Topic Modeling for Learning Analytics Researchers

Vitomir Kovanović
School of Informatics
University of Edinburgh
Edinburgh, United Kingdom
v.kovanovic@ed.ac.uk

Srećko Joksimović
School of Interactive Arts and Technology
Simon Fraser University
Vancouver, Canada
sjoksimo@sfu.ca

Dragan Gašević
Schools of Education and Informatics
University of Edinburgh
Edinburgh, United Kingdom
dgasevic@acm.org

March 17, 2015

Marist College,
Poughkeepsie, NY, USA

Welcome everyone!

Workshop outline

08:30 Tutorial Opening (15 min)

- Opening of the tutorial: (15 min): Introduction of the presenters, workshop objectives, obtaining datasets for the workshop and resolving any technical issues.

08:45 Introduction (60 min**)**

- Introduction to topic modeling (30 min): Overview of the topic modeling and learning analytics problems it can address.
- Brief introduction to R and RStudio (30 min): Gentle introduction to R & RStudio for the purposes of the tutorial.

09: 45 Coffee break (15 min)

10:00 Topic Modeling Pt I (60 min)

- LDA topic modeling (50 min): Introduction to the topic modeling algorithms(LSA, pLSA, LDA) and `topicmodels` R programming library.
- Real-world example of LDA topic modeling (10 min): LDA analysis of the large MOOC news corpora.

11:00 Coffee break (15 min)

11:15 Topic Modeling Pt II (60 min)

- Collaborative topic modeling (40 min): Step-by-step LDA topic modeling on the real-world dataset.
- Advanced topic modeling techniques (20 min): A brief overview of more advanced topic modeling techniques such as hierarchical, dynamic and correlated topic modeling.

12:15 Tutorial Closing & Discussion (15 min)

- Closing discussion and final remarks (15 min): Overview and discussion of what is learned in the tutorial and how to proceed further with the adoption of topic modeling in own research.

Download software

Download R and R studio:

MS Windows: `bit.do/tm_win`

OS X: `bit.do/tm_mac`

Download example code:

Both platforms: `bit.do/tm_code`

Presenters

Vitomir Kovanović

- Background in software engineering.
  - BSc & MSc in Informatics & Computer Science.
  - From 2008 to 2011 worked as a software developer on a large distributed system for sports betting.
- Currently third year PhD student at the School of Informatics, University of Edinburgh.
  - PhD research focuses on development of learning analytics models based on text mining, social network analysis and trace-data clustering in the context of communities of inquiry.
  - From Sep. 2011 to Dec. 2014 in a PhD program at Simon Fraser University.
- Member of SoLAR, active in learning analytics community.
- In 2014, with Christopher Brooks, Zachary Pardos, and Srećko Joksimović run tutorial *“Introduction to Data Mining for Educational Researchers”* at LASI at Harvard University.

Presenters

Srećko Joksimović

- Background in software engineering.
  - BSc & MSc in Informatics & Computer Science.

- Currently completing the second year of the PhD program at School of Interactive Arts and Technology, Simon Fraser University.
  - Research interests center around the analysis of teaching and learning in networked learning environments.

- Member of SoLAR, active in learning analytics community.
  - SoLAR Executive Student Member

- In 2014, with Christopher Brooks, Zachary Pardos, and Vitomir Kovanović run tutorial *“Introduction to Data Mining for Educational Researchers”* at LASI at Harvard University.

DISCLAIMER

This is our first time to organize this tutorial, so we are both excited and frightened!

Audience introduction

Very quick (and casual) run around the room:

- Introduce yourself to other participants

- *What is the most important thing you want to get out of this tutorial?*

The goals of the tutorial

- What is topic modelling?
- Why you might want to use it?
- What types of topic modeling there are? (e.g., LSA, pLSA, LDA)
- Advanced topic modeling techniques

- Latent Dirichlet Allocation (LDA)
  - How to conduct LDA analysis using R
  - How to interpret topic modeling results
  - Several real-world examples
  - Collaborative hands-on LDA analysis
- Very hands-on and practical (i.e., not too much math)

Ask questions anytime!

Technical stuff

Download R and R studio:

MS Windows: http://bit.do/lda_win

OS X: http://bit.do/lda_mac

Download example code:

Both platforms: http://bit.do/lda_code

Installing R/R studio

Introduction to Topic Modeling

What is it, and why should we care?

Goal of Topic Modeling

Fast and easy birds eye view of the large datasets.

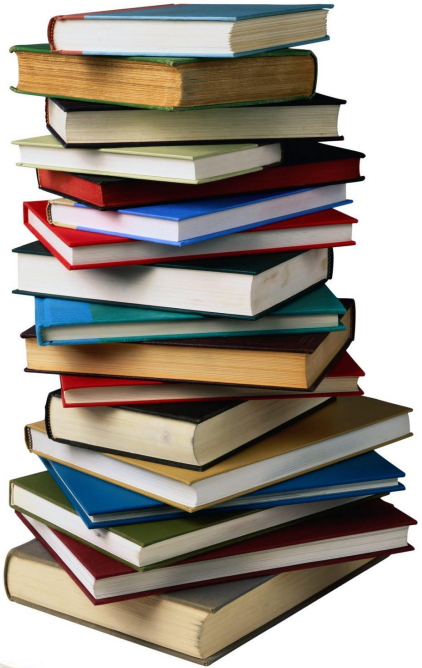
What is the data **about**?

What are the key **themes**?

Very powerful when coupled with different covariates (e.g., year of publication or author):

Longitudinal analysis: How the key themes change over time?

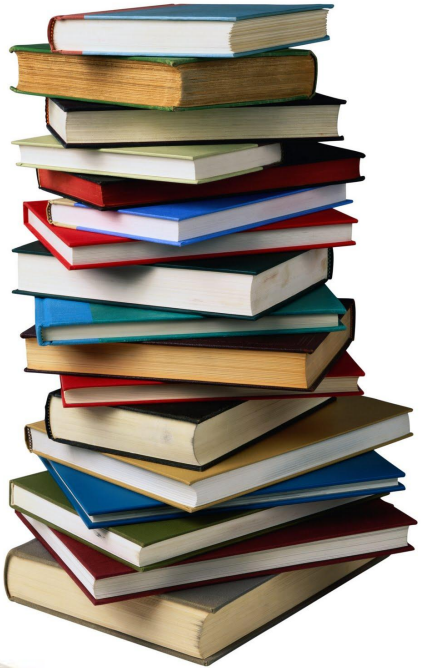
Focus of discussion: Who is focussing on one topic, who talks about variety of topics?



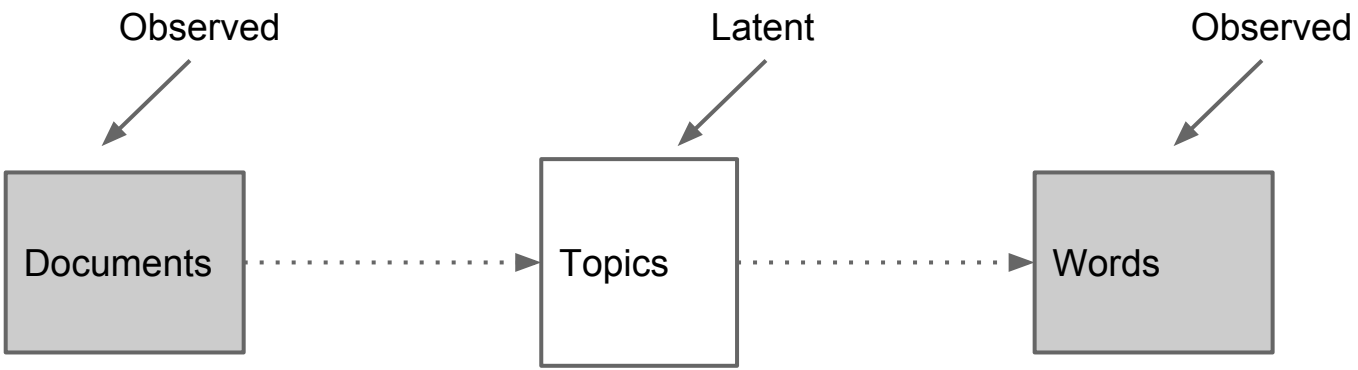
Goal of Topic Modeling

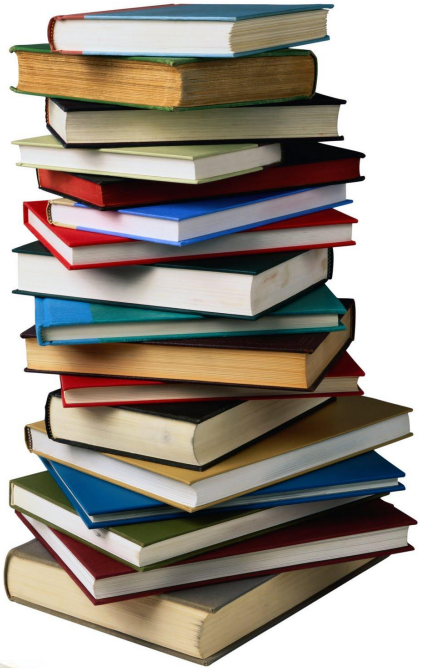
Meeks and Weingart (2012):

“It [Topic modeling] is distant reading in the most pure sense: focused on corpora and not individual texts, treating the works themselves as unceremonious “buckets of words,” and providing seductive but obscure results in the forms of easily interpreted (and manipulated) “topics.”” (p. 2)



Goal of Topic Modeling





Documents are about several topics at the same time.
Topics are associated with different words.

Topics in the documents are expressed through the words that are used.

Goal of Topic Modeling

Topics

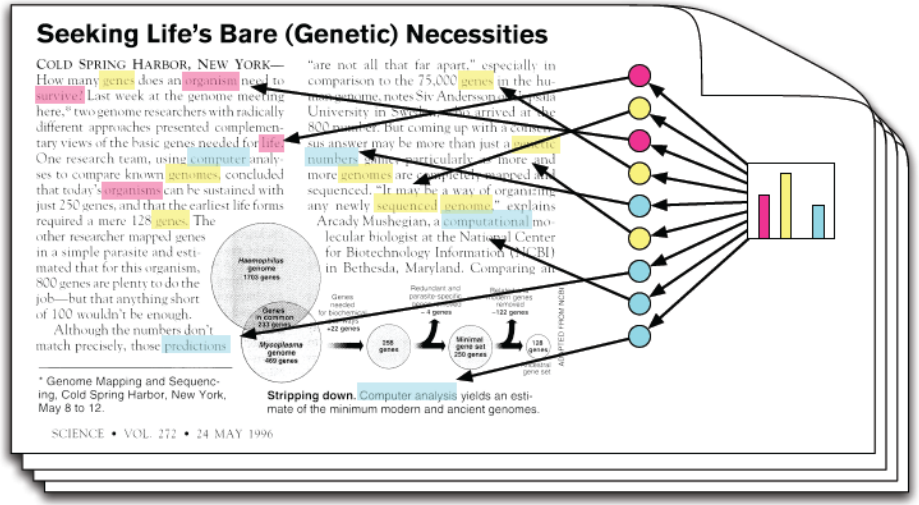
gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents



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.

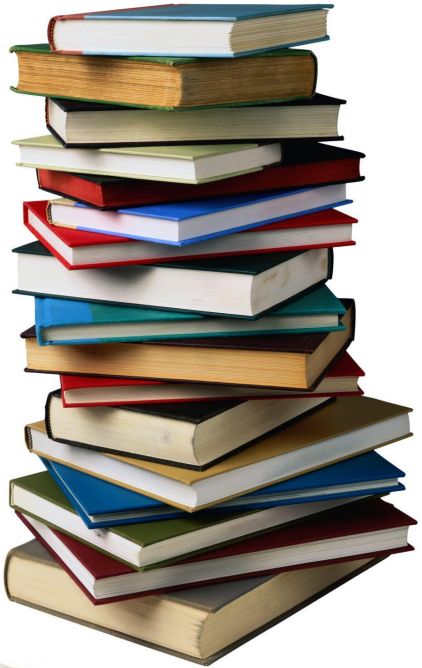
Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 round-off. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completed, mapped, and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an-

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

Topic proportions and assignments



Real world example:

The New York Times

LDA analysis of 1.8M New York Times articles:

music band songs rock album jazz pop song singer night	book life novel story books man stories love children family	art museum show exhibition artist artists paintings painting century works	game knicks nets points team season play games night coach	show film television movie series says life man character know
theater play production show stage street broadway director musical directed	clinton bush campaign gore political republican dole presidential senator house	stock market percent fund investors funds companies stocks investment trading	restaurant sauce menu food dishes street dining dinner chicken served	budget tax governor county mayor billion taxes plan legislature fiscal

Extensively used in Humanities, Social Sciences, History

Topic Modeling on Historical Newspapers

Tze-I Yang
Dept. of Comp. Sci. & Eng.
University of North Texas
tze-iyang@my.unt.edu

Andrew J. Torget
Dept. of History
University of North Texas
andrew.torget@unt.edu

Rada Mihalcea
Dept. of Comp. Sci. & Eng.
University of North Texas
rada@cs.unt.edu

Abstract

In this paper, we explore the task of automatic text processing applied to collections of historical newspapers, with the aim of assisting historical research. In particular, in this first stage of our project, we experiment with the use of topical models as a means to identify potential issues of interest for historians.

ical resources. The reason for this paradox is quite simple: the sheer volume and breadth of information available in historical newspapers has, ironically, made it extremely difficult for historians to go through them page-by-page for a given research project. A historian, for example, might need to wade through tens of thousands of newspaper pages in order to answer a single research question (with no guarantee of stumbling onto the necessary infor-

http://mappingtexts.stanford.edu/whitepaper/Topic_Modeling_History.pdf

Extensively used in Humanities, Social Sciences, History

Topic Modeling on Historical Newspapers

Tze-I Yang
Dept. of Comp. Sci. & Eng.
University of North Texas
tze-iyang@my.unt.edu

Andrew J. Torget
Dept. of History
University of North Texas
andrew.torget@unt.edu

Rada Mihalcea
Dept. of Comp. Sci. & Eng.
University of North Texas
rada@cs.unt.edu

Abstract

In this paper, we explore the task of automatic text processing applied to collections of historical newspapers, with the aim of assisting historical research. In particular, in this first stage of our project, we experiment with the use of topical models as a means to identify potential issues of interest for historians.

ical resources. The reason for this paradox is quite simple: the sheer volume and breadth of information available in historical newspapers has, ironically, made it extremely difficult for historians to go through them page-by-page for a given research project. A historian, for example, might need to wade through tens of thousands of newspaper pages in order to answer a single research question (with no guarantee of stumbling onto the necessary infor-

http://mappingtexts.stanford.edu/whitepaper/Topic_Modeling_History.pdf

Analysis of news articles published in Texas between 1865 and 1901.

