**. Sampling each z_{id} has complexity $O(K)$. **Walker's Alias**: fast way to repeatedly sample from dense discrete distributions that do not change much from iteration to iteration.

$$p(z_{di} = k|\cdot) = \underbrace{n_{k,d}^{-di} \frac{n_{k,w}^{-di} + \beta}{n_{k,\cdot}^{-di} + V\beta}}_{\text{Sampling in } O(K_d)} + \underbrace{\alpha \frac{n_{k,w}^{-di} + \beta}{n_{k,\cdot}^{-di} + V\beta}}_{\text{Alias sampling in amortized } O(1)}$$

MH-step needed for the full collapsed sampler. Simple Gibbs for partially collapsed.

PubMed example

- Abstracts from medical articles.
- Here we use 10% of the data. 78.5M tokens from 820K docs.

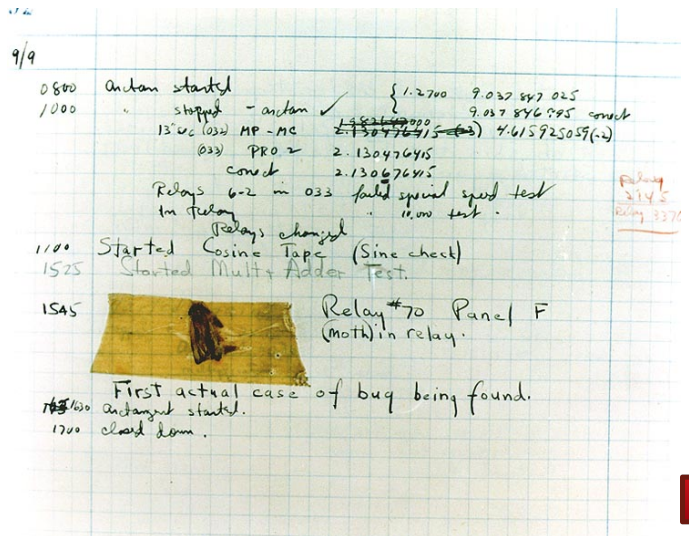


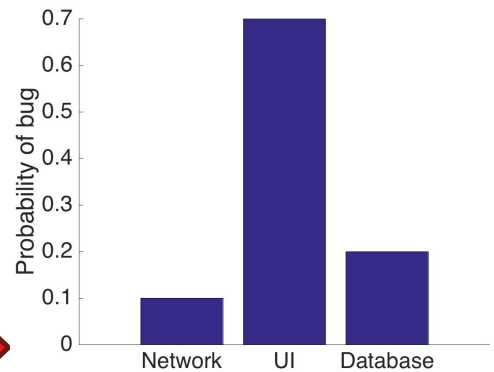
Topic model as building block in other models

- Predicting the location of bugs in computer code [2]. Ericsson.
- New **multi-class** (code component) prediction model for high-dimensional data.
- **Diagonal Orthant Multinomial Probit** [3] instead of the usual multinomial logit/probit. No reference category.
- Prediction machine: Bug report $\Rightarrow \Pr(\text{Component})$.
- **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)

