Machine Learning

Lecture 10 - Textual data, document classification and topic models

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University

mattiasvillani.com @matvil mattiasvillani

Lecture overview

- Text as data
- Classifying documents
- Topic models
- Word embeddings

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.
- **Feature construction** - turning text into numbers.

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

Classifying texts - feature construction

- Data: corpus of documents, each with a label
  - Journal articles under the headings: sports, politics, culture etc
  - Financial articles: positive/negative about the economy.
- Text features from documents:
  - Presence/absence of individual words, or pairs of words
  - Number of times an individual word is used
  - Lexical diversity
  - Number of web links from document (Page Rank).

Document	has('ball')	has('political_arena')	wordlen	Lex. Div.	Topic
Article1	Yes	No	4.1	5.4	Sports
Article2	No	No	6.5	13.4	Sports
⋮	⋮		⋮	⋮	⋮
ArticleN	No	Yes	7.4	11.1	News

- See `DocumentTermMatrix` function in `tm` package.

Multinomial model with Dirichlet prior

- **Topic model**. Factorization of the **Document-Term matrix**.
- **Categorical counts**: $\mathrm{y} = (y_1, ...y_C)$, where $\sum_{c=1}^{C} y_c = n$.
- y_c= number of observations in cth category. Brand choices.
- **Multinomial model**:

$$p(\mathrm{y}|\boldsymbol{\theta}) \propto \prod_{c=1}^{C} \theta_c^{y_c}, \text{ where } \sum_{c=1}^{C} \theta_c = 1.$$

- **Dirichlet prior**: $\boldsymbol{\theta} \sim \mathrm{Dirichlet}(\alpha_1, ..., \alpha_C)$

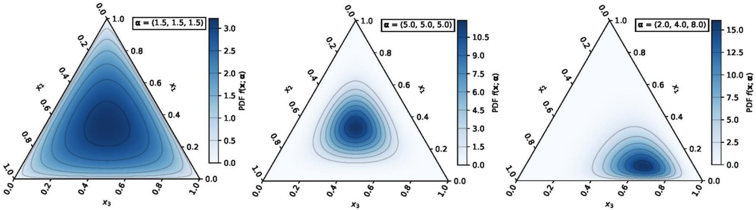
$$p(\boldsymbol{\theta}) \propto \prod_{c=1}^{C} \theta_c^{\alpha_c - 1}.$$

Dirichlet prior

$$(\theta_1, \ldots, \theta_C) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_C)$$

$$\mathbb{E}(\theta_c) = \frac{\alpha_c}{\sum_{j=1}^{C} \alpha_j}$$

$$\mathbb{V}(\theta_c) = \frac{\mathbb{E}(\theta_c)(1-\mathbb{E}(\theta_c))}{1+\sum_{j=1}^{C} \alpha_j}$$



- Uniform distribution on unit simplex: $\alpha_1 = \ldots = \alpha_K = 1$.

Multinomial model with Dirichlet prior

Multinomial data with Dirichlet prior

Model: $\mathbf{n}|\boldsymbol{\theta} \sim \text{Multinomial}(\boldsymbol{\theta})$, where

$\mathbf{n} = (n_1, \ldots, n_C)$ are counts in C categories

$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_C)$ are category probabilities.

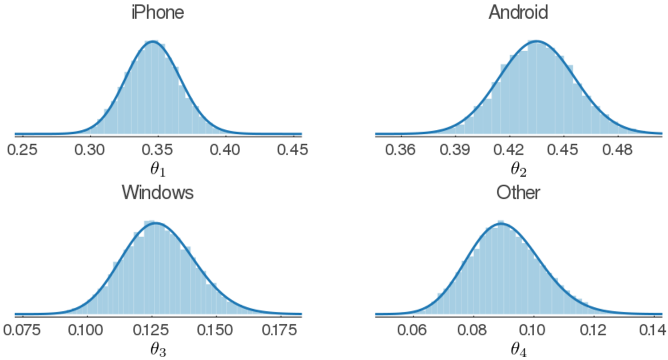
Prior: $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, for $\alpha = (\alpha_1, \ldots, \alpha_C)$