Results:

Topics	Explanation
black* price* worth* white* goods* yard* silk* made* lot* week ladies wool* inch* ladles* sale* prices* pair* suits* fine*	Reflects discussion of the market and sales of goods, with some words that relate to cotton and others that reflect other goods being sold alongside cotton (such as wool).
state* people* states* bill* law* made united* party* men* country* government* county* public* president* money* committee* general* great question*	Political language associated with the political debates that dominated much of newspaper content during this era. The association of the topic "money" is particularly telling, as economic and fiscal policy were particularly important discussion during the era.
clio worth mid city alie fort lino law lour lug thou hut fur court dally county anil tort iron	Noise and words with no clear association with one another.
tin inn mid tint mill* till oil* ills hit hint lull win hut ilia til ion lot lii foi	Mostly noise, with a few words associated with cotton milling and cotton seed.
texas* street* address* good wanted houston* office* work city* sale main* house* apply man county* avenue* room* rooms* land*	These topics appear to reflect geography. The inclusion of Houston may either reflect the city's importance as a cotton market or (more likely) the large number of newspapers from the collection that came from Houston.
worth* city* fort* texas* county* gazette tex* company* dallas* miss special yesterday night time john state made today louis*	These topics appear to reflect geography in north Texas, likely in relation to Fort Worth and Dallas (which appear as topics) and probably as a reflection that a large portion of the corpus of the collection came from the Dallas/Ft. Worth area.
houston* texas* today city* company post* hero* general* night morning york men* john held war* april* left san* meeting	These topics appear to an unlikely subject identified by the modeling. The words `Houston`, `hero`, `general`, `april` and `san` (perhaps part of `San Jacinto`) all fit together for a historian to suggest a sustained discussion in the newspapers of the April 1836 Battle of San Jacinto, when General Sam Houston defeated Santa Anna of Mexico in the Texas Revolution. This is entirely unexpected, but the topics appear to fit together closely. That this would rank so highly within all topics is, too, a surprise. (Most historians, for example, have argued that few Texans spent much time memorializing such events until after 1901. This would be quite a discovery if they were talking about it in such detail before 1901.)
man time great good men years life world long made people make young water woman back found women work	Not sure what the connections are here, although the topics clearly all fit together in discussion of the lives of women and men.
market* cotton* york* good* steady* closed* prices* corn* texas* wheat* fair* stock* choice* year* lower* receipts* ton* crop* higher*	All these topics reflect market-driven language related to the buying and selling cotton and, to a much smaller extent, other crops such as corn.
tube tie alie time thaw art ton ion aid ant ore end hat ire aad lour thee con til	Noise with no clear connections.

Table 6: 10 topic groups found for the 1865-1901 main set. Asterisks denote meaningful topic terms.

Results:

Topics	Explanation
black* price* worth* white* goods* yard* silk* made* lot* week ladies wool* inch* ladles* sale* prices* pair* suits* fine*	Reflects discussion of the market and sales of goods, with some words that relate to cotton and others that reflect other goods being sold alongside cotton (such as wool).
state* people* states* bill* law* made united* party* men* country* government* county* public* president* money* committee* general* great question*	Political language associated with the political debates that dominated much of newspaper content during this era. The association of the topic "money" is particularly telling, as economic and fiscal policy were particularly important discussion during the era.
clio worth mid city alie fort lino law lour lug thou hut fur court dally county anil tort iron	Noise and words with no clear association with one another.
tin inn mid tint mill* till oil* ills hit hint lull win hut ilia til ion lot lii foi	Mostly noise, with a few words associated with cotton milling and cotton seed.
texas* street* address* good wanted houston* office* work city* sale main* house* apply man county* avenue* room* rooms* land*	These topics appear to reflect geography. The inclusion of Houston may either reflect the city's importance as a cotton market or (more likely) the large number of newspapers from the collection that came from Houston.
worth* city* fort* texas* county* gazette tex* company* dallas* miss special yesterday night time john state made today louis*	These topics appear to reflect geography in north Texas, likely in relation to Fort Worth and Dallas (which appear as topics) and probably as a reflection that a large portion of the corpus of the collection came from the Dallas/Ft. Worth area.
houston* texas* today city* company post* hero*	These topics appear to an unlikely subject identified by

Results:

Period	Topics	Explanation
1865-1901	texas* city* worth* houston* good* county* fort* state* man* time* made* street* men* work* york today company great people	These keywords appear to be related to three things: (1) geography (reflected in both specific places like `Houston` and `Fort Worth` and more general places like `county`, `street`, and `city`), (2) discussions of people (`men` and `man`) and (3) time (`time` and `today`).
1892	texas* worth* gazette* city* tex* fort* county* state* good* march* man* special* made* people* time* york men days feb	As with the 1865-1901 set, these keywords also appear to be related to three things: (1) geography, (2) discussions of people and (3) time.
1893	worth* texas* tin* city* tube* clio* time* alie* man* good* fort* work* made street year men county state tex	As with the 1865-1901 set, these keywords also appear to be related to three things: (1) geography, (2) discussions of people and (3) time.
1929-1930	tin* texas* today* county* year* school* good* time* home* city* oil* man* men* made* work* phone night week sunday	As with the 1865-1901 set, these keywords also appear to be related to three things: (1) geography, (2) discussions of people and (3) time. The time discussion here appears to be heightened, and the appearance of economic issues for Texas (`oil`) makes sense in the context of the onset of the Great Depression in 1929-30.

Table 7: Main topics for years of interest for the main set

Social Sciences: Stanford Topic Modeling Toolbox

Also extensively used in social sciences

Topic Modeling for the Social Sciences

Daniel Ramage, Evan Rosen, Jason Chuang, Christopher D. Manning and Daniel A. McFarland*
Computer Science Department and School of Education*
Stanford University
Stanford, CA 94305
{dramage,emrosen,jcchuang,manning,dmcfarla}@stanford.edu

Abstract

As textual datasets grow in size and scope, social scientists need better tools to help make sense of that data. Despite the natural applicability of topic modeling to many such problems, word counts and tag clouds are often used as the primary means of gleaning information from textual data. We characterize two barriers to adoption encountered during a collaboration between the Stanford NLP group and social scientists in the school of education: accessibility and trust. Accessibility refers to the technical barriers that make text processing and topic modeling difficult. Trust comes when practitioners can explore and validate a model being used to discover or support a hypothesis. We introduce recent work aimed at solving these challenges including the Stanford Topic Modeling Toolbox software.

http://vis.stanford.edu/files/2009-TopicModels-NIPS-Workshop.pd

(Digital) Humanities

Special issue of Journal of Digital Humanities on topic modeling

http://journalofdigitalhumanities.org/2-1/

(Digital) Humanities

ELIJAH MEEKS AND SCOTT WEINGART

The Digital Humanities Contribution to Topic Modeling

Topic modeling could stand in as a synecdoche of digital humanities. It is distant reading in the most pure sense: focused on corpora and not individual texts, treating the works themselves as unceremonious “buckets of words," and providing seductive but obscure results in the forms of easily interpreted (and manipulated) “topics.” In its most commonly used tool, it runs in the command line. To achieve its results, it leverages occult statistical methods like “dirichlet priors” and “bayesian models.” Were a critic of digital humanities to dream up the worst stereotype of the field, he or she would likely create something very much like this, and then name a popular implementation of it after a hammer.

Topic Modeling + Learning Analytics

We have a vast amount of data in learning analytics that we want to analyze:

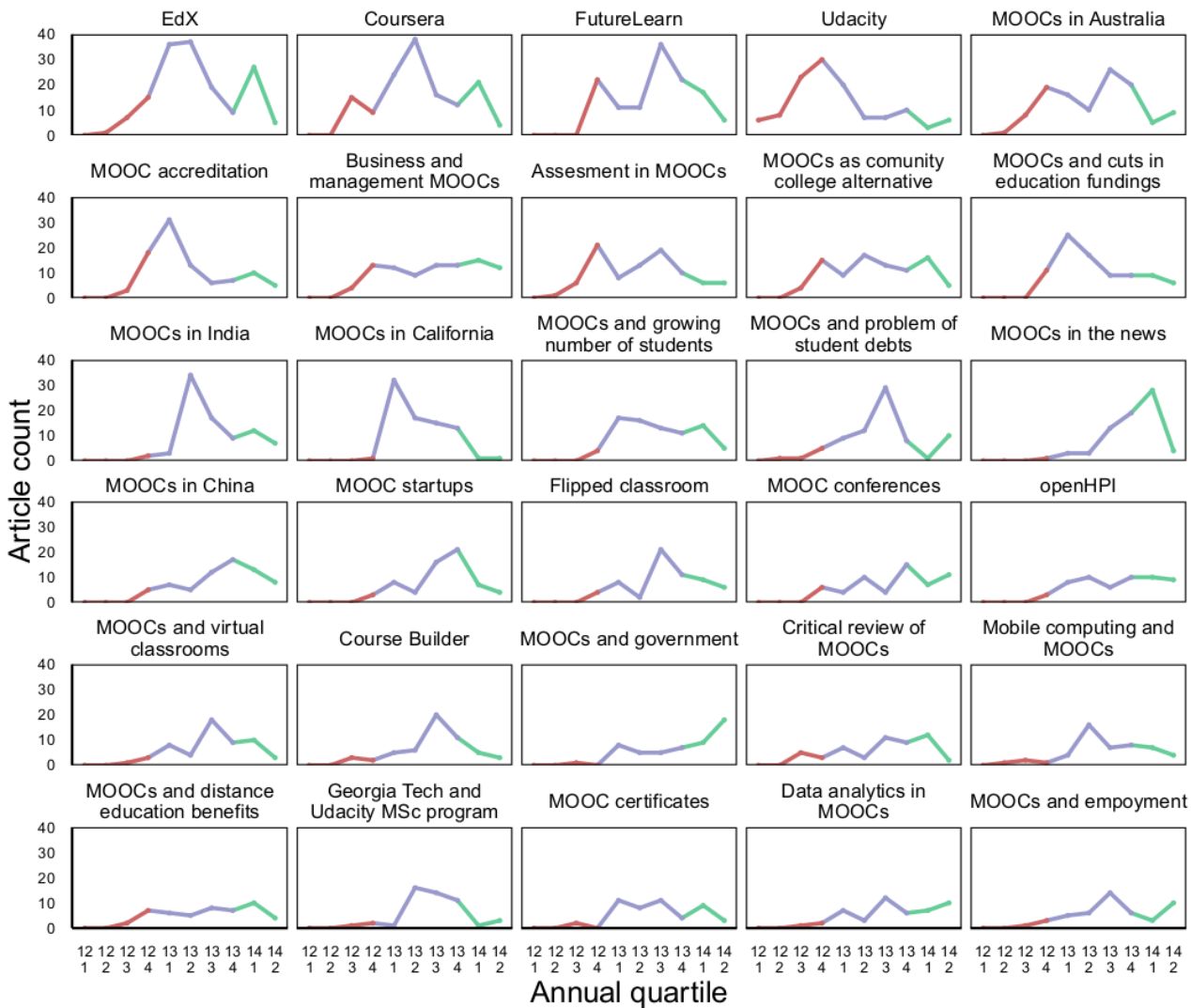
- MOOC/LMS online discussions,
- Student (micro)blog postings,
- Student essays/written exams,

- Research reports that can be used to inform about the field (semi-automated meta-analyses),
- News reports about LA/MOOCs/Online learning informing about public,
- .....

MOOC News Analysis

Kovanovic, V., Joksimovic, S., Gasevic, D., Siemens, G., & Hatala, M. (2014). What public media reveals about MOOCs? British Journal of Educational Technology (to appear)

- 4,000 News reports from all over the world
- Run LDA topic modeling to discover important topics
- Looked at how topics changed over time



Analysis of MOOC discussions

JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Do not touch this during review process. (xxxx). Paper title here. *Journal of Learning Analytics*, *xx* (x), xx–xx.

Computer-Assisted Reading and Discovery for Student Generated Text in Massive Open Online Courses

Justin Reich, Dustin Tingley, Jetson Leder-Luis, Margaret E. Roberts, Brandon M. Stewart
Harvard University, U.S.A.
justin_reich@harvard.edu

Dealing with the vast quantities of text that students generate in a Massive Open Online Course (MOOC) is a daunting challenge. Computational tools are needed to help instructional teams uncover themes and patterns as MOOC students write in forums, assignments, and surveys. This paper introduces to the learning analytics community the Structural Topic Model, an approach to language processing that can (1) find syntactic patterns with semantic meaning in unstructured text, (2) identify variation in those patterns across covariates, and (3) uncover archetypal texts that exemplify the documents within a topical pattern. We show examples of computationally-aided discovery and reading in three MOOC settings: mapping students' self-reported motivations, identifying themes in discussion forums, and uncovering patterns of feedback in course evaluations.

Keywords: Massive Open Online Courses, topic modeling, text analysis, computer-assisted reading

http://scholar.harvard.edu/files/dtingley/files/educationwriteup.pdf

Types of topic modeling

Vector-based techniques:

- Latent Semantic Analysis (LSA) (a.k.a Latent Semantic Indexing - LSI)

Probabilistic techniques

- All cool kids use bayesian statistics
- Better handles synonymy and polysemy
- Probabilistic Latent Semantic Analysis (pLSA)
- Latent Dirichlet Allocation (LDA)
    - Whole bunch of LDA extensions

Highly recommended introduction to all three algorithms:

Crain, S. P., Zhou, K., Yang, S.-H., & Zha, H. (2012). *Dimensionality Reduction and Topic Modeling: From Latent Semantic Indexing to Latent Dirichlet Allocation and Beyond*. In C. C. Aggarwal & C. Zhai (Eds.), Mining Text Data (pp. 129–161). Springer.

Software for topic modeling

Most require solid technical background:

- Python libraries
  a. `Gensim` - Topic Modelling for Humans
  b. `LDA` Python library
- C/C++ libraries
  a. `lda-c`
  b. `hlda`
  c. `ctm-c`
  d. `hdp`
- R packages
  a. `lsa` package
  b. `lda` package
  c. `topicmodels` package
  d. `stm` package
- Java libraries
  a. `S-Space` Package
  b. `MALLET`

Brief introduction to R

Srećko Joksimović

What is R?

http://www.r-project.org/

Features of R

- The core software is quite lean
  - functionality is divided into modular packages,

- Graphics capabilities very sophisticated,

- Useful for interactive work,

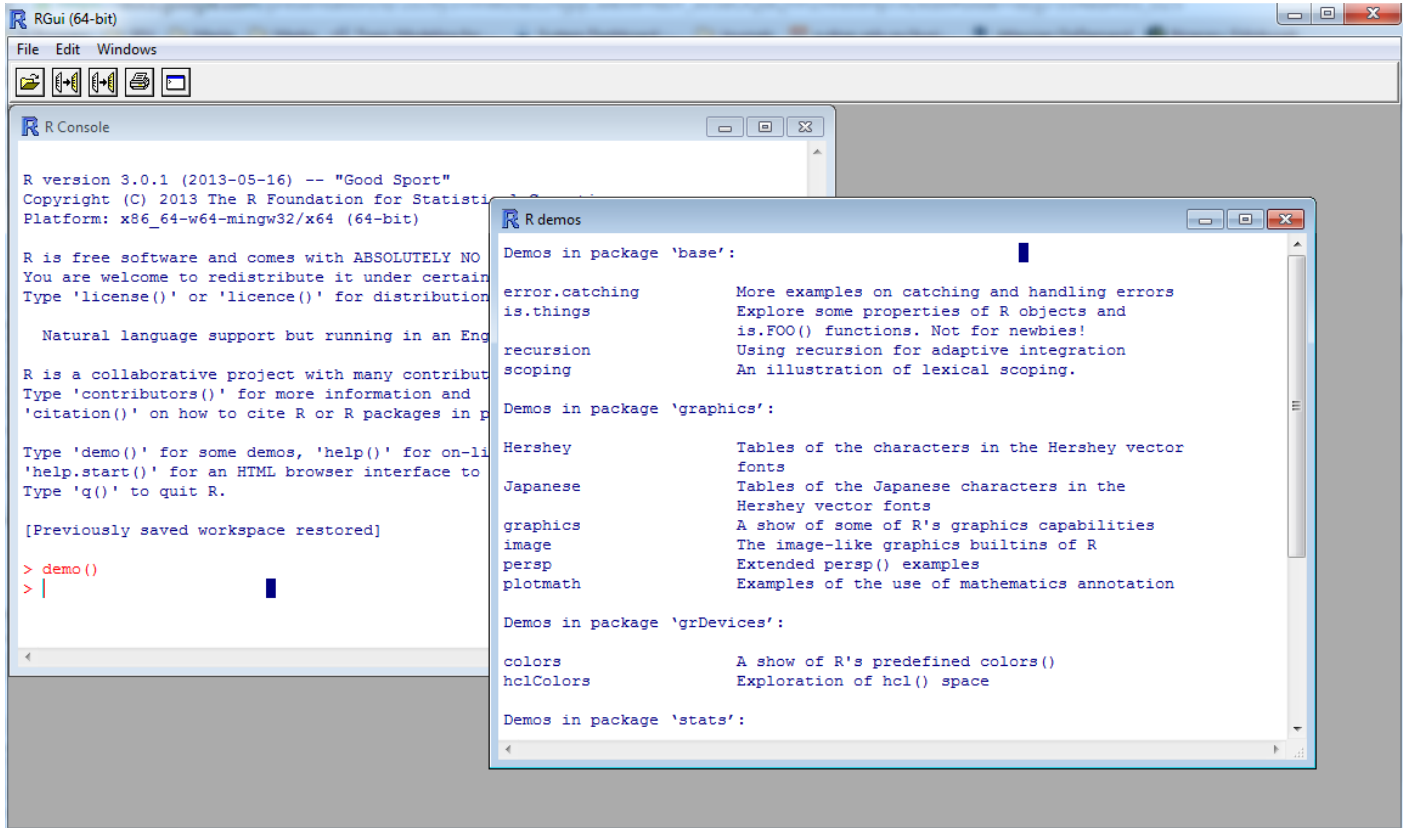
- Contains a powerful programming language for developing new tools ,

- Very active and vibrant user community
  - R-help and R-devel mailing lists and Stack Overflow, CrossValidated.