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





Example data

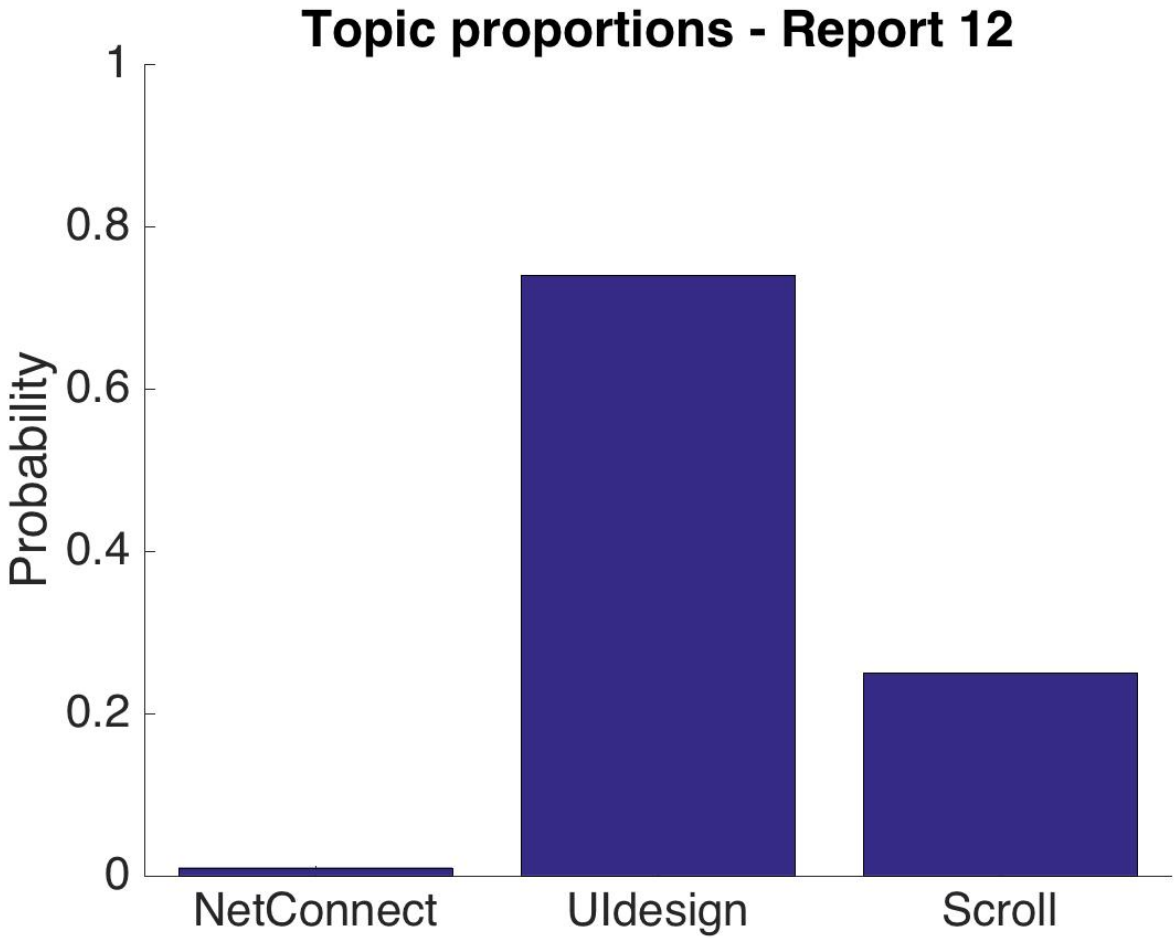
Dataset	No. Bug reports	No. components	Vocabulary size
Mozilla	15,000	118	3505
Eclipse	15,000	49	3367
Telecom	9,778	26	5286

Topics ϕ_k

- Automatically summarize a bug report by topics.