Posterior: $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{n})$

Example: smartphone market shares

- Survey among 513 smartphones owners:
  - 180 used mainly an iPhone
  - 230 used mainly an Android phone
  - 62 used mainly a Windows phone
  - 41 used mainly some other mobile phone.
- Old survey: iPhone 30%, Android 30%, Windows 20%, Other 20%.
- **Pr(Android has largest share | Data)**
- Prior: $\alpha_1 = 15, \alpha_2 = 15, \alpha_3 = 10$ and $\alpha_4 = 10$ (prior info is equivalent to a survey with only 50 respondents)
- Posterior: $(\theta_1, \theta_2, \theta_3, \theta_4)|y \sim \text{Dirichlet}(195, 245, 72, 51)$.
- **R Notebook**: Multinomial.Rmd

Example: smartphone market shares

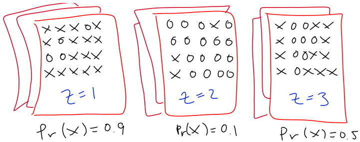


Mixture of unigrams

- Let $\phi_1, \phi_2, ..., \phi_K$ 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 $d = 1, ..., D$:
  1. Draw a **topic** z_d from a **topic distribution** $\theta = (\theta_1, ..., \theta_K)$.
  2. Given topic z_d, draw **words** from a **word distribution** ϕ_{z_d}.



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

Simulating documents 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 $z_{di} \sim \text{Categorical}(\theta_d)$
      ii. Draw a word $w_{di} \sim \text{Categorical}(\phi_{z_{di}})$

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

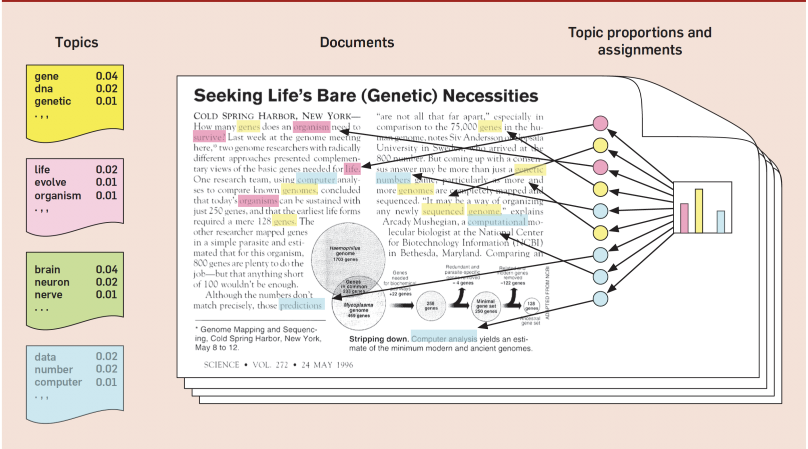
Doc 1	$\theta_1 = (0.2, 0.8)$		
	Word 1:	Topic=2	Word='gene'
	Word 2:	Topic=2	Word='gene'
	Word 3:	Topic=1	Word='data'

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

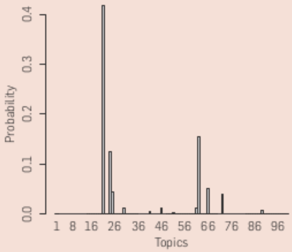
Example from Science (Blei, review paper)

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)

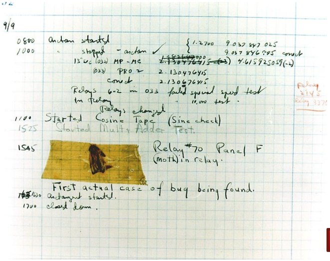
Figure 2. Real inference with LDA. We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left are the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in this article.