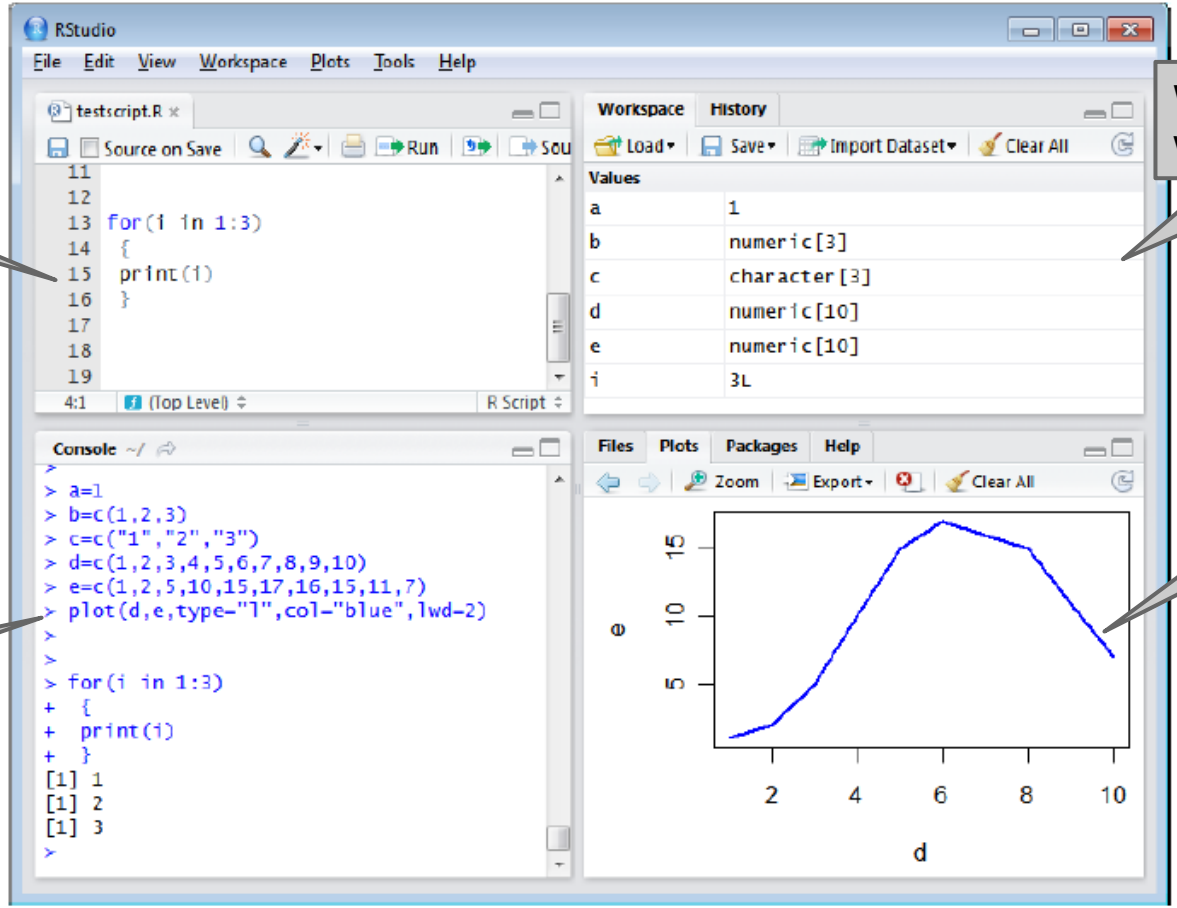
RStudio

- RStudio IDE is a powerful and productive user interface for R
- Free, open source, and works on Windows, Mac, and Linux.
- http://www.rstudio.com/

RGui is not RStudio!



RStudio layout



Editor window

Workspace/History window

Files/Plots/ Packages/Help window

Console window

Working directory and libraries

- Working directory
  - getwd()
  - setwd("directoryname")
  - Using Files window

- Libraries
  - Packages are collections of R functions
  - library() # see all packages installed
  - search() # see packages currently loaded
  - install.library("package") # install package
  - library("package") # load package

Scripts

- R script is just a plain text file with R commands.

- You can prepare a script in any text editor (e.g., vim, Notepad, TextEdit, WordPad)

- Creating scripts with R studio

- Running a simple script

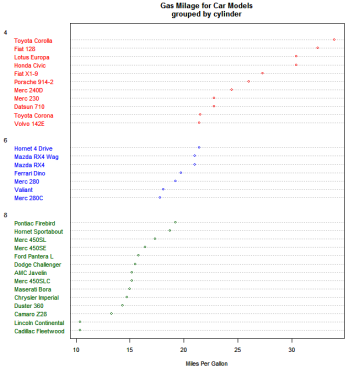
```
> source("foo.R")
```

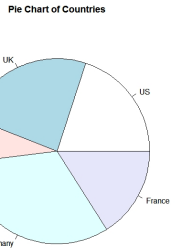
Graphics

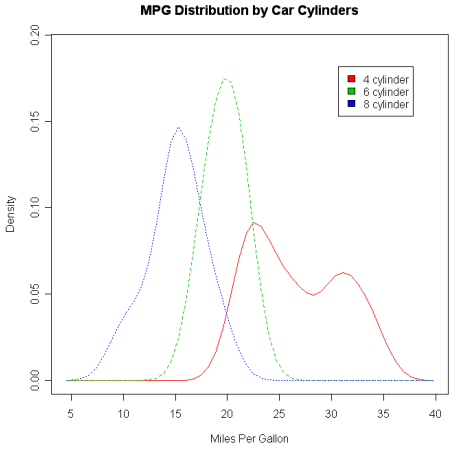
- Basic graph types:
  - density plots, dot plots, bar charts, line charts, pie charts, boxplots, scatter plots
- Advanced graphics
  - customization - customize a graph's axes, add reference lines, text annotations and a legend,
  - statistically complex graphs - e.g., plots conditioned on one or more variables, probability plots, mosaic plots, and correlograms
- Packages
  - lattice, ggplot2, graphics

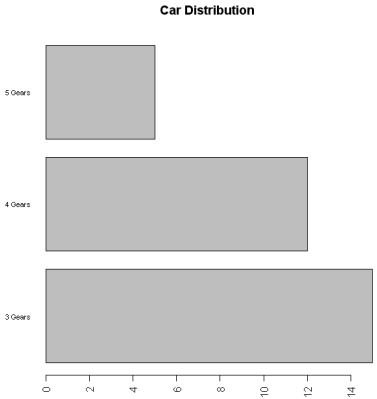
Graphics

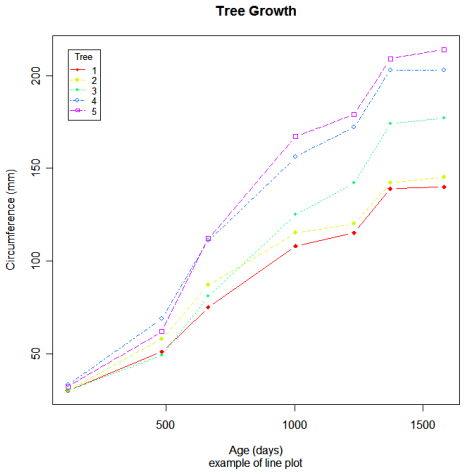
- Basic graph types

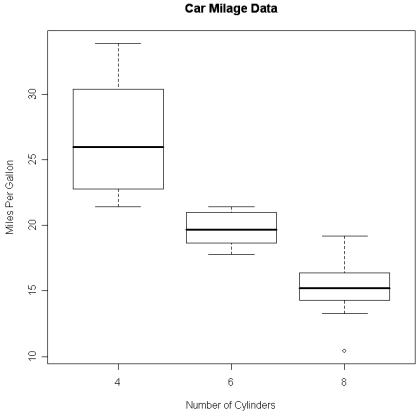


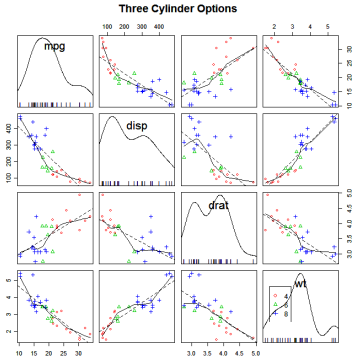












Basic commands

comments:

everything after # is ignored

variables:

Variable names are case sensitive. '.' is a valid character!

```
variable.name <- value # this is some comment
input.directory <- "/home/srecko/cck12_blogs"
number.of.topics <- 5
TF.threshold <- 0.95
stem.words <- FALSE
stem.words <- F
```

Basic commands

vectors (arrays):

`c()` is vector concatenation function

```
> names <- c("Huey", "Dewey", "Louie")
 > names
 [1] "Huey"  "Dewey" "Louie"
```

shorthands for numerical vectors: `':'` and `seq`

```
> first.twenty <- 1:20
> first.twenty
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> every.fifth <- seq(from=0, to=50, by=5)
> every.fifth
 [1]  0  5 10 15 20 25 30 35 40 45 50
```

Basic commands

vectors (arrays): all elements of the same type

accessing elements with [and]

indexing starts at 1

```
> names[1]
[1] "Huey"
> names[2]
[1] "Dewey"
> names[4] <- "Donald Duck"
> names
[1] "Huey"        "Dewey"       "Louie"       "Donald Duck"
```

Basic commands

lists: elements of the different type

created with `list()` function

accessing elements with $name

```
savings <- list(amount=10000, currency="USD")
> savings
$amount
[1] 10000
$currency
[1] "USD"
> savings$amount
[1] 10000
```

Basic commands

strings:

single or double quotes valid.

`paste` and `paste0` for string concatenation.

```
> paste("Three nephews", names[1], names[2], names[3])
[1] "Three nephews Huey Dewey Louie"
> paste0("Three nephews", names[1], names[2], names[3])
[1] "Three nephewsHueyDeweyLouie"
```

R can operate an all elements in a vector. Arguments are paired

```
> paste(names, names)
[1] "Huey Huey"             "Dewey Dewey"           "Louie Louie"
```

Everyting is a vector. Shorter vectors are cycled

```
paste("Hello", names)
[1] "Hello Huey"       "Hello Dewey"      "Hello Louie"
```

same as

```
> paste(c("Hello", "Hello", "Hello"), names)
[1] "Hello Huey"       "Hello Dewey"      "Hello Louie"
```

Data frames

table-like structure

```
> nephew.data <- data.frame(age=c(3, 5, 7),
                            name=c("Huey",  "Dewey", "Louie"),
                            favorite.drink=c("coke", "orange juice", "bourbon"))
> nephew.data
  age  name favorite.drink
1   3  Huey           coke
2   5 Dewey   orange juice
3   7 Louie        bourbon
```

use [row,col] to access individual elements

```
> nephew.data[2,2]
[1] Dewey
Levels: Dewey Huey Louie
```

Data frames

Access the whole row

```
> nephew.data[2, ]
  age  name favorite.drink
2   5 Dewey   orange juice
```

Access the whole column

```
> nephew.data[, 2]
[1] Huey  Dewey Louie
Levels: Dewey Huey Louie
```

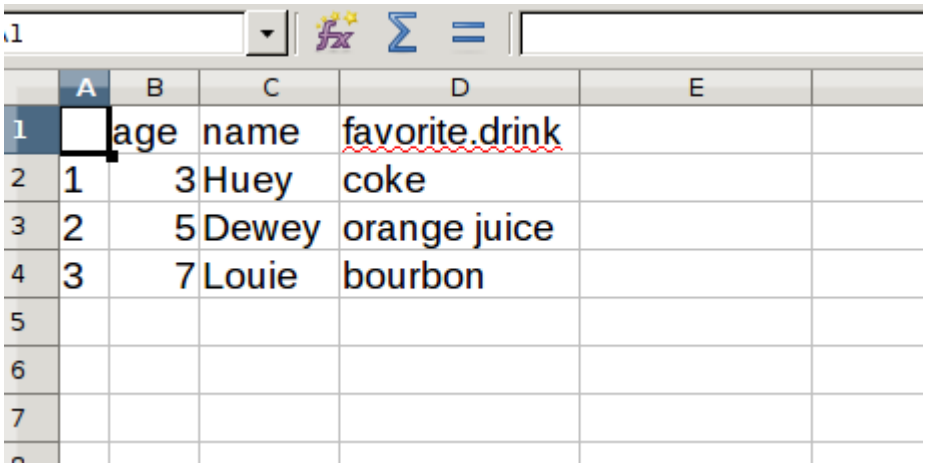
You can also use `$column.name` syntax

```
> nephew.data$name
[1] Huey  Dewey Louie
Levels: Dewey Huey Louie
```

Exporting the data

The most common ways to export the data:

```
> write.csv(nephew.data, file="nephews.csv")
```



	A	B	C	D	E
1		age	name	favorite.drink	
2	1	3	Huey	coke	
3	2	5	Dewey	orange juice	
4	3	7	Louie	bourbon	
5					
6					
7					

```
> nephew.data <- read.csv(file="nephews.csv")
```

Now you know R

Coffee break (15 min).

Introduction to topic modeling

Vitomir Kovanovic

Introduction to topic modeling

Original problem: How to synthesize the information in a large collections of documents?

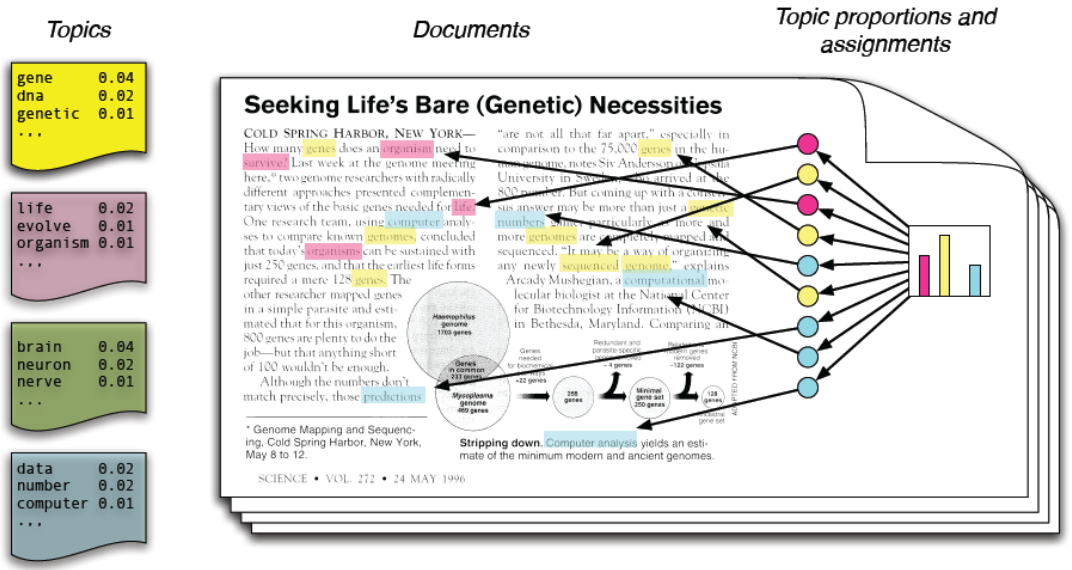
Example:

D1: modem the steering linux. modem, linux the modem. steering the modem. linux!
D2: linux; the linux. the linux modem linux. the modem, clutch the modem. petrol.
D3: petrol! clutch the steering, steering, linux. the steering clutch petrol. clutch the petrol; the clutch.
D4: the the the. clutch clutch clutch! steering petrol; steering petrol petrol; steering petrol!!!!