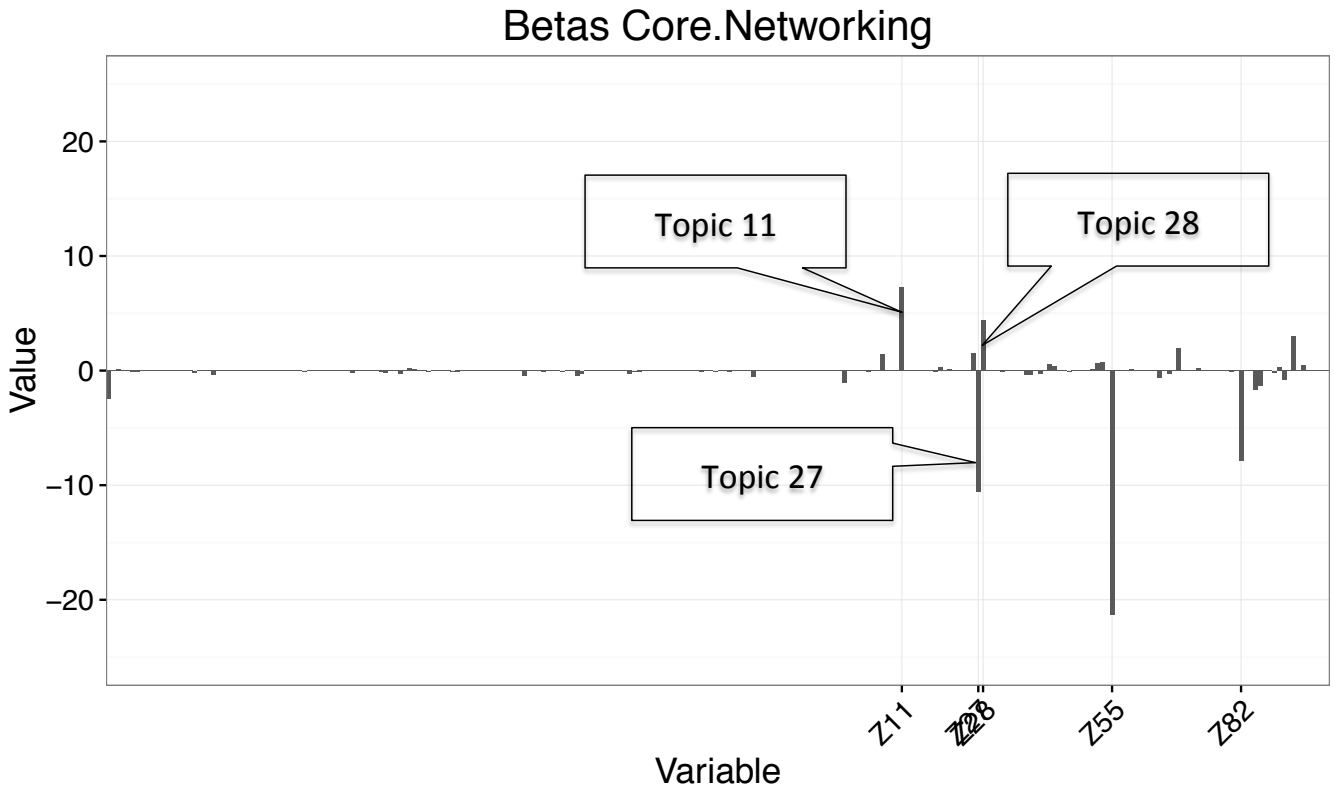
Topic	Topic label	Top 10 words in topic
11	HTTP	proxy server http network connection request connect error www host
27	Layout	div style px background color border css width height element html
28	Connection Headers	http cache accept en public localhost gmt max modified alive
55	Search	search google bar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

TOPIC PROPORTIONS θ_d



IDEAL: FEW TOPICS FOR EACH COMPONENT

- **Horseshoe regularization prior.**



Interpretable predictions

- Our model: **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 UIdesign and Scroll and very little about NetConnect.*"
- System: "*I am very uncertain where the bug is because bug report contains a jumble of topics. Please ask human.*"

No interpretation vs accuracy trade-off

Dataset	DOLDA-normal	DOLDA-horseshoe	StackingLDA	LDA-KL
Mozilla	46%	45%	39%	26%
Eclipse	61%	61%	55%	37%
Telecom	72%	71%	75%	41%


M. Magnusson, L. Jonsson, M. Villani, and D. Broman, "Sparse partially collapsed mcmc for parallel inference in topic models," *Journal of Computational and Graphical Statistics*, forthcoming.


L. Jonsson, M. Magnusson, D. Broman, K. Sandahl, M. Villani, and S. Eldh, "Automatic localization of bugs to faulty components in large scale software systems using bayesian classification," in *IEEE Conference on Software Quality and Reliability (QRS-16)*, 2016.


J. Johndrow, D. B. Dunson, and K. Lum, "Diagonal orthant multinomial probit models.," in *AISTATS*, pp. 29–38, 2013.