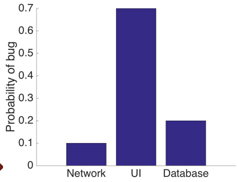


"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	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

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)

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





Three datasets

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

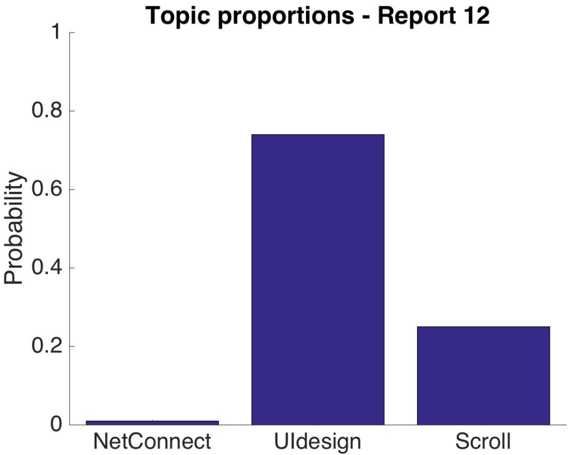
From Jonsson et al (2016). Automatic Localization of Bugs to Faulty Components in Large Scale Software Systems Using Bayesian Classification.

Topics ϕ_k

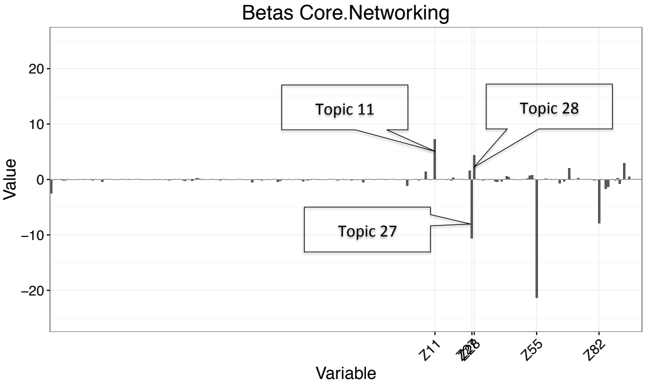
- Automatically summarize a bug report by topics.

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



Multi-class logistic regression on topics



Word embeddings

- **Represents word with dense numeric vectors** in $\mathbb{R}^p$ with typically $p \in [100, 1000]$.
- Much more compact representation than one-hot word vectors.
- **Cosine similarity** between vectors u and v

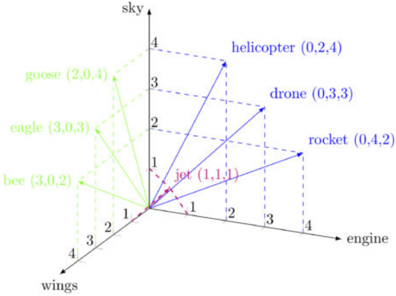
$$\cos\theta = \frac{\mathrm{u} \cdot \mathrm{v}}{\|\mathrm{u}\| \, \|\mathrm{v}\|}$$

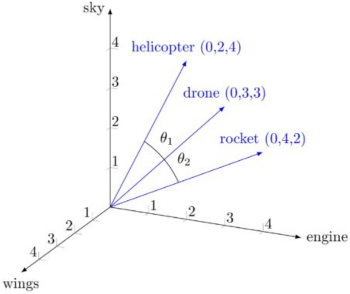
- Neighboring words ($\cos\theta$ small) have similar meanings.
- 'You shall know a word by the company it keeps!'
- Learn relations between words by surrounding words in text.
- **Vector addition** and subtraction makes sense:

$$\text{King} - \text{Man} + \text{Woman} \approx \text{Queen}$$

- **Word2Vec**: learn embeddings by shallow neural networks.

Word embeddings - toy example





From Guillaume Desagulier, "Word embeddings: the (very) basics," in Around the word, 25/04/2018, https://corpling.hypotheses.org/495.