Typically done using some clustering algorithm to find groups of similar documents.

Alternative approach: Topic modeling

- Find several latent topics of themes that are “link” between the documents and words.
- Topics explain why certain words appear in a given document.
- Preprocessing is very much the same for all topic modeling methods (LSA, pLSA, LDA)



Pre-processing steps

Example:

D1: modem the steering linux. modem, linux the modem. steering the modem. linux!
D2: linux; the linux. the linux modem linux. the modem, clutch the modem. petrol.
D3: petrol! clutch the steering, steering, linux. the steering clutch petrol. clutch the petrol; the clutch.
D4: the the the. clutch clutch clutch! steering petrol; steering petrol petrol; steering petrol!!!!

Pre-processing steps

Step 1: Remove all non alphanumerical characters

D1: modem the steering linux modem linux the modem steering the modem linux
D2: linux the linux the linux modem linux the modem clutch the modem petrol
D3: petrol clutch the steering steering linux the steering clutch petrol clutch the petrol the clutch
D4: the the the clutch clutch clutch steering petrol steering petrol petrol steering petrol

Pre-processing steps

Step 2: Find all words that exist in the corpus. That defines our ***Vocabulary***

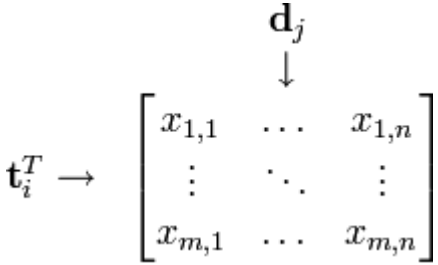
D1: modem the steering linux modem linux the modem steering the modem linux
D2: linux the linux the linux modem linux the modem clutch the modem petrol
D3: petrol clutch the steering steering linux the steering clutch petrol clutch the petrol the clutch
D4: the the the clutch clutch clutch steering petrol steering petrol petrol steering petrol

Vocabulary: modem, the, steering, linux, clutch, petrol

Topic modeling steps

Step 3: Create ***Document-Term-Matrix (DTM)***

	D1	D2	D3	D4
linux	3	4	1	0
modem	4	3	0	1
the	3	4	4	3
clutch	0	1	4	3
steering	2	0	3	3
petrol	0	1	3	4



Topic modeling steps

Step 3: Create ***Document-Term-Matrix (DTM)***

	D1	D2	D3	D4
linux	3	4	1	0
modem	4	3	0	1
the	3	4	4	3
clutch	0	1	4	3
steering	2	0	3	3
petrol	0	1	3	4

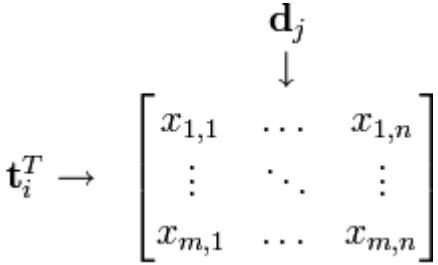
$$\mathbf{t}_i^T \rightarrow \begin{bmatrix} & & \mathbf{d}_j \\ & & \downarrow \\ x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix}$$

Back to starting problem

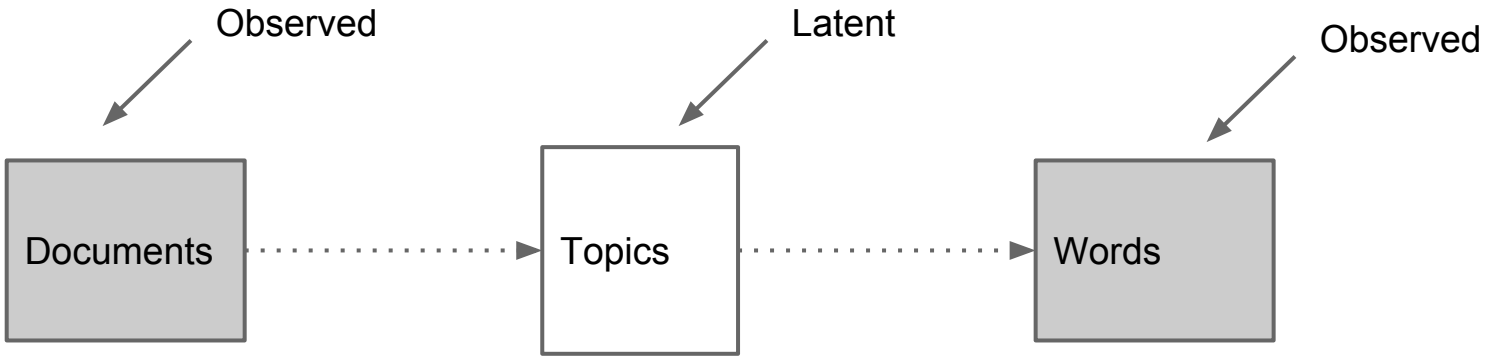
Can we somehow make matrices smaller?

Two topics => two rows in matrix

	D1	D2	D3	D4
linux	3	4	1	0
modem	4	3	0	1
the	3	4	4	3
clutch	0	1	4	3
steering	2	0	3	3
petrol	0	1	3	4



Enter Latent Semantic Analysis (LSA)



Example:
George Siemens's new blog post (i.e., a document) will likely be about:
MOOCs. (say 80%)
Connectivism (say 30%)

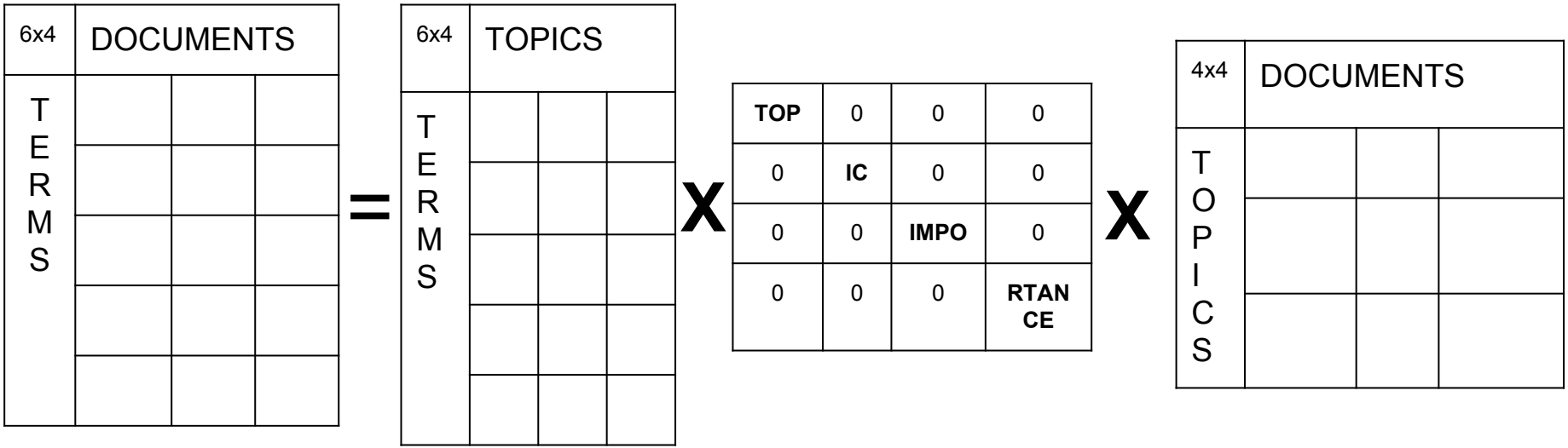
And given that it is 80% about MOOCs,
it will likely use words:
Massive (90%),
Coursera (50%),
Connectivism (85%),
Analytics (60%)
etc...

And given that it is 30% about Connectivism
it will likely use the words:
Network (80%)
Knowledge (40%)
Share (30%)
etc...

LSA

Nothing more than a **singular value decomposition (SVD)** of document-term matrix:

Find three matrices U, Σ and V so that: $X = U\Sigma V^t$



For example with 5 topics, 1000 documents and 1000 word vocabulary:
Original matrix: 1000 x 1000 = 10^6
LSA representation: 5x1000 + 5 + 5x1000 ~ 10^4
-> 100 times less space!

Latent Semantic Analysis (LSA)

Our example:

3	4	1	0
4	3	0	1
3	4	4	3
0	1	4	3
2	0	3	3
0	1	3	4

=

	To1	To2	To3	To4
Te1	-0.33	-0.53	0.37	-0.14
Te2	-0.32	-0.54	-0.49	0.35
Te3	-0.62	-0.10	0.26	-0.14
Te4	-0.38	0.42	0.30	-0.24
Te5	-0.36	0.25	-0.68	-0.47
Te6	-0.37	0.42	0.02	0.75

X

Topic Importance

11.4
6.27
2.22
1.28

X

	D1	D2	D3	D4
To1	-0.42	-0.48	-0.57	-0.51
To2	-0.56	-0.52	0.45	0.46
To3	-0.65	0.62	0.28	-0.35
To4	-0.30	0.34	-0.63	0.63

Latent Semantic Analysis (LSA)

First two topics much more important than other two topics!

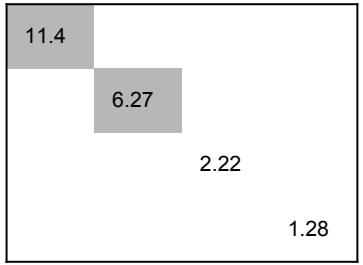
3	4	1	0
4	3	0	1
3	4	4	3
0	1	4	3
2	0	3	3
0	1	3	4

=

	To1	To2	To3	To4
Te1	-0.33	-0.53	0.37	-0.14
Te2	-0.32	-0.54	-0.49	0.35
Te3	-0.62	-0.10	0.26	-0.14
Te4	-0.38	0.42	0.30	-0.24
Te5	-0.36	0.25	-0.68	-0.47
Te6	-0.37	0.42	0.02	0.75

X

Topic Importance



X

	D1	D2	D3	D4
To1	-0.42	-0.48	-0.57	-0.51
To2	-0.56	-0.52	0.45	0.46
To3	-0.65	0.62	0.28	-0.35
To4	-0.30	0.34	-0.63	0.63

Latent Semantic Analysis (LSA)

We can drop columns/rows for the topics we are not interested.

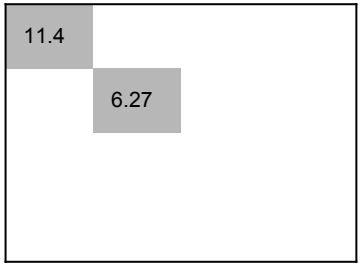
3	4	1	0
4	3	0	1
3	4	4	3
0	1	4	3
2	0	3	3
0	1	3	4

=

	To1	To2
Te1	-0.33	-0.53
Te2	-0.32	-0.54
Te3	-0.62	-0.10
Te4	-0.38	0.42
Te5	-0.36	0.25
Te6	-0.37	0.42

X

Topic Importance

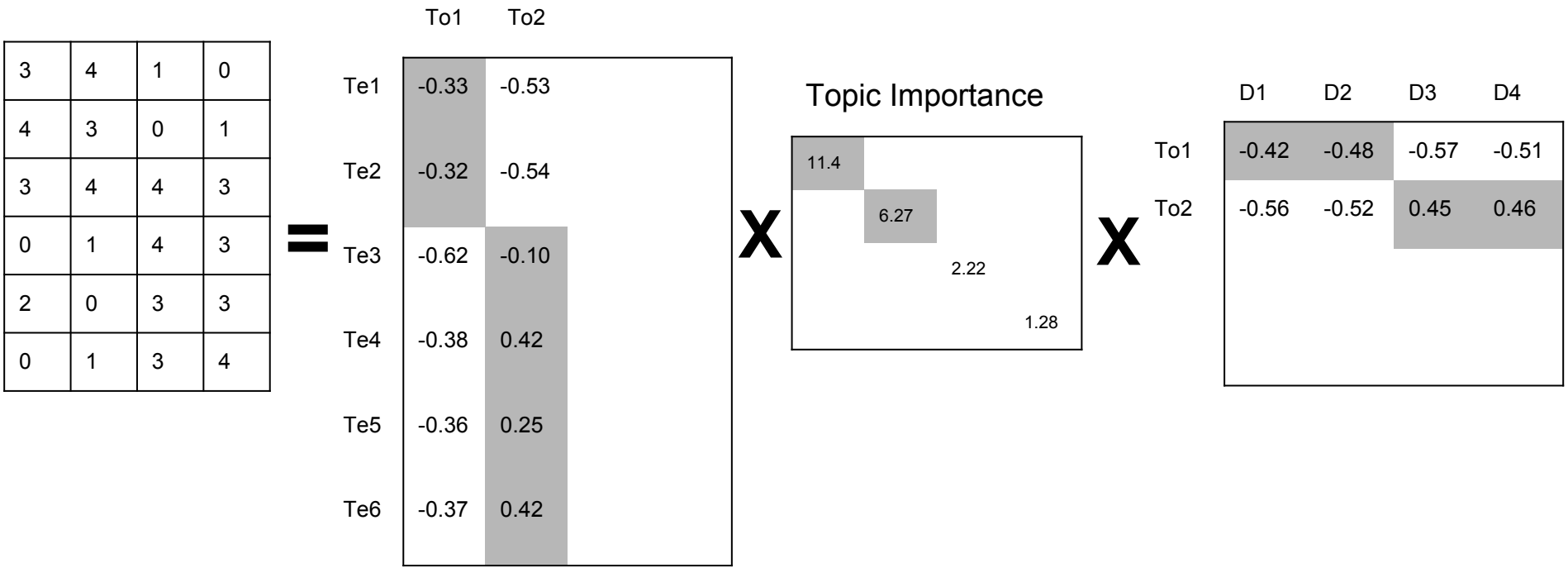


X

	D1	D2	D3	D4
To1	-0.42	-0.48	-0.57	-0.51
To2	-0.56	-0.52	0.45	0.46

Latent Semantic Analysis (LSA)

Pick highest assignments for each word to topic, and each topic to document



Latent Semantic Analysis (LSA)

LSA is essentially low-rank *approximation* of document term-matrix

3	4	1	0
4	3	0	1
3	4	4	3
0	1	4	3
2	0	3	3
0	1	3	4

=

Word assignment to topics

	IT	cars
linux	-0.33	-0.53
modem	-0.32	-0.54
the	-0.62	-0.10
clutch	-0.38	0.42
steering	-0.36	0.25
petrol	-0.37	0.42

X

Topic Importance

11.4	
	6.27

X

Topic distribution acr documents

	D1	D2	D3	D4
IT	-0.42	-0.48	-0.57	-0.51
cars	-0.56	-0.52	0.45	0.46

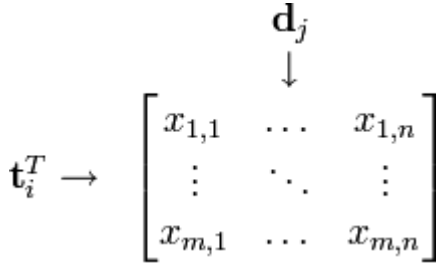
Probabilistic topic modeling

“All knowledge degenerates into probability”

David Hume

Back to Document Term Matrix

	D1	D2	D3	D4
linux	3	4	1	0
modem	4	3	0	1
the	3	4	4	3
clutch	0	1	4	3
steering	2	0	3	3
petrol	0	1	3	4



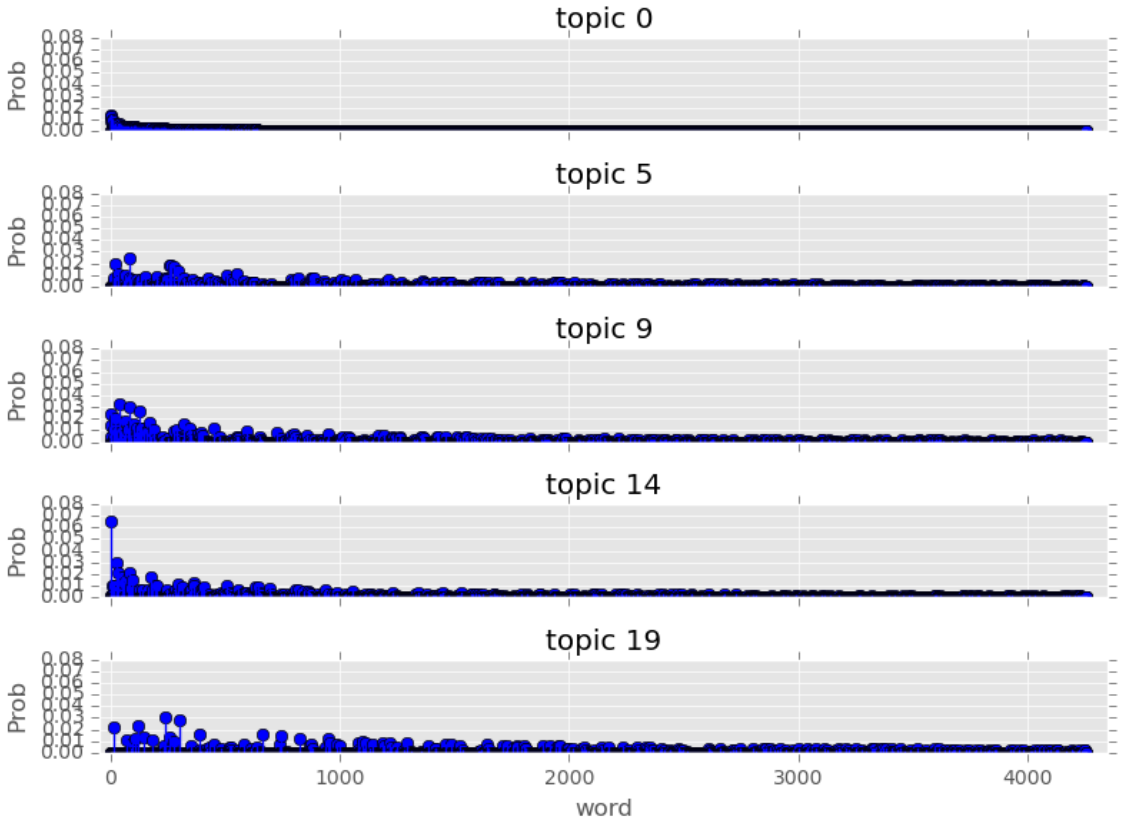
Probabilistic Topic Modeling

What is a topic?

A list of probabilities for each of the possible words in a vocabulary.

Example topic:

- dog: 5%
- cat: 5%
- hause: 3%
- hamster: 2%
- turtle: 1%
- calculus: 0.000001%
- analytics: 0.000001%
- .....
- .....
- .....



Probabilistic Topic Modeling

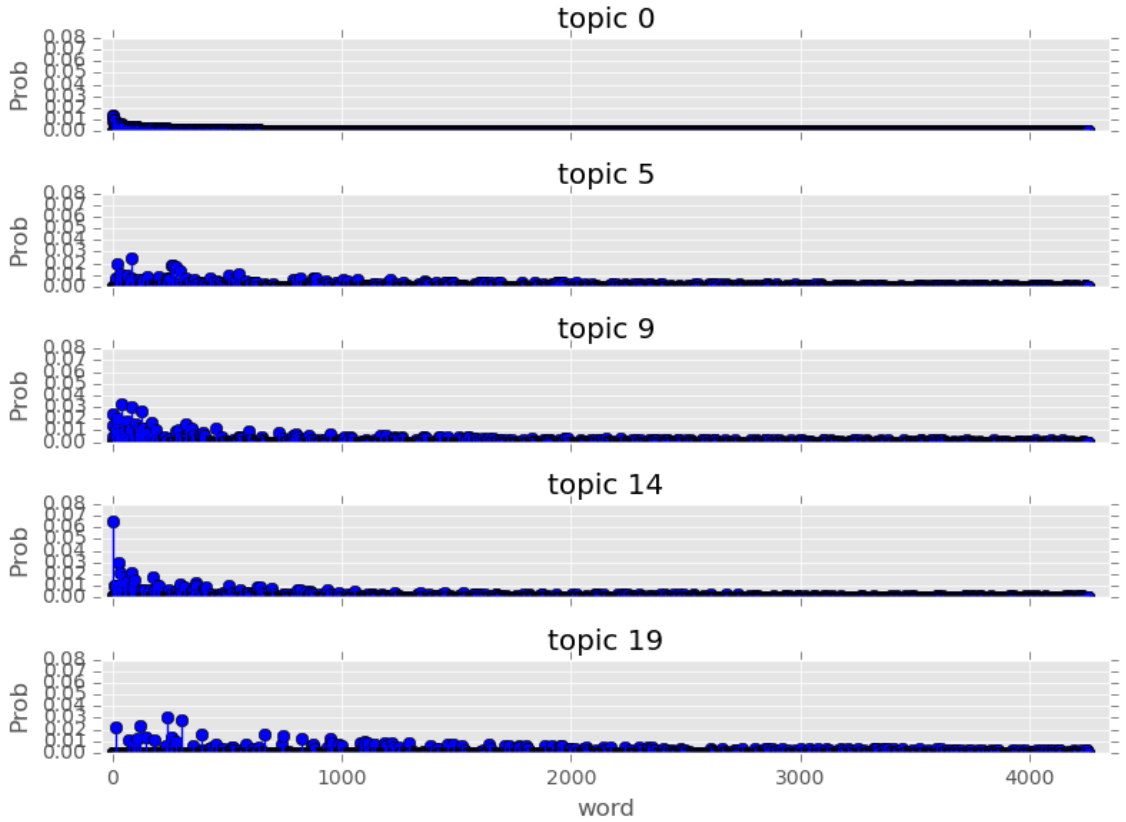
What is a topic?

A list of probabilities for each of the possible words in a given language.

Example topic:

Pets

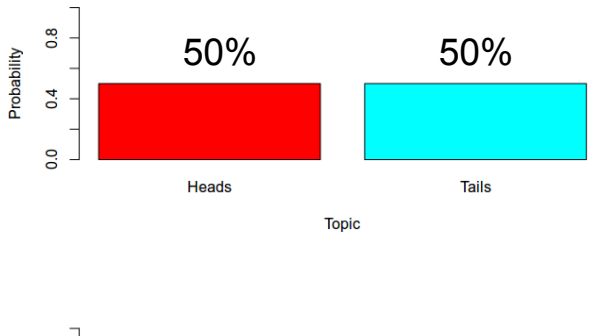
- dog: 5%
- cat: 5%
- hause: 3%
- hamster: 2%
- turtle: 1%
- calculus: 0.000001%
- analytics: 0.000001%
- .....
- .....
- .....

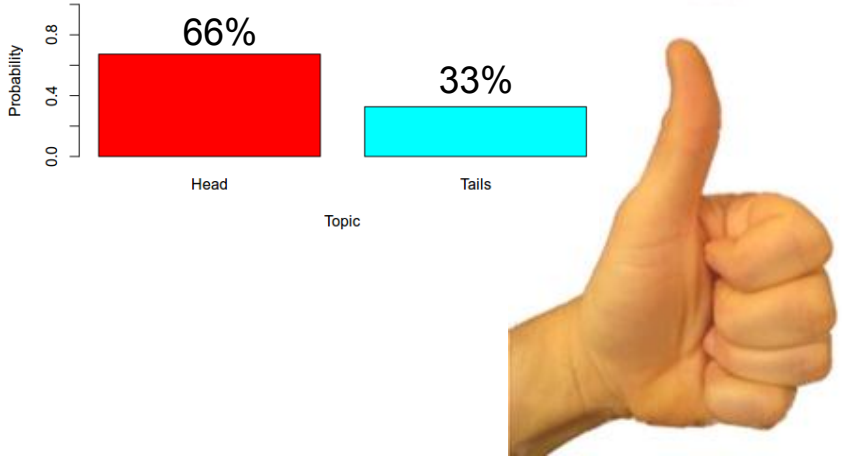


Digression: Coin tossing and dice rolling

- Two possible outcomes: Head 50%, Tail 50%
- Modeled by binomial distribution:
  - parameters:
    - number of trials n (e.g., 10),
    - probabilities for head/tails (e.g., 50/50)

- Coin can be weighted!
- Binomial distribution is still used!
  - only probabilities are different

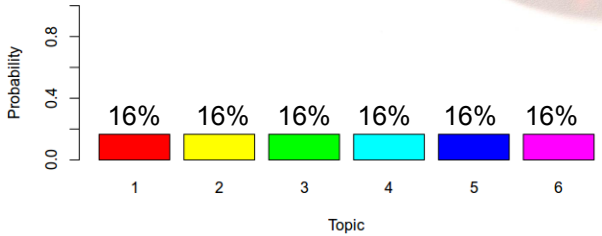


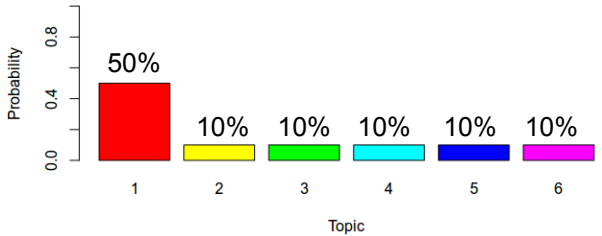


Digression: Coin tossing and dice rolling

- Analogously, rolling dices is modeled by multinomial distribution
  - parameters
    - number of trials n
    - probabilities (e.g. , 1/6 for a fair dice)

- Dices also can be rigged (e.g., weighted)
  - the same distribution applies, just with different probabilities.

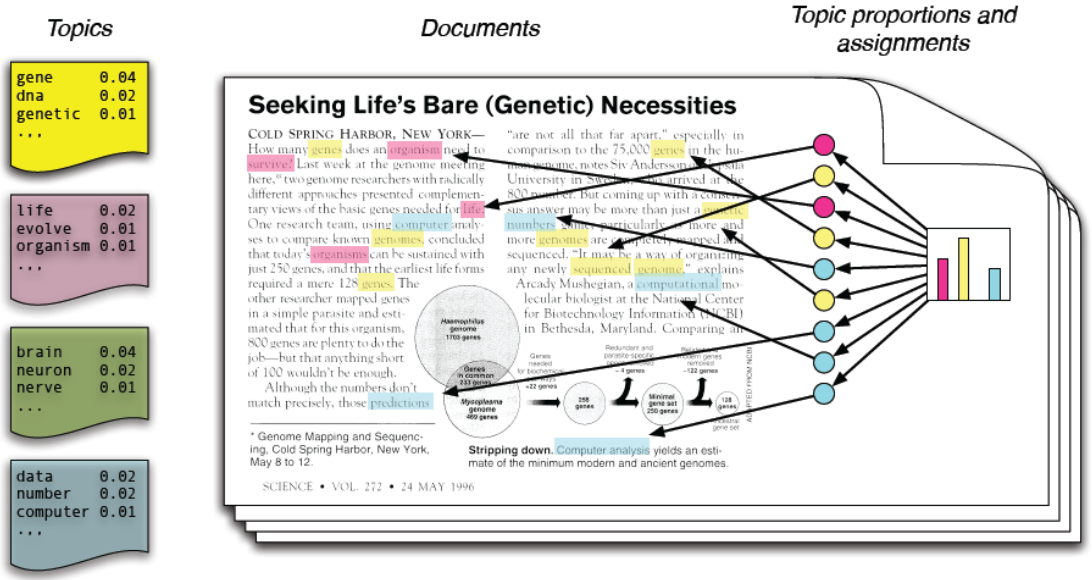




A probabilistic spin to LSA: pLSA

Instead of finding lower-ranked matrix representation, we can try to find a mixture of word->topic & topic->documents distributions that are most likely given the observed documents.

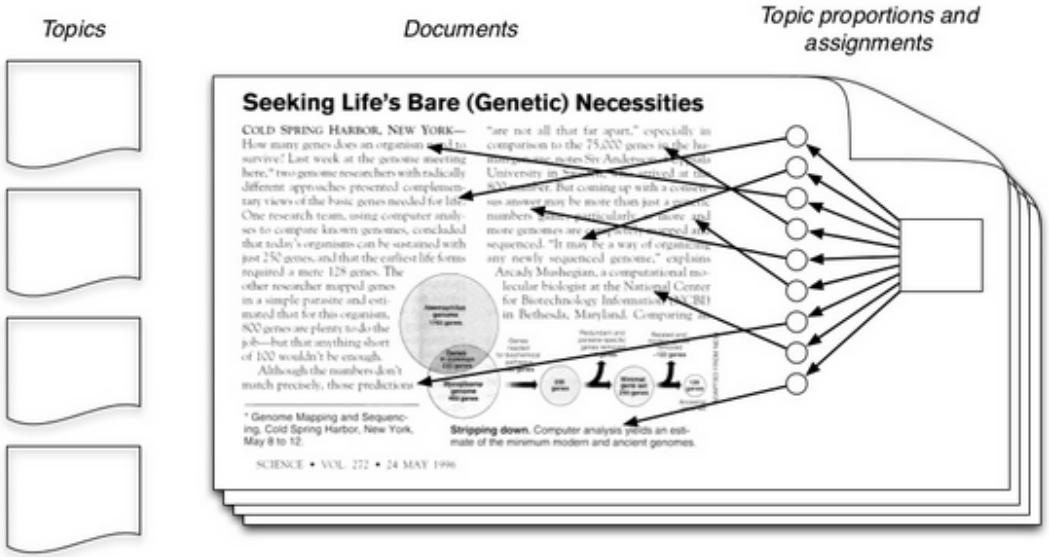
- We define a statistical model of how the documents are being made (generated).
  - This is called generative process in topic modeling terminology.
- Then we try to find parameters of that model that best fit the observed data.



A probabilistic spin to LSA: pLSA

Instead of finding lower-ranked matrix representation, we can try to find a mixture of word->topic & topic->documents distributions that are most likely given the observed documents.

- We define a statistical model of how the documents are being made (generated).
  - This is called generative process in topic modeling terminology.
- Then we try to find parameters of that model that best fit the observed data.



Generative process illustrated

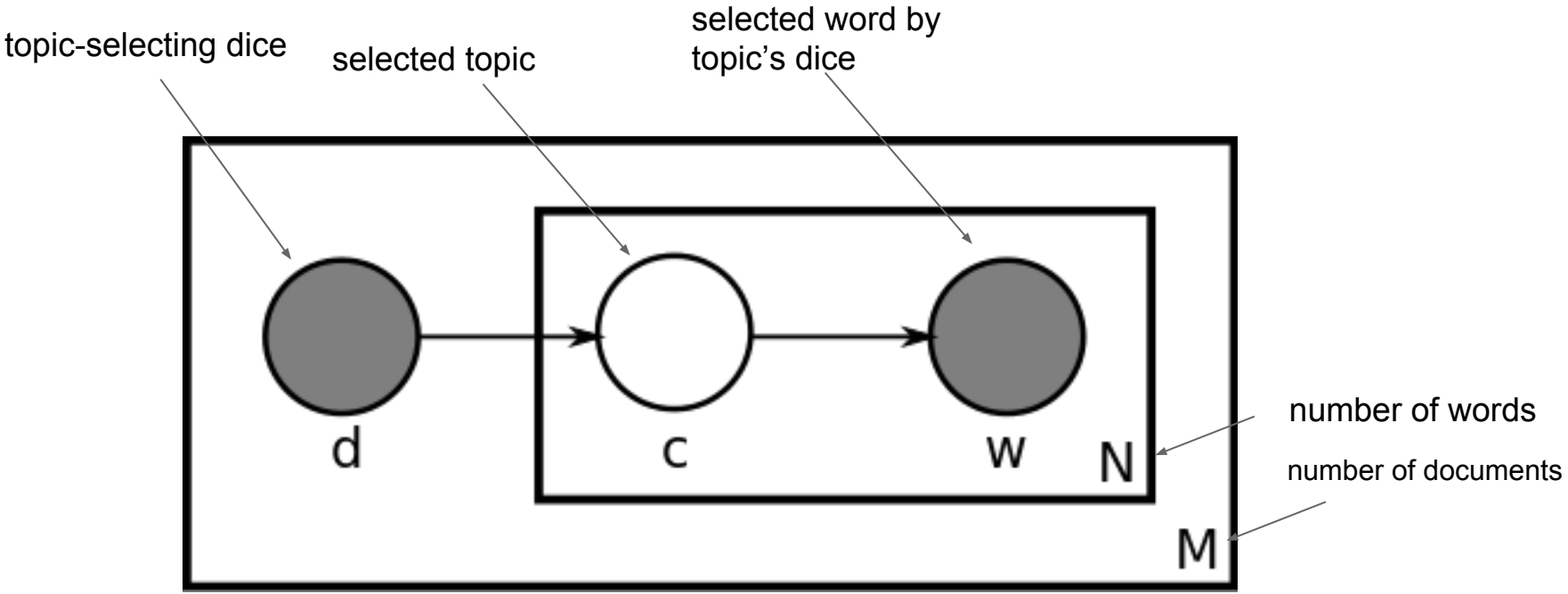
We just received a 50 word long document by our news reporter John Doe.
He is allowed to write only about one of the 6 possible topics, using only 6 words that exist in the world.

This is how we imagine that he wrote that article:

- For the first word, he throws a dice that tells him what is the topic of the first word. Say it is topic 1 (IT)
- Then he throws another dice to pick which word to use to describe topic 1. Say it is word 1 (linux)
- The process is repeated for all 50 words in the document.

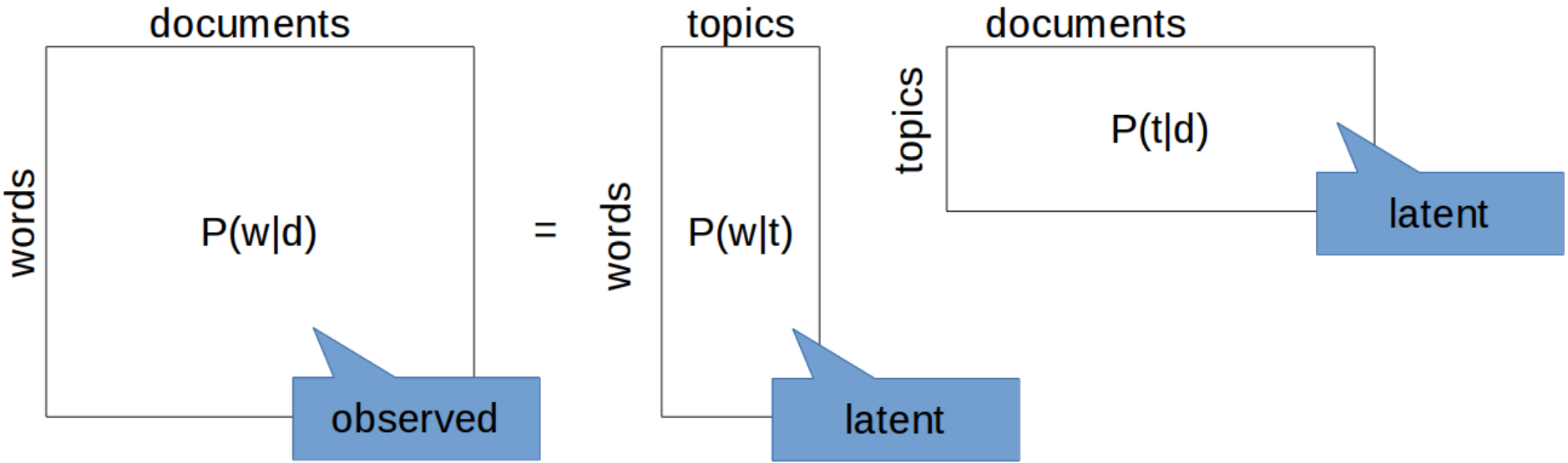
- One thing! **Dices are weighted**!!!
  - The first dice for picking topics puts more weight on IT topic that on the other 5 topics.
  - Also, dice for IT topic, puts more weight on words ‘linux’ and ‘modem’.
  - Likewise dice for topic 2 (cars) puts more weight on word ‘petrol’ and ‘steering’

Generative process illustrated



pLSA

$$P(w|d) = \sum_t P(t|d)\,P(w|t)$$



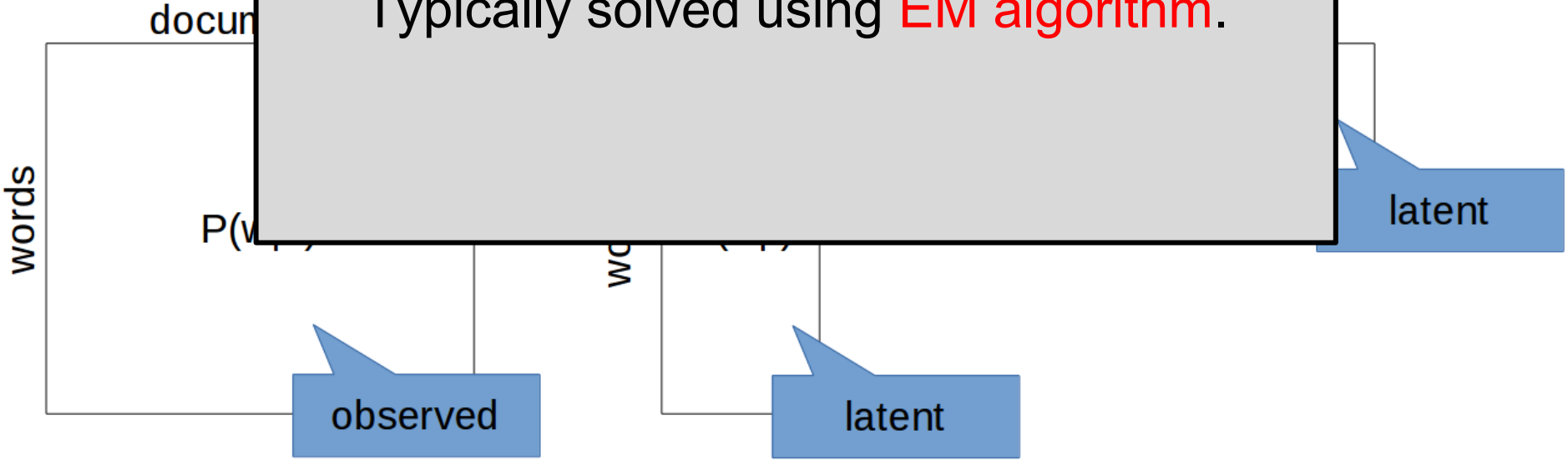
pLSA

Another more useful formulation:

$$P(w|d) = \sum_t P(t|d)\,P(w|t)$$

Find most likely values for P(t|d) and P(w|t).

Typically solved using EM algorithm.



pLSA challenges

- Tries to find topic distributions and word distributions that best fit the data
  - Overfitting
  - Some documents have strong associations to only couple of topics, while others have more evenly distributed associations.

LDA: an extension to pLSA

In LDA, w*e encode our assumptions about the data.*

Two important assumptions:

1. On average, how many topics are per document?
   a. Few or more?
2. On average, how are words distributed across topics?
   a. Are topics strongly associated with few words or not?

Those assumptions are defined by two vectors *α* and *β*:

α: K dimensional vector that defines **how K topics are distributed across documents**.
Smaller *αs* favor fewer topics strongly associated with each document.

β: V dimensional vector that defines **how V words are associated across topics**.
Smaller *βs* favor fewer words strongly associated with each topics.

IMPORTANT: Typically, all elements in *α* and *β* are the same (in `topicmodels` library 0.1 is default)
Uninformative prior: We don't know what are the prior distributions so we will say all are equally likely.

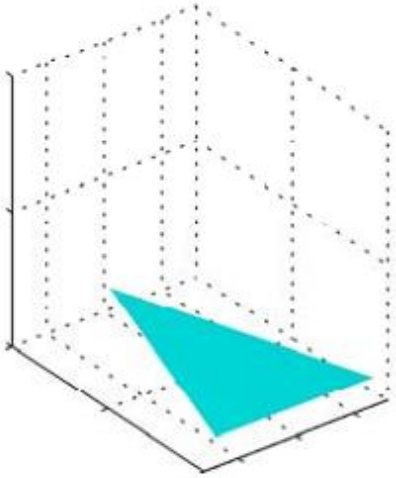
Those assumptions are then used to generate "dices" (topic-word associations, topic-document associations)

Dirichlet Distribution

- Multivariate distribution parametrized by a N-dimensional vector.
- *Drawing a sample from a dirichlet distribution parametrized with N-dimensional vector returns an N-dimensional vector that sums to one.*

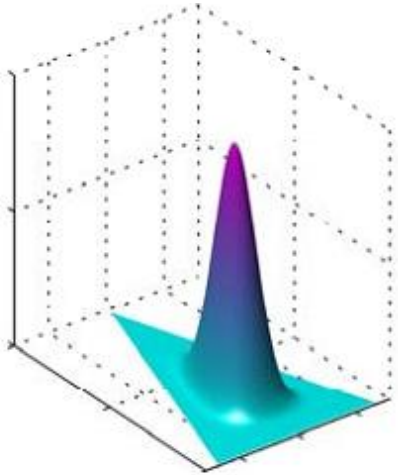
N=3:

Params = [1, 1, 1]



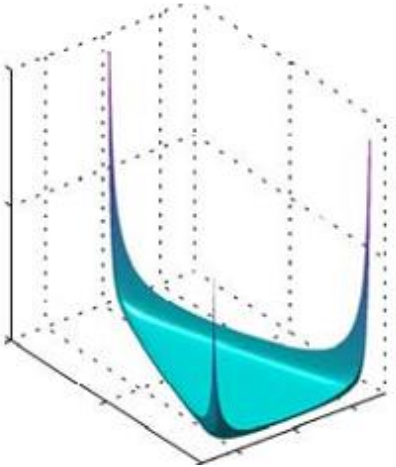
Bigger than 1

Params = [10, 10, 10]



Less than 1

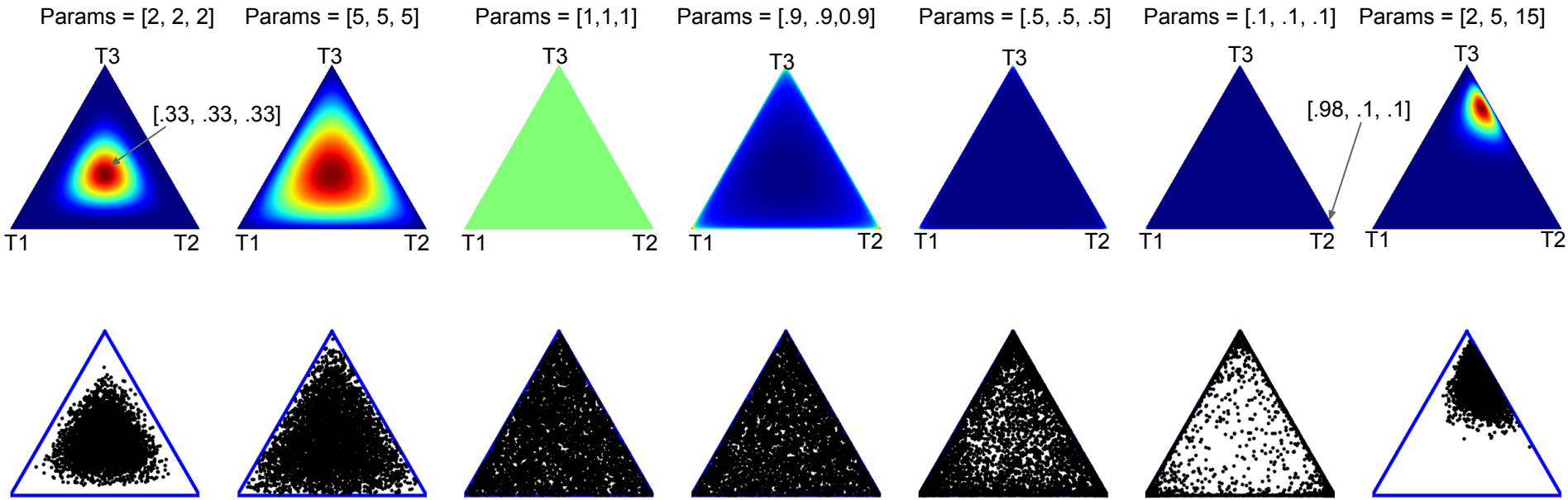
Params = [.1, .1, .1]



Dirichlet Distribution

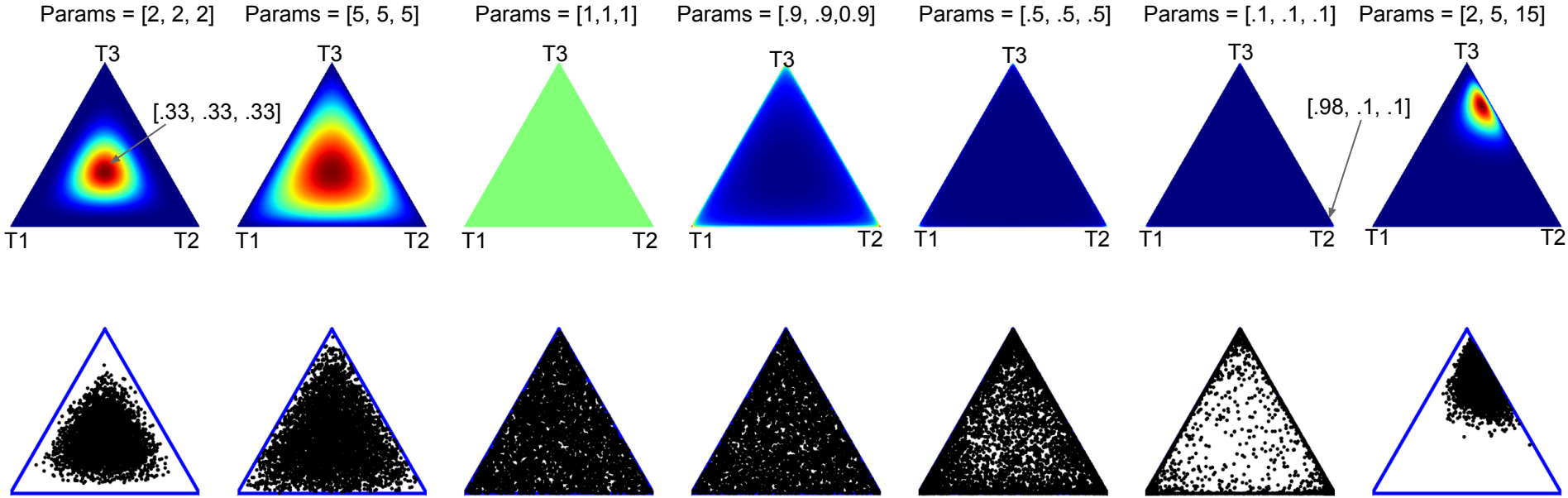
- Multivariate distribution parametrized by a N-dimensional vector.
- *Drawing a sample from a dirichlet distribution parametrized with N-dimensional vector returns an N-dimensional vector that sums to one.*

N=3:



Dirichlet Distribution

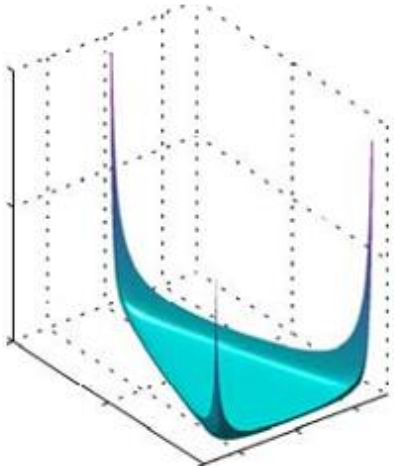
How topics are distributed across documents and how words are distributed across topics are drawn from Dirichlet distribution!



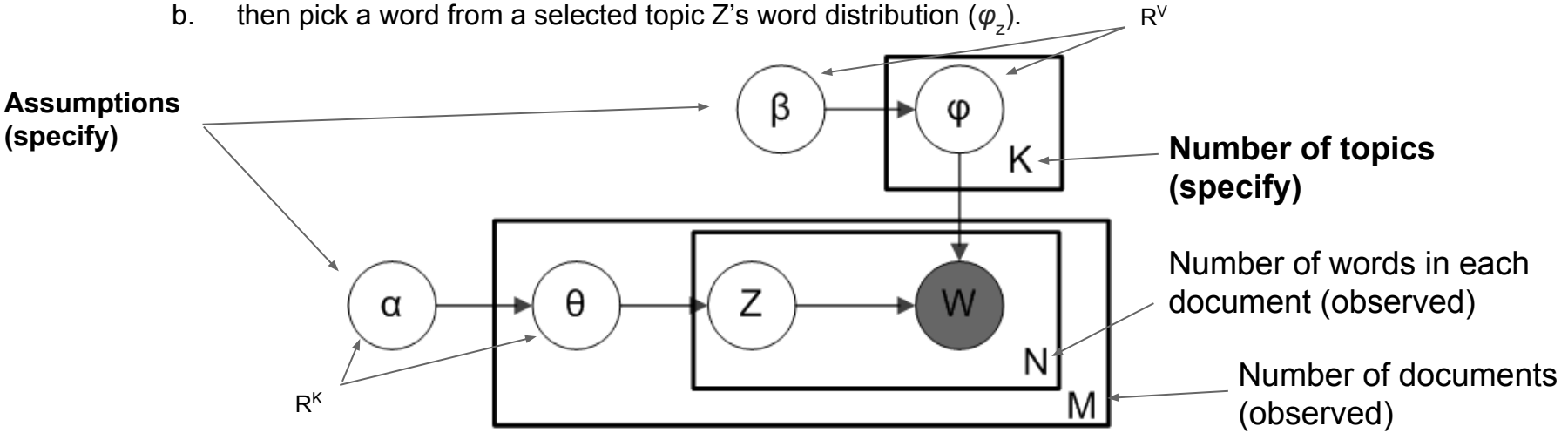
Generative process outline

How we imagine that documents were generated:

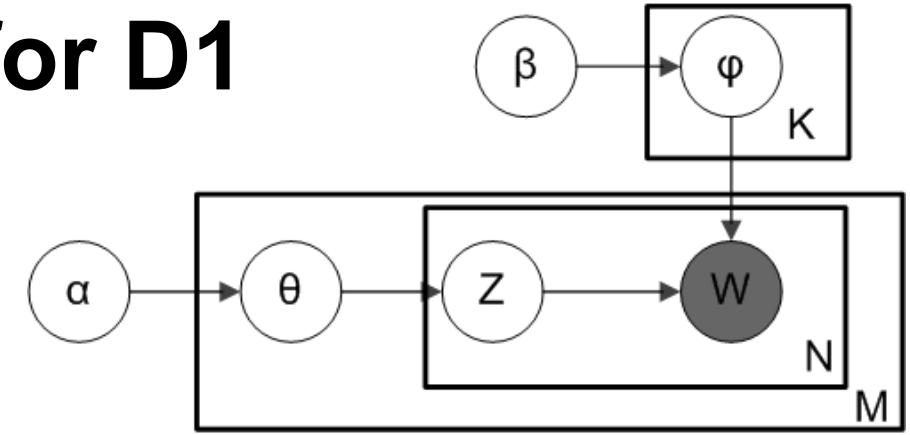
1. We specify number of topics K and our assumptions $\alpha \in R^K$ and $\beta \in R^V$ that capture general associations between document-topic and topic-word relationships.
2. For each document we pick one sample from a Dirichlet distribution (defined by α) that defines document's distribution over topics ($\theta \in R^K$).
3. For each topic we pick one sample from a Dirichlet distribution (defined by β) that defines topic's distribution over available words ($\varphi \in R^V$).
4. For each position in each document:
   a. Pick a topic Z from document's topic distribution
   b. then pick a word from a selected topic Z's word distribution (φ_z).



K=3, α = [.1, 0.1, .1]



Generative process for D1



D1: modem the steering linux modem linux the modem steering the modem linux

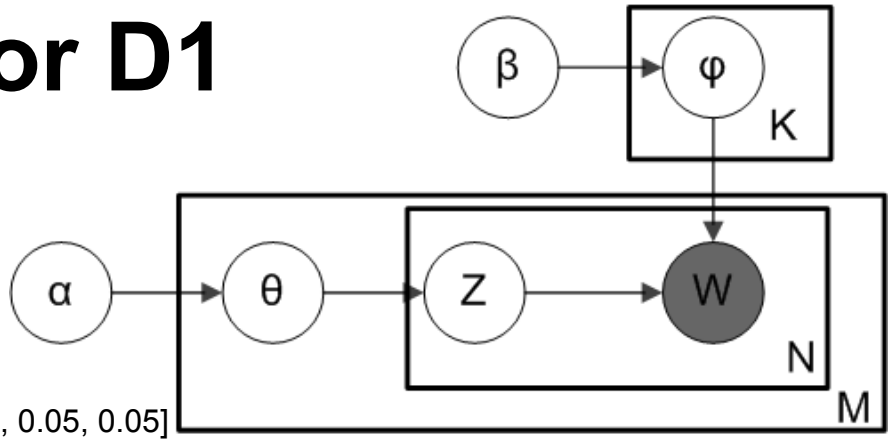
Our vocabulary is 6 words in total (V=6):
linux, modem, the, clutch, steering, petrol

We decide to use three topics (K=3). We call them IT, cars, and pets.

As we have three topics α has three elements. We set α to [.1, .1, .1].

As we have six words, β has six elements. We set β to [.1, .1, .1, .1, .1, .1].

Generative process for D1



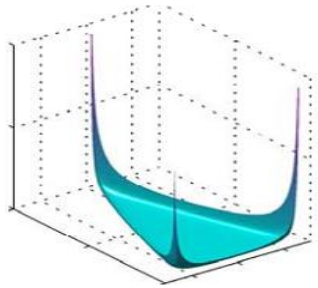
D1: modem the steering linux modem linux the
modem steering the modem linux

K=3, V=6, $\alpha \in R^3$ = [0.1, 0.1, 0.1],
$\beta \in R^6$ = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1],

From *Dirichlet*(β) we sample 3 times, once for each topic:

1. For topic IT we pick φ_{IT} . It happened to be = [0.3, 0.5, 0.05, 0.05, 0.05, 0.05]
2. For topic cars we pick φ_{cars}. It happened to be = [0.1, 0.05, 0.3, 0.15, 0.2, 0.2]
3. For topic pets we pick φ_{pets}. It happened to be = [0.05, 0.1, 0.1, 0.20, 0.3, 0.25]

From *Dirichlet*(α) we sample M times, once for every document:

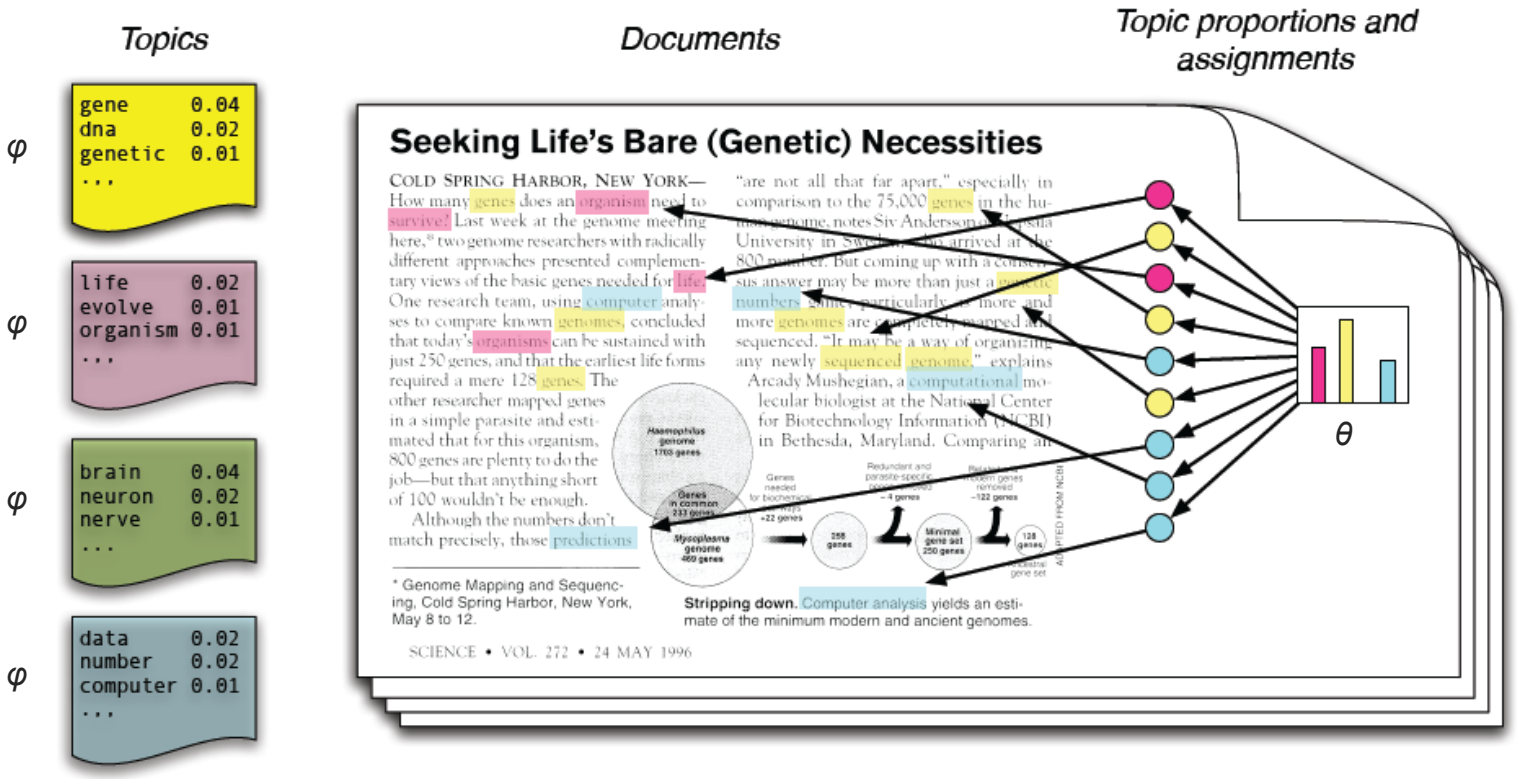


4. For document D1, we pick θ_{D1} . It happened to be [0.9, 0.05, 0.05]
5. *For document D2, we pick θ_{D2} . It happened to be .....*

   .....

6. For first position in document D1, we pick a topic from *Multinomial*([0.9, 0.05, 0.05]). Say it is IT.
   As the topic for the first position is IT, we pick a word from *Multinomial*([0.3, 0.5, 0.05, 0.05, 0.05, 0.05]).
   Oh look, it is 'modem'.
7. For second position in document D1, we pick a topic based on *Multinomial*(θ_{D1}). Say it is cars.
   As the topic for the second position is cars, we pick a word from *Multinomial*(φ_{cars}). Oh look, it is 'the'.
8. .....

Generative Process in LDA



What is LDA?

- Posterior inference of θs, φs, and Zs
- Going “backwards” from observation to latent elements in the model.
- The problem is intractable, so probabilities are estimated typically using:
  - MCMC through “Gibbs sampling, and
  - Variational Bayesian methods.

`topicmodels` R library

`topicmodels` R library

- “Interface” to `lda` library. Provides less functionality, but is easier to use.

- Analysis steps:
  a. reading data
  b. data cleanup (stopwords, numbers, punctuation, short words, stemming, lemmatization)
  c. building document term matrix
  d. filtering document term matrix based on TF and TF-IDF thresholds
     - TF: we don’t want words that appear only in few documents, they are useless for topic description
     - TF-IDF: we don’t want words that are appear a lot, but are too general (e.g., ‘have’)
  e. Either select K upfront, or evaluate several values of K for best fit.
     - Plot evaluation of different values of K
  f. Show top words for each topic
  g. Additional analyses (covariates etc)

`topicmodels` R library

```
library(topicmodels)
data <-  load files into memory
corpus <- Corpus(VectorSource(data));
preprocessing.params <- list(stemming = T,
                             stopwords = T,
                             minWordLength = 3,
                             removeNumbers = T,
                             removePunctuation = T)
dtm <- DocumentTermMatrix(corpus, control = preprocessing.params)
dtm <-  filter noise from DTM (TF, TF-IDF)
lda.params <- list(seed=1, iter= 2000)
LDA(dtm, k = 20, method = "Gibbs", control = lda.params)
```

plot graphics

do further analysis....

`topicmodels` R library

- Problem of picking number of topics K
  - Perplexity on the left-out test set
  - Harmonic mean evaluation

- A bit more coding for:
  - Filtering of words
  - Plotting perplexity scores requires coding
  - Calculating frequencies of each topic requires a bit of coding
  - Finding most likely topics for each document requires

lak_topic_models.R script

```
# include utility script
source("lak_topic_modeling.R")

# project name, used for saving analysis results
project.id <- "mooc news topic models"

# what values of K to check?
all.ks <- seq(2, 60, 2)

dtm <- build.dtm(load.dir="./data/mooc_news/",
                 tf.threshold=0.95,
                 tf.idf.threshold=0.9,
                 tf.idf.function="median",
                 stemming=T,
                 stopwords.removal=T,
                 minWordLength=3,
                 removeNumbers=T,
                 removePunctuation=T,
                 minDocFreq=3)
```

lak_topic_models.R script

```
# build all topic models with different values of K
all.topic.models <- build.topic.models(dtm, all.ks, alpha=0.1)

# get the best topic models based on harmonic mean metric
best.model <- get.best.model(all.topic.models)

# assign all documents to most likely topics
document.topic.assignments <- get.topic.assignments(best.model)

log.message("\n\n*****************************************")

# print LDA results
print.topics.and.term(document.topic.assignments, best.model, n.terms=10)

# print document headlines
print.nth.line(document.topic.assignments, load.dir)
```

Coffee break (15 min).

Collaborative Topic Modeling:

Step-by-step LDA topic modeling on the real-world dataset.

Research objective

- Discover emerging topics from learner generated blog data, within a cMOOC

Course and dataset

- Connectivism and Connective Knowledge 2012 (CCK12)

- Aim of the course

- Data
  - ~ 285 blogs (including comments)

Course topics

- What is Connectivism?
- Patterns of connectivity
- Connective Knowledge
- What Makes Connectivism Unique?
- Groups, Networks and Collectives
- Personal Learning Environments & Networks
- Complex Adaptive Systems
- Power & Authority
- Openness & Transparency
- Net Pedagogy: The role of the Educator
- Research & Analytics
- Changing views, changing systems: From grassroots to policy

Let's dive into the data...

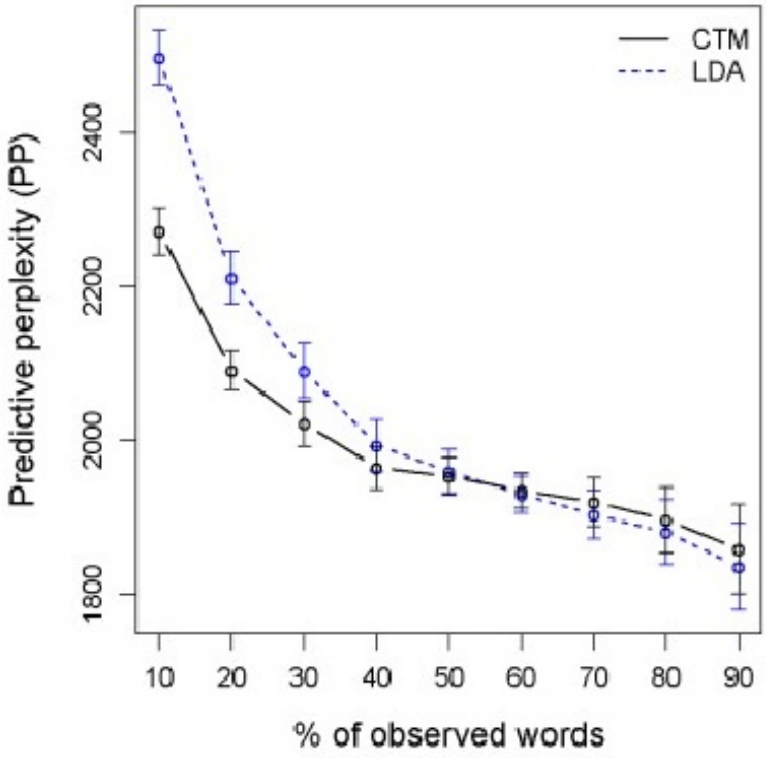
Advanced topic modeling techniques

Correlated Topic Model

- CTM allows the topics to be correlated

- CTM allows for better prediction
  - Logistic normal instead of Dirichlet distribution

- More robust to overfitting



CTM in R

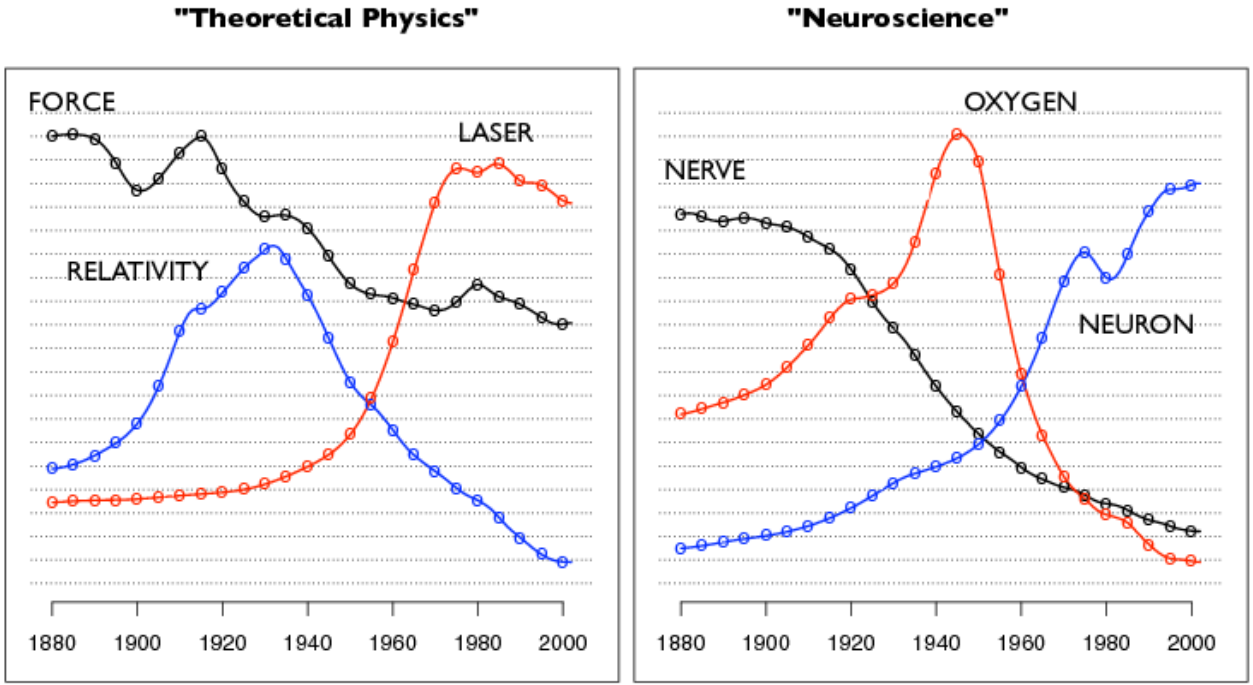
- Package "topicmodels"
  - function CTM

CTM(x, k, method = "VEM", control = NULL, model = NULL, ...)

- Arguments
  - x - DocumentTermMatrix object
  - k - Integer, number of topics
  - method - currently only "VEM" supported

Dynamic Topic Model

- DTM models how each individual topic changes over time



Supervised LDA

- Each document is associated with an external variable
  - for example, a movie review could be associated with a numerical ratings.

- Defines a one-to-one correspondence between latent topics and user tags

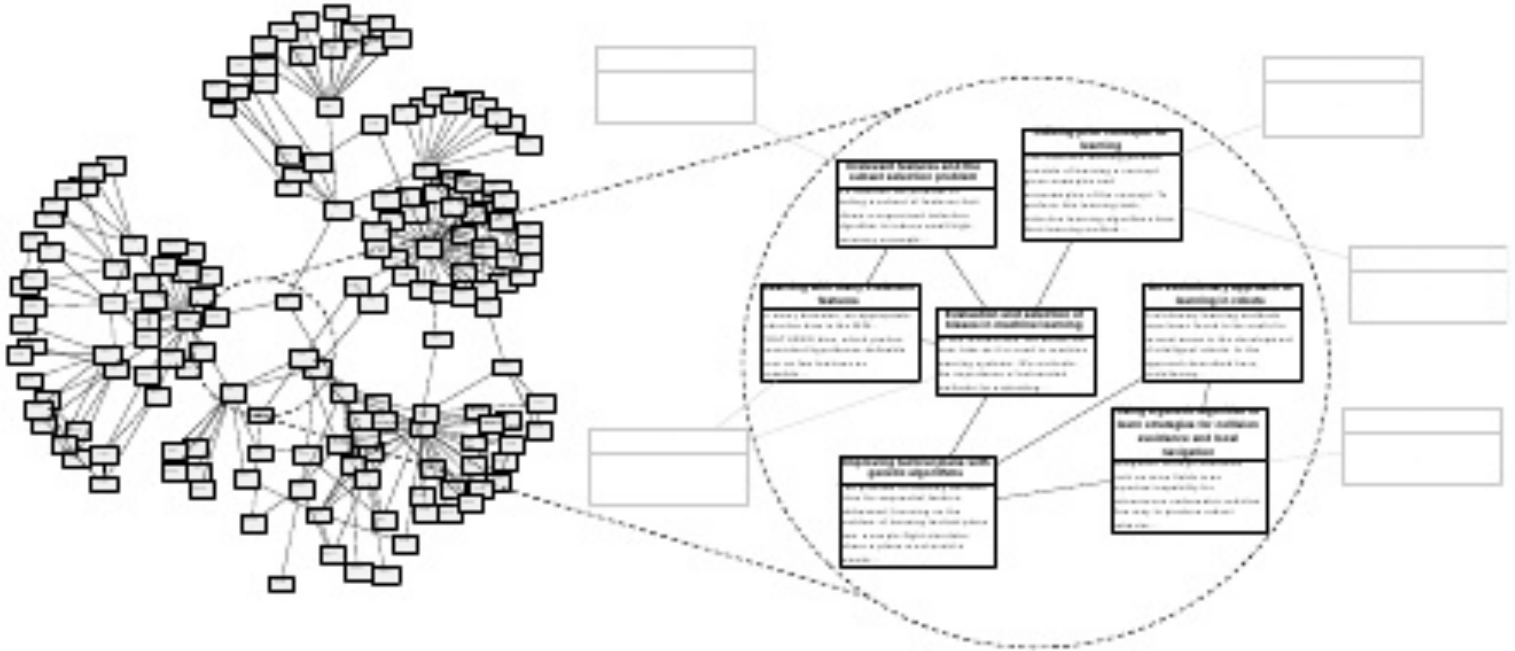
- Allows sLDA to directly learn word-tag correspondences.

sLDA in R

- Package "`lda`"

```
slda.em(documents,
        K,
        vocab,
        num.e.iterations,
        num.m.iterations,
        alpha,
        eta,
        annotations,
        params,
        variance,
        logistic = FALSE,
        lambda = 10,
        regularise = FALSE,
        method = "sLDA",
        trace = 0L)
```

Relational Topic Models



Relational Topic Models

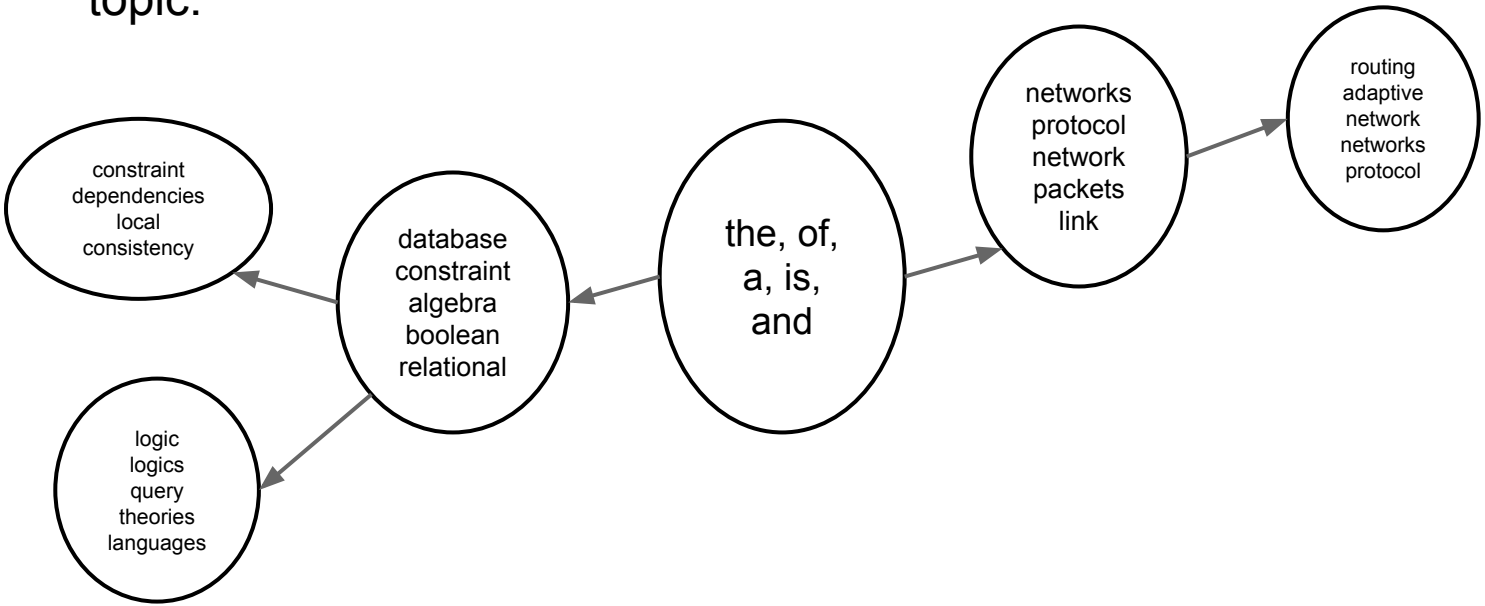
- Given a new document RTM predicts which documents it is likely to be linked to

- For example - tracking activities on Facebook in order to predict a reaction on an advertisement

- RTM is also good for certain types of data that have spatial/geographic dependencies

Hierarchical topic modeling

- LDA fails to draw the relationship between one topic and another.
- CTM considers the relationship but fail to indicate the level of abstract of a topic.



Structural topic modeling

- STM allows for the inclusion of covariates of interest.
- STM vs. LDA
  - topics can be correlated
  - each document has its own prior distribution over topics, defined by covariate X rather than sharing a global mean
  - word use within a topic can vary by covariate *U*

- The STM provides fast, transparent, replicable analyses that require few a priori assumptions about the texts under study

STM in R

- `stm`: R package for Structural Topic Models
  http://structuraltopicmodel.com

Topic Mapping

LDA

- accuracy, reproducibility

Network approach

- *Preprocessing*
  - stemming -> "stars" to "star"
  - stop word removal
- *Pruning of connections*
  - remove nonsignificant words
- *Clustering of words*
  - Infomap community detection algorithm
- *Topic-model estimation*

References

[1] David M. Blei and John D. Lafferty: "A Correlated Topic Model of Science", *The Annals of Applied Statistics*, Vol. 1, No. 1 (Jun., 2007) , pp. 17-35 Published by: Institute of Mathematical Statistics, URL: http://www.jstor.org/stable/4537420

[2] Blei, David M., and John D. Lafferty. "Dynamic topic models." Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

[3] Gerrish, Sean, and David Blei. "The ideal point topic model: Predicting legislative roll calls from text." Proceedings of the Computational Social Science and the Wisdom of Crowds Workshop. Neural Information Processing Symposium. 2010.

Summary

Recap

- Topic modeling is used in variety of domains
  - humanities,
  - history,
  - engineering & computing,
  - social sciences, and
  - education
- All try to model relationships between observed words and documents by a set of latent topics
- Used for exploration and getting “sense of the data”
- Require reasonably small data pre-processing
- Document-Term Matrix (DTM) is the foundation on which all topic modeling methods work
  - Bag-of words approaches as DTM does not contain ordering information
  - Punctuation, numbers, short, rnare and uninformative words are typically removed.
  - Stemming and lemmatization can also be applied

Algorithms

- Several different topic modeling algorithms
  - LSA
    - Finding smaller (lower-rank) matrices that closely approximate DTM
  - pLSA
    - Finding topic-word and topic-document associations that best match dataset and specified number of topics K
  - LDA
    - Finding topic-word and topic-document associations that best match dataset and specified number of topics that come from Dirichlet distribution with given *dirichlet priors*.

Practical application

- Practical applications of algorithms require use of different libraries
  - Most require certain level of programming proficiency
  - Require fiddling with parameters and human judgement
  - MALLET and R libraries most widely used.
  - C/C++ libraries typically have more advanced algorithms

Where to go next

- `lak_topic_modeling.R` provides a nice template for LDA analyses
- Play around with parameters and extend it for more advanced algorithms
- Try MALLET and other R libraries that provide correlated topic models or structured topic models
- Even better: find a (poor) grad students to do that for you
- Try out the basic topic modeling using topicmodels library
- Use covariates such as publication year or author to make interesting interpretations and insights into the dataset

Discussion & Questions
Topic Modeling in Learning Analytics

Q1: Do you plan on using topic modeling for some of your own research projects?

Q2: Do you see particular problems that you can address with topic modeling?

Thank you